GRAPH-BASED LEARNING FOR COMMUNITY DETECTION AND HUB NODE IDENTIFICATION

By

Meiby Ortiz-Bouza

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical and Computer Engineering—Doctor of Philosophy

2025

ABSTRACT

Many real-world systems can be represented using complex networks, where the different agents and their relations are represented as nodes and links, respectively. Traditional network models employ simple graphs where the graph is represented by a set of vertices and edges that connect them. With the advances in data acquisition technologies and the different types of data that are available, the simple graph model becomes insufficient to describe the higher dimensional relational data sets. For instance, in social networks, users can be defined as nodes with multiple types of interactions like friendship, collaboration, and economic exchange, connecting them. Furthermore, each node is often associated with attributes such as demographics or interests. To overcome the limitation of existing simple graph models, multi-dimensional graphs such as multiplex networks have been proposed. Similarly, in order to capture the node information that is available in most real-world networks, attributed graphs have been introduced.

Given a large scale complex network, one is usually interested in learning the underlying graph structure, such as the community structure or hub nodes. These structures uncover meaningful patterns and provide insights within complex networks. Community detection identifies groups of nodes that are more densely connected to each other than they are to the rest of the network. Hub nodes, on the other hand, correspond to nodes which are densely connected to the rest of the graph and play a critical role in information processing in the network. Although there are numerous works on community detection in single-layer networks, existing work on multiplex community detection mostly focuses on learning a common community structure across layers without taking the heterogeneity of the different layers into account. Beyond detecting communities within a single multiplex network, many applications may require comparing the community structures of two or more multiplex networks. Furthermore, most of the existing community detection methods focus solely on the graph connectivity information. In attributed graphs, where each node is associated with an attribute vector, the community detection methods that focus only on the edges and the data clustering methods that focus only on the attributes of the nodes become insufficient. Traditional hub detection methods rely mostly on graph connectivity without taking the node attributes into

account. This thesis addresses the limitations of learning these graph structures in high-dimensional and attributed networks. Novel algorithms for multiplex and attributed community detection, as well as approaches for discovering discriminative communities between two multiplex networks, are introduced using graph spectral theory and graph signal processing methods. Similarly, a graph signal processing approach that takes into account both the graph topology and node attributes is introduced for hub node identification.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to everyone who has supported me throughout this journey. First, I am deeply thankful to my advisor, Dr. Selin Aviyente, for her guidance, encouragement, invaluable insights, and the significant contributions she has made to my professional and personal growth throughout this process. Second, I'd like to thank my co-authors, Hanlu Yang, Duc Vu, Sema Athamnah, and Dr. Adbullah Karaslaanli, for their collaboration, valuable input, and insightful discussions. Additionally, I am grateful to my Ph.D. guidance committee members for their time, constructive feedback, and support at each milestone of this journey.

A special thank you to my parents, stepparents, brother, and sister for their unconditional love and support; to my friends and extended family for being there through every step of this journey; and to my husband, whose love and encouragement have been my greatest source of strength.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Many real-world complex systems spanning from social systems to biological entities can be effectively modeled as complex networks, where entities and their interactions are represented as nodes and links, respectively [14]. These networks can be mathematically described using graphs, which consist of a set of vertices connected by edges. Modeling relational datasets as graphs provides a powerful framework for representing, analyzing, and extracting valuable insights from complex data. It enables the exploration of relationships, patterns, and structures in a wide range of applications, making it an essential tool in data analysis. Traditional network models are based on simple graphs, where a single edge describes the relationship between the vertices. While this basic representation has been effective for many applications, it falls short when dealing with more complex and multidimensional datasets. The limitations of these single-graph models become evident in light of the increasing availability of diverse data types and the need to capture richer relationships within networks. In real-world scenarios, nodes may interact in various ways, and these interactions may carry different meanings. For example, in a transportation network, cities might be connected by different modes of transportation (buses, airplanes, trains). Similarly, the same group of people may interact in different ways on social media platforms such as Facebook, Twitter, and LinkedIn.

Since single graph models can only denote one edge between vertices, they are not capable of capturing the multiple modes of interaction that can exist between nodes. To address this issue, multiplex networks have emerged as a model where the traditional graph model is extended by introducing multiple layers, each of which represents a distinct mode of interaction. Multiplex networks have found applications in modeling a broad spectrum of complex systems, including living organisms, human societies, transportation systems, and critical infrastructures [229, 7].

Another significant limitation of single-graph models is their inability to capture the rich node information often available in real-world networks. In many applications, nodes have attributes or properties beyond their connectivity patterns [42, 207]. For instance, in protein networks, the

interactions between proteins can be modeled as a graph, but there are also known properties of such proteins that are available. In social networks, in addition to the different modes of interaction between the nodes, information about each node such as demographics, e.g., age, gender, location, and interests may be available. To accommodate this type of information, attributed graphs have been introduced, extending traditional graph structures by allowing nodes to have attributes or properties. This extension enriches the representation of nodes and edges, enabling a more comprehensive analysis of complex systems.

Given a large complex network, one of the most important problems is dimensionality reduction and network structure discovery. Community detection and hub node identification are two particular forms of network structure discovery problems. Communities allow us to identify groups of functionally related objects (i.e., nodes) and the interactions between them. For example, in social networks, communities correspond to groups of friends who attended the same school or who come from the same hometown [151]; in protein interaction networks, communities are functional modules of interacting proteins [6]; in co-authorship networks, communities correspond to scientific disciplines [99]. Various methods have been proposed for detecting the community structure of single layer networks [92]. Among these, the most commonly used ones are spectral clustering [155, 256], methods based on statistical inference [1], approaches based on optimization of a quality function [51], and techniques based on network dynamics [213]. Although there are numerous works on community detection in single-layer networks, existing work on multiplex community detection mostly focuses on learning a common community structure across layers without taking the heterogeneity of the different layers into account [52, 77, 245]. In the case of attributed networks, both the graph connectivity and node attributes are available. Thus, two sources of data can be used to perform the clustering task. The first is the data about the objects, i.e., node attributes. For example, known properties of proteins, users' social network profiles, or authors' publication histories may tell us which objects are similar, and to which communities they may belong. The second source of data is network connectivity, i.e., the set of edges between the nodes, such as the friendship relationships between users, interactions between proteins and

the collaboration between authors. However, classical clustering methods typically focus only on one of these two data modalities. While data clustering methods such as k-means can be used to assign class labels based on attribute similarity [123], community detection algorithms aim to find communities based on the network structure, e.g., to find groups of nodes that are densely connected [91, 248, 182]. Employing just one of these two sources of information in isolation can result in the algorithm failing to account for important structures in the data.

In the case of the hub node identification, the goal is to detect highly connected nodes, known as hubs, that play a central role in network connectivity and information processing. Hub nodes play a critical role in various domains, such as social networks, biological systems, and brain connectomics, where they act as key influencers [118], essential proteins [252], or important brain regions [253], respectively. Traditionally, hubs have been defined based on structural properties using common measures of centrality such as degree [188], clustering coefficient [189], vulnerability [131], betweenness [282], and eigenvector centrality [165], which rely solely on connectivity information. While these measures effectively capture nodes with high influence in terms of network topology, they overlook intrinsic node attributes that may be equally important in defining hub roles. One critical area where hub identification has significant implications is brain network analysis. Although there has been some works for identifying brain hubs [129], most the proposed methods rely only on the functional connectivity network without considering the coupling between the brain's anatomical wiring, i.e., structural network, and its dynamic functional properties, e.g., the BOLD signal [93].

1.1 Notations and Background

Vectors and matrices are indicated by bold lowercase letters, $\mathbf{x}$, and bold uppercase letters $\mathbf{X}$, respectively. Entries of a vector are denoted as x_i, and entries of a matrix are denoted as X_{ij}. The ith row and column of $\mathbf{X}$ are indicated as $\mathbf{X}_{i\cdot}$ and $\mathbf{X}_{\cdot i}$, respectively and they are both column vectors. Superscript $^\top$ indicates the transpose of vectors and matrices, identity matrix is shown by $\mathbf{I}$, and $||\cdot||_F$ and $||\cdot||_1$ are the Frobenious norm and ℓ_1 norm, respectively.

1.1.1 Graphs

Let $\mathcal{G} = (V, E)$ be a graph where V is the vertices set with $|V| = N$, $E \subseteq V \times V$ is the edge set. An edge from vertex i to j is represented by $e_{ij} \in E$ and it is associated with a weight. Graphs are natural representations of networks, where *vertices* correspond to network *nodes* and *edges* correspond to network *links*. A graph can be represented algebraically by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. If the graph is undirected, then the adjacency matrix is symmetric. For a weighted graph, $A_{ij} \in [0, 1]$, and for a binary graph, $A_{ij} \in \{0, 1\}$. In this thesis, we use undirected (symmetric) weighted and binary adjacency matrices. A degree matrix, $\mathbf{D}$, for an undirected graph is a diagonal matrix with elements D_{ii}, which are equal to the sum of weights of all edges connected to the vertex i, that is, the sum of elements in the $i-$th row of $\mathbf{A}$, $D_{ii} = \sum_j A_{ij}$. Therefore, for an unweighted and undirected graph, the value of the element D_{ii} is equal to the number of edges connected to the $i-$th vertex.

A graph can also be represented by the graph Laplacian, which combines the adjacency matrix and the degree matrix, given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The graph Laplacian matrix carries more physical interpretation than the corresponding adjacency matrix due to its intrinsic connection to the number of disjoint components or subgraphs within the graph. The Laplacian matrix is defined in such a way that the sum of elements in each row (column) is zero. As a consequence, this enforces the inner products of every row of $\mathbf{L}$ with any constant vector to be zero. This means that at least one eigenvalue of the Laplacian is zero, $\lambda_0 = 0$. The multiplicity of the eigenvalue $\lambda_0 = 0$ of the Laplacian is equal to the number of connected subgraphs in the corresponding graph. This property follows from the fact that the Laplacian matrix of disconnected graphs can be written in a block diagonal form and each subgraph of a disconnected graph behaves as an independent graph and has a $\lambda_0 = 0$. This property does not hold for the adjacency matrix.

The normalized Laplacian matrix $\mathbf{L}_n$ is defined as $\mathbf{L}_n = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-1/2} = \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ $= \mathbf{I}_N - \mathbf{A}_n$, where $\mathbf{I}_N$ is the identity matrix of size N and $\mathbf{A}_n$ is the normalized adjacency matrix. The spectrum of $\mathbf{L}_n$ is composed of the diagonal matrix of the eigenvalues, $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_N)$ with $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$, and the eigenvector matrix $\mathbf{U} = [u_1|u_2|\ldots|u_N]$ such that $\mathbf{L_n} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$

[57].

1.2 Graph Signal Processing

Graph signal processing (GSP) extends traditional signal processing tools to deal with data defined on graph or network structures. In GSP, a graph signal is defined as a numerical or informational quantity associated with each node (or vertex), mathematically described by a function $f : V \rightarrow \mathbb{R}$. The graph signal can be represented as a vector $\mathbf{f} \in \mathbb{R}^N$ where f_i is the signal value on node i. If for each node the dimension of the signal, i.e., attributes, is P, the graph signal can be represented as a matrix $\mathbf{F} \in \mathbb{R}^{N \times P}$. The graph signal value can be continuous or discrete, and it can convey various types of information depending on the application. Examples include temperature readings from sensor nodes, pixel values in an image graph, or feature vectors representing individuals in a social network. Graph signal processing techniques are applied to analyze, filter, transform, or extract meaningful information from these graph signals. GSP methods leverage the graph's connectivity structure to process signals in a way that takes into account the relationships between nodes.

Akin to conventional signal processing, a graph signal can be studied using its Fourier domain representation, which can be derived using the graph shift operator (GSO). A GSO is an $N \times N$ dimensional matrix representing the structure of the graph, such as the adjacency, Laplacian or normalized Laplacian matrices [224]. In this work, the latter is employed as the GSO and its eigenvectors and eigenvalues provide a similar notion of frequency as that in the classical Fourier domain. For connected graphs, the Laplacian eigenvector $\mathbf{u}_0$ associated with the eigenvalue $\lambda_0 = 0$ is constant and equal to $\frac{1}{\sqrt{N}}$ at each vertex. The graph Laplacian eigenvectors associated with low frequencies vary slowly across the graph and the values of the eigenvector at those locations are likely to be similar. The eigenvectors associated with larger eigenvalues oscillate more rapidly and are more likely to have dissimilar values on vertices connected by a high-weight edge. These eigenvectors and eigenvalues are used to define Graph Fourier Transform (GFT) of $\mathbf{f}$ as $\widehat{\mathbf{f}} = \mathbf{U}^\top \mathbf{f}$ where $\hat{f}_i$ is the Fourier coefficient at the ith frequency component Λ_{ii}. When the dimension of the signals is P, i.e., $\mathbf{F} \in \mathbb{R}^{N \times P}$ with $\mathbf{F}_{\cdot p} = \mathbf{f}_p$, GFT can be computed as $\widehat{\mathbf{F}} = \mathbf{U}^\top \mathbf{F}$.

GFT can be utilized to define a notion of signal variability over the graph such that $\mathbf{F}$ is a smooth graph signal, i.e., $\mathbf{F}$ has low variation over the graph if most of the energy of $\widehat{\mathbf{F}}$ lies in the low-frequency components. The smoothness of $\mathbf{F}$ can then be calculated using the total variation of $\mathbf{F}$ measured in terms of the spectral density of its Fourier transform as:

$$\text{tr}(\widehat{\mathbf{F}}^\top \mathbf{\Lambda} \widehat{\mathbf{F}}) = \text{tr}(\mathbf{F}^\top \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \mathbf{F}) = \text{tr}(\mathbf{F}^\top \mathbf{L}_n \mathbf{F}). \tag{1.1}$$

The quadratic term $\text{tr}(\mathbf{F}^\top \mathbf{L}_n \mathbf{F})$ on the right-hand side of (1.1) is equal to $\sum_{i \neq j} A_{ij} \left(\frac{\mathbf{F}_{i\cdot}}{\sqrt{d_i}} - \frac{\mathbf{F}_{j\cdot}}{\sqrt{d_j}} \right)^2$, whose smaller values indicate that the graph signal is smooth. In particular, for a smooth signal $\mathbf{F}$, signal values on strongly connected nodes, i.e., large A_{ij}, are similar to each other.

1.2.1 Graph Filtering

A (convolutional) graph filter is an information processing unit that preserves specific properties of its input graph signal $\mathbf{f}$ by applying a shift-and-sum operation on $\mathbf{f}$ [121]. Shifting $\mathbf{f}$ requires propagating information in $\mathbf{f}$ across the graph topology, which can be done by a linear transformation of $\mathbf{f}$ with the GSO, e.g., $\mathbf{L}_n \mathbf{f}$, when the GSO is the normalized Laplacian. A graph filter of order T is then defined as:

$$\mathcal{H}(\mathbf{L}_n) = \sum_{t=0}^{T-1} h_t \mathbf{L}_n^t, \tag{1.2}$$

where $\mathbf{h} = [h_0, \ldots, h_{T-1}]^\top$ is the vector of filter coefficients and $\mathbf{L}_n^t$ is the tth power of the normalized Laplacian representing t-times shift operation. In the graph Fourier domain, the filtering operation can be interpreted as preserving the spectral content that is relevant to the task at hand. Namely, for a graph signal $\mathbf{f}$, let $\tilde{\mathbf{f}} = \mathcal{H}(\mathbf{L}_n)\mathbf{f}$. In graph Fourier domain:

$$\tilde{\mathbf{f}} = \mathbf{U} \mathcal{H}(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{f} = \mathbf{U} \mathcal{H}(\mathbf{\Lambda}) \widehat{\mathbf{f}} = \mathbf{U} \widehat{\mathbf{f}}_o, \tag{1.3}$$

where $\widehat{\mathbf{f}}_o$ is the GFT of $\mathbf{f}$ after being filtered by $\mathcal{H}(\mathbf{\Lambda})$, and whose ith Fourier coefficient is:

$$[\hat{\mathbf{f}}_o]_i = \mathcal{H}(\Lambda_{ii}) \hat{f}_i = \sum_{t=0}^{T-1} h_t \Lambda_{ii}^t \hat{f}_i. \tag{1.4}$$

Depending on the values of $\mathbf{h}$, one can attenuate or amplify specific spectral components, thus, yielding an output graph signal $\tilde{\mathbf{f}}$ that has a desired Fourier representation. For example, a smooth $\tilde{\mathbf{f}}$

can be obtained by filtering out high-frequency components while amplifying low-frequency ones. This concept of graph filtering can be extended to multi-dimensional graph signals, $\mathbf{F} \in \mathbb{R}^{N \times P}$, such that the filtered signal is obtained as $\tilde{\mathbf{F}} = \mathbf{U}\mathcal{H}(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{F}$.

1.3 Multiplex Networks

Multiplex networks are complex systems that consist of multiple layers, each representing different types of relationships or interactions among entities. The multiplex networks are used in a broad of applications such as social networks, transportation networks, and biological ones [229, 7]. For example, in social media, a multiplex network can represent a user's interactions across multiple platforms, such as Facebook, Twitter, and LinkedIn. Each layer corresponds to a different social media platform, as seen in Figure 1.1, and nodes represent users. Edges within each layer represent interactions like friendships, follows, or messages. In transportation networks, a multiplex graph can model various modes of transportation, including road networks, railway systems, and air travel. Each layer represents a different mode of transport, with nodes representing locations (cities or transportation hubs) and edges denoting transportation routes.

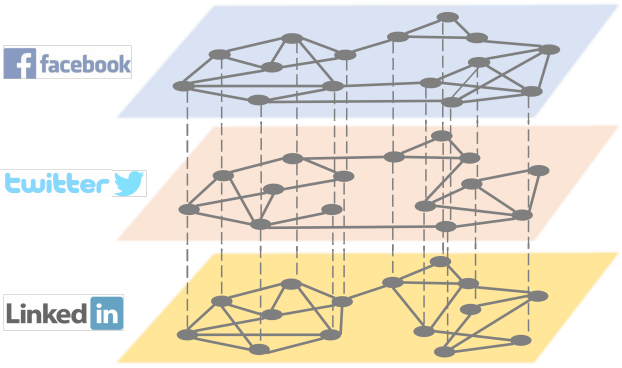


Figure 1.1: Example of a Multiplex Social Network.

Multiplex networks can be represented using a finite sequence of graphs $\{\mathcal{G}_l\}$, where $l \in \{1, 2, \ldots, L\}$, $\mathcal{G}_l = (V, E_l, \mathbf{A}_l)$ [62]. V is the set of vertices which is the same for all layers l, E_l and $\mathbf{A}_l \in \mathbb{R}^{n \times n}$ are the set of edges and the adjacency matrix for layer l, respectively.

1.4 Community Detection

One of the most fundamental tasks in analyzing large-scale networks is community detection. A community is a dense subnetwork within a larger network, within which nodes are connected more densely among themselves than to those outside the community. Numerous methods have been proposed for detecting the community structure of networks in single layer networks [92]. Among these, the most commonly used ones are spectral clustering [155, 256], methods based on statistical inference [1], approaches based on optimization of a quality function, [51], and techniques based on network dynamics [213]. In addition to these classical approaches focusing on nonoverlapping community structure, extensions for overlapping communities have also been considered [86]. In this thesis, we focus on nonoverlapping community detection, which is the partitioning of a node set V as $C = \{C_1, \ldots, C_K\}$ where K is the number of communities.

1.4.1 Spectral Clustering

One of the most popular algorithms for partitioning graphs is the minimum cut method, where the network is partitioned such that the number of edges between different communities is minimized. Given a single layer graph, $\mathcal{G} = \{V, E, \mathbf{A}\}$ and K clusters, minimizing the cut consists of finding a partition $\{C_1, C_2, \ldots, C_K\}$ that minimizes:

$$\text{Cut}(C_1, C_2, \ldots, C_K) = \frac{1}{2} \sum_{k=1}^{K} \text{links}(C_k, V \setminus C_k), \tag{1.5}$$

where $\text{links}(C_k, V \setminus C_k) = \sum_{i \in C_k, j \notin C_k} A_{ij}$. In practice, minimizing the cut results in clusters that are unbalanced. In order to address this issue, different variations of the cut definition, e.g., Ratio Cut (RCut), Normalized Cut (NCut), and MinMax Cut [74, 258], have been proposed.

The Normalized Cut is defined as:

$$\text{NCut}(C_1, C_2, \ldots, C_K) = \sum_{k=1}^{K} \frac{\text{links}(C_k, V \setminus C_k)}{\text{vol}(C_k)}, \tag{1.6}$$

where $\text{vol}(C_k)$ is the total degree of all nodes in C_k.

Similar to minimizing the cut metrics, one can determine the partition by maximizing the

corresponding association metrics [73]. For example, the normalized association is defined as:

$$\text{NAssoc}(C_1, C_2, \ldots, C_K) = \sum_{k=1}^{K} \frac{\text{links}(C_k, C_k)}{\text{vol}(C_k)}. \tag{1.7}$$

Minimizing or maximizing these cost functions are NP hard. However, it has been shown [74, 258] that spectral clustering and nonnegative matrix factorization provide solutions to relaxed versions of these problems. Introducing a community assignment matrix $\mathbf{Z} \in \mathbb{R}^{N\times K}$, the normalized cut can be rewritten as $\sum_{k=1}^{K} \frac{\text{links}(C_k, V\setminus C_k)}{\text{vol}(C_k)} = \sum_{k=1}^{K} \frac{\mathbf{z}_k^\top(\mathbf{D}-\mathbf{A})\mathbf{z}_k}{\mathbf{z}_k^\top\mathbf{D}\mathbf{z}_k} = \sum_{k=1}^{K} \tilde{\mathbf{z}}_k^\top\mathbf{L}\tilde{\mathbf{z}}_k$ where $\tilde{\mathbf{z}}_k = \frac{\mathbf{z}_k}{(\mathbf{z}_k^\top\mathbf{D}\mathbf{z}_k)^{1/2}}$. Relaxing the problem by allowing $\mathbf{z}_k$ to take arbitrary real values, the corresponding optimization problem becomes:

$$\min_{\bar{\mathbf{Z}}, \bar{\mathbf{Z}}^\top\bar{\mathbf{Z}}=\mathbf{I}} \text{tr}(\bar{\mathbf{Z}}^\top\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\bar{\mathbf{Z}}), \tag{1.8}$$

where $\bar{\mathbf{Z}} = \mathbf{D}^{1/2}\tilde{\mathbf{Z}}$. This is the standard form of a trace minimization problem, and the Rayleigh-Ritz theorem states that the solution is given by choosing $\bar{\mathbf{Z}}$ as the matrix containing the K eigenvectors corresponding to the smallest eigenvalues of the normalized Laplacian $\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ as columns.

Minimizing the NCut can equivalently be written in terms of maximizing the NAssoc written in trace form:

$$\max_{\bar{\mathbf{Z}}, \bar{\mathbf{Z}}^\top\bar{\mathbf{Z}}=\mathbf{I}} \text{tr}(\bar{\mathbf{Z}}^\top\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\bar{\mathbf{Z}}), \tag{1.9}$$

where $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the normalized adjacency matrix, which will be denoted by $\mathbf{A}_n$ in the remainder of the thesis. In both of these formulations, the eigenvectors encode the low-dimensional embeddings of the graph's nodes and k-means algorithm is applied to determine the communities.

Authors in [74] show that NMF is equivalent to Laplacian based spectral clustering, and that Normalized Cut using the normalized adjacency matrix, $\mathbf{A}_n = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, is equivalent to the nonnegative matrix factorization problem $\underset{\mathbf{H}\geq 0}{\text{argmin}} ||\mathbf{A}_n - \mathbf{H}\mathbf{H}^\top||_F^2$.

1.4.2 Non-negative Matrix Factorization

Methods based on Nonnegative matrix factorization (NMF) and its variants have been popular for community detection [111]. Compared to other community detection methods, these methods have some unique advantages including high interpretability, and applicability to general complex networks including directed, temporal and multilayer networks [111].

Nonnegative Matrix Factorization decomposes a nonnegative matrix $\mathbf{W} \in \mathbb{R}^{N\times M}$ into the product of two low-rank nonnegative matrices $\mathbf{V} \in \mathbb{R}^{N\times k}$ and $\mathbf{U} \in \mathbb{R}^{M\times k}$, such that $\mathbf{W} \approx \mathbf{V}\mathbf{U}^\top$ and $k \ll N, M$. $\mathbf{V}$ and $\mathbf{U}$ are found by solving the optimization problem

$$\underset{\mathbf{V},\mathbf{U}>0}{\operatorname{argmin}} ||\mathbf{W} - \mathbf{V}\mathbf{U}^\top||_F^2. \tag{1.10}$$

Compared with other types of matrix factorization techniques, NMF is more suitable for the task of community detection. This is because it has two unique capabilities. One is the potential clustering capability possessed by NMF. In [74], NMF and its extensions are proved to have equivalent relationships with some classical clustering models. For example, if $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$, then NMF becomes equivalent to k-means clustering. If $\mathbf{W}$ is symmetric, then the transformation can be written as $\mathbf{W} \approx \mathbf{U}\mathbf{U}^\top$, and NMF becomes equivalent to spectral clustering. The other aspect is the generative capability of NMF that can give a good interpretation to community structure [206].

In NMF-based community detection, $\mathbf{V}$ and $\mathbf{U}$ are the community feature matrix and the community indicator matrix, respectively. The conventional NMF model, $\mathbf{W} \approx \mathbf{V}\mathbf{U}^\top$ can be directly used to detect communities. However, it cannot model the interactions among communities, which is useful to determine whether communities are overlapping. Nonnegative matrix tri-factorization (NMTF) was introduced to address this issue, where $\mathbf{W} \approx \mathbf{U}\mathbf{S}\mathbf{U}^\top$, with $\mathbf{S}$ modeling the interactions among communities.

Another variant of NMF, the orthogonal NMF (ONMF) imposes an additional orthogonality constraint on one of the factor matrices and improves the performance as orthogonality and non-negativity force each row of $\mathbf{V}$ ($\mathbf{U}$) to have only one nonzero element which implies that each node belongs only to one community. Like the k-means, the orthogonally constrained factor matrix functions the same as an indicator matrix that shows how the data samples are assigned to different clusters [75, 201]. Therefore, the ONMF model can be regarded as a continuous relaxation of k-means [262]. ONMF has been broadly used in community detection and has been shown to outperform k-means and NMF based clustering [264, 167]. In this thesis, we use Orthogonal Nonnegative Matrix Tri-factorization (ONMTF) which combines the advantages of orthogonality

in ONMF and the degrees of freedom provided by NMTF to formulate the community detection problem in multiplex networks in Chapter 2.

1.5 Organization and Contributions

In this thesis, we developed algorithms for two important tasks in network analysis: community detection and hub node identification. Although there are numerous works on community detection in single-layer networks, existing work on multiplex community detection mostly focuses on learning a common community structure across layers without taking the heterogeneity of the different layers into account. Beyond detecting communities within a single multiplex network, in a lot of applications, one may be interested in comparing the community structure of two multiplex networks. For instance, in settings where we have multiplex networks constructed from different conditions, e.g., a treatment and a control experiment, we are interested in visualizing and exploring communities that are specific to one multiplex network or communities that discriminate between the two groups. Furthermore, most of the existing community detection methods focus solely on the graph connectivity information. In attributed graphs, where in addition to connectivity information, nodes have associated features or signals, classical clustering methods focus on detecting communities using the attributes of the nodes [123], ignoring the relationships between the nodes, while community detection methods focus only on the topology of the network [248, 182]. However, these methods usually fall short in attributed graph clustering, as they do not exploit informative node features such as user profiles in social networks and document contents in citation networks. In the case of identifying hubs, traditional hub detection methods rely on graph connectivity, e.g., degree, betweenness, and eigenvector centrality, without considering the node attributes. Specifically, in brain networks, this corresponds to considering only the functional connectivity network without considering the coupling between the brain's anatomical wiring, i.e., structural network, and its dynamic functional properties, e.g., the BOLD signal.

In Chapter 2, we introduce a new multiplex community detection method that identifies communities that are common across layers as well as those that are unique to each layer. The proposed method, Multiplex Orthogonal Nonnegative Matrix Tri-Factorization (MX-ONMTF), represents

the adjacency matrix of each layer as the sum of two low-rank matrix factorizations corresponding to the common and private communities, respectively. Unlike most of the existing methods, which require the number of communities to be pre-determined, the proposed method also introduces a two stage method to determine the number of common and private communities. The proposed algorithm is based on multiplicative update rules and a proof of convergence is provided. Additionally, we present an in-depth analysis of the algorithm, including studies of overfitting and ablation, recovery guarantees, and consistency. MX-ONMTF is evaluated on synthetic and real multiplex networks, as well as for multiview clustering applications, and compared to state-of-the-art techniques. In addition, MX-ONMTF is applied to functional connectivity networks that are extracted from multi-subject resting-state fMRI data to identify subgroups of subjects that exhibit significant differences in key functional areas of the brain.

In Chapter 3, we introduce a discriminative community detection approach based on spectral clustering for detecting community structures that distinguish between two multiplex networks in both static and dynamic settings. In particular, we introduce three different formulations: the first approach finds discriminative subspaces between two multiplex networks; the second method offers a more comprehensive approach where the consensus, discriminative and individual layerwise subspaces are learned simultaneously across the two groups; and the third method learns the discriminative subgraphs between two temporal multiplex networks. These methods are evaluated on synthetic and real networks, including EEG and dynamic fMRI functional brain networks, comparing across experimental conditions and tasks.

In Chapter 4, we present two graph signal processing based frameworks for community detection in attributed networks, Graph Filtering for Clustering Attributed Graphs (GraFiCA) and Multi-Scale Graph Wavelets for Clustering (MSGWC). A cost function quantifying the separability of filtered attributes is proposed, along with a general framework for learning optimal graph filters. For GraFiCA, the parameters of Finite Impulse Response (FIR) and Autoregressive Moving Average (ARMA) filters are learned. While for MSGWC, we learn the optimal combination of the multi-scale features from graph spectral wavelet and scaling filters. The proposed methods are evaluated

on real-world attributed networks with both binary and numerical attributes and compared to the state-of-the-art graph clustering algorithms. GraFiCA is also extended to multiplex networks and evaluated on an EEG brain network dataset.

In Chapter 5, we introduce a GSP-based framework for hub node identification in brain networks utilizing both the structural connectome and functional BOLD signals. The proposed approach is based on learning the optimal graph filter for detecting hub nodes with the following assumptions: (i) hub nodes are sparse and have high activation patterns simultaneously with a more diverse set of connections, i.e., their activity corresponds to the high-frequency component of the BOLD signal, and (ii) the non-hub nodes' activation patterns are low-frequency/smooth with respect to the structural connectome, thus can be modeled as the output of a graph diffusion kernel, e.g., polynomial graph filter. The proposed method is evaluated on both simulated data and rs-fMRI data from HCP. The results are compared to the state-of-the-art hub node identification methods and recently published meta-analysis of hub nodes in rs-fMRI [271].

Finally, conclusions and future directions are presented in Chapter 6, where we discuss extensions of the presented methods.

CHAPTER 2

COMMUNITY DETECTION IN MULTIPLEX NETWORKS

2.1 Introduction

Community detection methods for multiplex networks [170] can be grouped into three main classes. The first class of methods merges the layers in a multiplex network, using a flattening algorithm, then apply single-layer community detection to the aggregated network [23, 52, 245]. While these methods are computationally efficient, they can only identify communities that are common across all layers. Moreover, due to the flattening process, some spurious communities may emerge. The second class of methods applies community detection to each layer individually and then merges the results [24, 243, 77]. These methods include nodes in the same community only when they are part of the same community in at least one layer. Finally, the third class of methods operates directly on the multiplex network model [145, 69, 182, 11, 290, 193].

Existing multiplex community detection approaches typically assume that the community structure is the same across layers and find the partition that best fits all layers. Thus, they do not differentiate between communities that are common across layers from those that are unique to each layer. This is particularly important for real world applications where the networks are heterogeneous, and the different layers correspond to different modes of interaction. For example, in social networks, a group of individuals may be well connected via friendships on Facebook; however, this group of individuals will likely not work at the same company. Thus, in a situation like this, a given community will only be present in a subset of the layers, and different communities may be present in different subsets of layers.

In this work, common communities are defined as communities that are observed in more than one layer, i.e., communities that are common across any subset of two or more layers, and private communities as communities that are unique to each layer. The problem of detecting common and private communities is then formulated using a novel framework titled Multiplex Orthogonal Nonnegative Matrix Trifactorization (MX-ONMTF). In the proposed framework, each layer's adjacency matrix is represented as the sum of two low-rank matrix factorizations corresponding to

the common and private communities, respectively. The resulting optimization problem is solved using an iterative multiplicative update algorithm. The proposed approach also addresses the problem of determining the number of communities. Unlike most existing work, where the number of communities is determined through a greedy search, in this work, a two-step approach is proposed. The proposed algorithm is first evaluated on synthetic benchmark multiplex networks with different numbers of layers, nodes, communities, noise levels, and inter-layer dependency probability. Next, the proposed method is applied to real networks including social and biological networks. Finally, the algorithm is evaluated for multiview clustering task, where the communities across all layers are assumed to be the same.

The rest of the chapter is organized as follows. Section 2.2 presents a summary of related works. Sections 2.3 and 2.4 present the proposed multiplex community detection algorithm and its convergence analysis. Section 2.5 establishes the theoretical properties of the algorithm, while Section 2.6 illustrates results on both simulated and real networks. Section 2.7 presents an application of our method to functional connectivity networks that are extracted from multi-subject resting-state fMRI data. Finally, Section 2.8 provides conclusions and discussion on future work.

2.2 Related works

The method proposed in this chapter belongs to the third class of algorithms, which operate directly on the multiplex network model. There are different types of algorithms that fall in this class: random walk, statistical generative network models, label propagation, objective function optimization, and Nonnegative Matrix Factorization (NMF).

Methods based on random walkers model the dynamic process on networks as random walks where the process is more likely to persist on the vertices in the same community and far less on the vertices in different communities. For instance, LART [145] is initialized by assigning each node in each layer to its own community. Hierarchical clustering is then used to merge nodes based on a distance matrix, and the partition with the highest multiplex modularity is chosen. In [69], Infomap, which is based on a compression of network flows, is proposed to identify communities within and across layers. However, Infomap tends to assign each physical node across layers to the

same community, not differentiating the topological differences across layers.

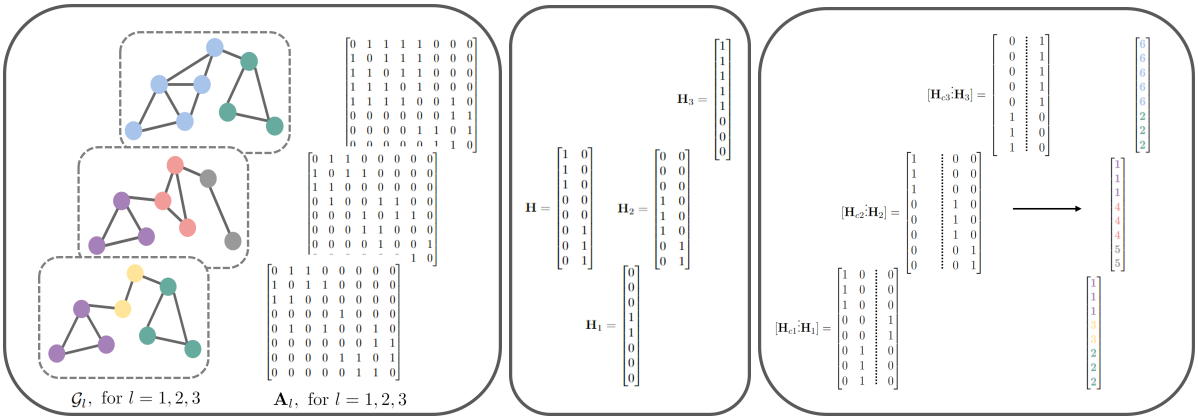
Statistical methods use variants of Stochastic Block Model (SBM) to model the latent variables. Among these, multilayer stochastic block model (MLSBM) [197, 25, 110, 149, 36] model each layer's adjacency matrix through a common community membership matrix, $\mathbf{Z}$, and a layer specific connectivity probability matrix $\mathbf{B}^m$, where the goal is to infer $\mathbf{Z}$. Consistency properties of various methods such as orthogonal linked matrix factorization (OLMF) [244, 197, 77] and spectral clustering under MLSBM have been investigated [197, 25, 110]. More recently, mixture multilayer stochastic block model (MMLSBM) has been proposed to model the heterogeneity of multiplex networks [128, 89, 241, 233]. MMLSBM assumes that there is a mixture of m latent network models and each network is sampled independently from this mixture of models with each of the m classes following SBM. While these methods provide some flexibility in modeling heterogeneous networks, they still make assumptions such that the layers can be clustered into subgroups where each subgroup has exactly the same community structure and connectivity. Similarly, Weighted Stochastic Block Model (WSBM) [8] has been proposed to detect common and private communities in heterogeneous weighted networks. Although this method addresses the heterogeneity of networks across layers, the method is limited to detecting only common communities that are shared by all layers, ignoring communities that may be shared by only a subset or different subsets of layers. In [66], authors propose a generative model and an expectation maximization algorithm for community detection and link prediction in multilayer networks. Although the method allows for different connectivity patterns in each layer, the interdependence between layers is only taken into account for link prediction, while the layers are assumed to share a common community structure.

The third class of methods, Label Propagation Algorithms (LPA), are based on the intuition that a label can become dominant in a densely connected group of nodes but will have trouble crossing a sparsely connected region. In [32], an LPA-based method for community detection in multidimensional networks is proposed to identify communities and the subset of layers in which each of these communities is observed, simultaneously. However, this algorithm fails to detect communities that are private to each layer and communities that may be common among a small

number of layers.

The fourth type of multiplex community detection methods is based on defining an objective function and identifying the community structure that maximizes/minimizes the objective function. For example, Generalized Louvain (GenLouvain) [182] uses an extended definition of modularity and is one of the fastest methods for community detection in multiplex networks. As GenLouvain assigns each node-layer tuple to its own community, it cannot identify common communities across layers. More recently, multiobjective genetic and evolutionary algorithms such as MultiMOGA [11] and MOEA/D-TS [135] have been used to jointly maximize the modularity of each layer and the similarity between the community structures across layers. These methods find a shared community structure across all layers, not differentiating communities that may be unique to each layer. In [49], extension of normalized cut to multiplex networks is proposed by constructing a block Laplacian matrix with each block corresponding to a layer. This method relies on selecting a parameter β that controls the consistency of the community structure across different layers.

The last class of methods is based on NMF which, because of its interpretability and good performance, has been broadly used for community detection in single-layer, multiplex, multilayer, and dynamic networks [74, 260, 172, 237]. In [169], Semi-Supervised joint Nonnegative Matrix Factorization (S2-jNMF) is proposed for detecting the common communities across layers in a multiplex network. A greedy search of dense subgraphs is performed and these subgraphs are used as *a priori* information to create new adjacency matrices for each layer. In [101], a two-step approach is proposed, where first a nonnegative low dimensional feature representation of each layer is found using one of the four different NMF models. These community structures are then used to obtain a consensus community structure. Authors in [187] use NMF for detecting communities in multiplex social networks, where both unifying and coupling approaches are proposed. The unifying approach finds a common community structure by aggregating all layers, while the coupling approach finds mostly consistent community structures. Most of the aforementioned NMF based methods find a common structure across all layers or for a majority of layers and do not consider cases where common communities may be present in different subsets of layers. Moreover, they do not detect



(a) (b) (c)

Figure 2.1: Illustration of the proposed community detection algorithm. (a) Toy example of a multiplex network with 3 layers with 8 nodes, and 2 common communities (purple and green) across different pair of layers. (b) Community membership matrices obtained from MX-ONMTF cost function. (c) Post-processing described in Algorithm 2.3 applied to $\mathbf{H}$ to detect which of the common communities are present in each layer, and final community labels.

private communities. These methods also require that the number of communities is provided *a priori*.

2.3 Proposed Method (MX-ONMTF)

The proposed method, MX-ONMTF, models each layer's adjacency matrix as a sum of low-rank representations of common and private communities using Orthogonal Nonnegative Matrix Tri-Factorization (ONMTF). Figure 2.1 illustrates the overview of the proposed algorithm for a multiplex network with three layers and two common communities. In this example, there are two common communities that do not exist in all layers. The purple community is common across layers 1 and 3, while the green one is common across layers 1 and 3. Figure 2.1b shows the low-rank representations corresponding to the common and private communities and 2.1c shows the post-processing applied to $\mathbf{H}$ in order to determine which layers each common community is present in.

2.3.1 Problem Formulation

In this work, we define common communities as communities that are observed in more than one layer.

Definition 1: An ideal common community C in a multiplex network $\{\mathcal{G}_l\}$ is defined as a subgraph

with the same set of nodes for a subset of layers $\mathbf{m} \subseteq \{1, 2, \ldots, L\}$, where $|\mathbf{m}| > 1$. Mathematically, C can be defined as

$$C = \{(V_l^C, E_l^C) : V_l^C \subseteq V_l, E_l^C \subseteq V_l^C \times V_l^C, V_l^C = V_k^C, \text{ with } l, k \in \mathbf{m}, \mathbf{m} \subseteq \{1, 2, \ldots, L\}, |\mathbf{m}| > 1\}.$$

Definition 2: A private community C in a multiplex network $\{\mathcal{G}_l\}$ is defined as any community that is not common across at least two layers.

For a multiplex network with L layers and adjacency matrices, $\mathbf{A}_l \in \mathbb{R}^{N \times N}$, $l \in \{1, 2, \ldots, L\}$, we model each layer's adjacency matrix in terms of common and individual communities using ONMTF. The resulting objective function can be formulated as

$$\begin{aligned} &\underset{\mathbf{H} \geq 0, \mathbf{H}_l \geq 0, \mathbf{S}_l \geq 0, \mathbf{G}_l \geq 0}{\operatorname{argmin}} \sum_{l=1}^{L} ||\mathbf{A}_l - \mathbf{H}\mathbf{S}_l\mathbf{H}^\top - \mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top||_F^2 \\ &\text{s.t } \mathbf{H}^\top\mathbf{H} = \mathbf{I}, \mathbf{H}_l^\top\mathbf{H}_l = \mathbf{I}, \text{with } \; l \in \{1, 2, \ldots, L\}, \end{aligned} \tag{2.1}$$

where $\mathbf{H} \in \mathbb{R}^{N \times k_c}$ and $\mathbf{H}_l \in \mathbb{R}^{N \times k_{p_l}}, l \in \{1, 2, \ldots, L\}$ are the community membership matrices corresponding to the common and private communities, respectively, and $\mathbf{S}_l$ and $\mathbf{G}_l$ are diagonal matrices whose entries indicate the strength of the common and private communities across layers, respectively. In this work, it is assumed that the L layers have a total of k_c common communities and k_{p_l} private communities in each layer l. The goal is to simultaneously identify communities that are common across any subset of two or more layers and communities that are unique to each layer. Therefore, $\mathbf{H}$ will contain information for all common communities.

2.3.2 Optimization solution

ONMTF optimization problem in (2.1) can be solved using a multiplicative update algorithm (MUA) [75]. Multiplicative update algorithms for solving NMF problems were introduced in [148], while solving NMTF with orthogonal constraints was first addressed by [75]. In this work, we follow their approach to derive the multiplicative update rules for each variable.

To find the update rules for $\mathbf{H}$, $\mathbf{H}_l$, $\mathbf{S}_l$, and $\mathbf{G}_l$, the following Lagrangian function with Lagrange multipliers $\mathbf{\Lambda}$ and $\mathbf{\Lambda}_l$ is minimized:

$$\mathcal{L}(\mathbf{H}, \mathbf{H}_l, \mathbf{S}_l, \mathbf{G}_l) = \sum_{l=1}^{L} ||\mathbf{A}_l - \mathbf{H}\mathbf{S}_l\mathbf{H}^\top - \mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top||_F^2 + \text{tr}(\mathbf{\Lambda}(\mathbf{H}^\top\mathbf{H} - \mathbf{I})) + \sum_{l=1}^{L} \text{tr}(\mathbf{\Lambda}_l(\mathbf{H}_l^\top\mathbf{H}_l - \mathbf{I})). \tag{2.2}$$

For updating $\mathbf{H}$, we find $\nabla_{\mathbf{H}}\mathcal{L}$ as

$$\nabla_{\mathbf{H}}\mathcal{L} = \sum_{l=1}^{L}(\mathbf{4}\mathbf{H}\mathbf{S}_l^\top\mathbf{H}^\top\mathbf{H}\mathbf{S}_l + 4\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l - 4\mathbf{A}_l\mathbf{H}\mathbf{S}_l) + \mathbf{4}\mathbf{H}\mathbf{\Lambda}. \tag{2.3}$$

Setting $\nabla_{\mathbf{H}}\mathcal{L} = 0$ and $\nabla_{\mathbf{\Lambda}}\mathcal{L} = 0$, we obtain:

(i) $\mathbf{\Lambda} = \sum_{l=1}^{L}(-\mathbf{S}_l^\top\mathbf{S}_l - \mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l + \mathbf{H}^\top\mathbf{A}_l\mathbf{H}\mathbf{S}_l)$.

(ii) $\mathbf{H}^\top\mathbf{H} = \mathbf{I}$.

Substituting (i) and (ii) in Eq. (2.3), we get

$$\nabla_{\mathbf{H}}\mathcal{L} = \sum_{l=1}^{L}(\mathbf{4}\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l - 4\mathbf{A}_l\mathbf{H}\mathbf{S}_l + 4\mathbf{H}\mathbf{H}^\top\mathbf{A}_l\mathbf{H}\mathbf{S}_l - \mathbf{4}\mathbf{H}\mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l). \tag{2.4}$$

As discussed in [280], if the gradient of an error function, ε, is of the form $\nabla\varepsilon = \nabla\varepsilon^+ - \nabla\varepsilon^-$, where $\nabla\varepsilon^+ > \mathbf{0}$ and $\nabla\varepsilon^- > \mathbf{0}$, then the multiplicative update for parameter $\mathbf{\Theta}$ has the form $\mathbf{\Theta} = \mathbf{\Theta} \odot \frac{\nabla\varepsilon^-}{\nabla\varepsilon^+}$. It can be easily seen that the multiplicative update preserves the nonnegativity of $\mathbf{\Theta}$, while $\nabla\varepsilon = 0$ when the convergence is achieved. Following this procedure, from the gradient of the error function in Eq. (2.4), we derive the following multiplicative update rule for $\mathbf{H}$

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\sum_{l=1}^{L}(\mathbf{A}_l\mathbf{H}\mathbf{S}_l + \mathbf{H}\mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l)}{\sum_{l=1}^{L}(\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l + \mathbf{H}\mathbf{H}^\top\mathbf{A}_l\mathbf{H}\mathbf{S}_l)}, \tag{2.5}$$

where the multiplication and division are performed element-wise and both numerator and denominator are positive. Note that the update in Eq. (2.5) satisfies the Karush-Kuhn Tucker (KKT) complementary slackness condition for nonnegativity of $\mathbf{H}$, $\nabla_{\mathbf{H}}\mathcal{L}_{ij}\mathbf{H}_{ij} = 0$, given as

$$\left(\sum_{l=1}^{L}(\mathbf{4}\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l - 4\mathbf{A}_l\mathbf{H}\mathbf{S}_l + 4\mathbf{H}\mathbf{H}^\top\mathbf{A}_l\mathbf{H}\mathbf{S}_l - \mathbf{4}\mathbf{H}\mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l)\right)_{ij}\mathbf{H}_{ij} = 0. \tag{2.6}$$

This is the fixed point condition that any local minima $\mathbf{H}^*$ must satisfy. This shows that if the update rule (2.5) converges, the converged solution is a local minimum of the optimization problem.

Similarly, we obtain the following update rules for $\mathbf{H}_l$, $\mathbf{S}_l$, and $\mathbf{G}_l$, for each $l \in \{1, 2, \ldots, L\}$:

$$\mathbf{H}_l \leftarrow \mathbf{H}_l \odot \frac{\mathbf{A}_l\mathbf{H}_l\mathbf{G}_l + \mathbf{H}_l\mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l^\top\mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l}{\mathbf{H}\mathbf{S}_l^\top\mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l^\top + \mathbf{H}_l\mathbf{H}_l^\top\mathbf{A}_l\mathbf{H}_l\mathbf{G}_l}, \tag{2.7}$$

$$\mathbf{S}_l \leftarrow \mathbf{S}_l \odot \frac{\mathbf{H}^\top\mathbf{A}_l\mathbf{H}}{\mathbf{H}^\top\mathbf{H}\mathbf{S}_l\mathbf{H}^\top\mathbf{H} + \mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}}, \tag{2.8}$$

$$\mathbf{G}_l \leftarrow \mathbf{G}_l \odot \frac{\mathbf{H}_l^\top\mathbf{A}_l\mathbf{H}_l}{\mathbf{H}_l^\top\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}_l + \mathbf{H}_l^\top\mathbf{H}\mathbf{S}_l\mathbf{H}^\top\mathbf{H}_l}. \tag{2.9}$$

Since NMF algorithms are initialized with random matrices, different runs yield local minima. For this reason, we run the algorithm 50 times and report the best results [153, 168]. As shown in Algorithm 2.1, for each random initialization of $\mathbf{H}$, $\mathbf{H}_l$, $\mathbf{S}_l$, and $\mathbf{G}_l$, the multiplicative update rules described in Eqs. (2.5)-(2.9) are repeated for 1000 iterations or until convergence. We then select the solution that yields the maximum value of the performance metric across the different runs. For synthetic networks for which a ground truth is available, Normalized Mutual Information (NMI) [65] is used. For networks without ground truth, Modularity Density (Q_D) [157] is used as the performance metric.

2.3.3 Number of communities

In most NMF-based community detection algorithms, the number of communities (k) is an input parameter [111]. This problem is usually addressed by detecting communities with different values of k and selecting the one that gives the solution with the best pre-determined performance metric, such as modularity [203].

In this chapter, a two-step approach is proposed to determine the number of communities per layer and the number of common communities. First, the number of communities per layer (k_1, k_2,..., k_L) are found using the eigengap rule [160] which determines the number of communities by the value that maximizes the eigengap, i.e. the difference between consecutive eigenvalues. A suitable null model, e.g., Laplacianized Erdős–Rényi adjacency matrices $\mathbf{L}^{null}$ with size and density matching the Laplacian of the network, can be used . The threshold, δ, can be set to be the 0.95 quantile of the largest eigengap (see lines 1 to 4 of Algorithm 2.2).

Algorithm 2.1: MX-ONMTF.

Input: Adjacency matrices $\mathbf{A}_l$, $l \in \{1, 2, \ldots, L\}$.
Output: Community membership matrices $\mathbf{H}$, $\mathbf{H}_l$.

Use Algorithm 2.2 to find k_c, k_l, and k_{p_l}.
for r=1 to 50 **do**
    Randomly initialize $\mathbf{H}, \mathbf{H}_l, \mathbf{S}_l, \mathbf{G}_l > 0$
    for 1000 iterations or until convergence **do**
        update $\mathbf{H}$ according to Eq. (2.5)
        update $\mathbf{H}_l$ for each $l \in \{1, 2, \ldots, L\}$ according to Eq. (2.7)
        update $\mathbf{S}_l$ for each $l \in \{1, 2, \ldots, L\}$ according to Eq. (2.8)
        update $\mathbf{G}_l$ for each $l \in \{1, 2, \ldots, L\}$ according to Eq. (2.9)
    end for
    for each layer l **do**
        Apply Algorithm 2.3 with $\mathbf{A}_l$, $\mathbf{H}$, and k_{p_l} as inputs to find $\mathbf{H}_{c_l}$.
        for each i **do**
            $j^* \leftarrow argmax_j \mathbf{H}_{c_l}(i, j)$
            if $\mathbf{H}_{c_l}(i, j^*) > \mathbf{H}_l(i, j^*)$ **then**
                $idx(i) \leftarrow j^*$
            else
                $idx(i) \leftarrow (argmax_j \mathbf{H}_l(i, j)) + k_c + \sum_{n=1}^{l-1} k_{p_n}$
            end if
        end for
    end for
    Compute NMI_r or Q_{D_r}.
end for
Choose the partition $r^* = argmax_r \mathrm{NMI}_r$ ($r^* = argmax_r Q_{D_r}$).

Next, ONMTF is applied to each layer [75] and the low-rank embedding matrices, $\mathbf{U}_l \in \mathbb{R}^{N \times k_l}$, are obtained. Once we have $\mathbf{U}_l$ corresponding to each layer l, our goal is to reduce the embedding subspace by finding columns that are similar to each other, i.e., embedding vectors for the common communities. Each element of the embedding matrices, $\mathbf{U}_l(i, j)$, represents the likelihood of node i belonging to community j. An agglomerative hierarchical clustering algorithm using Euclidean distance is applied to the columns of $\mathbf{X} = [\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_L] \in \mathbb{R}^{N \times m}$, where $m = \sum_{l=1}^{L} k_l$, to obtain the number of common communities. At each step of the algorithm, the two columns with the smallest distance are aggregated, and the distances between the newly formed cluster and the remaining ones are updated. The assumption is that if two or more layers share a common

community, the columns of the respective $\mathbf{U}_l$'s corresponding to this community will be close to each other. A dendrogram like the one shown in Figure 2.2b can be used to represent the different iterations of this algorithm. The m leaves of the dendrogram correspond to the total number of communities across the L layers.

Algorithm 2.2: Finding k_l, k_c and k_{p_l}, for $l \in \{1, 2, \ldots, L\}$.

Input: Adjacency matrices $\mathbf{A}_l$ for $l \in \{1, 2, \ldots, L\}$.
Output: Number of common communities k_c, total number of communities per layer k_l, and number of private communities per layer k_{p_l}, for $l \in \{1, 2, \ldots, L\}$.

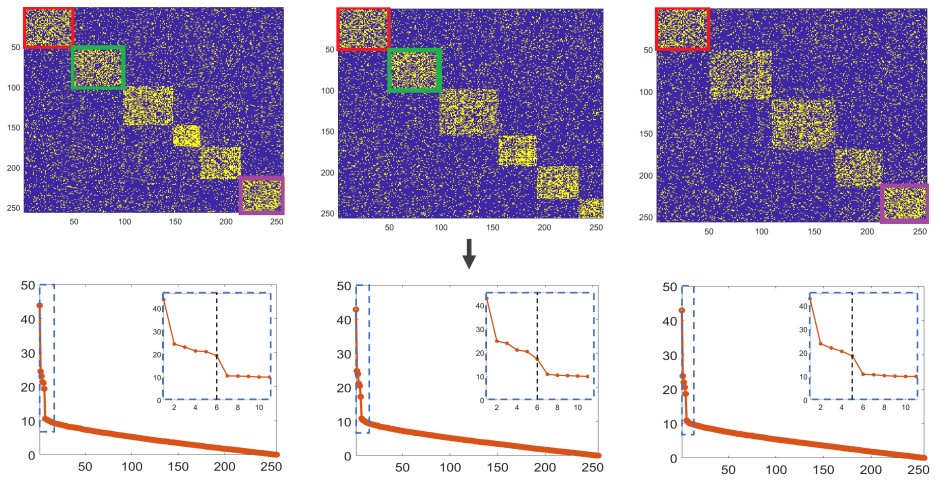
Let $\mathbf{L}_l^{null}$ be the normalized Laplacian of the Erdős–Rényi null model.
$\mathbf{L}_l^{null} = \mathbf{V}_l \mathbf{\Lambda}_l \mathbf{V}_l^T$
$\delta \leftarrow quantile_{0.95}[max\{|\lambda_i^{null}| - |\lambda_{i+1}^{null}|, i \geq 2\}]$
$k_l \leftarrow min\{k : |\lambda_i^l| - |\lambda_{i+1}^l| > \delta, \forall i > k\}$
Randomly initialize $\mathbf{U}_l \geq 0, \mathbf{S}_l \geq 0$.
for 1000 iterations or until convergence **do**
    update $\mathbf{U}_l$ using $\mathbf{U}_l = \mathbf{U}_l * \frac{(\mathbf{A}_l \mathbf{U}_l \mathbf{S}_l)_{ij}}{(\mathbf{U}_l \mathbf{U}_l^\top \mathbf{A}_l \mathbf{U}_l \mathbf{S}_l)_{ij}}$
    update $\mathbf{S}_l$ using $\mathbf{S}_l = \mathbf{S}_l * \frac{(\mathbf{U}_l^\top \mathbf{A}_l \mathbf{U}_l)_{ij}}{(\mathbf{U}_l^\top \mathbf{U}_l \mathbf{S}_l \mathbf{U}_l^\top \mathbf{U}_l)_{ij}}$
end for
$\mathbf{X} = [\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_L] \in \mathbb{R}^{N \times m}$, $m = \sum_{l=1}^{L} k_l$
$\mathbf{F} \leftarrow AgglomerativeHierarchicalClustering(\mathbf{X})$
$k_c = 0$
for $i = 2$ to m **do**
    $d_i \leftarrow \frac{\mathbf{F}(i,3) - \mathbf{F}(i-1,3)}{\mathbf{F}(i-1,3)}$
    if $d_i \geq 0.5$ **then**
        if $max\{\mathbf{F}(i,1), \mathbf{F}(i,2)\} \leq m$ **then**
            $k_c \leftarrow k_c + 1$
        else
            $k_c \leftarrow k_c$
        end if
    else
        $cut \leftarrow i$ and stop **for**
    end if
end for
$C \leftarrow$find$(1 + \sum_{j=1}^{l-1} k_j \leq \mathbf{F}(1 : cut, 1 : 2) \leq \sum_{j=1}^{l} k_j)$
$k_{p_l} \leftarrow k_l - |C|$, for $l \in \{1, 2, \ldots, L\}$

This agglomerative hierarchical clustering algorithm outputs a matrix $\mathbf{F} \in \mathbb{R}^{m-1 \times 3}$. The first two columns of $\mathbf{F}(i, :)$ correspond to the labels of the two leaves of the dendrogram that form cluster

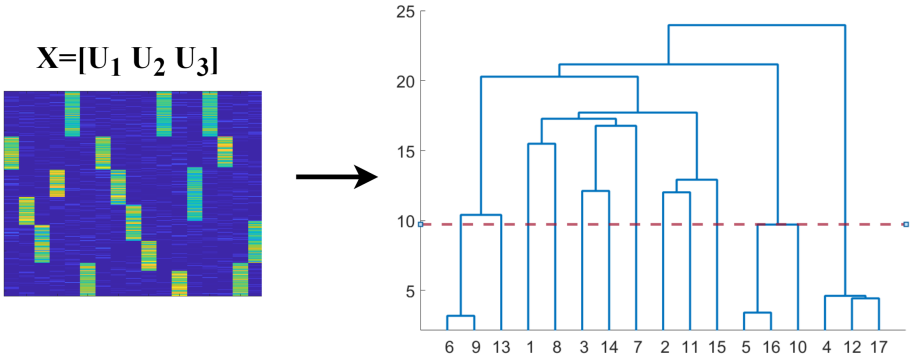
$m+i$, and the third column contains the distance between these two leaves. The number of clusters resulting from this procedure correspond to k_c, while the columns of $\mathbf{X}$ corresponding to each layer l that are not assigned to any of the clusters correspond to k_{p_l} (see lines 13 to 26 in Algorithm 2.2). The algorithm iterates until the minimum distance between any two clusters increases by more than 50% of the minimum distance from the previous iteration. Figure 2.2 illustrates how the number of communities per layer and the common communities are obtained for an example of a 3-layer network with $k_1 = 6$, $k_2 = 6$, and $k_3 = 5$, and three common communities highlighted in red, green, and purple. For this example, $k_c = 3$, $k_{p_1} = 3$, $k_{p_2} = 4$, and $k_{p_3} = 3$.

2.3.4 Determining the common community labels for each layer

$\mathbf{H} \in \mathbb{R}^{N \times k_c}$ is the community membership matrix corresponding to the common communities. In order to determine whether a node from a particular layer belongs to any of the k_c common communities, $\mathbf{H}$ needs to go through some post-processing as described in Algorithm 2.3. First, for each node, i, in each layer, l, the common community membership matrix, $\mathbf{H}$, and the layer specific community membership matrix, $\mathbf{H}_l$ are concatenated and the column j with the maximum entry is identified (see line 3 in Algorithm 2.3). This determines the initial community assignment for that node. Next, we construct a binary common community membership matrix, $\mathbf{Z}_l$, for each layer where each entry is equal to 1 if a particular node belongs to one of the k_c common communities in that layer. For each layer, we compute the ratio of the average strength within a particular common community to the average strength outside that common community (lines 10-15). Finally, the common communities for that layer are determined as the ones which have the top $k_l - k_{p_l}$ ratios (lines 16-18). As shown in Figure 2.1c, the new embedding matrices corresponding to the common communities in each layer, $\mathbf{H}_{c_1}$, $\mathbf{H}_{c_2}$, and $\mathbf{H}_{c_3}$, will only have the columns that contain the information corresponding to the common communities present in that layer. In this example, $\mathbf{H}_{c_1}$ keeps the two columns (purple and green communities) from $\mathbf{H}$, while $\mathbf{H}_{c_2}$ keeps only the first column (purple), and $\mathbf{H}_{c_3}$ only the second column (green).



(a)



(b)

Figure 2.2: Example illustrating how the number of total communities per layer and number of common communities are determined. (a) Illustration of the eigengaps for each layer from left to right, top row showing the adjacency matrices, $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$, and bottom row showing the corresponding eigenvalues. The eigengap rule yields $k_1 = 6$, $k_2 = 6$, and $k_3 = 5$, respectively, as indicated by the black dotted lines; (b) Embedding matrices, $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$, are found using these k values and hierarchical clustering is then applied to the columns of $\mathbf{X} = [\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3]$. The red dotted line indicates where the algorithm stops, and resulting in the number of common communities as $k_c = 3$.

2.3.5 Time complexity

The time complexity of the proposed algorithm is mostly due to the Multiplicative Updates Rules, Eqs. (2.5)-(2.9). The time complexity for the product of two matrices, e.g., the product of a $m \times k$ matrix by a $k \times n$ matrix, is $O(mkn)$. Table 4.1 shows the time complexities of Eqs. (2.5)-(2.9) and the total complexity, with $l \in \{1, 2, \ldots, L\}$.

Algorithm 2.3: Identify the membership of k_c common communities across layers $\{1, 2, \ldots, L\}$.

Input: Community membership matrices $\mathbf{H}, \mathbf{H}_l$, adjacency matrices $\mathbf{A}_l$, and number of communities k_{p_l}, k_l, and k_c, for each layer $l \in \{1, 2, \ldots, L\}$.

Output: Layer specific community membership matrix $\mathbf{H}_{c_l}$ containing information about the common communities present in layer l.

for $l = 1$ to L **do**
  for each node i **do**
    $j^* \leftarrow argmax_j([\mathbf{H}, \mathbf{H}_l])_{ij}$
    $idx(i) \leftarrow j^*$
  end for
  for $t = 1$ to k_c **do**
    $\mathbf{Z}_l(find(idx == t), t) = 1$
    $\mathbf{Z}_l(find(idx \neq t), t) = 0$
  end for
  for $t = 1$ to k_c **do**
    $m = |\mathbf{Z}_l(:, t) == 1|$
    $\mathbf{B}_0 \leftarrow \mathbf{A}_l \odot (\mathbf{Z}_l(:, t)\mathbf{Z}_l(:, t)^\top)$
    $\mathbf{B}_1 \leftarrow \mathbf{A}_l \odot (\mathbf{1}_{n\times n} - \mathbf{Z}_l(:, t)\mathbf{Z}_l(:, t)^\top)$
    $q(t) \leftarrow \frac{\sum_{v,w}(\mathbf{B}_0)_{vw}/(m(m-1))}{\sum_{v,w}(\mathbf{B}_1)_{vw}/(m(N-m))}$
  end for
  $[\mathbf{q}_{sorted}, J] \leftarrow sort(\mathbf{q}, descending)$. J contains the sorted indices of the elements of $\mathbf{q}$.
  $pos \leftarrow J(1 : (k_l - k_{p_l}))$.
  $\mathbf{H}_{c_l} \leftarrow \mathbf{H}(:, [pos])$
end for

Table 2.1: Computational complexity of updating each variable per iteration.

$\mathbf{H}$ update (Eq. (2.5))	$O(N^2(max\{k_c, k_{p_1}, \cdots, k_{p_L}\})$
$\mathbf{H}_l$ update (Eq. (2.7))	$O(N^2(max\{k_c, k_{p_l}\}))$
$\mathbf{S}_l$ update (Eq. (2.8))	$O(N^2(max\{k_c, k_{p_l}\}))$
$\mathbf{G}_l$ update (Eq. (2.9))	$O(N^2(max\{k_c, k_{p_l}\}))$
Total	$O(N^2(max\{k_c, k_{p_1}, \cdots, k_{p_L}\}))$

2.3.6 Storage complexity

The storage complexity of our algorithm is determined by the sizes of the matrices $\mathbf{H}$, $\mathbf{H}_l$, $\mathbf{S}_l$, and $\mathbf{G}_l$. It can be seen that the total storage complexity is $O(N(max\{k_c, k_{p_1}, \cdots, k_{p_L}\}))$. For a multiplex network of size $N \times N \times L$, this is a significant reduction in memory cost.

Table 2.2: Storage complexity of each variable.

$\mathbf{H}$	$O(Nk_c)$
$\mathbf{H}_l$	$O(Nk_{p_l})$
$\mathbf{S}_l$	$O({k_c}^2)$
$\mathbf{G}_l$	$O({k_{p_l}}^2)$
Total	$O(N(max\{k_c, k_{p_1}, \cdots, k_{p_L}\}))$

2.4 Convergence Analysis

In this section, we will prove the convergence of the multiplicative update rule defined by Eq. (2.5) using the auxiliary function approach. As the other update rules are similar, we will not explicitly prove their convergence. We first introduce the definition of auxiliary function as follows.

Definition 1: A function $Z(\mathbf{H}, \mathbf{H}^t)$ is called an auxiliary function of $\mathcal{L}(\mathbf{H})$ if it satisfies

$$Z(\mathbf{H}, \mathbf{H}^t) \geq \mathcal{L}(\mathbf{H}) \text{ and } Z(\mathbf{H}, \mathbf{H}) = \mathcal{L}(\mathbf{H}).$$

The auxiliary function is a useful concept because of the following lemma which is proved in [148].

Lemma 1. *If Z is an auxiliary function, then $\mathcal{L}$ is non-increasing under the update*

$$\mathbf{H}^{t+1} = \underset{\mathbf{H}}{\operatorname{argmin}}\, Z(\mathbf{H}, \mathbf{H}^t).$$

Theorem 1. *Given $\mathbf{H}_l$, $\mathbf{S}_l$, and $\mathbf{G}_l$ the Lagrangian function $\mathcal{L}(\mathbf{H})$ is monotonically decreasing under the update rule* (2.5)*.*

Proof. For convenience, let $\mathcal{L}(h)$ denote the part of $\mathcal{L}(\mathbf{H})$ dependent on H_{ij}. From Eq. (2.4) we have

$$\mathcal{L}'(h) = \sum_{l=1}^{L} (\mathbf{4H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{HS}_l - 4\mathbf{A}_l\mathbf{HS}_l + 4\mathbf{HH}^\top\mathbf{A}_l\mathbf{HS}_l - \mathbf{4HH}^\top\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{HS}_l)_{ij}.$$

The second-order derivative of $\mathcal{L}(h)$ with respect to h_{ij} is

$$\begin{aligned}\mathcal{L}''(h) = \sum_{l=1}^{L} \{&4(\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top)_{ii}\mathbf{S}_{l_{jj}} - 4\mathbf{A}_{l_{ii}}\mathbf{S}_{l_{jj}} + 4[(\mathbf{H}^\top\mathbf{A}_l\mathbf{HS}_l)_{ij} + h_{ij}(\mathbf{A}_l\mathbf{HS}_l)_{ij} + (\mathbf{HH}^\top\mathbf{A}_l)_{ii}\mathbf{S}_{l_{jj}}] \\ &-4[(\mathbf{H}^\top\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{HS}_l)_{ij} + h_{ij}(\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{HS}_l)_{ij} + (\mathbf{HH}^\top\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top)_{ii}\mathbf{S}_{l_{jj}}]\}.\end{aligned}$$

Let h_{ij}^t denote the updated value of h_{ij} after the tth iteration, then the Taylor series expansion of $\mathcal{L}(h)$ at h_{ij}^t can be written as

$$\mathcal{L}(h) = \mathcal{L}(h_{ij}^t) + \mathcal{L}'(h_{ij}^t)(h - h_{ij}^t) + \frac{1}{2}\mathcal{L}''(h_{ij}^t)(h - h_{ij}^t)^2.$$

Now, the key is to find an appropriate auxiliary function $Z(h, h_{ij}^t)$. We choose the following $Z(h, h_{ij}^t)$ and prove in Appendix A, that it satisfies the conditions to be an auxiliary function of $\mathcal{L}(h)$,

$$Z(h, h_{ij}^t) = \mathcal{L}(h_{ij}^t) + 3\mathcal{L}'(h_{ij}^t)(h - h_{ij}^t) + \frac{3}{2}\frac{\sum_{l=1}^{L}(4\mathbf{H}_l^t\mathbf{G}_l\mathbf{H}_l^{t\top}\mathbf{H}^t\mathbf{S}_l + 4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t}(h - h_{ij}^t)^2. \tag{2.10}$$

According to Lemma 1, we must find the minimum of $Z(h, h_{ij}^t)$ with respect to h.

$$\frac{\partial Z(h, h_{ij}^t)}{\partial h} = 3\mathcal{L}'(h_{ij}^t) + 3\frac{\sum_{l=1}^{L}(4\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t}(h - h_{ij}^t) + 3\frac{\sum_{l=1}^{L}(4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t}(h - h_{ij}^t) = 0$$

Replacing $\mathcal{L}'(h_{ij}^t)$ in the equation above and canceling the common terms, we obtain

$$\sum_{l=1}^{L}(-4\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l - 4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l)_{ij} + \sum_{l=1}^{L}(4\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l + 4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}\frac{h}{h_{ij}^t} = 0.$$

Replacing h by h_{ij}^{t+1} we obtain the following update rule

$$h_{ij}^{t+1} = h_{ij}^t\frac{\sum_{l=1}^{L}(\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l + \mathbf{H}^t\mathbf{H}^{t\top}\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l)_{ij}}{\sum_{l=1}^{L}(\mathbf{H}_l\mathbf{G}_l^\top\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l + \mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}},$$

which is the same as the update rule shown in Eq. (2.5). □

2.5 Recovery and Consistency Analysis

In this section, we will establish the theoretical properties of the proposed community detection method. In particular, we investigate the recovery guarantees of the proposed objective function and the consistency of the algorithm as N and L increase under MLSBM.

2.5.1 Recovery Guarantees

In this section, we will investigate the recovery guarantees of the global optimizer of the objective function under the MLSBM when there is no noise. The optimization problem in (2.1) can be rewritten as

$$\underset{\mathbf{H'}_l,\mathbf{F}_l}{\operatorname{argmin}} \sum_{l=1}^{L} ||\mathbf{A}_l - \mathbf{H'}_l\mathbf{F}_l\mathbf{H'}_l^\top||_F^2 \text{ s.t } \mathbf{H'}_l^\top\mathbf{H'}_l = \mathbf{I}, \tag{2.11}$$

where $\mathbf{F}_l$ is a block matrix defined as

$$\mathbf{F}_l = \begin{bmatrix} \mathbf{S}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_l \end{bmatrix}$$

and $\mathbf{H'}_l = [\mathbf{H}|\mathbf{H}_l]$ is the concatenation of the community membership matrices of the common and private communities, $\mathbf{H} \in \mathbb{R}^{N\times kc}$ and $\mathbf{H}_l \in \mathbb{R}^{N\times k_{p_l}}$, respectively.

For the $N \times N \times L$ adjacency tensor $\mathbb{A} = \{\mathbf{A}_1, ..., \mathbf{A}_L\}$, we can define a multiplex SBM $[\mathbf{Z}, \Theta]$ as in [8], with each of the L slices $\mathbf{A}_l \in \mathbb{R}^{N\times N}$. The multiplex SBM with parameters $[\mathcal{Z} = \{\mathbf{Z}_1, ..., \mathbf{Z}_L\}, \mathcal{B} = \{\mathbf{\Theta}_1, ..., \mathbf{\Theta}_L\}]$, can be written in matrix form as,

$$\mathbb{E}(\mathbf{A}_l) = \mathcal{A}_l = \mathbf{Z}_l\mathbf{\Theta}_l\mathbf{Z}_l^\top,$$

with $\mathbf{\Theta}_l \in [0,1]^{(kc+k_{p_l})\times(kc+k_{p_l})}$ and $\mathbf{Z}_l \in \{0,1\}^{N\times(kc+k_{p_l})}$ for each layer l. $\mathbf{\Theta}_l$ is a block matrix defined as

$$\mathbf{\Theta}_l = \begin{bmatrix} \mathbf{\Theta}_c & \mathbf{0} \\ \mathbf{0} & \mathbf{\Theta}_{p_l} \end{bmatrix},$$

with $\mathbf{\Theta}_c$ and $\mathbf{\Theta}_{p_l}$ being the affinity probability matrices of the common and private communities, respectively. $\mathbf{Z}_l = [\mathbf{Z}_c|\mathbf{Z}_{p_l}]$ is the concatenation of the community membership matrices of the common and private communities, $\mathbf{Z}_c \in \mathbb{R}^{N\times kc}$ and $\mathbf{Z}_{p_l} \in \mathbb{R}^{N\times k_{p_l}}$, respectively. $\mathcal{A}_l$ is the population adjacency matrix for the l-th layer and the tensor $\mathscr{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_L\}$ is the population adjacency tensor.

To prove that our method can correctly recover the community assignments, we propose the following lemma following the work in [197].

Lemma 2. *The optimization problem in* (2.1) *applied to the population tensor* $\mathscr{A}$ *has* $\mathbf{H}'_l = \mathbf{Z}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}$ *and* $\mathbf{F}_l = (\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}\mathbf{\Theta}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}$, $l = 1, ..., L$, *as the unique solution up to an orthogonal matrix, provided at least one of the* $\mathbf{\Theta}_l$ *is full rank.*

Proof. To prove Lemma 2, we can show that $\mathbf{H}'_l = \mathbf{Z}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}$, $\mathbf{F}_l = (\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}\mathbf{\Theta}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}$, $l = 1, \cdots, L$, is a solution to the optimization problem in (2.11) to the population tensor $\mathscr{A}$. Substituting the solution to (2.11), we have

$$\sum_{l=1}^{L} ||\mathcal{A}_l - \mathbf{Z}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}(\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}\mathbf{\Theta}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}\mathbf{Z}_l^\top||_F^2 = \sum_{l=1}^{L} ||\mathcal{A}_l - \mathbf{Z}_l\mathbf{\Theta}_l\mathbf{Z}_l^\top||_F^2$$

and, since $\mathcal{A}_l = \mathbb{E}(\mathbf{A}_l) = \mathbf{Z}_l\mathbf{\Theta}_l\mathbf{Z}_l^\top$, the value of this minimization objective function is 0, and $\mathbf{H}'^\top_l\mathbf{H}'_l = (\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}\mathbf{Z}_l^\top\mathbf{Z}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2} = (\mathbf{Z}_l^\top\mathbf{Z}_l)^{1/2}(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2} = \mathbf{I}$

Now, we need to show the uniqueness of this solution. By assumption, at least one of the $\mathbf{\Theta}_l$ is full rank. For a non-singular matrix $\mathbf{P} \in \mathbb{R}^{(k_c+k_{p_l})\times(k_c+k_{p_l})}$ we can say that $\mathbf{H}'_l\mathbf{P}$ and $\mathbf{P}^{-1}\mathbf{\Theta}_l\mathbf{P}^{-1^\top}$ is also a solution. Due to the orthogonality constraint, we must have $(\mathbf{H}'_l\mathbf{P})^\top\mathbf{H}'_l\mathbf{P} = \mathbf{P}^\top(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}\mathbf{Z}_l^\top\mathbf{Z}_l(\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}\mathbf{P} = \mathbf{I}$, which implies $\mathbf{P}^\top\mathbf{P} = \mathbf{I}$, and therefore the solution is unique up to an orthogonal matrix. Moreover, since $\mathbf{Q}^{-1/2} = (\mathbf{Z}_l^\top\mathbf{Z}_l)^{-1/2}$ is a diagonal matrix with positive elements and therefore invertible, we have that $\mathbf{Z}_i\mathbf{Q}^{-1/2} = \mathbf{Z}_j\mathbf{Q}^{-1/2}$ implies $\mathbf{Z}_i = \mathbf{Z}_j$. □

2.5.2 Consistency

In this section, we investigate the asymptotic consistency of our method following [197]. The first step to prove consistency is to show that it is possible to recover the communities by maximizing the population version of the objective function, which was proven in the previous section. We consider the following asymptotic setup where we let N and L grow and assume no relationship exists between their growth rate. We also let the number of communities per layer, k_l, grow with both N and L.

The following results are proven for a multiplex network with L layers and the $N \times N \times L$ population adjacency tensor $\mathscr{A} = \{\mathcal{A}_1, ..., \mathcal{A}_L\}$ described in the previous section. λ_l denotes the minimum in absolute value nonzero eigenvalue of the l-th layer population adjacency matrix, Δ_l the maximum expected degree per layer, with $\bar{\Delta} = \frac{1}{L}\sum_{l=1}^{L}\Delta_l$ and $\bar{\Delta}' = \frac{1}{L}\sum_{l=1}^{L}\Delta_l^2$.

Theorem 2. *Let* $[(\hat{\mathbf{H}}'_1, \cdots, \hat{\mathbf{H}}'_L), (\hat{\mathbf{F}}_1, \cdots, \hat{\mathbf{F}}_L)]$ *be the solution that minimizes the MX-ONMTF objective function in* (2.11) *applied to the tensor adjacency* $\mathbb{A}$*. Let* r_{MX} *denote the fraction of misclustered nodes. Assume that* $\Delta_l > \frac{4}{9} log(2N/\epsilon)$*, and at least one of the* Θ_l*'s is of full rank, then with probability at least* $1 - \epsilon$*,*

$$r_{MX} \leq \frac{96 N_{max} k_{max} L^{1/4} \bar{\Delta}^{5/4} (log(2N/\epsilon))^{1/2}}{N \frac{1}{L} \sum_{l=1}^{L} (\lambda_l)^2}.$$

Here, a four parameter MLSBM defined by $\mathbf{p} = \{p_1, \cdots, p_L\}$, $\mathbf{q} = \{q_1, \cdots, q_L\}$, k_{max}, N_{max} is considered. p_l and q_l are the connection probabilities within and between communities, respectively. It is assumed that $p_l \neq q_l$ but they are of the same asymptotic order with respect to N, for all l. $k_{max} = max\{k_1, k_2, \cdots, k_L\}$, and N_{max} denotes the number of nodes in the largest true community with $N_{max} \asymp N/k_{max}$. For this MLSBM, $\lambda_l = N_{max}(p_l - q_l)$. Let $a_l \frac{\Delta_l}{N} = p_l$ and $b_l \frac{\Delta_l}{N} = q_l$. Then, $\lambda_l = \frac{\Delta_l}{k_{max}}(a_l - b_l)$, and $a_l \asymp b_l \asymp 1$. Define $f(\mathbf{a}, \mathbf{b}) = \frac{1}{L} \sum_{l=1}^{L} (a_l - b_l)^2$. Under the four parameter MLSBM, with Δ_l's all being of the same order, $\Delta_l \asymp \bar{\Delta}$ and $\bar{\Delta}' \asymp \bar{\Delta}^2$, the bound in Theorem 2 can be simplified to

$$\begin{aligned} r_{MX} &\lesssim \frac{L^{1/4} \bar{\Delta}^{5/4} (log(2N/\epsilon))^{1/2}}{\frac{1}{L} \sum_{l=1}^{L} \frac{\Delta_l^2 (a_l - b_l)^2}{k_{max}^2}} \asymp \frac{L^{1/4} k_{max}^2 \bar{\Delta}^{5/4} (log(2N/\epsilon))^{1/2}}{\bar{\Delta}' f(\mathbf{a}, \mathbf{b})}, \\ &\asymp \frac{L^{1/4} k_{max}^2 \bar{\Delta}^{5/4} (log(2N/\epsilon))^{1/2}}{\bar{\Delta}^2 f(\mathbf{a}, \mathbf{b})} \asymp \frac{L^{1/4} k_{max}^2 (log(2N/\epsilon))^{1/2}}{\bar{\Delta}^{3/4} f(\mathbf{a}, \mathbf{b})}. \end{aligned}$$

In the dense case where $\bar{\Delta} \asymp N$, we have

$$r_{MX} \lesssim \frac{L^{1/4} k_{max}^2 (log(2N/\epsilon))^{1/2}}{\bar{\Delta}^{3/4} f(\mathbf{a}, \mathbf{b})} \asymp \frac{k_{max}^2}{L^{-1/4} N^{3/4} f(\mathbf{a}, \mathbf{b}) (log(2N/\epsilon))^{-1/2}}.$$

and,

$$k_{max} = O\left(\left(\frac{N^3}{L} \right)^{1/8} (f(\mathbf{a}, \mathbf{b}))^{1/2} (log(2N/\epsilon))^{-1/4} \right).$$

Hence consistent estimation is possible as long as k_{max} grows slower than $\left(\frac{N^3}{L} \right)^{1/8}$. Proof of Theorem 2 is provided in Appendix B.

2.6 Experiments: Simulated Data and Real-world Networks

2.6.1 Synthetic Multiplex Networks

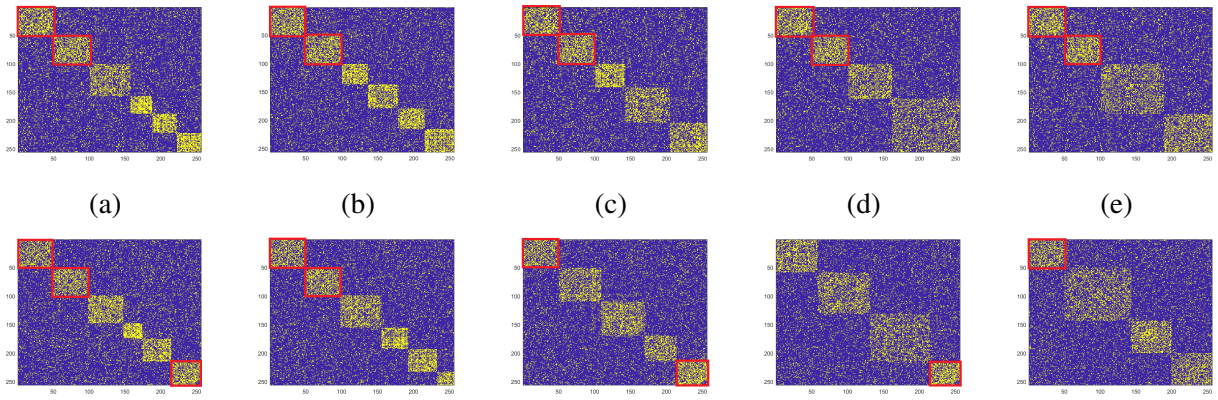
Multiplex benchmark networks based on the model described in [16, 124] were generated. The authors in [16] propose a two-step approach to generate multilayer networks with a community structure. First, a multilayer partition with the user-defined number of nodes in each layer, number of layers, and an interlayer dependency tensor that specifies the desired dependency structure between layers is generated. Next, for the given multilayer partition, edges in each layer are generated following a degree-corrected block model [136] parameterized by the distribution of expected degrees and a community mixing parameter $\mu \in [0, 1]$. The mixing parameter μ controls the modularity of the network. When $\mu = 0$, all edges lie within communities, whereas $\mu = 1$ implies that edges are distributed independently. For multiplex networks, the probabilities in the interlayer dependency tensor are the same for all pairs of layers and are specified by $p \in [0, 1]$. When $p = 0$, the partitions are independent across layers while $p = 1$ indicates an identical partition across layers.

In this work, we extend the model described above to generate multiplex benchmark networks with common and private communities. We first generate the common communities by randomly selecting N_c nodes across all layers and setting the inter-layer dependency probability to p_1. For each common community, we decide whether it exists in a particular layer or not. Next, we independently generate the private communities for each layer with the remaining nodes in that layer. We generated 100 different random realizations of each multiplex network in order to report the average performance metric on the experiments.

Evaluation: We compared the performance of our method to well-known multiplex community detection algorithms. In particular, we compared with ONMTF applied to the aggregated multiplex networks using the average of the adjacency matrices (Aggregated Average), Spectral Clustering on Multi-Layer graphs (SC-ML) [77], Generalized Louvain (GenLouvain) multilayer community detection algorithm [130, 182], Infomap [69], Collective Symmetric Nonnegative Matrix Factorization (CSNMF) [101], Collective Projective Nonnegative Matrix Factorization (CPNMF) [101],

Collective Symmetric Nonnegative Matrix Tri-factorization (CSNMTF) [101], and Orthogonal Link Matrix Factorization (OLMF) [244].

Experiment 1: In this experiment, we generated two different types of networks, one where the common communities are present across all layers and another where the common communities are present in different subsets of layers. Figure 2.3 illustrates a single realization of the adjacency matrices generated with $\mu = 0.1$ for two 5-layer networks, one of each type (first row and second row). The network in Figure 2.3a-2.3e has two common communities (the first two communities in each layer) across all layers and $k_{p_1} = 4$, $k_{p_2} = 4$, $k_{p_3} = 3$, $k_{p_4} = 2$, and $k_{p_5} = 2$, while the network in Figure 2.3f-2.3j has a total of 3 common communities (highlighted in red) that are present in different subsets of layers and $k_{p_1} = 3$, $k_{p_2} = 4$, $k_{p_3} = 3$, $k_{p_4} = 3$, and $k_{p_5} = 3$. In order to evaluate the performance of our algorithm for different noise levels, these two types of networks were generated with varying values of $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. The inter-layer dependency probability is $p_1 = 1$, for the common communities.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 2.3: Example illustrating a single realization of two 5-layer networks generated with $\mu = 0.1$. (a)-(e) Adjacency matrices for a network with two common communities (highlighted in red) across all layers; (f)-(j) Adjacency matrices for a network with three common communities (highlighted in red) across different subsets of layers.

Figure 2.4 shows the results for the networks with 2 common communities across all layers for 3 (2.4a), 4 (2.4b), and 5 layers (2.4c), and for the networks with 3 common communities across different subsets of layers for 3 (2.4d), 4 (2.4e), and 5 layers (2.4f). The results indicate that our

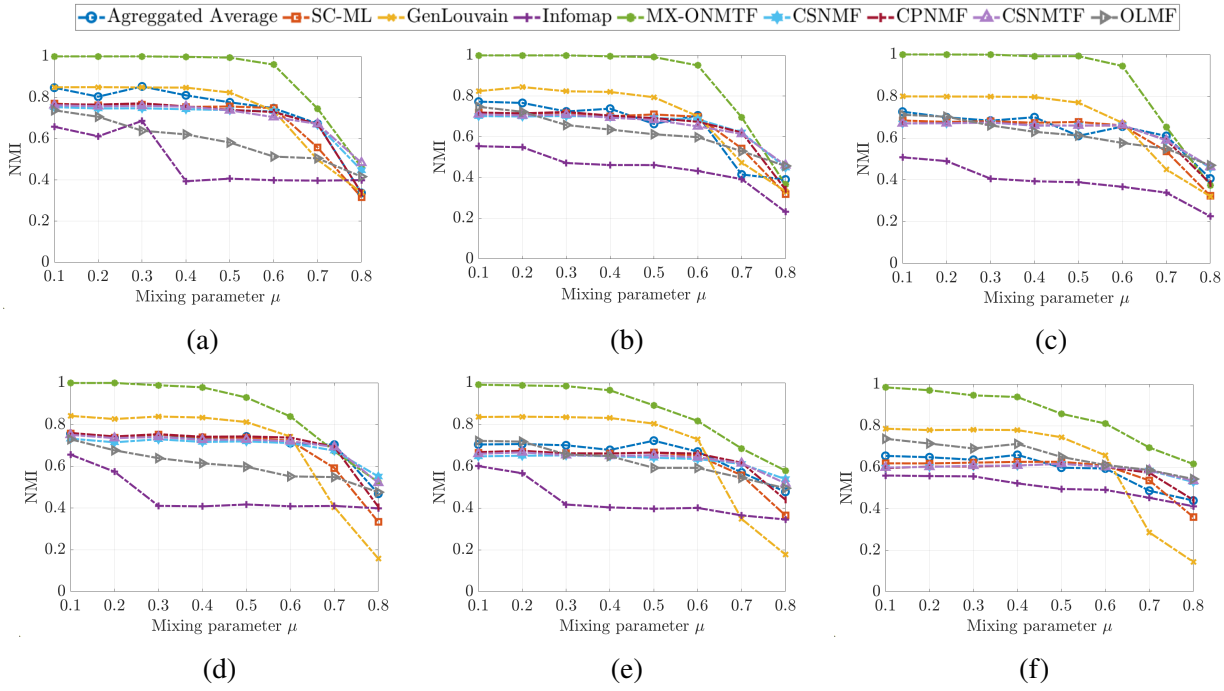


Figure 2.4: Mean NMI over 100 realizations of (a)-(c) 3-layer, 4-layer and 5-layer benchmark networks, respectively, for the scenario with 2 common communities across all layers; (d)-(f) 3-layer, 4-layer and 5-layer benchmark networks, respectively, for the scenario with 3 common communities across different subsets of layers. All networks are generated with 8 different values of the mixing parameter μ and $N = 256$.

method performs well for both networks with common communities across all layers as well as for networks with common communities that do not span all layers. Our method discovers the complete structure of the network rather than forcing it to have a consensus partition. Moreover, our method is robust to noise for larger values of μ compared to the other methods. We can also conclude that our algorithm performs better when the common communities are across all layers (see Figure 2.4a, 2.4b, 2.4c) than when the multiplex community structure is more complex with common communities across subsets of layers (see Figure 2.4d, 2.4e, 2.4f), but it still outperforms the rest of the methods. From Figure 2.4 we can see that GenLouvain performs well when μ is small, but its performance deteriorates for μ values above 0.6. Another observation is that when the number of layers is small, the NMF-based methods perform closer to GenLouvain but when the number of layers increases, NMF algorithms perform worse. This is because these NMF methods perform aggregation on either the adjacency or the community indicator matrices. When there is

more variation across layers, these methods fail to capture this heterogeneity.

Experiment 2: In the second experiment, we evaluated the robustness of the algorithm against variations in the common community structure by fixing $\mu = 0.1$ and varying the inter-layer dependency probability, p_1, i.e., the common communities are allowed to vary across layers. The performance of all methods for a 5-layer network are reported in Figure 2.5 based on the average NMI over 100 realizations of the network. As we can see in Figure 2.5, our method still outperforms the other eight methods when there is some variation in the common community structure. This demonstrates that our method is robust to variations of the common community structure across layers. However, our algorithm is more sensitive to the drop in p_1 than the rest of the methods. This is because when the common communities have a high variation our algorithm may try to assign some of those nodes to the private communities.

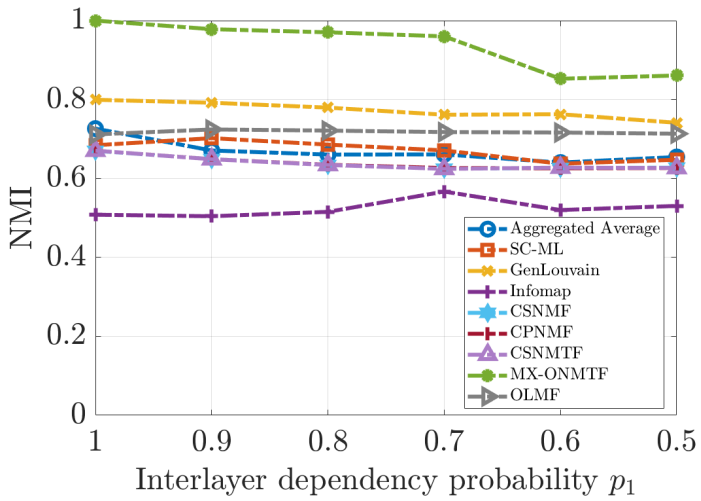


Figure 2.5: 5-layer network generated with 6 different values of the interlayer dependency probability p_1, with $\mu = 0.1$, and $N = 256$.

Experiment 3: Another parameter in our model is the number of common communities k_c. In this experiment, we fixed $N = 256$, $\mu = 0.3$, and the number of communities in each layer, and varied k_c from 1 to 7. When $k_c = 7$, all communities are common across layers and there are no private communities. As we can see in Figure 2.6, as k_c is increased, the performance of all the other methods improves, as expected, because these methods are designed to detect the common

community structure. When the communities are common across layers, most of the methods, except Infomap, converge to the same NMI value. The performance of MX-ONMTF is not affected by increasing k_c, and when all communities are common it performs similarly to the other methods.

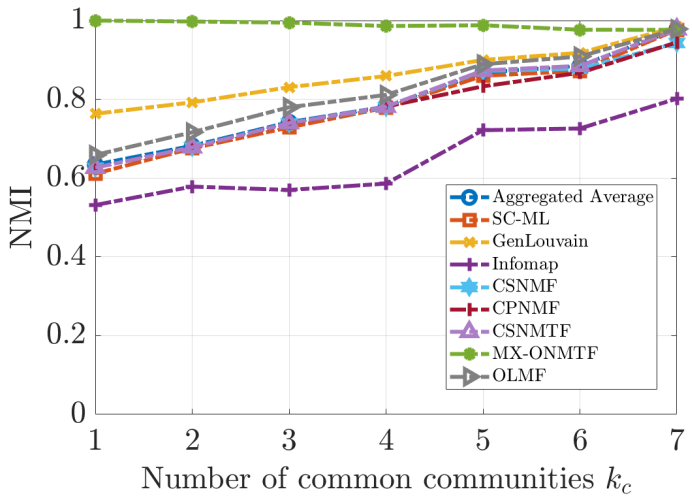


Figure 2.6: 5-layer network generated with different values of common communities k_c across layers.

Scalabilty Analysis: In this experiment, we evaluate the effect of network size on the run time of the proposed algorithm. For this purpose, we fixed $\mu = 0.3$, $L = 5$, $k_c = 3$ and varied N from 32 to 8192. From Figure 2.7, it can be seen that our method's run time is almost log-linear. This is comparable with all the other NMF based methods. However, our as shown in the previous experiments, our method performs better. Most of this time complexity is due to the multiplicative update rule used in NMF-based algorithms and can be reduced using alternative approaches as discussed in [59].

2.6.2 Ablation Study

In this section, we consider the importance of the different variables and constraints in our method, i.e., orthogonality constraint and tri-factorization, by modifying our cost function and its constraints.

- MX-NMTF: This is equivalent to our problem without the orthogonality constraints. The

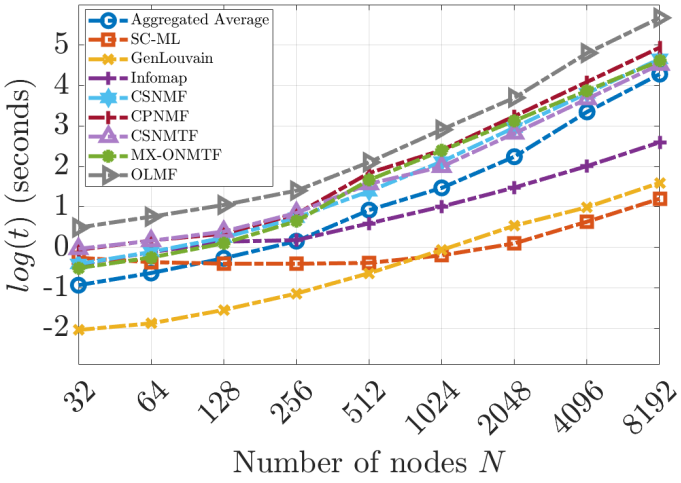


Figure 2.7: 5-layer network with 9 different values of N, $k_c = 3$, and $\mu = 0.3$.

modified problem formulation is

$$\underset{\mathbf{H}\geq 0,\mathbf{H}_l\geq 0,\mathbf{S}_l\geq 0,\mathbf{G}_l\geq 0}{\operatorname{argmin}} \sum_{l=1}^{L} ||\mathbf{A}_l - \mathbf{H}\mathbf{S}_l\mathbf{H}^\top - \mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top||_F^2.$$

- MX-ONMF: This is equivalent to symmetric NMF without the tri-factorization, with the orthogonality constraints preserved.

$$\underset{\mathbf{H}\geq 0,\mathbf{H}_l\geq 0}{\operatorname{argmin}} \sum_{l=1}^{L} ||\mathbf{A}_l - \mathbf{H}\mathbf{H}^\top - \mathbf{H}_l\mathbf{H}_l^\top||_F^2, \text{s.t } \mathbf{H}^\top\mathbf{H} = \mathbf{I}, \mathbf{H}_l^\top\mathbf{H}_l = \mathbf{I}.$$

- MX-NMF: This is equivalent to our method without the orthogonality constraints and the tri-factorization.

$$\underset{\mathbf{H}\geq 0,\mathbf{H}_l\geq 0}{\operatorname{argmin}} \sum_{l=1}^{L} ||\mathbf{A}_l - \mathbf{H}\mathbf{H}^\top - \mathbf{H}_l\mathbf{H}_l^\top||_F^2.$$

Table 2.3 shows the NMI results for MX-ONMTF and the different variations presented above for the multiplex network with five layers and three common communities across different subsets of layers used in Experiment 1. MX-ONMTF, which uses both orthogonality and tri-factorization performs better than the other variations. It can also be concluded that orthogonality improves the results with respect to regular NMF more than adding only tri-factorization.

2.6.3 Real World Multiplex Networks

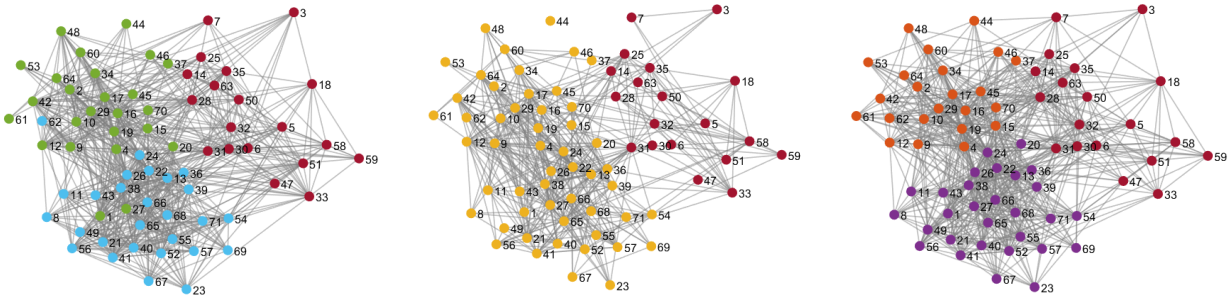
Lazega Law Firm Multiplex Social Network: Lazega Law Firm [147] is a multiplex social network with 71 nodes and three layers representing Co-work, Friendship and Advice relationships

Table 2.3: Effect of orthogonality and tri-factorization in the proposed framework.

μ	MX-ONMTF	MX-NMTF	MX-ONMF	MX-NMF
0.1	**0.9854**	0.5391	0.9326	0.8384
0.2	**0.9717**	0.5466	0.9236	0.8484
0.3	**0.9469**	0.5435	0.9160	0.8298
0.4	**0.9489**	0.5357	0.9080	0.8114
0.5	**0.8582**	0.5202	0.8784	0.7835
0.6	**0.8110**	0.5084	0.8157	0.7404
0.7	**0.6952**	0.4868	0.6960	0.6394
0.8	**0.6158**	0.4750	0.6180	0.5705

between partners and associates of a corporate law firm. This data set also includes information about some attributes of each node such as status, gender, office location, years with the firm, age, type of practice, and law school.

Applying MX-ONMTF to this network, we obtain one common community across all layers composed of the nodes colored in red as well as private communities for each layer, as shown in Figure 2.8. This network does not have ground truth community structure, but we can compute the NMI between the detected community structure and each type of node attributes, i.e., metadata, to gain better insight into the results and to be able to provide quantitative results [193]. For each of the attributes, the nodes are divided into communities based on that particular attribute. For example, for the status, the network is divided into two communities, partners and associates. For Age and Seniority, the nodes were grouped into five-year bins. The community structure for each attribute is used as ground truth to compute the NMI between each attribute and the community structure detected by our method. The NMI values given in Table 2.4 for the partition obtained by our method, suggest that office location and type of practice (litigation or corporate) are highly correlated with community membership across co-work, friendship and advice relationships. We can also see that the partition detected by MX-ONMTF has greater NMI values for each attribute. Therefore, our method detects a community structure that takes all of the attributes into account instead of partitioning with respect to just one attribute as the Aggregated Average does.



(a) Advice (b) Friendship (c) Co-work

Figure 2.8: Communities detected in Lazega Law Firm network across the three layers, advice, friendship, and co-work relationships. Red nodes are in the common community across the three layers.

Table 2.4: NMI of the obtained community partition for each method and the metadata available for the Lazega Law Firm Multiplex Network.

Method	Status	Gender	Office	Seniority	Age	Practice	Law School
GenLouvain	0.0345	0.0307	0.5294	0.0807	0.0431	0.5468	0.0040
Aggregated Average	0.0383	0.0197	0.5379	0.1307	0.0798	0.4411	0.0201
SC-ML	0.0138	0.0259	0.0731	0.1225	0.0464	0.0249	0.0140
Infomap	0.0179	0.0043	0.1668	0.2880	0.0083	0.0003	0.0093
CSNMF	0.0418	0.0291	0.5732	0.1155	0.0736	0.4227	0.0172
CPNMF	0.0081	0.0524	0.1139	0.0514	0.0291	0.0187	0.0221
CSNMTF	0.0395	0.0217	0.0798	0.0795	0.0487	0.1335	0.0279
OLMF	0.0534	0.0296	0.4776	0.1023	0.0401	0.1665	0.0163
MX-ONMTF	**0.4752**	**0.4906**	**0.7386**	**0.4135**	**0.4203**	**0.6162**	**0.4226**

C. Elegans Network: C. Elegans Network [68, 48] is a multiplex network with 279 nodes and 3 layers representing different synaptic junctions (electric, chemical monadic, and polyadic) of 279 neurons of the Caenorhabditis Elegans connectome. Information about different attributes of the neurons in this dataset such as the group of neuron they belong to (bodywall, mechanosensory, ring interneurons, head motor neurons, etc.), the type of neuron (motor neurons, sensory neurons, interneurons), and the color (blue, red, yellow, orange, etc.) is available.

Table 2.5 shows the NMI values between the community structures detected by each method and each of the three attributes available for this dataset. The partition detected by MX-ONMTF has greater NMI values for each of the attributes compared to the other eight methods.

Table 2.5: NMI of the obtained community partition for each method and the metadata available for C. Elegans Network.

Method	Neuron Group	Neuron Type	Color
GenLouvain	0.3756	0.1297	0.2362
Aggregated Average	0.3839	0.1590	0.2977
SC-ML	0.0185	0.0103	0.2690
Infomap	0.2265	0.2345	0.2355
CSNMF	0.1635	0.075	0.1211
CPNMF	0.0854	0.0277	0.0628
CSNMTF	0.1113	0.0402	0.0914
OLMF	0.3288	0.2149	0.2669
MX-ONMTF	**0.4074**	**0.4001**	**0.4593**

YeastLandscape Multiplex Network: Yeast Landscape is a multiplex genetic interaction network of a specie of yeast, Saccharomyces Cerevisiae [60, 68]. This network has 4458 nodes and 4 layers representing the positive and negative interaction networks of genes in Saccharomyces cerevisiae and positive and negative correlation based networks in which genes with similar interaction profiles are connected to each other. For this work, we use the bioprocess annotations of the genes available on the supplementary data file S6 of [60] as ground truth. We divided the genes into 18 groups according to their primary bioprocess. There were 1580 genes in this network without attributes.

Table 2.6 shows the NMI values between the community structures detected by each method and the bioprocess of the genes. MX-ONMTF gives the highest NMI value followed by the other NMF-based community detection methods.

2.6.4 Analysis of Overfitting

In this section, we evaluate the efficacy of MX-ONMTF in terms of overfitting/underfitting. We characterize the algorithm's model fitting performance in terms of link prediction and link description as described in [98].

In link prediction, each layer of a multiplex network is sampled using an α fraction of the edges of that layer creating a new sequence of graphs $\{\mathcal{G}'_l\}$, where $l \in \{1, 2, \ldots, L\}$. The goal

Table 2.6: NMI of the obtained community partition for each method with respect to the metadata available for YeastLandscape Network.

Method	Bioprocess
GenLouvain	0.0794
Aggregated Average	0.1108
SC-ML	0.1564
Infomap	0.2987
CSNMF	0.3553
CPNMF	0.3559
CSNMTF	0.3549
OLMF	0.3145
MX-ONMTF	**0.4123**

of link prediction is to accurately distinguish missing links, $E_m = E \setminus E'$ (true positives), from non-existent edges, $E_{ne} = U \setminus E$ (true negatives), within the set of unobserved connections $U \setminus E'$, where U is the set of all possible edges in $\mathcal{G}_l$ and $|E'| = \alpha|E|$ is a uniformly random subset of edges in the original graph $\mathcal{G}_l = (V_l, E_l)$. MX-ONMTF is then applied to the new multiplex network and a model-specific score function s_{ij} that estimates the likelihood that a pair of nodes i, j is connected, is then computed. In our case, we use the estimated adjacency matrix defined as $\hat{\mathbf{A}}_l = \mathbf{H}'\mathbf{S}'_l\mathbf{H}'^{\top} + \mathbf{H}'_l\mathbf{G}'_l\mathbf{H}'^{\top}_l$, using the embedding matrices obtained from MX-ONMTF such that $s_{ij} = \hat{\mathbf{A}}_l(i, j)$. Provided the rank of all non-observed links, the AUC value can be interpreted as the probability that a randomly chosen missing link, i.e., a link in E_m, is given a higher score than a randomly chosen nonexistent link, i.e., a link in E_{ne}. At each time we randomly pick a missing link and a nonexistent link to compare their scores, if among n independent comparisons, there are n' times the missing link has a higher score, and n'' times they have the same score, the AUC value is

$$\text{AUC} = \frac{n' + 0.5n''}{n}.$$

The AUC curve as a function of α shows how MX-ONMTF performs across the sampled graph ranging from when very few edges are observed to when only a few edges are missing.

On the other hand, link description evaluates how well the method learns an observed network. Its goal is to distinguish accurately observed edges E' (true positives) and observed non-edges

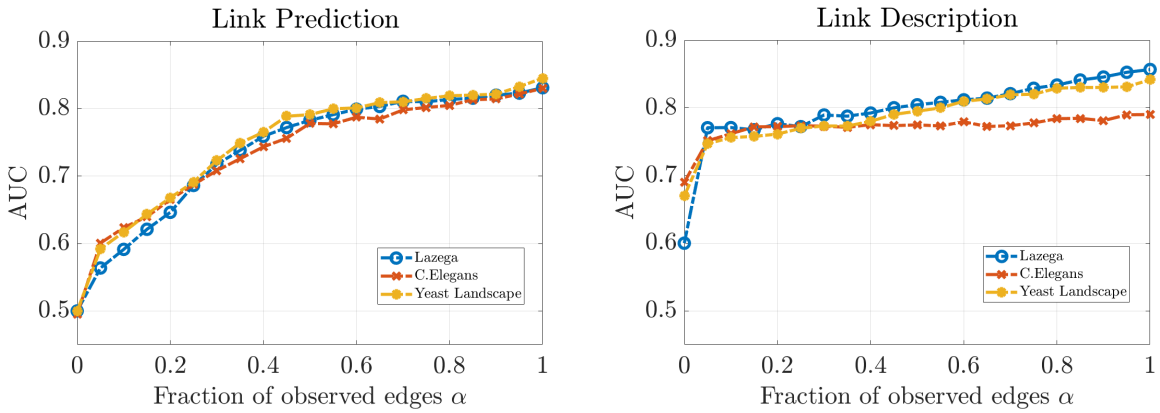


Figure 2.9: AUC curves for link prediction and description tasks. Each curve shows the mean AUC for MX-ONMTF over 20 realizations of a real-world network for a given fraction α of observed edges in the network.

$U \setminus E'$ (true negatives) within the set of all possible edges U. Similar to link prediction, we employ $\mathcal{G}_l'$'s and use the same scoring function s_{ij} to evaluate our algorithm's accuracy at distinguishing edges from non-edges.

In our evaluation, 20 randomly subsampled multiplex networks $\{\mathcal{G}_l'\}$ are generated for each value of α and the mean AUC is computed at each α. This analysis is applied to all three real multiplex networks. As it can be seen in Figure 2.9, based on the guidelines provided in [98], the performance of our method in link prediction can be described as good and its performance in link description as poor, and we can conclude that our method does not overfit or underfit.

2.6.5 Multiview Networks

In order to evaluate the performance of our method on networks where the communities are common across all layers, we use two multiview data sets, UCI Handwritten Digits[1] [83] and Caltech [158].

The UCI Handwritten Digits data set consists of features of handwritten digits from (0- 9) extracted from a collection of Dutch utility maps. There is a total of 2000 patterns that have been digitized in binary images, 200 patterns per digit. These digits are represented by six different feature sets: Fourier coefficients of the character shapes, profile correlations, Karhunen-Loève coefficients, pixel averages in 2×3 windows, Zernike moments, and morphological features. Each

[1] https://archive.ics.uci.edu/ml/datasets/Multiple+Features

layer of the multiplex network represents one of the 6 features. The graphs are constructed using k-nearest neighbors graphs with the nearest 50 neighbors and Euclidean distance.

Caltech-101 is a well-known object recognition dataset that consists of pictures of objects belonging to 102 categories. There are about 40 to 800 images per category for a total of 9144 images. This dataset consists of 6 types of features extracted from each image. A multiplex network with 6 layers representing each of the features, 102 classes, and 9144 nodes is constructed from this dataset using k-nearest neighbors graphs with the nearest 50 neighbors. A smaller version of this dataset is also used in these experiments, where only 20 objects are selected, resulting in a multiplex network with 6 layers, 20 communities, and 2386 nodes.

Table 2.7: NMI of the obtained community partition for multiview networks.

Method	Handwritten	Caltech-20	Caltech-101
GenLouvain	0.8791	0.5921	0.3406
Aggregated Average	0.7957	0.5358	0.3941
SC-ML	0.8435	0.6476	0.5016
Infomap	0.5367	0.3876	0.2583
CSNMF	0.4499	0.4250	0.3842
CPNMF	0.4421	0.4208	0.3816
CSNMTF	0.4478	0.4264	0.3831
OLMF	0.4876	0.4321	0.3117
MX-ONMTF	**0.9432**	**0.6861**	**0.5660**

In this case, as we have the true class assignment, we compute the NMI with respect to this ground truth. As it can be seen in Table 2.7, our method performs better than the rest of the methods for the three networks. This indicates that even in cases where there are no private communities, our method is successful at obtaining the consensus community structure, thus can be used as an alternative to multiview clustering.

2.7 Application to fMRI data: Subgroup identification

Identifying homogeneous subgroups with similar symptoms or neuropsychological patterns is essential for understanding the heterogeneity of psychotic disorders and advancing precision

medicine, which enables tailored treatments based on patients' unique profiles. Given the complexity of psychiatric disorders, exploring relationships across multiple functional networks can provide deeper insights into diagnostic heterogeneity. In this section, we apply our method, MX-ONMTF, to functional connectivity networks that are extracted from multi-subject resting-state fMRI data, with each network representing a distinct functional interaction pattern. These networks form a multiplex framework, where each layer corresponds to a functional network, and nodes correspond to individual subjects. By applying the community detection method proposed in this chapter, we identify communities that capture shared functional patterns across multiple networks (common communities) while also preserving network-specific subgroup structures (private communities).

2.7.1 Resting-State fMRI Data

The resting-state fMRI datasets and corresponding clinical scores used in this study are obtained from the Bipolar-Schizophrenia Network on Intermediate Phenotypes (B-SNIP) study [239, 238]. The study follows a standardized diagnostic and recruitment process across multiple sites, including Baltimore, Chicago, Dallas, Detroit, and Hartford. All subjects underwent a single 5-minute resting-state fMRI session using a 3-T scanner while maintaining fixation on a crosshair displayed on a monitor to minimize motion artifacts.

The first three time points were discarded and head motion correction was performed followed by slice-timing correction. The corrected fMRI data were then transformed into the standard Montreal Neurological Institute (MNI) space using an echo-planar imaging template and then were resampled to $3 \times 3 \times 3$ mm^3 isotropic voxels. The resampled fMRI data were further smoothed using a Gaussian kernel, with a full width at half maximum (FWHM) equal to 6 mm. Quality control procedures [82] were applied to select subjects. In this study, 464 individuals diagnosed with psychotic disorders including 176 individuals with schizophrenia, 159 with psychotic bipolar disorder, and 129 with schizoaffective disorder, were used.

Figure 2.10 illustrates the workflow of the proposed method for subgroup identification in this resting-state fMRI dataset. Figure 2.10a illustrates the preprocessing and decomposition pipeline used in this study. The fMRI data for each subject are first transformed into vectorized brain

volumes, forming the input matrices X_l across multiple functional networks. These matrices are then constrained by reference templates and decomposed using entropy bound minimization (c-EBM) [275, 276] to obtain both estimated components and subject-specific spatially constrained variations (SCVs). The spatial constraints, *i.e.,* references, for c-EBM are generated based on the fSIG pipeline [276], which includes 49 resting-state networks (RSNs) spanning auditory (AUD: 1 RSN), sensorimotor (MOT: 8 RSNs), visual (VIS: 10 RSNs), default-mode (DMN: 11 RSNs), attentional (ATTN: 8 RSNs), frontal (FRONT: 8 RSNs), cerebellar (CB: 2 RSNs), and basal ganglia (BG: 1 RSN) networks. Next, as shown in Figure 2.10b, an element-wise squared partial correlation matrix $\hat{C}$ is constructed from the estimated SCVs while removing reference effects. This matrix is then transformed into an undirected graph by setting diagonal elements to zero and thresholding partial correlation values to define network edges. The resulting graph layers are categorized based on their topological properties as explained in the next subsection, ensuring consistent classification across networks. Finally, community detection is performed on each group of layers using MX-ONMTF to identify both common and private communities.

2.7.2 Layer Classification Based on Graph-Theoretical Metrics

First, the layers in the multiplex network are classified based on their topological properties using graph-theoretical metrics. The element-wise squared partial correlation matrix $\hat{C}_l$ captures subject similarity within a functional network while excluding reference influences. Networks with similar correlation patterns are assigned to the same layer category, as they provide consistent subgroup information. To achieve this, $\hat{C}_l$ is converted into an undirected binary graph $\mathcal{G}_l$ with adjacency matrix $\mathbf{B}_l$, where subjects are nodes and edges represent significant partial correlations after thresholding. The threshold e is selected to maintain a link density (Γ) between 20% and 70%, ensuring balanced connectivity.

Graph-theoretical metrics including (1) path length, (2) global efficiency, (3) centrality, (4) clustering coefficient, and (5) small-worldness, are used to characterize each layer, and a feature matrix summarizing these topological characteristics of $\mathcal{G}_l$ is formed. k-means clustering is applied to this feature matrix with the number of layer categories varying from 2 to 10. The final number of

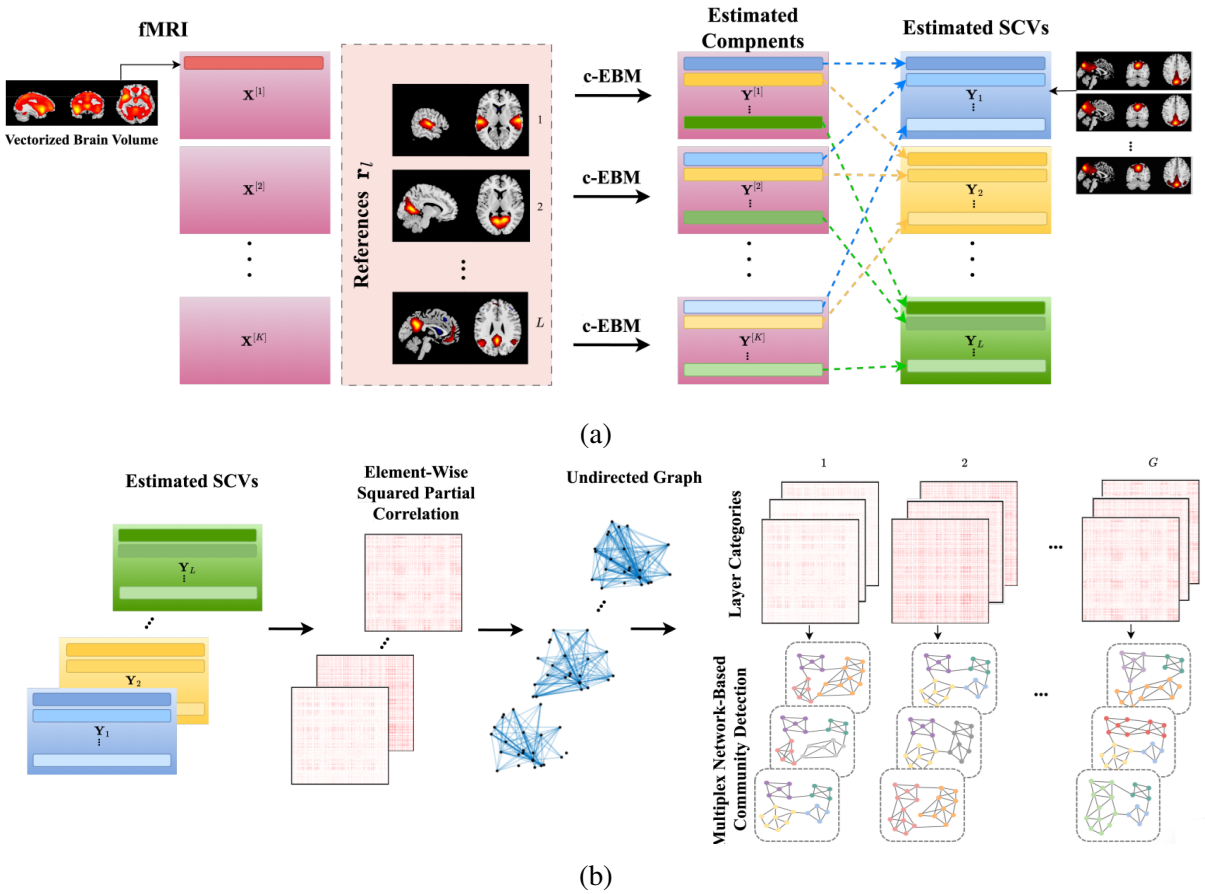


Figure 2.10: (a): Flowchart of applying c-EBM to individual datasets. The estimated components $\mathbf{Y}^{[k]}$ are aligned across datasets by utilizing reference priors $\mathbf{r}_l$. The resulting SCVs are formed by concatenating the corresponding estimated components from all subjects, such as the default mode network component across individuals. Each SCV summarizes information about a specific functional network across all subjects. (b): Flowchart of the proposed multiplex community-based subgroup identification includes **Element-Wise Squared Partial Correlation Matrix Formation**: The matrix $\hat{C}$ is constructed from the estimated SCVs derived from c-EBM, with the effects of references removed; **Graph Transformation**: $\hat{C}$ is converted into an undirected graph with adjacency matrix $\mathbf{B}$ by removing its diagonal elements, thresholding and binarizing the partial correlation values to define the edges, with the subjects represented as nodes; **Layer Categorization**: Based on the topological properties of the graphs, layers are classified into different layer categories. This ensures that connection patterns across subjects are consistent within each layer category across various functional networks; **Community Detection**: MX-ONMTF is applied to each layer category to identify common and private communities.

layer categories is determined based on the within-cluster sums of point-to-centroid distances. The purpose of dividing the 49 layers into different categories is to make sure the connection pattern across subjects is similar across different functional networks that are within a layer category. For this dataset, the layer categorization resulted in $G = 3$ categories with $L_1 = 6$, $L_2 = 23$, and $L_3 = 20$ layers, respectively.

Once the 49 layers are divided into three categories, the number of communities in each category is determined by applying the algorithms described in Section 2.3.3. MX-ONMTF is then applied individually to the three multiplex networks, resulting in two common communities for each layer category.

2.7.3 Discussion on fMRI Results

To investigate the behavioral differences across the detected communities, i.e. subgroups, we analyzed the corresponding Positive and Negative Syndrome Scale (PANSS) [137] scores collected from the same group of individuals, since abnormal functional status observed through fMRI analysis is often associated with certain behavioral symptoms. The PANSS evaluates 30 distinct symptoms categorized into positive symptoms, negative symptoms, and general psychopathology symptoms, commonly observed in psychotic patients. Each symptom is rated on a scale of 1 to 7, where 1 signifies no symptoms and 7 represents severe symptoms. Positive symptoms, such as hallucinations and delusions, reflect an excess or distortion of normal functions and are typically more pronounced in Type 1 schizophrenia. Negative symptoms, including emotional withdrawal and difficulty in abstract thinking, represent a loss of normal functions and are more severe in Type 2 schizophrenia. General psychopathology symptoms encompass issues not specifically categorized as positive or negative, such as poor attention, anxiety, guilt, tension, lack of insight, and active social avoidance [122].

In Figure 2.11, we present the two common communities identified separately from Layer Categories 1 and 2. The detected common communities exhibit significant group differences compared to the remaining subjects in both activations across functional networks and PANSS clinical scores. A two-sample t-test was conducted to analyze voxel-wise activation values of the

spatial maps for subjects within each subgroup, determining whether the spatial activation patterns of RSNs showed significant differences between subgroups. False discovery rate (FDR) correction [20] is applied to all results.

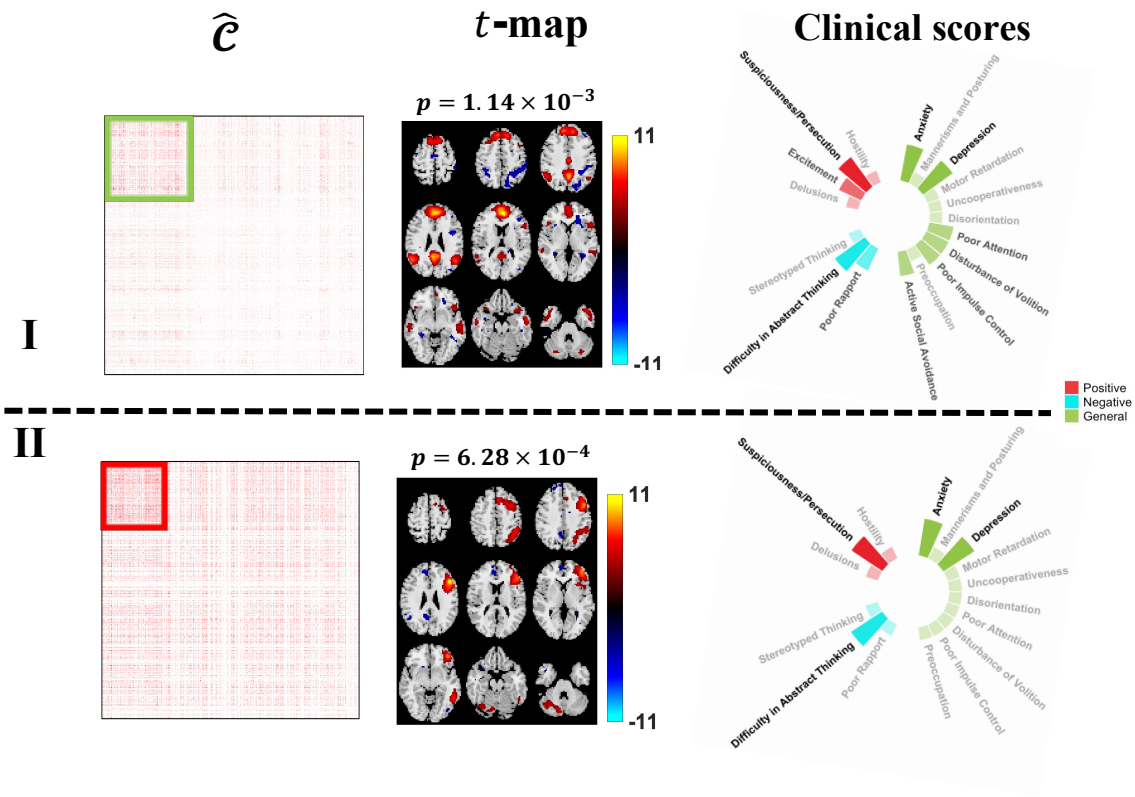


Figure 2.11: The two identified common communities (**I, II**) show significant group differences compared to the remaining subjects, both in terms of meaningful functional brain areas and clinical scores. The two identified common communities are detected from a set of 23 layers and a set of 6 layers separately. The neuro-activity maps (t-maps) highlight the functional areas with significant activation differences, derived from two-sample t-tests with $p < 0.05$ after FDR correction, between the subgroups. From left to right, the visualization includes the element-wise squared partial correlation matrix of a SCV $\hat{C}$, the t-map, and the clinical score differences between the identified subgroups. Subjects belonging to the identified common communities are indicated by red and green blocks in $\hat{C}$.

The common community in Figure 2.11 **I**, highlighted by a green square, exhibits elevated scores across multiple behavioral variables. These include poor attention, disturbance of volition, poor impulse control, and active social avoidance from the general psychopathology subscale of the PANSS score, as well as higher excitement scores from the positive subscale and poor rapport from the negative subscale. These behavioral differences can be linked to functional variations

($p = 1.14 \times 10^{-3}$) observed in specific brain regions, including the anterior prefrontal cortex (antPFC, BA 10), dorsolateral prefrontal cortex (dlPFC, BA 9), dorsal anterior cingulate cortex (dACC, BA 32), dorsal posterior cingulate cortex (dPCC, BA 31), angular gyrus (BA 39), and supramarginal gyrus (BA 40). Notably, the angular gyrus and supramarginal gyrus are part of the inferior parietal lobule (IPL), which has been consistently reported associated with attention deficits in psychiatric patients [178, 218]. This aligns with the clinical observation that individuals in the identified community display higher (worse) scores in poor attention symptoms. Additionally, Paulus et al. [198] reported significantly greater activation of the supramarginal gyrus in patients compared to controls during a decision-making study, further supporting its role in attention-related deficits. Dysfunction in the antPFC has also been frequently linked to psychiatric disorders, particularly negative symptoms such as avolition (lack of motivation) [235]. This is consistent with the observed higher scores for disturbance of volition in the identified community. Notably, the identified common community exhibits more severe symptoms compared to the community in **II**. This increased severity may be linked to dysfunctions observed in the default mode network (DMN), which includes regions such as the posterior cingulate cortex and bilateral inferior parietal lobule. The DMN has been reported to positively correlate with internally-directed thought processes, including future planning and affective decision-making, while negatively correlating with task-related processes [38, 232, 261]. Mingoia et al. reported increased functional connectivity within the DMN in individuals with schizophrenia from resting-state fMRI [179].

In the other detected common community, shown in Figure 2.11 **II** and highlighted by a red square, subjects in this group displayed significantly higher activation ($p = 6.28 \times 10^{-4}$) in the dorsolateral prefrontal cortex (dlPFC, BA 46, BA 9), middle temporal gyrus (BA 21), primary sensory cortex (BA 1), superior parietal lobule (BA 7), and Broca's operculum (BA 44). These brain regions are closely associated with various psychotic disorders such as schizophrenia. The dlPFC, in particular, plays a critical role in complex cognitive functions such as working memory, planning, and decision-making [285, 200]. Dysfunctional local connectivity in the dlPFC has been linked to psychiatric disorders, where patients often exhibit pronounced deficits in working

memory [196]. Alterations in the dlPFC have also been associated with negative symptoms in schizophrenia [257]. This dysfunctionality may contribute to the observed higher (worse) negative symptom scores, such as difficulty in abstract thinking, in subjects within this common community.

2.8 Conclusions

In this chapter, we proposed a multiplex community detection method based on ONMTF. The proposed method, MX-ONMTF, is able to detect both common and private communities across layers, allowing us to differentiate between the topologies across layers. The proposed algorithm is based on multiplicative update rules and a proof of convergence is provided, along with an in-depth analysis of the algorithm, including studies of overfitting and ablation, recovery guarantees and consistency. A new approach based on the eigengap criterion is introduced for determining the number of communities. Results for both synthetic and real-world networks show that our method performs better than existing community detection methods for multiplex networks as it is able to handle the heterogeneity of the network topology across layers. Moreover, experiments on multiview networks show that our method also performs well in cases where a consensus community structure is needed. In addition, MX-ONMTF is applied to an fMRI dataset where the nodes are 464 psychotic patients and the layers represent different functional areas of the brain, identifying subgroups of subjects that exhibit significant differences in key functional areas, such as the default mode network (DMN) and anterior prefrontal cortex (antPFC), as well as in their corresponding clinical scores. These findings align with prior clinical studies, demonstrating the ability of the proposed approach to uncover clinically relevant subgroups and enhance understanding of psychotic disorder heterogeneity.

CHAPTER 3

DISCRIMINATIVE COMMUNITY DETECTION FOR MULTIPLEX NETWORKS

3.1 Introduction

A multiplex network is a multilayer network where all layers share the same set of nodes with edges representing different interactions [140]. These multiplex networks model complex systems like living organisms, human societies, and transportation systems [229]. Community detection is a core task in network analysis, where communities are defined as groups of nodes that are more densely connected to each other than they are to the rest of the network [92]. Detecting the community structure is useful for understanding the structure and function of complex networks.

In a lot of applications, one may have multiplex networks constructed from two different datasets. For example, in the study of brain networks through multiplex representations [67], each layer may correspond to a different subject, and each group may correspond to a different population, e.g., healthy vs. disease. In these settings, one is interested not only in the community structure of each multiplex network but also in the network components, i.e., communities, that discriminate between the two groups. Moreover, in a lot of cases the multiplex network structure may be changing with time leading to temporal multiplex networks. For example, recent studies show that functional brain networks are dynamic with the topology evolving with time, so one could also be interested in identifying network components that evolve differently across two groups over time.

In this chapter, we introduce a discriminative community detection approach based on spectral clustering for detecting community structures that distinguish between two multiplex networks in both static and dynamic cases. In particular, we introduce three different formulations. The first approach, Multiplex Discriminative Spectral Clustering (MX-DSC), focuses on minimizing the normalized cut of the difference between the two groups with a regularization term that ensures that the projection distance between the discriminative subspaces is maximized. The second method, Multiplex Discriminative and Consensus Spectral Clustering (MX-DCSC), extends this approach by simultaneously learning consensus, discriminative, and individual layerwise subspaces across both groups. And, the third method, Discriminative Subgraph Detection for Temporal Multiplex

Networks (TMX-DiSG), identifies discriminative subgraphs between two temporal multiplex networks by minimizing the normalized cut over time, incorporating regularization terms to maximize projection distance and ensure temporal smoothness. These methods are evaluated on synthetic and real multiplex and temporal multiplex networks, including EEG and dynamic fMRI functional brain networks, comparing across experimental conditions and tasks.

3.1.1 Related Work

The problem of community detection in multiplex networks is closely tied to the literature in multiview clustering [46], which deals with the problem of clustering the data points given multiple sets of features. The main approach to multiview clustering is to optimize the objective function to find the best clustering solution for the given data with N samples and m views, yielding a membership matrix $\mathbf{H} \in \mathbb{R}^{N \times k}$ that indicates group membership. Some examples of this approach include multiview spectral clustering [77], multiview subspace clustering [37], multiview NMF clustering [161] and canonical correlation analysis based methods [47]. The methods proposed here are most similar to multiview spectral clustering, which constructs a similarity matrix and minimizes the normalized cut between clusters. However, existing multiview spectral clustering methods focus on learning either consensus or both layer-specific and consensus cluster structures [144]. Thus, there is no direct emphasis on differentiating between two groups of multiview data.

Another class of methods that are closely related to the proposed frameworks are contrastive principal component analysis (cPCA) [2] and discriminative principal component analysis (dPCA) [50]. These methods deal with the dimensionality reduction problem similar to PCA. However, unlike PCA which copes with one dataset at a time, they analyze multiple datasets jointly. They extract the most discriminative information from one dataset of particular interest, i.e., target data, relative to the other(s), i.e., background data. The method proposed in this chapter can be thought of as an extension of cPCA and dPCA from the Euclidean domain to the graph domain, where the discriminative subspaces now correspond to the discriminative community structure.

The problem of learning from multiple datasets has also been addressed in the area of brain imaging, where the increasing availability of data across multiple tasks in recent years provides

complementary information [236]. Methods that jointly analyze multiple datasets can leverage their complementary information, improving overall learning performance. However, a key challenge in analyzing these datasets is to distinguish between shared (joint) and unique (discriminative) patterns. Traditional methods in this domain focus on matrix and tensor decompositions [287] which use latent variable models that capture the interactions among the multiple datasets [5, 134]. Extensions of independent component analysis (ICA) such as linked ICA [105], tensor ICA [17], independent vector analysis (IVA) [4], group ICA [43] and multi-paradigm sparse tensor decomposition [287] have been used for multimodal data fusion, multi-subject, and multiple task fMRI analysis. More recently, deep learning methods have been employed for learning from multi-modal and multi-task fMRI data [279]. While these methods extract useful biomarkers, their lack of interpretability, especially as network depth increases, limits their usefulness. In recent work, a supervised dictionary learning method has been proposed for multi-subject fMRI data analysis to extract brain activation maps that are common and discriminative across different groups of subjects [127].

However, these methods still have shortcomings. First, most of the current methods focus on finding the common information across different networks, i.e., data fusion. Second, existing methods focus on the whole brain fMRI time series data and not the actual network. Thus, they cannot particularly answer how the topological organization of the network changes between two tasks or populations. Finally, the current methods aggregate the data over time, thus temporal variation of the common and unique components cannot be determined.

3.2 Background

3.2.1 Multiplex Network Community Detection

Multiplex networks can be represented using a finite sequence of graphs $\{\mathcal{G}_l\}$, where $l \in \{1, 2, \ldots, L\}$, $\mathcal{G}_l = (V, E_l, \mathbf{A}_l)$ [62]. V is the set of nodes which is the same for all layers, E_l and $\mathbf{A}_l \in \mathbb{R}^{N \times N}$ are the edges set and the adjacency matrix for layer l, respectively. A large group of community detection methods for multiplex networks aim to find a consensus community structure across all layers by first merging the layers and then applying a single-layer community

detection algorithm to the aggregated networks. When the networks are aggregated through the mean operation, the trace minimization problem in Eq. (1.8) can be written as [144]:

$$\underset{\mathbf{U}_*\in\mathbb{R}^{N\times k},\mathbf{U}_*^\top\mathbf{U}_*=\mathbf{I}}{\text{minimize}} \quad \text{tr}\Big(\mathbf{U}_*^\top \sum_{l=1}^{L} \mathbf{L}_l\mathbf{U}_*\Big), \tag{3.1}$$

where $\mathbf{L}_l \in \mathbb{R}^{N\times N}$ is the graph Laplacian matrix for layer l. The goal is to find a subspace $\mathbf{U}^*$ that is representative of all the layers in the multiplex network, and the consensus community structure can be found by applying k-means to this $\mathbf{U}_*$.

Multiplex networks that vary over time can be represented as $\mathcal{G}_{l,t} = (V, E_{l,t}, \mathbf{A}_{l,t}\}$ where V is the set of nodes, $E_{l,t}$ is the edge set and $\mathbf{A}_{l,t} \in \mathbb{R}^{N\times N}$ is the adjacency matrix for layer l and time t with $l \in \{1,\dots,L\}$ and $t \in \{1,\dots,T\}$. For multiplex temporal networks, the low-dimensional spectral embedding representative of all the layers at time t, $\mathbf{U}_{*,t}$, can be obtained by applying the trace minimization problem in Eq. (1.8) to layer aggregated Laplacian matrices as follows [144]:

$$\underset{\mathbf{U}_{*,t}\in\mathbb{R}^{N\times k},\mathbf{U}_{*,t}^\top\mathbf{U}_{*,t}=\mathbf{I}}{\text{minimize}} \quad \text{tr}\Big(\mathbf{U}_{*,t}^\top \sum_{l=1}^{L} \mathbf{L}_{l,t}\mathbf{U}_{*,t}\Big). \tag{3.2}$$

3.3 Discriminative Community Detection Methods

Given two multiplex networks $\mathcal{G}_l^1 = (V^1, E_l^1, \mathbf{A}_l^1)$ and $\mathcal{G}_m^2 = (V^2, E_m^2, \mathbf{A}_m^2)$ with $l \in \{1,2,\dots,L\}$ and $m \in \{1,2,\dots,M\}$ and graph Laplacians $\mathbf{L}_l^1$ and $\mathbf{L}_m^2$, the goal is to extract two embedding subspaces, $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$, that discriminate between the two multiplex networks with respect to the other. We propose two formulations for achieving this goal.

3.3.1 Multiplex Discriminative Spectral Clustering (MX-DSC)

In the first approach, we focus on obtaining $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$ that discriminate between the groups. Let $\bar{\mathbf{U}}^1 \in \mathbb{R}^{N\times\bar{k}_1}$ be the embedding subspace that minimizes the normalized cut of the first group, i.e., minimize $\text{tr}(\bar{\mathbf{U}}^{1\top}(\sum_{l=1}^{L}\mathbf{L}_l^1)\bar{\mathbf{U}}^1)$, while maximizing the second group's normalized cut, i.e., maximizing $\text{tr}(\bar{\mathbf{U}}^{1\top}(\sum_{m=1}^{M}\mathbf{L}_m^2)\bar{\mathbf{U}}^1)$. These two goals can be simultaneously satisfied through the following optimization:

$$\underset{\bar{\mathbf{U}}^1,\bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^1=\mathbf{I}}{\text{minimize}} \quad \text{tr}\Big(\bar{\mathbf{U}}^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1\Big)\bar{\mathbf{U}}^1\Big) - \alpha\text{tr}\Big(\bar{\mathbf{U}}^{1\top}\Big(\sum_{m=1}^{M}\mathbf{L}_m^2\Big)\bar{\mathbf{U}}^1\Big),$$

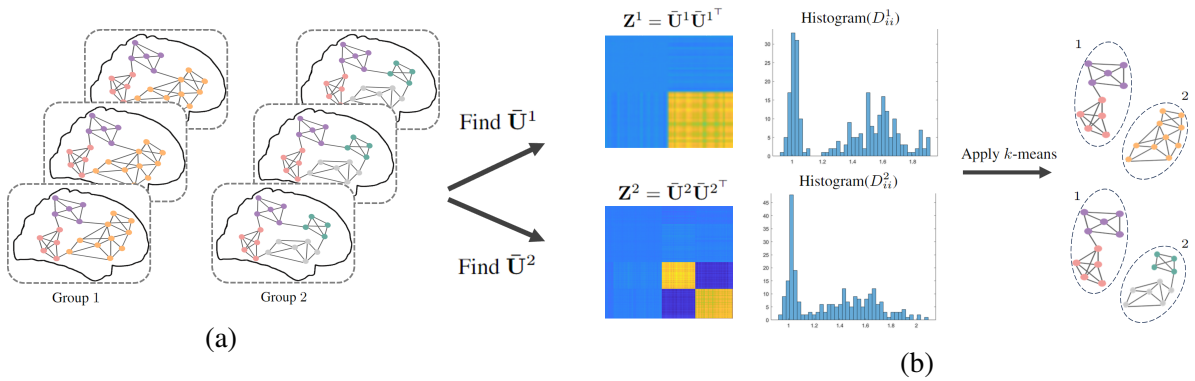


Figure 3.1: Overview of the proposed multiplex discriminative community detection methods, MX-DSC and MX-DCSC. (a) Toy example of two multiplex networks with shared (non-discriminative) communities colored in pink and purpe, and discriminative communities colored in orange, green and gray. (b) Discriminative embedding matrices are learned for each group, $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$. k-means with $k = 2$ is applied to the degrees of $|\mathbf{Z}^1|$ and $|\mathbf{Z}^2|$ to separate the nodes in the shared and in the discriminative subspace, respectively.

where $\bar{\mathbf{U}}^1$ captures what is discriminative in the first multiplex network with respect to the other. Similarly, we can define $\bar{\mathbf{U}}^2$ as the embedding matrix that contains information about the discriminative subspace of the second multiplex network with respect to the first.

Considering the two embedding matrices jointly, $\bar{\mathbf{U}}^1 \in \mathbf{R}^{N\times\bar{k}_1}$ and $\bar{\mathbf{U}}^2 \in \mathbf{R}^{N\times\bar{k}_2}$, results in

$$\begin{aligned}\underset{\bar{\mathbf{U}}^1,\bar{\mathbf{U}}^2}{\text{minimize}} \quad & \operatorname{tr}\Big(\bar{\mathbf{U}}^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 - \alpha\sum_{m=1}^{M}\mathbf{L}_m^2\Big)\bar{\mathbf{U}}^1\Big) + \operatorname{tr}\Big(\bar{\mathbf{U}}^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_m^2 - \alpha\sum_{l=1}^{L}\mathbf{L}_l^1\Big)\bar{\mathbf{U}}^2\Big) \\ & +\gamma_1\operatorname{tr}\Big(\bar{\mathbf{U}}^1\bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^2\bar{\mathbf{U}}^{2\top}\Big), \qquad \text{s.t. } \bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^1 = \mathbf{I}, \bar{\mathbf{U}}^{2\top}\bar{\mathbf{U}}^2 = \mathbf{I},\end{aligned} \tag{3.3}$$

where the first term determines $\bar{\mathbf{U}}^1$ that discriminates the first multiplex network from the second, the second term defines $\bar{\mathbf{U}}^2$ that discriminates the second network from the first, and the last term is a regularization that maximizes the projection distance between $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$. The hyperparameters α and γ_1 control the level of discrimination and the dissimilarity between the two subspaces, respectively.

The optimization problem in (3.7) can be solved in an alternating manner, first solving for $\bar{\mathbf{U}}^1$ and then for $\bar{\mathbf{U}}^2$.

$$
\begin{aligned}
\bar{\mathbf{U}}^{1(k+1)} &:= \underset{\bar{\mathbf{U}}^1 \in \mathbb{R}^{N\times k^1}}{\operatorname{argmin}} \operatorname{tr}(\bar{\mathbf{U}}^{1\top}(\sum_{l=1}^{L}\mathbf{L}_l^1 - \alpha\sum_{m=1}^{M}\mathbf{L}_m^2 + \gamma_1\bar{\mathbf{U}}^{2(k)}\bar{\mathbf{U}}^{2(k)\top})\bar{\mathbf{U}}^1), \\
\bar{\mathbf{U}}^{2(k+1)} &:= \underset{\bar{\mathbf{U}}^2 \in \mathbb{R}^{N\times k^2}}{\operatorname{argmin}} \operatorname{tr}(\bar{\mathbf{U}}^{2\top}(\sum_{m=1}^{M}\mathbf{L}_m^2 - \alpha\sum_{l=1}^{L}\mathbf{L}_l^1 + \gamma_1\bar{\mathbf{U}}^{1(k+1)}\bar{\mathbf{U}}^{1(k+1)\top})\bar{\mathbf{U}}^2) \\
&\text{s.t. } \bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^1 = \mathbf{I}, \bar{\mathbf{U}}^{2\top}\bar{\mathbf{U}}^2 = \mathbf{I}.
\end{aligned} \tag{3.4}
$$

The solution to updating $\bar{\mathbf{U}}^1$ is the eigenvectors corresponding to the k^1 smallest eigenvalues of $(\sum_{l=1}^{L}\mathbf{L}_l^1 - \alpha\sum_{m=1}^{M}\mathbf{L}_m^2 + \gamma_1\bar{\mathbf{U}}^2\bar{\mathbf{U}}^{2\top})$, which is the global optimum solution to the $\bar{\mathbf{U}}^1$ sub-problem in (3.8) [34]. The solution for $\bar{\mathbf{U}}^2$ can be found in a similar manner, and it is the global optimum for the $\bar{\mathbf{U}}^2$ sub-problem in (3.8). We solve iteratively for both variables until convergence.

3.3.2 Multiplex Discriminative and Consensus Spectral Clustering (MX-DCSC)

In this section, we propose a formulation where we learn both the discriminative subspaces between groups, $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$, while also learning the consensus subspaces, $\mathbf{U}_*^1 \in \mathbf{R}^{N\times k^1}$ and $\mathbf{U}_*^2 \in \mathbf{R}^{N\times k^2}$ and the individual layerwise embeddings, $\mathbf{U}_l^1 \in \mathbf{R}^{N\times k^1}$ and $\mathbf{U}_m^2 \in \mathbf{R}^{N\times k^2}$, within each group.

For the discriminative part, we propose to use a variation of Eq. (3.7), where we find $\bar{\mathbf{U}}^1$ that captures what is discriminative in the first multiplex network with respect to the other. We can define the squared projection distance between the target representative subspace $\bar{\mathbf{U}}^1$ of the first group and the individual subspaces of the second group, $\mathbf{U}_m^2$ as in [77]

$$
\begin{aligned}
d_{proj}^2(\bar{\mathbf{U}}^1, \{\mathbf{U}_m^2\}_{m=1}^M) &= \sum_{m=1}^{M} d_{proj}^2(\bar{\mathbf{U}}^1, \mathbf{U}_m^2) = \sum_{m=1}^{M}(k - \operatorname{tr}(\bar{\mathbf{U}}^1\bar{\mathbf{U}}^{1\top}\mathbf{U}_m^2\mathbf{U}_m^{2\top})) \\
&= kM - \operatorname{tr}(\bar{\mathbf{U}}^1\bar{\mathbf{U}}^{1\top}\mathbf{U}_m^2\mathbf{U}_m^{2\top}).
\end{aligned}
$$

We want to find a $\bar{\mathbf{U}}^1$ that minimizes the trace in Eq. (3.2) for the graph Laplacians of its group while maximizing its projection distance with the individual subspaces of the second group. Combining these two goals yields the following cost function

$$
\mathcal{L}_{\text{dis}}(\bar{\mathbf{U}}^1) = \operatorname{tr}\Big(\bar{\mathbf{U}}^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 + \alpha\sum_{m=1}^{M}\mathbf{U}_m^2\mathbf{U}_m^{2\top}\Big)\bar{\mathbf{U}}^1\Big),
$$

where the regularization parameter α balances the trade-off between the two terms.

To learn the community structure of each layer, we use trace minimization corresponding to spectral clustering

$$\mathcal{L}_{\text{lw}}(\mathbf{U}_l^1) = \text{tr}(\mathbf{U}_l^{1\top}\mathbf{L}_l^1\mathbf{U}_l^1).$$

Finally, in order to capture the consensus community structure for each group we use the multiview spectral clustering formulation in [77]

$$\mathcal{L}_{\text{con}}(\mathbf{U}_*^1) = \text{tr}\Big(\mathbf{U}_*^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 - \beta\sum_{m=1}^{M}\mathbf{U}_l^1\mathbf{U}_l^{1\top}\Big)\mathbf{U}_*^1\Big).$$

Combining these three terms, $\mathcal{L}_{\text{dis}}$, $\mathcal{L}_{\text{lw}}$ and $\mathcal{L}_{\text{con}}$, for each group and the regularization term that maximizes the projection distance between $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$, we propose the following formulation for MX-DCSC, to find $\bar{\mathbf{U}}^1 \in \mathbb{R}^{N\times\bar{k}_1}$, $\bar{\mathbf{U}}^2 \in \mathbb{R}^{N\times\bar{k}_2}$, $\mathbf{U}_l^1 \in \mathbb{R}^{N\times k^1}$, $\mathbf{U}_m^2 \in \mathbb{R}^{N\times k^2}$, $\mathbf{U}_*^1 \in \mathbb{R}^{N\times k^1}$, and $\mathbf{U}_*^2 \in \mathbb{R}^{N\times k^2}$

$$\begin{aligned}
\underset{\bar{\mathbf{U}}^1,\bar{\mathbf{U}}^2,\mathbf{U}_l^1,\mathbf{U}_m^2,\mathbf{U}_*^1,\mathbf{U}_*^2}{\text{minimize}} \quad & \text{tr}\Big(\bar{\mathbf{U}}^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 + \alpha\sum_{m=1}^{M}\mathbf{U}_m^2\mathbf{U}_m^{2\top}\Big)\bar{\mathbf{U}}^1\Big) + \text{tr}\Big(\bar{\mathbf{U}}^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_m^2 + \alpha\sum_{l=1}^{L}\mathbf{U}_l^1\mathbf{U}_l^{1\top}\Big)\bar{\mathbf{U}}^2\Big) \\
& + \gamma_1\text{tr}\Big(\bar{\mathbf{U}}^1\bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^2\bar{\mathbf{U}}^{2\top}\Big) + \sum_{l=1}^{L}\text{tr}\Big(\mathbf{U}_l^{1\top}\mathbf{L}_l^1\mathbf{U}_l^1\Big) + \sum_{m=1}^{M}\text{tr}\Big(\mathbf{U}_m^{2\top}\mathbf{L}_m^2\mathbf{U}_m^2\Big) \\
& + \text{tr}\Big(\mathbf{U}_*^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 - \beta\sum_{l=1}^{L}\mathbf{U}_l^1\mathbf{U}_l^{1\top}\Big)\mathbf{U}_*^1\Big) + \text{tr}\Big(\mathbf{U}_*^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_m^2 - \beta\sum_{m=1}^{M}\mathbf{U}_m^2\mathbf{U}_m^{2\top}\Big)\mathbf{U}_*^2\Big), \\
\text{s.t. } & \bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^1 = \mathbf{I}, \bar{\mathbf{U}}^{2\top}\bar{\mathbf{U}}^2 = \mathbf{I}, \mathbf{U}_l^{1\top}\mathbf{U}_l^1 = \mathbf{I}, \mathbf{U}_m^{2\top}\mathbf{U}_m^2 = \mathbf{I}, \text{for } l = 1, 2, \cdots, L \text{ and } m = 1, 2, \cdots, M.
\end{aligned} \tag{3.5}$$

The optimization problem in (3.5) can be solved in an alternating manner as follows

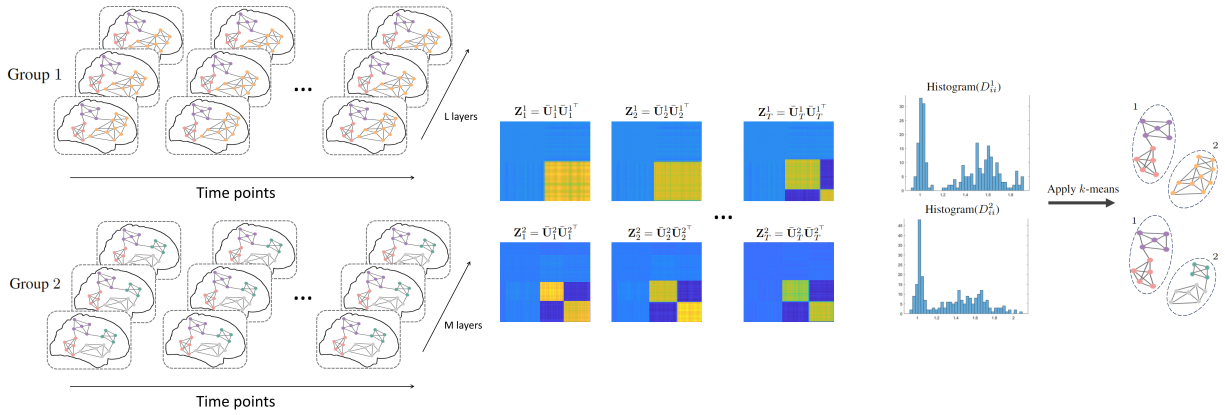
$$\begin{aligned}
\bar{\mathbf{U}}^{1(k+1)} &:= \underset{\bar{\mathbf{U}}^1}{\text{argmin}} \ \text{tr}\Big(\bar{\mathbf{U}}^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 + \alpha\sum_{m=1}^{M}\mathbf{U}_m^{2(k)}\mathbf{U}_m^{2(k)\top} + \gamma_1\bar{\mathbf{U}}^{2(k)}\bar{\mathbf{U}}^{2(k)\top}\Big)\bar{\mathbf{U}}^1\Big), \\
\bar{\mathbf{U}}^{2(k+1)} &:= \underset{\bar{\mathbf{U}}^2}{\text{argmin}} \ \text{tr}\Big(\bar{\mathbf{U}}^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_m^2 + \alpha\sum_{l=1}^{L}\mathbf{U}_l^{1(k)}\mathbf{U}_l^{1(k)\top} + \gamma_1\bar{\mathbf{U}}^{1(k)}\bar{\mathbf{U}}^{1(k)\top}\Big)\bar{\mathbf{U}}^2\Big), \\
\mathbf{U}_l^{1(k+1)} &:= \underset{\mathbf{U}_l^1}{\text{argmin}} \ \text{tr}\Big(\mathbf{U}_l^{1\top}\Big(\mathbf{L}_l^1 + \alpha\bar{\mathbf{U}}^{2(k+1)}\bar{\mathbf{U}}^{2(k+1)\top} - \beta\mathbf{U}_*^{1(k)}\mathbf{U}_*^{1(k)\top}\Big)\mathbf{U}_l^1\Big), \\
\mathbf{U}_m^{2(k+1)} &:= \underset{\mathbf{U}_m^2}{\text{argmin}} \ \text{tr}\Big(\mathbf{U}_m^{2\top}\Big(\mathbf{L}_m^2 + \alpha\bar{\mathbf{U}}^{1(k+1)}\bar{\mathbf{U}}^{1(k+1)\top} - \beta\mathbf{U}_*^{2(k)}\mathbf{U}_*^{2(k)\top}\Big)\mathbf{U}_m^2\Big),
\end{aligned}$$

$$\mathbf{U}_*^{1(k+1)} := \underset{\mathbf{U}_*^1}{\text{argmin}} \ \text{tr}\Big(\mathbf{U}_*^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_l^1 - \beta\sum_{l=1}^{L}\mathbf{U}_l^{1(k+1)}\mathbf{U}_l^{1(k+1)\top}\Big)\mathbf{U}_*^1\Big),$$
$$\mathbf{U}_*^{2(k+1)} := \underset{\mathbf{U}_*^2}{\text{argmin}} \ \text{tr}\Big(\mathbf{U}_*^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_m^2 - \beta\sum_{m=1}^{M}\mathbf{U}_m^{2(k+1)}\mathbf{U}_m^{2(k+1)\top}\Big)\mathbf{U}_*^2\Big),$$

$$\text{s.t. } \bar{\mathbf{U}}^{1\top}\bar{\mathbf{U}}^1 = \mathbf{I}, \bar{\mathbf{U}}^{2\top}\bar{\mathbf{U}}^2 = \mathbf{I}, \mathbf{U}_l^{1\top}\mathbf{U}_l^1 = \mathbf{I}, \mathbf{U}_m^{2\top}\mathbf{U}_m^2 = \mathbf{I}, \text{for } l = 1, 2, \cdots, L \text{ and } m = 1, 2, \cdots, M. \tag{3.6}$$

3.3.3 Discriminative Subgraph Detection for Temporal Multiplex Networks (TMX-DiSG)

Given two temporal multiplex networks $\mathcal{G}_{l,t}^1 = (V^1, E_{l,t}^1, \mathbf{A}_{l,t}^1)$ and $\mathcal{G}_{m,t}^2 = (V^2, E_{m,t}^2, \mathbf{A}_{m,t}^2)$ at time t for $l \in \{1, 2, \ldots, L\}$ and $m \in \{1, 2, \ldots, M\}$, and graph Laplacians $\mathbf{L}_{l,t}^1$ and $\mathbf{L}_{m,t}^2$, the goal is to find two spectral embedding matrices, $\bar{\mathbf{U}}_t^1$ and $\bar{\mathbf{U}}_t^2$, that discriminate between the two temporal multiplex networks at each time t. Figure 3.2a shows a toy example of two temporal-multiplex networks.



(a) (b) (c)

Figure 3.2: Overview of TMX-DiSG (a) Illustrative example of a temporal multiplex network (b) Discriminative embedding matrices,$\bar{\mathbf{U}}_t^1$ and $\bar{\mathbf{U}}_t^2$, are learned for each group at each time point t (c) k-means with $k = 2$ is applied to the degrees of $|\mathbf{Z}_t^1|$ and $|\mathbf{Z}_t^2|$ to separate the nodes in the shared and the discriminative subspaces, respectively.

Let $\bar{\mathbf{U}}_t^1 \in \mathbb{R}^{N\times\bar{k}_t^1}$ be the embedding subspace that represents the first group, i.e., minimize $\text{tr}(\bar{\mathbf{U}}_t^{1\top}(\sum_{l=1}^{L}\mathbf{L}_{l,t}^1)\bar{\mathbf{U}}_t^1)$, while maximizing its distinction from the second group, i.e., maximize $\text{tr}(\bar{\mathbf{U}}_t^{1\top}(\sum_{m=1}^{M}\mathbf{L}_{m,t}^2)\bar{\mathbf{U}}_t^1)$. These two goals can be simultaneously satisfied through the following

optimization:

$$\underset{\bar{\mathbf{U}}_t^1, \bar{\mathbf{U}}_t^{1\top}\bar{\mathbf{U}}_t^1 = \mathbf{I}}{\text{minimize}} \operatorname{tr}\Big(\bar{\mathbf{U}}_t^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_{l,t}^1\Big)\bar{\mathbf{U}}_t^1\Big) - \alpha\operatorname{tr}\Big(\bar{\mathbf{U}}_t^{1\top}\Big(\sum_{m=1}^{M}\mathbf{L}_{m,t}^2\Big)\bar{\mathbf{U}}_t^1\Big).$$

Similarly, we can define $\bar{\mathbf{U}}_t^2 \in \mathbb{R}^{N\times\bar{k}_t^2}$ as the embedding matrix that contains information about the discriminative subspace of the second group with respect to the first.

Considering the two embedding matrices jointly, $\bar{\mathbf{U}}_t^1$ and $\bar{\mathbf{U}}_t^2$, and introducing some regularizations, we propose the following optimization problem:

$$\begin{aligned}\underset{\bar{\mathbf{U}}_t^1, \bar{\mathbf{U}}_t^2}{\text{minimize}} \quad & \sum_{t=1}^{T}\operatorname{tr}\Big(\bar{\mathbf{U}}_t^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_{l,t}^1 - \alpha\sum_{m=1}^{M}\mathbf{L}_{m,t}^2\Big)\bar{\mathbf{U}}_t^1\Big) + \sum_{t=1}^{T}\operatorname{tr}\Big(\bar{\mathbf{U}}_t^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_{m,t}^2 - \alpha\sum_{l=1}^{L}\mathbf{L}_{l,t}^1\Big)\bar{\mathbf{U}}_t^2\Big) \\ & +\gamma_1\sum_{t=1}^{T}\operatorname{tr}\Big(\bar{\mathbf{U}}_t^1\bar{\mathbf{U}}_t^{1\top}\bar{\mathbf{U}}_t^2\bar{\mathbf{U}}_t^{2\top}\Big) - \gamma_2\sum_{t=2}^{T}\operatorname{tr}\Big(\bar{\mathbf{U}}_t^1\bar{\mathbf{U}}_t^{1\top}\bar{\mathbf{U}}_{t-1}^1\bar{\mathbf{U}}_{t-1}^{1\top}\Big) - \gamma_3\sum_{t=2}^{T}\operatorname{tr}\Big(\bar{\mathbf{U}}_t^2\bar{\mathbf{U}}_t^{2\top}\bar{\mathbf{U}}_{t-1}^2\bar{\mathbf{U}}_{t-1}^{2\top}\Big), \\ & \text{s.t. } \bar{\mathbf{U}}_t^{1\top}\bar{\mathbf{U}}_t^1 = \mathbf{I}, \bar{\mathbf{U}}_t^{2\top}\bar{\mathbf{U}}_t^2 = \mathbf{I}, \text{for } t = 1, 2, \ldots, T,\end{aligned} \tag{3.7}$$

where the first term determines the subspace $\bar{\mathbf{U}}_t^1$ that discriminates the first group of networks from the second at time t, the second term determines $\bar{\mathbf{U}}_t^2$ that discriminates the second group from the first at time t, the third term is a regularization that maximizes the projection distance between $\bar{\mathbf{U}}_t^1$ and $\bar{\mathbf{U}}_t^2$, thus ensuring that the overlap between the two spectral embeddings is minimized, and the last two terms enforce temporal smoothness of the subspaces.

The optimization problem in (3.7) can be solved in an alternating manner, first solving for $\bar{\mathbf{U}}_t^1$ and then for $\bar{\mathbf{U}}_t^2$:

$$\begin{aligned}\bar{\mathbf{U}}_t^{1(k+1)} &:= \underset{\bar{\mathbf{U}}_t^1, \bar{\mathbf{U}}_t^{1\top}\bar{\mathbf{U}}_t^1 = \mathbf{I}}{\operatorname{argmin}} \operatorname{tr}\Big(\bar{\mathbf{U}}_t^{1\top}\Big(\sum_{l=1}^{L}\mathbf{L}_{l,t}^1 - \alpha\sum_{m=1}^{M}\mathbf{L}_{m,t}^2 + \gamma_1\bar{\mathbf{U}}_t^{2(k)}\bar{\mathbf{U}}_t^{2(k)\top} - \gamma_2\bar{\mathbf{U}}_{t-1}^1\bar{\mathbf{U}}_{t-1}^{1\top}\Big)\bar{\mathbf{U}}_t^1\Big), \\ \bar{\mathbf{U}}_t^{2(k+1)} &:= \underset{\bar{\mathbf{U}}_t^2, \bar{\mathbf{U}}_t^{2\top}\bar{\mathbf{U}}_t^2 = \mathbf{I}}{\operatorname{argmin}} \operatorname{tr}\Big(\bar{\mathbf{U}}_t^{2\top}\Big(\sum_{m=1}^{M}\mathbf{L}_{m,t}^2 - \alpha\sum_{l=1}^{L}\mathbf{L}_{l,t}^1 + \gamma_1\bar{\mathbf{U}}_t^{1(k)}\bar{\mathbf{U}}_t^{1(k)\top} - \gamma_3\bar{\mathbf{U}}_{t-1}^2\bar{\mathbf{U}}_{t-1}^{2\top}\Big)\bar{\mathbf{U}}_t^2\Big),\end{aligned} \tag{3.8}$$

The solution to updating $\bar{\mathbf{U}}_t^1$ is the eigenvectors corresponding to the k_t^1 smallest eigenvalues of $\Big(\sum_{l=1}^{L}\mathbf{L}_{l,t}^1 - \alpha\sum_{m=1}^{M}\mathbf{L}_{m,t}^2 - \gamma_1\bar{\mathbf{U}}_t^2\bar{\mathbf{U}}_t^{2\top} - \gamma_2\bar{\mathbf{U}}_{t-1}^1\bar{\mathbf{U}}_{t-1}^{1\top}\Big)$, which is the global optimum solution to the $\bar{\mathbf{U}}_t^1$ sub-problem in (3.8) [34]. The solution for $\bar{\mathbf{U}}_t^2$ can be found in a similar manner, and is the global optimum for the $\bar{\mathbf{U}}_t^2$ sub-problem in (3.8). We solve iteratively for both variables until convergence.

3.3.4 Finding the embedding dimensions

In most clustering algorithms, the number of communities (k) is an input parameter. This is typically addressed by testing different k values and selecting the best based on a performance metric. In this chapter, the embedding dimensions, i.e., discriminative and consensus, are determined following the eigengap rule and hierarchical clustering-based method proposed in Chapter 2. First, we compute embedding matrices $\mathbf{U}^1 \in \mathbf{R}^{N\times k^1}$ and $\mathbf{U}^2 \in \mathbf{R}^{N\times k^2}$ for each group, with k^1 and k^2 found using the eigengap rule on the graph Laplacian. We then concatenate the embeddings from both groups as $\mathbf{X} = [\mathbf{U}^1, \mathbf{U}^2]$ and apply a hierarchical clustering algorithm to the columns of $\mathbf{X}$, grouping similar eigenvectors that represent shared structure between the groups. The number of clusters corresponding to shared components is denoted as k_c. Finally, we compute the dimensions of the discriminative subspaces, $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^1$, by subtracting the shared components from the original embedding dimensions $\bar{k}^1 = k^1 - k_c$, $\bar{k}^2 = k^2 - k_c$. Thus, the final embeddings $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$ capture only the distinctive features that differentiate between the two groups. For MX-DSC and MX-DCSC, the embedding process is computed once, producing dimensions $\bar{k}^1$, $\bar{k}^2$, k^1, and k^2. For TMX-DiSG, this process is repeated at each time step t, leading to time-dependent embeddings $\bar{\mathbf{U}}^1_t$ and $\bar{\mathbf{U}}^2_t$, and dimensions $\bar{k}^1_t$, $\bar{k}^2_t$

3.3.5 Subgraph Identification

Once the low-dimensional discriminative subspaces, $\bar{\mathbf{U}}^1$ and $\bar{\mathbf{U}}^2$, are learned, we can construct $N \times N$ matrices, $\mathbf{Z}^1 = \bar{\mathbf{U}}^1\bar{\mathbf{U}}^{1\top}$ and $\mathbf{Z}^2 = \bar{\mathbf{U}}^2\bar{\mathbf{U}}^{2\top}$ that capture the discriminative subgraphs for each group, as shown in the toy example in Figure 3.1. We compute the degrees of nodes in both groups as $D^1_i = \sum_j |Z^1_{ij}|$ and $D^2_i = \sum_j |Z^2_{ij}|$. As seen in the histograms in Figure 3.1, there are two groups of nodes with different degree distributions. These two clusters, discriminative and non-discriminative nodes, in each group, are identified by k-means with $k = 2$ applied to $\mathbf{D}^1$ and $\mathbf{D}^2$, where the cluster with the low degree nodes corresponds to the non-discriminative structure, and the cluster with the high degree nodes correspond to the discriminative subgraph. For TMX-DiSG, this process is repeated at each time step t, using the time-dependent embeddings $\bar{\mathbf{U}}^1_t$ and $\bar{\mathbf{U}}^2_t$.

3.3.6 Computational Complexity

The computational complexity of the algorithm is mostly due to the eigendecompositions at each iteration. At each iteration, we find the embeddings by computing the eigenvectors corresponding to the k smallest eigenvalues of and $N \times N$ matrix, which has a complexity of $O(N^2k)$. Therefore, the total complexity of the algorithm is dominated by $O(N^2 max\{k_t^1, k_t^2, \bar{k}_t^1, \bar{k}_t^2\})$.

3.4 Experiments: Multiplex Networks

In this section, MX-DSC and MX-DCSC are evaluated for both synthetic and real multiplex networks. For synthetic data, three experiments are conducted where different parameters of the simulated data are changed. MX-DSC is applied to two real multiplex networks datasets to find the corresponding discriminative subgraphs.

3.4.1 Synthetic Multiplex Networks

Multiplex benchmark networks based on the model described in [16, 124] were generated. First, a multilayer partition is generated with a user-defined number of nodes, layers, and an inter-layer dependency tensor specifying the layer relationships. Next, for the given multilayer partition, edges in each layer are generated following a degree-corrected block model [136] parameterized by the distribution of expected degrees and a community mixing parameter $\mu \in [0, 1]$ that controls the network modularity. When $\mu = 0$, all edges lie within communities, whereas $\mu = 1$ implies the edges are distributed uniformly. For multiplex networks, the probabilities in the inter-layer dependency tensor are the same for all pairs of layers and are specified by $p \in [0, 1]$. When $p = 0$, the partitions are independent across layers while $p = 1$ indicates an identical partition across layers.

In this work, we extend the model described above to generate two multiplex benchmark networks with shared (non-discriminative) and discriminative communities among them two. We first generate the shared communities by randomly selecting n_c nodes across all layers and for both groups and setting the inter-layer dependency probability to p_1. Next, we independently generate the discriminative communities for each group with the remaining nodes.

Evaluation: The performance of MX-DSC is evaluated based on the accuracy of detecting discriminative subgraphs, while MX-DCSC is assessed on both subgraph and community detection accuracy. In order to evaluate the performance in detecting the discriminative subgraphs, we use AUC-ROC as the evaluation metric. Three experiments with different parameters are repeated 50 times, and the average AUC-ROC for MX-DSC and MX-DCSC are reported in Table 3.2. We evaluate the accuracy of community detection in terms of Normalized Mutual Information (NMI) [65]. The accuracy of MX-DCSC, which learns the consensus community structure per each group, is compared with existing methods, as shown in Figure 3.3. SC-ML [77] is applied, combining both multiplex networks into one, and individually to each multiplex network, SC-ML$_{\text{ind}}$, and GenLouvain [182] to the combined networks. All the experiments are run with $\alpha, \beta, \gamma_1 \in [0, 1]$, and the results with the highest performance are reported.

Experiment 1: Varying Noise Level, μ: In this experiment, we generated two groups of multiplex networks with $N = 256$, 10 layers, and 6 and 5 communities per layer in each group, respectively. These two multiplex networks have two shared communities, and the number of discriminative communities per group is $k^1 = 4$ and $k^2 = 3$, respectively. To evaluate our algorithm's performance under different noise levels, these two networks are generated with varying values of the mixing parameter $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. The inter-layer dependency probability for the shared communities is $p_1 = 1$. Table 3.2 shows that both MX-DSC and MX-DCSC have high detection accuracy, with MX-DSC performing slightly better. From Figure 3.3, it can be seen that MX-DCSC outperforms existing community detection methods with increasing noise as the consensus community structure takes the discriminative information into account.

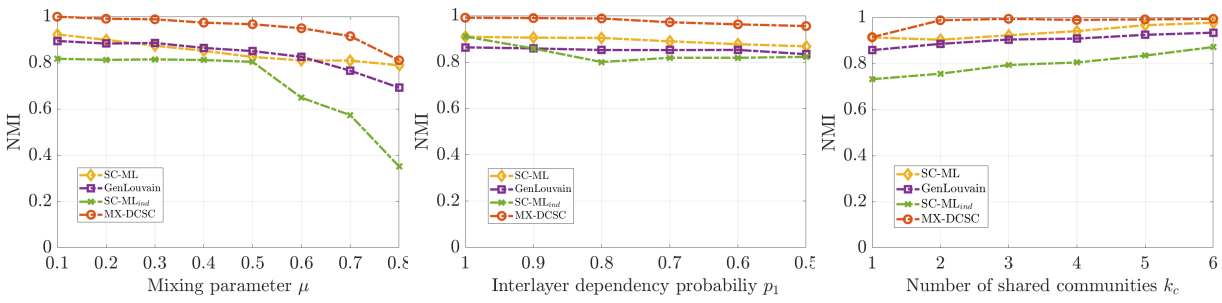
Experiment 2: Change in Variability, p_1: In the second experiment, we evaluated the robustness of the algorithm against variations in the shared community structure by fixing $\mu = 0.3$ and varying the inter-layer dependency probability p_1, i.e., the shared communities are allowed to vary across groups and layers. Table 3.2 shows that both methods are robust to variations in the shared community structure even when $p_1 = 0.5$, implying that the variations in the shared community structure between the two groups do not affect the discriminative subgraph detection accuracy.

Additionally, MX-DCSC performs best in community detection accuracy.

Experiment 3: Change in k_c: In this experiment, we evaluated the robustness of our method to the number of shared communities by fixing $\mu = 0.3$ and the total number of communities per layer, while varying k_c from 1 to 6. Both methods are robust to the value of k_c except for $k_c = 1$ since most of the nodes in the two groups will have different community structures, which makes it hard to detect the discriminative subgraphs. In terms of community detection accuracy, MX-DCSC performs well even when the two groups do not share a common community.

Table 3.1: Average AUC values for synthetic multiplex networks.

Experiment 1			Experiment 2			Experiment 3		
μ	MX-DSC	MX-DCSC	p_1	MX-DSC	MX-DCSC	k_c	MX-DSC	MX-DCSC
0.1	1.0000	1.0000	0.90	0.9980	0.9978	1	0.7052	0.7115
0.2	0.9864	0.9801	0.80	0.9940	0.9988	2	0.9812	0.9773
0.3	0.9763	0.9761	0.70	0.9992	0.9999	3	0.9941	0.9877
0.4	0.9683	0.9723	0.60	0.9956	0.9835	4	0.9802	0.9838
0.5	0.9652	0.9722	0.50	0.9821	0.9638	5	0.9949	0.9958
0.6	0.9398	0.9224				6	0.9991	0.9945
0.7	0.9167	0.8781						
0.8	0.8228	0.8118						

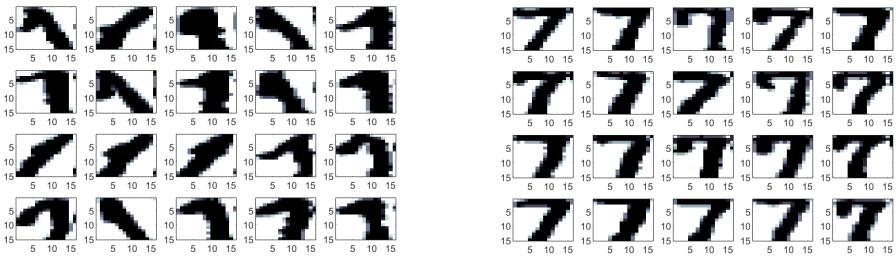


(a) (b) (c)

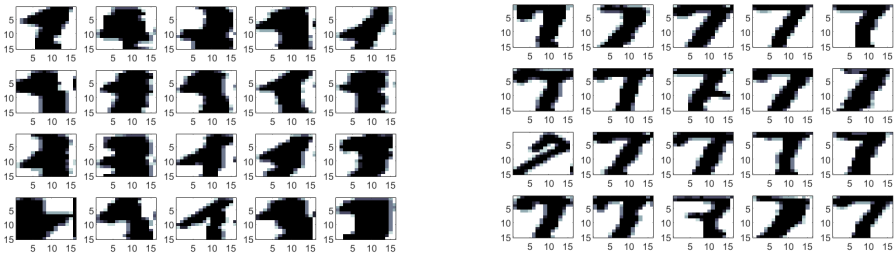
Figure 3.3: NMI Results for MX-DCSC with respect to other methods: (a) Experiment 1, (b) Experiment 2, (c) Experiment 3.

3.4.2 UCI Handwritten Dataset

In this section, we evaluate the performance of MX-DSC on a multiview dataset, UCI Handwritten Digits[1] [83]. The UCI Handwritten Digits dataset consists of features of handwritten digits from (0- 9) extracted from a collection of Dutch utility maps. There are a total of 2000 patterns that have been digitized in binary images, 200 patterns per digit. These digits are represented by six different feature sets: Fourier coefficients of the character shapes, profile correlations, Karhunen-Loève coefficients, pixel averages in 2×3 windows, Zernike moments, and morphological features. Each layer of the multiplex network represents one of the 6 features. The graphs are constructed using k-nearest neighbors graphs with the nearest 50 neighbors and Euclidean distance. For the purposes of this work, we selected two groups corresponding to digits 1 and 7 to construct the first and second multiplex networks, respectively. These are two digits that are often misclustered due to their similar patterns.



(a)



(b)

Figure 3.4: UCI Handwritten dataset results. Images that were selected as (a) discriminative and (b) shared nodes for both groups (digits 1 and 7).

MX-DSC is applied to these two multiplex networks each with 6 layers with $\alpha = 0.5$, $\gamma_1 = 0.5$,

[1] https://archive.ics.uci.edu/ml/datasets/Multiple+Features

$k^1 = 2$, and $k^2 = 2$. Figure 3.4 shows the images that were selected as the discriminative and non-discriminative samples between the two groups. The samples in the discriminative subgraph correspond to images where both numbers 1 and 7 are clearly written and well-defined. On the other hand, the samples that are classified as non-discriminative are images with noisy patterns. We also evaluate the performance by applying spectral clustering to the features of the discriminative 1's and 7's and the non-discriminative 1's and 7's separately. The NMI for the discriminative samples is 0.7037 whereas it is 0.3091 for the non-discriminative samples. This shows that our method provides an accurate clustering of samples into discriminative and non-discriminative groups which offers better separability.

3.4.3 Electroencephalogram (EEG) Networks

In this work, we applied MX-DSC to functional connectivity networks (FCNs) of the brain constructed from EEG data collected from a cognitive control-related error processing study [108]. Each participant was presented with a string of five letters at each trial. Letters could be congruent (e.g., SSSSS) or incongruent stimuli (e.g., SSTSS) and the participants were instructed to respond to the center letter with a mouse. The EEG was recorded following the international 10/20 system for the placement of 64 Ag–AgCl electrodes at a 512 Hz sampling frequency. For each response type (error and correct) the FCNs can be modeled as a multiplex network with 64 nodes (brain regions) and 17 layers (subjects).

MX-DSC is applied with $\alpha = 0.5$, $\gamma = 0.5$, $k^1 = 3$, and $k^2 = 2$ to the multiplex networks corresponding to error and correct responses. Figure 3.5 shows the discriminative communities corresponding to error and correct responses. For the error response, we detect a discriminative community centered around fronto-central nodes (FCz, FC1, FC2, Cz, C1). This is consistent with prior work showing that medial frontal regions are more activated for error compared to correct trials [191]. The other discriminative communities are similar in both response types and correspond to the parietal-occipital region which is activated due to the visual stimulus.

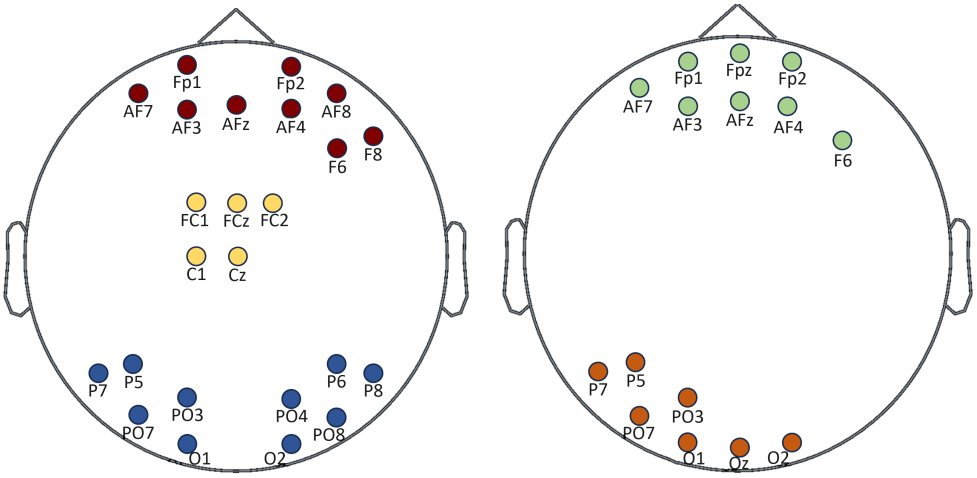


Figure 3.5: Discriminative Communities for Error (left) and Correct (right) responses.

3.5 Experiments: Temporal Mutiplex Networks

In this section, TMX-DiSG is evaluated in synthetic temporal multiplex networks. Two experiments are conducted where different parameters of the simulated data are changed. In addition, TMX-DiSG is applied to a real temporal multiplex network dataset to detect the discriminative subgraphs across time.

3.5.1 Synthetic Temporal Multiplex Networks

In this section, we extended the model described in Section 3.4.1 to generate two temporal multiplex benchmark networks with non-discriminative and discriminative communities among the two groups. We use the same approach as in Section 3.4.1 to generate the multiplex networks and the process is repeated for $T = 60$ time points.

Evaluation: The performance of TMX-DiSG in detecting the discriminative subgraphs is evaluated under two different settings using AUC-ROC. Both experiments are run with $\alpha, \gamma_1, \gamma_2, \gamma_3 \in [0, 1]$, and the results with the highest performance are reported. The accuracy of TMX-DiSG is compared to contrastive principal component analysis (cPCA) [2]. Since cPCA is applied to covariance matrices, adapting it to network data requires using the graph Laplacian in place of the covariance matrix. This adaptation makes cPCA equivalent to the first two terms of TMX-DiSG, i.e., learning the discriminative embedding matrices without the regularization terms.

For both experiments, we generated two groups of multiplex networks with $N = 300$ nodes,

$L, M = 10$ layers in each group, $k_c = 2$ shared communities. The number of discriminative communities, $\bar{k}_t^1$ and $\bar{k}_t^2$, is varied over time for both groups. Specifically, for the first group: $\bar{k}_t^1 = 3$ for $t \in [1, 20]$, $\bar{k}_t^1 = 4$ for $t \in [21 : 40]$, and $\bar{k}_t^1 = 3$ for $t \in [41, 60]$. For the second group, $\bar{k}_t^2 = 3$ for $t \in [1 : 10]$, , $\bar{k}_t^2 = 4$ for $t \in [11 : 20]$, $\bar{k}_t^2 = 3$ for $t \in [21 : 40]$, $\bar{k}_t^2 = 4$ for $t \in [41 : 50]$, and $\bar{k}_t^2 = 3$ for $t \in [51 : 60]$. This variation tests the method's adaptability to temporal changes in discriminative communities.

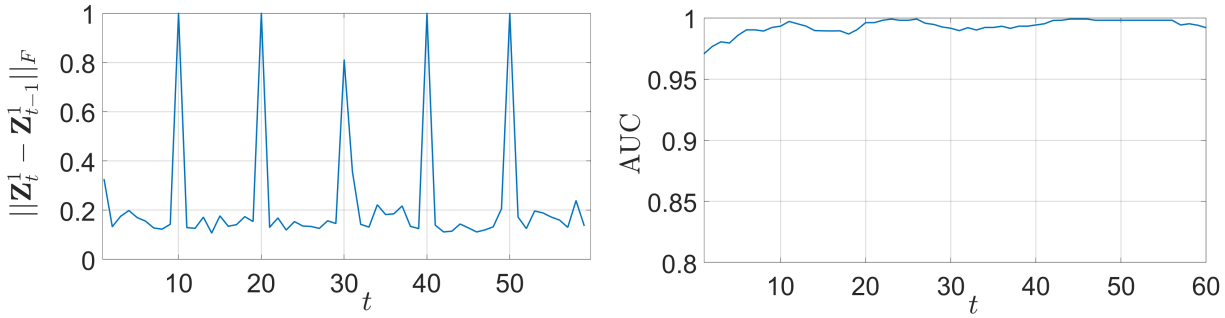
Experiment 1: Varying Noise Level: To evaluate robustness against noise, the two temporal multiplex networks are generated with varying values of $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$. Inter-layer dependencies are set as $p_1 = 0.9$ to allow for some variation in the discriminative community structure over time, $p_2 = 1$ to preserve shared communities across layers and time, and $p_3 = 1$ to preserve the discriminative communities across layers. This ensures that shared communities remain unchanged, while discriminative subgraphs vary only over time. Table 3.2 presents the average AUC values across time and groups. As expected, TMX-DiSG achieves higher detection accuracy compared to cPCA, where there are no regularizations to ensure the dissimilarity between the two subspaces, and the smoothness across time. The performance of both methods declines as the noise level increases.

Table 3.2: Average AUC values across time and groups for synthetic temporal multiplex networks.

Experiment 1			Experiment 2		
μ	cPCA	TMX-DiSG	p_1	cPCA	TMX-DiSG
0.1	0.9546	0.9976	0.90	0.9004	0.9521
0.2	0.9367	0.9786	0.80	0.8999	0.9542
0.3	0.9004	0.9521	0.70	0.9002	0.9501
0.4	0.9017	0.9513	0.60	0.8997	0.9489
0.5	0.8998	0.9501	0.50	0.8867	0.9297
0.6	0.8760	0.9245			
0.7	0.8591	0.8978			

Experiment 2: Change in Variability, p_1: In the second experiment, we evaluated the robustness of the algorithm against variations in the discriminative community structure across time by fixing

$\mu = 0.3$ and varying the inter-layer dependency probability p_1, i.e., the discriminative communities are allowed to vary. Table 3.2 shows that TMX-DiSG is robust to variations in the discriminative community structure even when $p_1 = 0.5$, implying that the variations across time do not affect the discriminative subgraph detection accuracy. Figure 3.6 shows that even when the network structure changes every ten time points due to variations in $\bar{k}_t^1$ and $\bar{k}_t^2$, as evidenced by Figure 3.6a, the accuracy remains constant as shown in Figure 3.6b.



(a) (b)

Figure 3.6: Illustrative example of the temporal variation in network structure and its impact on performance. (a) Frobenius norm of the difference between consecutive embedding-based matrices $\mathbf{Z}_t^1 = \bar{\mathbf{U}}_t^1 \bar{\mathbf{U}}_t^{1\top}$, computed for a single experiment and one run. (b) AUC values across time.

3.5.2 Dynamic fMRI Dataset: Task vs. Resting State

In this section, we evaluated TMX-DiSG on dynamic multiplex brain networks constructed from the Midnight Scan Club (MSC) dataset which includes fMRI data from ten subjects during resting state and three tasks [103]. Dynamic functional connectivity matrices were estimated using Pearson's correlation between windowed time courses (TCs) extracted with a sliding window approach [10]. A tapered window was created by convolving a rectangular window (width = 22 repetition times (TRs)) with a Gaussian kernel (σ = 3 TRs) and slid in steps of 1 TR. TCs were derived using the Gordon parcellation (333 parcels and 12 brain networks).

We compared the temporal multiplex graphs between resting state and two different tasks. In the first case, we constructed two groups of dynamic multiplex graphs representing motor vs. rest conditions, with $N = 333$ brain regions (nodes), 10 subjects (layers), and 187 time points. In the second case, we constructed the dynamic multiplex graphs representing memory vs. rest conditions,

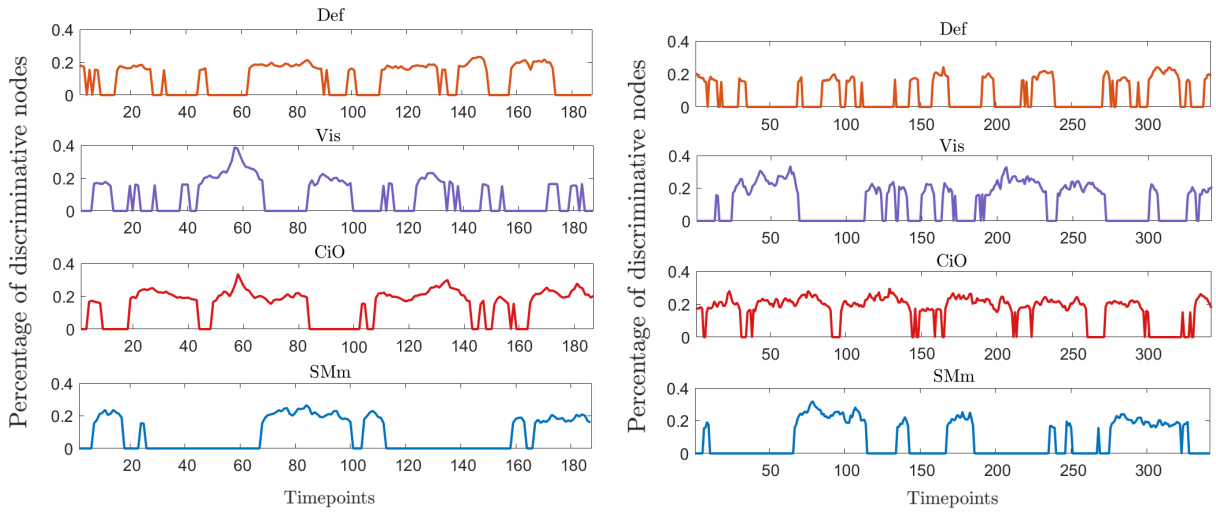


Figure 3.7: The percentage of discriminative nodes for both groups across default, visual, Cingulo Opercular, and Somatomotor Medial (Hand) networks and timepoints for (a) motor vs. rest case and (b) memory vs. rest case, respectively.

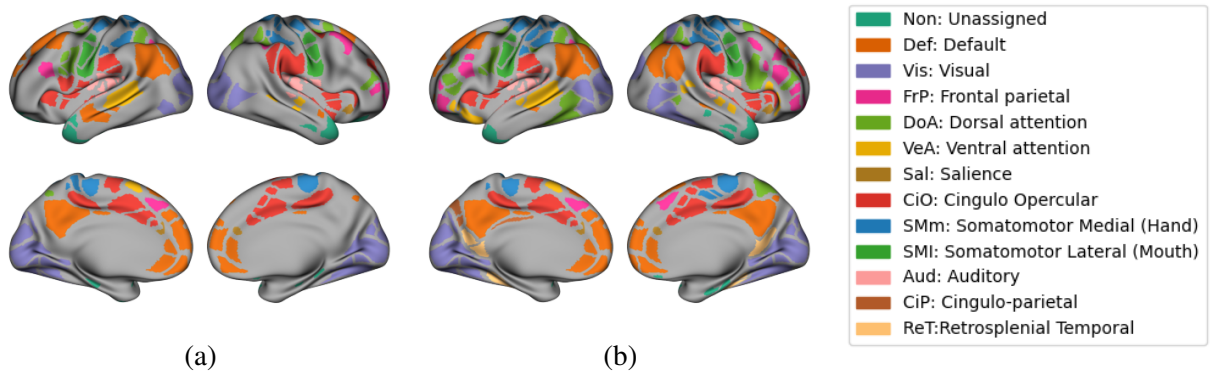


Figure 3.8: The brain topology of the aggregated discriminative nodes for both groups across all timepoints for (a) motor vs. rest case and (b) memory vs. rest case, respectively.

with $N = 333$ brain regions (nodes), 10 subjects (layers), and 342 time points. TMX-DiSG was applied to both set of temporal multiplex networks using parameters $\alpha = 0.5$ and $\gamma_1, \gamma_2, \gamma_3 = 0.1$. For each comparison, we computed the percentage of discriminative nodes within each of the 12 brain systems at every time point. Figure 3.7a illustrates how the percentage of discriminative nodes fluctuates over time in the Default Mode Network (DMN), Visual, Cingulo-Opercular (CiO), and Somatomotor Medial (Hand) (SMm) networks during motor vs. rest. Notably, the CiO and SMm networks show distinct peaks, reflecting their critical role in motor task execution and attentional control. The DMN and Visual networks also demonstrate variability, suggesting their involvement

in cognitive processes linked to motor function. Figure 3.8a visualizes the spatial distribution of discriminative nodes aggregated across all time points for motor vs. rest, highlighting the DMN, Visual, SMm, CiO, and Somatomotor Lateral (Mouth) (SMI) networks as key discriminative areas. These networks play a crucial role in high-level cognitive and attentional control processes related to motor task execution as expected.

Figure 3.7b presents the same analysis for memory vs. resting state. Compared to the first case, the fluctuations appear more evenly distributed across time, with the DMN, Visual, CiO, and SMm networks reflecting the continuous cognitive processing required for memory tasks. Figure 3.8b shows that the topology of discriminative nodes in memory-related tasks overlaps with motor-related networks but also involves additional regions, such as the Fronto-Parietal Network (FPN), and Retrosplenial Temporal Network (ReT). This suggests that memory tasks engage broader network interactions, integrating sensorimotor and cognitive control systems to a greater extent than motor tasks alone. Moreover, these regions are involved in high-level cognitive and attentional processes, including visual processing, sustained attention, and decision-making, essential for recognizing faces, scenes, and words in memory tasks.

3.6 Conclusions

In this chapter, we introduced two spectral clustering based community detection methods for identifying the discriminative subspaces between two multiplex networks and one for identifying the discriminative subspaces between two temporal multiplex networks. The first method, MX-DSC, focuses on only learning the discriminative subspaces, while the second one, MX-DCSC, learns the discriminative, consensus, and individual layerwise subspaces simultaneously. The evaluation of the methods on simulated data shows that the two methods perform similarly in terms of discriminative subspace detection accuracy. With MX-DCSC, one can also obtain a more accurate community detection performance compared to existing multiplex community detection algorithms. The application of the proposed framework to real data illustrates the possible applications of the method to multivariate network data. The third method, TMX-DiSG, provides an extension to temporal multiplex networks and detects discriminative subgraphs between two groups across

time. TMX-DiSG was evaluated on both synthetic temporal multiplex networks and dynamic functional connectivity networks constructed from two different task comparisons, i.e., motor and memory-related tasks versus resting state. Our analysis demonstrated that TMX-DiSG effectively identifies task-specific discriminative subgraphs. Future work will consider the extension of these frameworks to more than two groups.

CHAPTER 4

GRAPH FILTERING FOR CLUSTERING ATTRIBUTED GRAPHS

4.1 Introduction

Many real-world systems with relational data such as social interactions, citation and co-author relationships, and biological systems are represented as networks [14]. An important aspect of analyzing networks is the discovery of communities [99] which allow us to identify groups of functionally related objects and the interactions between them [151, 6].

In most real-world networks, both the node attributes and graph connectivity are available. For example, known properties of proteins, users' social network profiles, or authors' publication histories may tell us which objects are similar, and to which communities they may belong. Similarly, the set of edges between the nodes, such as the friendship relationships between users, interactions between proteins, and the collaboration between authors, can help identify groups based on connectivity. Classical data clustering methods such as K-means assign class labels based on only attribute similarity [123]. On the other hand, community detection algorithms find groups of nodes that are densely connected [91, 248, 182], ignoring node attributes. Employing just one of these two sources of information can result in the algorithm failing to account for important structures in the data. For instance, while it would be hard to determine the community membership of a sparsely connected node by solely relying on the network's connectivity, attributes may help reveal the community affiliation. Conversely, the network structure may indicate that two nodes belong to the same community, even if one of the nodes lacks attribute information. Hence, it is crucial to take both information sources into account and view network communities as clusters of closely linked nodes that also share common attributes.

4.1.1 Related Work

In recent years, several methods have been proposed for attributed graph clustering by combining the node attributes and link information [151, 125, 286, 9, 277]. The first class of methods for attributed graph clustering focuses on combining link and node information by formulating an objective function that integrates the two types of similarity: adjacency matrix that captures link

information and the similarity matrix that quantifies the affinity between the attributes [166, 9, 171]. These methods are indirect in the sense that they rely on the construction of an arbitrary similarity matrix from the node attributes. The second class of methods incorporates the graph structure and node attributes simultaneously into the community detection framework benefiting from the representation capability of graph neural networks (GNNs) [281]. The goal of these methods is to encode nodes in the graph with the neural networks and assign node labels. For example, methods such as graph autoencoders (GAE) [138], variational GAE (VGAE) [139], adversarially regularized graph autoencoder (ARGA), adversarially regularized variational graph autoencoder (ARVGA) [194] and marginalized graph autoencoder for graph clustering (MGAE) [259] have demonstrated state-of-the-art performance on attributed graph clustering. Although these methods achieve promising performance, they are not designed for the specific clustering performance, i.e., the network parameters are optimized to minimize the reconstruction error rather than maximizing the separation between different clusters. Moreover, in these methods, each convolutional layer is coupled with a projection layer, making it difficult to stack many layers and train a deep model. Thus, they only take into account neighbors of each node in two or three hops away and hence may be inadequate to capture global cluster structure of large graphs. Finally, these methods lack interpretability as they do not explicitly show the relationship between the learned models and the structure of the graph.

In order to take advantage of graph convolutional features while addressing the shortcomings of GNN-based methods, the last class of methods rely on graph signal processing (GSP) and in particular spectral graph filtering [286, 152, 266, 133, 265]. Over the past decade graph filters [120] have played a key role in different signal processing and machine learning tasks such as graph signal denoising [72, 190], smoothing [283], classification [219, 53], sampling [12], recovery [55], and graph clustering [249]. Different types of graph filter structures, including FIR graph filters [220, 225], ARMA filters [119], graph filter banks [186], and graph wavelets [109, 185] have been considered. For the task of graph node classification, spectral filtering based methods have certain advantages over spatial-based methods [29]. First, spectral filtering transforms all node features into

weighted sums of different eigenvectors via graph Fourier transform, which naturally captures global information, i.e., long range dependencies, unlike spatial methods that emphasize local information and suffer from oversmoothing [174]. Second, spectral filtering methods provide interpretability as the learned graph filters can directly state the most important frequency information associated with the labels, e.g., low-, medium-, and high-frequencies. Finally, spectral filters have been shown to obtain more cluster-friendly representations [289] and deal with the heterogeneity in the graph. Existing spectral graph filters either use pre-defined filters and learn the weights [291, 286, 267] or are based on adaptive spectral filter learning [112]. The first class of methods learn the weights of pre-determined filters, e.g., low-pass or high pass. The second class of methods learns the coefficients of polynomial filters with respect to different bases, e.g., ChebyNet [71], BernNet [112], ARMA [26]. However, in most cases the filters are designed empirically without any necessary constraints. As a result, these methods result in filters whose weights often have poor controllability.

Some example applications of spectral graph filters include semi-supervised learning on graphs, where graph filters are used to weigh and propagate the label information of multi-hop neighbors to the unknown nodes. This problem is formulated as an optimization problem, where the filter parameters are estimated to minimize the error between the estimated labels and the true labels on the labeled nodes with regularization on the filter parameters or the output, e.g., smooth label variation [53, 88, 22]. These works only consider the node label and graph connectivity information but do not necessarily address the issue of unsupervised clustering of attributed graphs. In the realm of unsupervised learning, GSP techniques have been used to address clustering and community mining problems. In [248], spectral graph wavelets are utilized to develop a fast, multiscale community mining protocol. In [249], graph-spectral filtering of random graph signals is used to construct feature vectors for each vertex so that the distances between vertices based on these feature vectors resemble those based on standard spectral clustering feature vectors. More recently, [79] uses spectral graph wavelets to learn structural embeddings that help identify vertices that have similar structural roles in the network. In all of these cases, the problem of community

detection is only addressed for either graphs without attributes or regular data clustering without graph structure. Moreover, the graph filter parameters are fixed. More recently, adaptive graph convolution (AGC) [286] was proposed for attributed graph clustering. Instead of stacking layers as in GCN, a K-order graph convolution that acts as a low-pass graph filter on node features is proposed to obtain smooth feature representations followed by spectral clustering on the learned features. In [133], this approach is further refined by learning the best similarity graph from the filtered features rather than constructing a graph using pre-determined similarity metrics. While these methods are intuitive and provide some interpretability to the node features, the filters are always low-pass, ignoring the useful information in higher frequency bands [265], and do not support learning arbitrary interpretable spectral filters that are optimized for the particular data. Moreover, the two steps of the algorithm, i.e., filtering and clustering, are completely decoupled from each other. Thus, there is no guarantee that the extracted features are optimal for clustering.

In this chapter, we address the shortcomings of the existing methods by proposing two graph filtering based methods for community detection in attributed networks, Graph Filtering for Clustering Attributed Graphs (GraFiCA) and Multi-Scale Graph Wavelets for Clustering (MSGWC). A cost function quantifying the separability of the filtered attributes is proposed. For GraFiCA, a general framework for learning the parameters of both Finite Impulse Response (FIR) and Autoregressive Moving Average (ARMA) filters is introduced. FIR filters are the most general form of graph convolutional filters implementing a polynomial frequency response. Their descriptive power increases as the filter order T grows. However, using higher orders implies handling higher matrix powers, which introduces numerical instabilities and in turn leads to poor performance. A more versatile class of filters is the family of ARMA filters [184], which offer a larger variety of frequency responses and can account for higher-order neighborhoods compared to polynomial filters with the same number of parameters. For MSGWC, we learn the optimal combination of the multi-scale features from graph spectral wavelet and scaling filters by minimizing the proposed cost function that quantifies the separability between clusters. Both methods are formulated as a two-step alternating minimization problem, where the first step learns the optimal graph partition-

ing for the given node attributes while the second step learns the optimal graph filter parameters for FIR and ARMA filters, and the optimal combination of graph filters at multiple scales, for GraFiCA and MSGWC, respectively.

The main contributions of the proposed framework are as follows:

- GraFiCA is the first that addresses the problem of parametric graph filter design in the form of both FIR and IIR filters for the purpose of attributed graph clustering. While there has been prior work in graph filter design for denoising or graph signal recovery [214], GraFiCA is the first unsupervised approach that learns the parameters of the filters for the purpose of clustering.
- Most of the methods based on spectral graph wavelets [248, 268] focus only on the output of the wavelet filter across scales without considering the scaling filter which captures the local neighborhood information. MSGWC learns the optimal combination of the multi-scale features from both graph spectral wavelet and scaling filters.
- The filters learned by the proposed approaches are not limited to low-pass filters as the structure of the filter is determined directly by the data. The filters take into account the useful information in middle and high frequency bands, i.e., higher-order neighborhoods, providing interpretability to the learned filters.
- The proposed cost function quantifies the discriminability between different classes unlike GCN-based approaches where the loss function is usually the reconstruction error. Thus, the filter coefficients are updated at each step of the algorithm to ensure that the smoothed node attributes are representative of the node assignments.

4.2 Background

4.2.1 Graph Filtering

In graph signal processing, two fundamental filter types, Finite Impulse Response (FIR) and Autoregressive Moving Average (ARMA) [119] graph filters, are considered [120]. FIR polynomial

graph filter is described as the linear operator

$$\mathcal{H}(\mathbf{L}) = \sum_{t=0}^{T-1} h_t \mathbf{L}^t = \mathbf{U}(\sum_{t=0}^{T-1} h_t \mathbf{\Lambda}^t)\mathbf{U}^\top,$$

where T is the filter order and h_t's are the coefficients. On the other hand, an ARMA filter is defined as

$$\mathcal{H}(\mathbf{L}) = \mathbf{U}\frac{\sum_{t=0}^{T-1} a_t \mathbf{\Lambda}^t}{\mathbf{I} + \sum_{q=1}^{Q-1} b_q \mathbf{\Lambda}^q}\mathbf{U}^\top,$$

where (T, Q) are the filter orders and a_t's and b_q's are the filter coefficients. Note that the FIR polynomial filter is a special case of ARMA filter for $Q = 1$.

Signals defined on the nodes of an attributed graph can be represented as a matrix $\mathbf{F} \in \mathbb{R}^{N \times P}$, where P is the number of attributes for each node. The filtered graph signal $\tilde{\mathbf{F}}$ is obtained as $\tilde{\mathbf{F}} = \mathcal{H}(\mathbf{L})\mathbf{F} = \mathbf{U}\mathcal{H}(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{F}$, where $\mathcal{H}(\mathbf{\Lambda}) = \text{diag}(\mathcal{H}(\lambda_1), \ldots, \mathcal{H}(\lambda_N))$ is the frequency response of the graph filter.

4.2.2 Spectral Graph Wavelet Transform

The Spectral Graph Wavelet Transform (SGWT) is a continuous multi-scale transform on graphs, enabling localization of graph signals in both the vertex and spectral domains, simultaneously [109]. The wavelet, $\psi_{s,a}$, at scale s centered at node a is generated by stretching a band-pass filter kernel $g(\cdot)$ with a parameter $s > 0$. The frequency response of the wavelet at scale s can be written as $\mathbf{G}_s(\mathbf{\Lambda}) = \text{diag}(g(s\lambda_1), g(s\lambda_2), ..., g(s\lambda_N))$. The wavelet basis at scale s is then given by

$$\mathbf{\Psi}_s = \{\psi_{s,1}|\psi_{s,2}|...|\psi_{s,N}\} = \mathbf{U}\mathbf{G}_s(\mathbf{\Lambda})\mathbf{U}^\top.$$

The wavelet coefficients at scale s for a graph signal $\mathbf{F}$, are defined as $\mathbf{\Psi}_s^\top\mathbf{F}$.

By this definition, a wavelet centered at node a corresponds to a signal on the graph diffused away from that node, remaining highly localized in the vertex domain. At small scales, the kernel is stretched out and lets through high-frequency modes. Therefore, the corresponding wavelet extends only to the close neighborhood of the node in the graph. At large scales, the filter function is localized around low-frequency modes, and the corresponding wavelet spans a larger neighborhood.

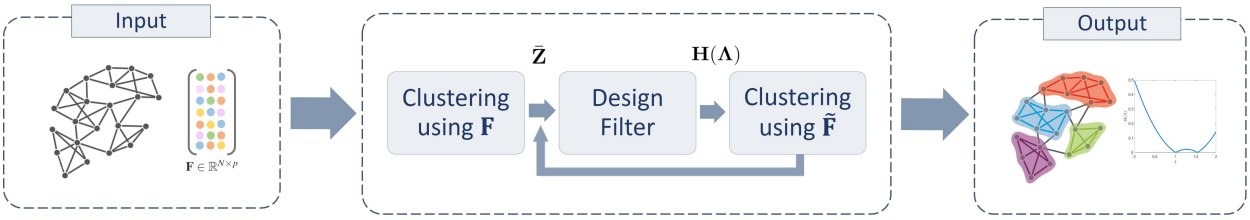


Figure 4.1: Framework of the proposed method.

Similarly, the scaling basis can be generated using a low-pass scaling filter kernel $h(\cdot)$, designed to smoothly represent the low-frequency content of the graph signal, as

$$\mathbf{\Phi}_s = \mathbf{U}\text{diag}(h(s\lambda_1), h(s\lambda_2), \ldots, h(s\lambda_N))\mathbf{U}^\top = \mathbf{U}\mathbf{H}_s(\mathbf{\Lambda})\mathbf{U}^\top,$$

and the corresponding scaling coefficients as $\mathbf{\Phi}_s^\top \mathbf{F}$.

The wavelet and scaling function kernels, $g(\cdot)$ and $h(\cdot)$, used in this work are the band-pass and low-pass filter kernels, respectively, as defined in [109, 248], along with their associated parameters.

4.3 Graph Filtering for Clustering Attributed Graphs (GraFiCA)

In this work, we propose to learn the optimal graph filter such that the within-cluster association of the filtered attributes, i.e., the total within-cluster dissimilarity or distance, is minimized while the between-class distance of the filtered attributes is maximized. Using an alternating minimization approach, in the first step, given the node attributes, we find the best cluster assignment to minimize the clustering cost function. In the second step, the graph partition is fixed and the cost function is optimized with respect to the filter coefficients. In this manner, the resulting graph filters are optimized for the clustering task (see Figure 4.1 for an overview of the method). This results in an attributed graph clustering method that takes into account both the topology and the node attributes.

In this section, we will first introduce the general problem formulation and the corresponding optimization problem. We will then present solutions for the optimal filter design for two different filter types; FIR and ARMA.

4.3.1 Problem Formulation

Given a graph $\mathcal{G}$ with normalized adjacency matrix $\mathbf{A}_n \in \mathbb{R}^{N \times N}$ and graph signal $\mathbf{F} \in \mathbb{R}^{N \times P}$, the goal is to find the best partition, i.e., K non-overlapping clusters, $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, and the

optimal graph filter $\mathcal{H}(\mathbf{\Lambda};\beta)$ with parameters β. We quantify the quality of the clustering based on the filtered graph attributes, $\tilde{\mathbf{F}} = \mathbf{U}\mathcal{H}(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{F}$, as follows:

$$\mathcal{L}(\boldsymbol{C},\mathcal{H}(\mathbf{\Lambda};\beta)) = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j\in C_k} ||\tilde{F}_{i\cdot} - \tilde{F}_{j\cdot}||^2 - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i\in C_k \\ j\notin C_k}} ||\tilde{F}_{i\cdot} - \tilde{F}_{j\cdot}||^2, \tag{4.1}$$

where the first and second terms quantify the dissimilarity/distance of the filtered node attributes within and between clusters, respectively. $\mathcal{D}(C_k)$ is total dissimilarity of nodes in C_k with respect to the whole graph quantified by $\mathcal{D}(C_k) = \sum_{i\in C_k, j\in V} ||\tilde{F}_{i\cdot} - \tilde{F}_{j\cdot}||^2$. Thus, we want to minimize the the ratio of dissimilarity attributed to within cluster connections to the total dissimilarity across the whole graph, i.e., association, while maximizing the separation between clusters, i.e., cut. Defining the dissimilarity matrix based on $\tilde{\mathbf{F}}$ as $\tilde{W}_{ij} = ||\tilde{F}_{i\cdot} - \tilde{F}_{j\cdot}||^2$, (4.1) can be rewritten as

$$\mathcal{L}(\boldsymbol{C},\mathcal{H}(\mathbf{\Lambda};\beta)) = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j\in C_k} \tilde{W}_{ij} - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i\in C_k \\ j\notin C_k}} \tilde{W}_{ij}. \tag{4.2}$$

Our goal is to minimize this cost function in terms of both the graph partition, $\boldsymbol{C}$ and the graph filter parameters β. The corresponding optimization problem can be formulated as

$$\underset{\boldsymbol{C},\beta}{\text{minimize}}\ \mathcal{L}(\boldsymbol{C},\mathcal{H}(\mathbf{\Lambda};\beta)) + \alpha\mathcal{R}(\boldsymbol{C}), \tag{4.3}$$

where the regularization term $\mathcal{R}(\boldsymbol{C})$ will be specified to put additional constraints on the partition such that the connectivity information is also taken into account. We propose a two-step alternating minimization approach to solve this problem, where at each iteration l we first learn the optimal graph partitioning for the given graph signal while the second step learns the optimal graph filter parameters:

$$\begin{aligned} \boldsymbol{C}^{(l+1)} &:= \underset{\boldsymbol{C}}{\text{argmin}}\ \mathcal{L}(\boldsymbol{C},\mathcal{H}(\mathbf{\Lambda};\beta^{(l)})) + \alpha\mathcal{R}(\boldsymbol{C}), \\ \beta^{(l+1)} &:= \underset{\beta}{\text{argmin}}\ \mathcal{L}(\boldsymbol{C}^{(l+1)},\mathcal{H}(\mathbf{\Lambda};\beta)). \end{aligned} \tag{4.4}$$

4.3.2 C update: Clustering

For the clustering task, given the filtered attributes, $\tilde{\mathbf{F}}$, we aim to find the graph partition, $\boldsymbol{C}$. Fixing $\mathcal{H}(\boldsymbol{\Lambda};\beta)$, the cost function in (4.2) can be rewritten as:

$$\mathcal{L}(\boldsymbol{C},\mathcal{H}(\boldsymbol{\Lambda};\beta)) = \sum_{k=1}^{K}\frac{\text{diss}(C_k,C_k)}{\mathcal{D}(C_k)} - \gamma\sum_{k=1}^{K}\frac{\text{diss}(C_k,V\setminus C_k)}{\mathcal{D}(C_k)}. \tag{4.5}$$

where $\text{diss}(C_k,C_k) = \sum_{i,j\in C_k}\tilde{W}_{ij}$ is the total dissimilarity within a cluster and $\text{diss}(C_k,V\setminus C_k) = \sum_{\substack{i\in C_k\\ j\notin C_k}}\tilde{W}_{ij}$ is the total distance/dissimilarity of C_k with respect to rest of the graph. Since it can be shown that $\text{diss}(C_k,V\setminus C_k) = \mathcal{D}(C_k) - \text{diss}(C_k,C_k)$, we can rewrite (4.5) in terms of within-cluster dissimilarity as

$$\mathcal{L}(\boldsymbol{C},\mathcal{H}(\boldsymbol{\Lambda};\beta)) = (1+\gamma)\sum_{k=1}^{K}\frac{\text{diss}(C_k,C_k)}{\mathcal{D}(C_k)} + K. \tag{4.6}$$

This problem can be rewritten as a trace optimization problem. Therefore, minimizing the cost function in Eq. (4.6) is equivalent to minimizing the following

$$\mathcal{L}(\bar{\mathbf{Z}},\mathcal{H}(\boldsymbol{\Lambda};\beta)) = \text{tr}(\bar{\mathbf{Z}}^\top\mathbf{D}^{-1/2}\tilde{\mathbf{W}}\mathbf{D}^{-1/2}\bar{\mathbf{Z}}), \tag{4.7}$$

subject to $\bar{\mathbf{Z}}^\top\bar{\mathbf{Z}} = \mathbf{I}$.

Regularization $\mathcal{R}$: In order to incorporate the connectivity information into the clustering problem, we propose to use the regularization term $\mathcal{R}(\bar{\mathbf{Z}}) = -\text{tr}(\bar{\mathbf{Z}}^\top\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\bar{\mathbf{Z}})$, such that minimizing our regularization term is equivalent to maximizing NAssoc in Eq. (1.9). Thus, $\boldsymbol{C}$ update in Eq. (4.4) can be equivalently expressed by the following $\bar{\mathbf{Z}}$ update equation:

$$\begin{aligned}\bar{\mathbf{Z}}^{(l+1)} &:= \underset{\bar{\mathbf{Z}},\bar{\mathbf{Z}}^\top\bar{\mathbf{Z}}=\mathbf{I}}{\text{argmin}}\ \text{tr}(\bar{\mathbf{Z}}^\top\mathbf{D}^{-1/2}\tilde{\mathbf{W}}\mathbf{D}^{-1/2}\bar{\mathbf{Z}}) - \alpha\text{tr}(\bar{\mathbf{Z}}^\top\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\bar{\mathbf{Z}}),\\ &:= \underset{\bar{\mathbf{Z}},\bar{\mathbf{Z}}^\top\bar{\mathbf{Z}}=\mathbf{I}}{\text{argmin}}\ \text{tr}(\bar{\mathbf{Z}}^\top(\tilde{\mathbf{W}}_n - \alpha\mathbf{A}_n)\bar{\mathbf{Z}}).\end{aligned} \tag{4.8}$$

The first term in (4.8) corresponds to minimizing the normalized dissimilarity matrix constructed from the graph attributes while the second term corresponds to normalized association with respect to graph connectivity. While the two terms are similar mathematically, they correspond to two different sources of information, node attributes and graph topology, respectively.

The optimal solution to this problem is the set of eigenvectors corresponding to the K smallest eigenvalues of $\tilde{\mathbf{W}}_n - \alpha \mathbf{A}_n$. The graph partition, $\boldsymbol{C}^{(l+1)}$, is then updated by applying k-means to the rows of $\bar{\mathbf{Z}}$.

4.3.3 β Update: Optimal Filter Design

Once we have the cluster assignments, we want to determine the coefficients of the optimal filter $\mathcal{H}(\boldsymbol{\Lambda}; \beta)$. In this chapter, we present the derivations for both FIR and IIR filter types, with $\beta = \mathbf{h}$ for FIR and $\beta = \{\mathbf{a}, \mathbf{b}\}$ for the ARMA filter.

FIR Filter: In this section, we present the optimization problem for learning the parameters of the optimal polynomial filter with a given filter order T, $\mathcal{H}(\boldsymbol{\Lambda}) = \sum_{t=0}^{T-1} h_t \boldsymbol{\Lambda}^t$, for the clustering task.

Following the definitions in [220], we can define the t-th shifted input signal, $\mathbf{S}^{(t)} \in \mathbb{R}^{N \times P}$, as $\mathbf{S}^{(t)} := \mathbf{U}\Lambda^t\mathbf{U}^\top\mathbf{F} = \mathbf{L}_n^t\mathbf{F}$ and $\mathbf{S}_{(i)}$ can then be defined as a $T \times P$ matrix corresponding to the i-th node where each row corresponds to the t-th shifted input signal at that node with $[\mathbf{S}_{(i)}]_t := [\mathbf{S}^{(t)}]_{(i)}$. With $\tilde{\mathbf{F}}$ denoting the output of a graph filter for the input signal $\mathbf{F}$, it follows that

$$\tilde{\mathbf{F}} = \mathbf{U}\left(\sum_{t=0}^{T-1} h_t \Lambda^t\right)\mathbf{U}^\top\mathbf{F} = \sum_{t=0}^{T-1} h_t \mathbf{S}^{(t)}. \tag{4.9}$$

Hence, the filtered graph signal corresponding to the i-th node can be computed as $\tilde{\mathbf{F}}_i = \sum_{t=0}^{T-1} h_t [\mathbf{S}^{(t)}]_i = \mathbf{h}^\top \mathbf{S}_{(i)}$, with $\mathbf{h} = [h_0, h_1, \cdots, h_{T-1}]$. The cost function in (4.1) can then be rewritten in terms of the filter coefficient vector, $\mathbf{h}$, as follows

$$\begin{aligned}\mathcal{L}(\boldsymbol{C}, \mathbf{h}) &= \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} ||\mathbf{h}^\top \mathbf{S}_{(i)} - \mathbf{h}^\top \mathbf{S}_{(j)}||^2 - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} ||\mathbf{h}^\top \mathbf{S}_{(i)} - \mathbf{h}^\top \mathbf{S}_{(j)}||^2, \\ &= \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} \mathbf{h}^\top (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top \mathbf{h} - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} \mathbf{h}^\top (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top \mathbf{h}.\end{aligned} \tag{4.10}$$

Eq. (4.30) can be rewritten as follows:

$$\mathcal{L}(\boldsymbol{C}, \mathbf{h}) = (\mathbf{h}^\top (\mathbf{B} - \gamma \mathbf{C})\mathbf{h}), \tag{4.11}$$

where $\mathbf{B}$ and $\mathbf{C}$ are $T \times T$ matrices defined as

$$\mathbf{B} = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top, \tag{4.12}$$

$$\mathbf{C} = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top. \tag{4.13}$$

Our optimization problem becomes

$$\underset{\mathbf{h}, \mathbf{h}^\top \mathbf{h} = 1}{\text{minimize}} \ (\mathbf{h}^\top (\mathbf{B} - \gamma \mathbf{C}) \mathbf{h}). \tag{4.14}$$

The solution to the optimization problem is the eigenvector of $\mathbf{B} - \gamma\mathbf{C}$ corresponding to the smallest eigenvalue. Once $\mathbf{h}$ is obtained, the filtered signal can be updated as $\tilde{\mathbf{F}} = \mathbf{U} \sum_{t=0}^{T-1} h_t \mathbf{\Lambda}^t \mathbf{U}^\top \mathbf{F}$.

ARMA Filter: In this section, we present the optimization problem for learning the parameters of an ARMA filter with the following graph frequency response

$$\mathcal{H}(\mathbf{\Lambda}) = \frac{\sum_{t=0}^{T-1} a_t \mathbf{\Lambda}^t}{\mathbf{I} + \sum_{q=1}^{Q-1} b_q \mathbf{\Lambda}^q}, \tag{4.15}$$

where $\mathbf{a} = [a_0, a_1, \ldots, a_{T-1}]$ and $\mathbf{b} = [b_1, b_2, \ldots, b_{Q-1}]$ are the filter coefficients, and (T, Q) is the pair of filter orders.

In order to find the filter parameters $\mathbf{a}$ and $\mathbf{b}$ we introduce an auxiliary polynomial $\Gamma_M(\lambda) = \sum_{m=0}^{M-1} c_m \lambda^m$. $\Gamma_M(\lambda)$ can be viewed as the reciprocal polynomial of $(1 + \sum_{q=1}^{Q-1} b_q \lambda^q)$. In general, the reciprocal polynomial has a larger order than the denominator polynomial, i.e., $M \geq Q$ [269]. Letting $\sum_{m=0}^{M-1} c_m \mathbf{\Lambda}^m = \frac{1}{\mathbf{I} + \sum_{q=1}^{Q-1} b_q \mathbf{\Lambda}^q}$, Eq. (4.15) becomes

$$\mathcal{H}(\mathbf{\Lambda}) = \sum_{m=0}^{M-1} c_m \mathbf{\Lambda}^m \sum_{t=0}^{T-1} a_t \mathbf{\Lambda}^t. \tag{4.16}$$

To solve for the filter parameters, we introduce the constraint $(\sum_{m=0}^{M-1} c_m \mathbf{\Lambda}^m)(\mathbf{I} + \sum_{q=1}^{Q-1} b_q \mathbf{\Lambda}^q) = \mathbf{I}$. We can rewrite the polynomials as $\sum_{m=0}^{M-1} c_m \mathbf{\Lambda}^m = \text{diag}(\mathbf{\Psi}_M \mathbf{c})$ and $\sum_{q=1}^{Q-1} b_q \mathbf{\Lambda}^q = \text{diag}(\bar{\mathbf{\Psi}}_Q \mathbf{b})$, where $\mathbf{\Psi}_M$ and $\bar{\mathbf{\Psi}}_Q$ are defined as

$$\boldsymbol{\Psi}_M = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{M-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & \cdots & \lambda_N^{M-1} \end{bmatrix}, \bar{\boldsymbol{\Psi}}_Q = \begin{bmatrix} \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{Q-1} \\ \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{Q-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_N & \lambda_N^2 & \cdots & \lambda_N^{Q-1} \end{bmatrix}.$$

Using these definitions we can then rewrite $(\sum_{m=0}^{M-1} c_m \boldsymbol{\Lambda}^m)(\mathbf{I} + \sum_{q=1}^{Q-1} b_q \boldsymbol{\Lambda}^q) = \mathbf{I}$ as $(\text{diag}(\boldsymbol{\Psi}_M \mathbf{c}))(\mathbf{I} + \text{diag}(\bar{\boldsymbol{\Psi}}_Q \mathbf{b})) = \mathbf{I}$ and the optimization problem becomes

$$\begin{aligned} &\underset{\mathbf{a},\mathbf{b},\mathbf{c}}{\text{minimize}} \;\; \mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; [\mathbf{a}, \mathbf{c}])) \\ &+ ||\text{diag}(\boldsymbol{\Psi}_M \mathbf{c})[\mathbf{I} + \text{diag}(\bar{\boldsymbol{\Psi}}_Q \mathbf{b})] - \mathbf{I}||_F^2. \end{aligned} \tag{4.17}$$

In order to find the parameters $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, we propose an alternating minimization approach where in order to learn each of the variables we fix the other two. Thus, the β update in (4.4) becomes

$$\begin{aligned} \mathbf{a}^{(l+1)} &:= \underset{\mathbf{a}}{\text{argmin}}\; \mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; [\mathbf{a}, \mathbf{c}^{(l)}])), \\ \mathbf{b}^{(l+1)} &:= \underset{\mathbf{b}}{\text{argmin}}\; ||\text{diag}(\boldsymbol{\Psi}_M \mathbf{c}^{(l)})[\mathbf{I} + \text{diag}(\bar{\boldsymbol{\Psi}}_Q \mathbf{b})] - \mathbf{I}||_F^2, \\ \mathbf{c}^{(l+1)} &:= \underset{\mathbf{c}}{\text{argmin}}\; \mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; [\mathbf{a}^{(l+1)}, \mathbf{c}])) + ||\text{diag}(\boldsymbol{\Psi}_M \mathbf{c})[\mathbf{I} + \text{diag}(\bar{\boldsymbol{\Psi}}_Q \mathbf{b}^{(l+1)})] - \mathbf{I}||_F^2. \end{aligned} \tag{4.18}$$

Update a: In order to update $\mathbf{a}$, we define the t-th shifted input signal as $\mathbf{S}^{(t)} := \mathbf{U}(\sum_{m=0}^{M-1} c_m \boldsymbol{\Lambda}^m)\boldsymbol{\Lambda}^t \mathbf{U}^\top \mathbf{F}$ which can be rewritten as $\mathbf{S}^{(t)} := \sum_{m=0}^{M-1} c_m \mathbf{L}^{m+t} \mathbf{F}$ and $\mathbf{S}_{(i)}$ as a $T \times P$ matrix corresponding to the i-th node where each row corresponds to the t-th shifted input signal at that node, $[\mathbf{S}^{(t)}]_{(i)}$. Therefore, $\tilde{\mathbf{F}}_i = \sum_{t=0}^{T-1} a_t [\mathbf{S}^{(t)}]_i = \mathbf{a}^\top \mathbf{S}_{(i)}$.

The cost function in (4.1) can be rewritten in terms of the filter coefficients $\mathbf{a}$ and the newly defined $\mathbf{S}_{(i)}$ as follows

$$\mathcal{L}(\boldsymbol{C}, \mathbf{a}, \mathbf{c}) = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} ||\mathbf{a}^\top \mathbf{S}_{(i)} - \mathbf{a}^\top \mathbf{S}_{(j)}||^2 - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} ||\mathbf{a}^\top \mathbf{S}_{(i)} - \mathbf{a}^\top \mathbf{S}_{(j)}||^2,$$

$$= \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} \mathbf{a}^\top (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top \mathbf{a} - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} \mathbf{a}^\top (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top \mathbf{a}. \tag{4.19}$$

Eq. (4.19) can be rewritten as follows:

$$\mathcal{L}(C, \mathbf{a}, \mathbf{c}) = (\mathbf{a}^\top (\mathbf{B} - \gamma \mathbf{C})\mathbf{a}), \tag{4.20}$$

where $\mathbf{B}$ and $\mathbf{C}$ are $T \times T$ matrices defined as in Eqs. (4.32) and (4.33) using the newly defined $\mathbf{S}_{(i)}$ which depends on both C and $\mathbf{c}$. Our optimization problem becomes

$$\underset{\mathbf{a}, \mathbf{a}^\top \mathbf{a}=1}{\text{minimize}} \ (\mathbf{a}^\top (\mathbf{B} - \gamma \mathbf{C})\mathbf{a}). \tag{4.21}$$

whose solution is the eigenvector of $\mathbf{B} - \gamma \mathbf{C}$ corresponding to the smallest eigenvalue.

Update b: Vectorizing each term in (4.18) for the $\mathbf{b}$ update, we have the following equivalent optimization problem

$$\underset{\mathbf{b}}{\text{minimize}} \ ||\mathbf{\Psi}_M \mathbf{c} + \text{diag}(\mathbf{\Psi}_M \mathbf{c})\bar{\mathbf{\Psi}}_Q \mathbf{b} - \mathbf{1}_N||^2, \tag{4.22}$$

where $\mathbf{1}_N \in \mathbf{R}^N$ is an all ones vector. To find $\mathbf{b} = [b_1, b_2, \cdots, b_{Q-1}]$, the following objective function $\mathcal{L}_1(\mathbf{b}) = ||\mathbf{\Psi}_M \mathbf{c} + \text{diag}(\mathbf{\Psi}_M \mathbf{c})\bar{\mathbf{\Psi}}_Q \mathbf{b} - \mathbf{1}_N||^2$ is minimized with respect to $\mathbf{b}$ by setting

$$\nabla_b \mathcal{L}_1 = 2\bar{\mathbf{\Psi}}_Q^\top \text{diag}(\mathbf{\Psi}_M \mathbf{c})[\mathbf{\Psi}_M \mathbf{c} + \text{diag}(\mathbf{\Psi}_M \mathbf{c})\bar{\mathbf{\Psi}}_Q \mathbf{b}] - 2\bar{\mathbf{\Psi}}_Q^\top \text{diag}(\mathbf{\Psi}_M \mathbf{c})\mathbf{1}_N = 0.$$

After some algebraic manipulations, $\mathbf{b}$ can be updated as

$$\mathbf{b} = \mathbf{Y}_1^{-1} \mathbf{v}_1, \tag{4.23}$$

where $\mathbf{Y}_1 \in \mathbb{R}^{Q-1 \times Q-1}$ and $\mathbf{v}_1 \in \mathbb{R}^{Q-1}$ are defined as $\mathbf{Y}_1 = (\bar{\mathbf{\Psi}}_Q^\top \text{diag}(\mathbf{\Psi}_M \mathbf{c})\text{diag}(\mathbf{\Psi}_M \mathbf{c})\bar{\mathbf{\Psi}}_Q)$ and $\mathbf{v}_1 = \bar{\mathbf{\Psi}}_Q^\top \text{diag}(\mathbf{\Psi}_M \mathbf{c})[\mathbf{1}_N - \mathbf{\Psi}_M \mathbf{c}]$, respectively. $\mathbf{Y}_1 \in \mathbb{R}^{Q-1 \times Q-1}$ is full rank and therefore invertible if and only if the Vandermonde matrix has full rank. Since $Q - 1 \ll N$, $\bar{\mathbf{\Psi}}_Q$ will be full rank if there are at least $Q - 1$ distinct eigenvalues of the normalized graph Laplacian. The full explanation and proof for this statement is provided in Appendix C.

Update c: We can define the m-th shifted input signal as $\mathbf{S}^{(m)} := \mathbf{U}\mathbf{\Lambda}^m(\sum_{t=0}^{T-1} a_t\mathbf{\Lambda}^t)\mathbf{U}^\top\mathbf{F}$ and $\tilde{\mathbf{F}}_i = \sum_{m=0}^{M-1} c_m[\mathbf{S}^{(m)}]_i = \mathbf{c}^\top\mathbf{S}_{(i)}$, with $\mathbf{S}_{(i)}$ being a $M \times P$ matrix corresponding to the i-th node where each row corresponds to the m-th shifted input signal at that node, $[\mathbf{S}^{(m)}]_{(i)}$.

The $\mathbf{c}$ update in (4.18) can be rewritten as

$$\underset{\mathbf{c}}{\text{minimize}}\ \mathbf{c}^\top(\mathbf{B} - \gamma\mathbf{C})\mathbf{c} + ||\mathbf{\Psi}_M\mathbf{c} + \text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b})\mathbf{\Psi}_M\mathbf{c} - \mathbf{1}_N||^2. \tag{4.24}$$

To find $\mathbf{c} = [c_0, c_1, \cdots, c_{M-1}]$, the following objective function $\mathcal{L}_2(\mathbf{c}) = \mathbf{c}^\top(\mathbf{B}-\gamma\mathbf{C})\mathbf{c}+||\mathbf{\Psi}_M\mathbf{c}+$ $\text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b})\mathbf{\Psi}_M\mathbf{c} - \mathbf{1}_N||^2$ is minimized with respect to $\mathbf{c}$ by setting

$$\begin{aligned}\nabla_c\mathcal{L}_2 = {} & 2(\mathbf{B} - \gamma\mathbf{C})\mathbf{c} + 2\mathbf{\Psi}_M^\top(\mathbf{I} + \text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b}))^\top(\mathbf{I} + \text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b}))\mathbf{\Psi}_M\mathbf{c} \\ & - 2\mathbf{\Psi}_M^\top(\mathbf{I} + \text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b})^\top)\mathbf{1}_N = 0.\end{aligned}$$

After some algebraic manipulations, $\mathbf{c}$ can be updated as

$$\mathbf{c} = \mathbf{Y}_2^{-1}\mathbf{v}_2, \tag{4.25}$$

where $\mathbf{Y}_2 \in \mathbb{R}^{M\times M}$ and $\mathbf{v}_2 \in \mathbb{R}^M$ are defined as $\mathbf{Y}_2 = (\mathbf{B} - \gamma\mathbf{C}) + \mathbf{\Psi}_M^\top(\mathbf{I} + \text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b}))^\top(\mathbf{I} +$ $\text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b}))\mathbf{\Psi}_M$ and $\mathbf{v}_2 = \mathbf{\Psi}_M^\top(\mathbf{I} + \text{diag}(\bar{\mathbf{\Psi}}_Q\mathbf{b})^\top)\mathbf{1}_N$, respectively. $\mathbf{Y}_2$ is full rank and therefore invertible if there are at least M ($M \ll N$) distinct eigenvalues of the normalized graph Laplacian.

We solve iteratively for the filter coefficients $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ until convergence. Once the filter coefficients are obtained at the l-th iteration, we update the filtered signal $\tilde{\mathbf{F}}^{(l)}$. Both variable updates, $\boldsymbol{C}$ and β, for Clustering and Optimal Filter Design steps, respectively, are repeated until convergence as described in Algorithm 4.1.

4.4 Multi-Scale Graph Wavelets for Clustering (MSGWC)

In this section, we propose to learn the optimal combination of multi-scale features using graph scaling and wavelet bases. Given the input signal $\mathbf{F}$, let $\tilde{\mathbf{F}}$ denote the multi-scale features defined as

$$\tilde{\mathbf{F}} = \left(w_1\mathbf{\Phi}_1^\top + \sum_{s=2}^{T} w_s\mathbf{\Psi}_s^\top\right)\mathbf{F} = \sum_{s=1}^{T} w_s\mathbf{S}^{(s)}, \tag{4.26}$$

Algorithm 4.1: GraFiCA.

Input: Normalized adjacency matrix $\mathbf{A}_n$, graph signal $\mathbf{F}$, number of clusters K, parameters α, γ, filter orders (T, Q), and $M \geq Q$.
Output: Cluster partition $\boldsymbol{C}$, graph filter $\mathcal{H}(\mathbf{L})$.

$\mathbf{L}_n = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$
[NMI$^{(0)}$, $\boldsymbol{C}^{(0)}$]=ClusteringStep($\mathbf{F}$)
Initialize $\mathbf{c}^{(0)}$ for ARMA
$l = 0$
while $|\text{NMI}^{(l)} - \text{NMI}^{(l-1)}| > 10^{-3}$ **do**
    if $Q = 1$ **then** ▷ FIR Filter
        Compute $\mathbf{B}$ and $\mathbf{C}$ using (4.32) and (4.33)
        $\mathbf{S} \leftarrow (\mathbf{B} - \gamma\mathbf{C})$
        $\mathbf{S} = \mathbf{H}\mathbf{\Gamma}\mathbf{H}^\top$
        $\mathbf{h} \leftarrow \mathbf{H}_{\cdot 1}$
        $\tilde{\mathbf{F}} \leftarrow \sum_{t=0}^{T-1} h_t \mathbf{L}_n^t \mathbf{F}$
    else ▷ ARMA Filter
        $r = 0$
        while $||\mathbf{a}^{(r)} - \mathbf{a}^{(r-1)}||^2 > 10^{-3}$, $||\mathbf{b}^{(r)} - \mathbf{b}^{(r-1)}||^2 > 10^{-3}$, and $||\mathbf{c}^{(r)} - \mathbf{c}^{(r-1)}||^2 > 10^{-3}$ **do**
            Compute $\mathbf{B}$ and $\mathbf{C}$ using (4.32) and (4.33)
            $\mathbf{S} \leftarrow (\mathbf{B} - \gamma\mathbf{C})$
            $\mathbf{S} = \mathbf{H}\mathbf{\Gamma}\mathbf{H}^\top$
            $\mathbf{a}^{(r)} \leftarrow \mathbf{H}_{\cdot 1}$
            $\mathbf{b}^{(r)} \leftarrow \mathbf{Y}_1^{-1}\mathbf{v}_1$ using (4.23)
            $\mathbf{c}^{(r)} \leftarrow \mathbf{Y}_2^{-1}\mathbf{v}_2$ using (4.25)
            $r = r + 1$
        end while
        $\tilde{\mathbf{F}} \leftarrow \mathbf{U}\frac{\sum_{t=0}^{T-1} a_t \mathbf{\Lambda}^t}{\mathbf{I} + \sum_{q=1}^{Q-1} b_q \mathbf{\Lambda}^q}\mathbf{U}^\top\mathbf{F}$
    end if
    [NMI$^{(l)}$,$\boldsymbol{C}^{(l)}$]=ClusteringStep($\tilde{\mathbf{F}}$)
    $l = l + 1$
end while
function ClusteringStep($\mathbf{F}$)
    $W_{ij} \leftarrow ||F_{i\cdot} - F_{j\cdot}||^2$
    $\mathbf{W}' \leftarrow \mathbf{W}_n - \alpha\mathbf{A}_n$
    $\mathbf{W}' = \mathbf{V}\mathbf{\Gamma}\mathbf{V}^\top$
    $\mathbf{V} \leftarrow \mathbf{V}(:, 1:K)$
    $\boldsymbol{C} \leftarrow k$-means($\mathbf{V}, K$)
    Compute NMI
    return NMI and $\boldsymbol{C}$
end function

where $\mathbf{S}^{(1)} := \mathbf{\Phi}_1^\top \mathbf{F}$ is the output of the graph scaling filter and $\mathbf{S}^{(s)} := \mathbf{\Psi}_s^\top \mathbf{F}$ is the output of the graph wavelet filter at scale s for $s \geq 2$. The multi-scale graph signal at node i can be written as $\tilde{\mathbf{F}}_i = \sum_{s=1}^{T} w_s [\mathbf{S}^{(s)}]_i = \mathbf{w}^\top \mathbf{S}_{(i)}$, where $\mathbf{S}_{(i)}$ is a $T \times P$ matrix corresponding to the i-th node where each row is the s-th scale feature with $[\mathbf{S}_{(i)}]_s := [\mathbf{S}^{(s)}]_{(i)}$ [220].

To further interpret the multi-scale features $\tilde{\mathbf{F}}$ in the frequency domain, we can write $\tilde{\mathbf{F}}$ using the scaling and wavelet filter responses, $\mathbf{H}_s(\mathbf{\Lambda})$ and $\mathbf{G}_s(\mathbf{\Lambda})$, as

$$\tilde{\mathbf{F}} = \mathbf{U} \left(w_1 \mathbf{H}_1(\mathbf{\Lambda}) + \sum_{s=2}^{T} w_s \mathbf{G}_s(\mathbf{\Lambda}) \right) \mathbf{U}^\top \mathbf{F}. \tag{4.27}$$

This formulation shows how the multi-scale signal $\tilde{\mathbf{F}}$ is formed by a combination of low-pass and band-pass filters across different scales, with each scale's contribution weighted by w_s's, resulting in a graph filter with frequency response $\mathcal{H}(\mathbf{\Lambda}, \mathbf{w}) = w_1 \mathbf{H}_1(\mathbf{\Lambda}) + \sum_{s=2}^{T} w_s \mathbf{G}_s(\mathbf{\Lambda})$.

4.4.1 Problem Formulation

Given a graph $\mathcal{G}$ with normalized adjacency matrix $\mathbf{A}_n \in \mathbb{R}^{N \times N}$ and graph signal $\mathbf{F} \in \mathbb{R}^{N \times P}$, our goal is to find the best partition, i.e., K non-overlapping clusters, $\boldsymbol{C} = \{C_1, C_2, \ldots, C_K\}$, and the optimal weights $\mathbf{w} = [w_1, w_2, \cdots, w_T]$. The quality of the clustering is quantified based on the separability of the multi-scale features, $\tilde{\mathbf{F}}$ as in Eq. 4.27, using the cost function proposed in Eq. 4.1 as follows:

$$\mathcal{L}(\boldsymbol{C}, \mathcal{H}(\mathbf{\Lambda}; \mathbf{w})) = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} ||\tilde{F}_{i\cdot} - \tilde{F}_{j\cdot}||^2 - \gamma \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} ||\tilde{F}_{i\cdot} - \tilde{F}_{j\cdot}||^2.$$

Similarly to Section 4.3, our goal here is to minimize this cost function in terms of both the graph partition, $\boldsymbol{C}$ and the optimal weights $\mathbf{w}$. The corresponding optimization problem can be formulated as:

$$\underset{\boldsymbol{C}, \mathbf{w}}{\text{minimize}}\ \mathcal{L}(\boldsymbol{C}, \mathcal{H}(\mathbf{\Lambda}; \mathbf{w})) + \alpha \mathcal{R}(\boldsymbol{C}), \tag{4.28}$$

where the regularization term $\mathcal{R}(\boldsymbol{C})$ will be specified to put additional constraints on the partition such that the connectivity information is also taken into account. We propose a two-step alternating minimization approach to solve this problem, where at each iteration l we first learn the optimal

graph partitioning for the given graph signal while the second step learns the optimal graph filter parameters:

$$\begin{aligned} \boldsymbol{C}^{(l+1)} &:= \underset{\boldsymbol{C}}{\operatorname{argmin}}\ \mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; \mathbf{w}^{(l)})) + \alpha \mathcal{R}(\boldsymbol{C}), \\ \mathbf{w}^{(l+1)} &:= \underset{\mathbf{w}}{\operatorname{argmin}}\ \mathcal{L}(\boldsymbol{C}^{(l+1)}, \mathcal{H}(\boldsymbol{\Lambda}; \mathbf{w})). \end{aligned} \tag{4.29}$$

4.4.2 Learning the partition, $\boldsymbol{C}$

Similarly to Section 4.3, the $\boldsymbol{C}$ update in Eq. (4.29) can be equivalently expressed by the $\bar{\mathbf{Z}}$ update equation, as in Eq. 4.8:

$$\begin{aligned} \bar{\mathbf{Z}}^{(l+1)} &:= \underset{\bar{\mathbf{Z}}, \bar{\mathbf{Z}}^\top \bar{\mathbf{Z}} = \mathbf{I}}{\operatorname{argmin}}\ \operatorname{tr}(\bar{\mathbf{Z}}^\top \mathbf{D}^{-1/2} \tilde{\mathbf{W}} \mathbf{D}^{-1/2} \bar{\mathbf{Z}}) - \alpha \operatorname{tr}(\bar{\mathbf{Z}}^\top \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \bar{\mathbf{Z}}), \\ &:= \underset{\bar{\mathbf{Z}}, \bar{\mathbf{Z}}^\top \bar{\mathbf{Z}} = \mathbf{I}}{\operatorname{argmin}}\ \operatorname{tr}(\bar{\mathbf{Z}}^\top (\tilde{\mathbf{W}}_n - \alpha \mathbf{A}_n) \bar{\mathbf{Z}}), \end{aligned}$$

whose optimal solution is the set of eigenvectors corresponding to the K smallest eigenvalues of $\tilde{\mathbf{W}}_n - \alpha \mathbf{A}_n$. The graph partition, $\boldsymbol{C}^{(l+1)}$, is then updated by applying k-means to the rows of $\bar{\mathbf{Z}}$.

4.4.3 Learning optimal weights, w

Once the partition $\boldsymbol{C}$ is learned, the goal is to determine the optimal weights for the multi-scale features that achieve the best separability between clusters.

Using $\tilde{\mathbf{F}}_i = \sum_{s=1}^{T} w_s [\mathbf{S}^{(s)}]_i = \mathbf{w}^\top \mathbf{S}_{(i)}$, with $\mathbf{S}^{(1)} := \boldsymbol{\Phi}_1^\top \mathbf{F}$ is the output of the graph scaling filter and $\mathbf{S}^{(s)} := \boldsymbol{\Psi}_s^\top \mathbf{F}$ is the output of the graph wavelet filter at scale s for $s \geq 2$, the cost function in (4.1) can be expressed in terms of the optimal weights as follows:

$$\sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} \mathbf{w}^\top (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top \mathbf{w} - \gamma \sum_{c=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} \mathbf{w}^\top (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top \mathbf{w}, \tag{4.30}$$

which can be rewritten to obtain the following optimization problem:

$$\underset{\mathbf{w}, \mathbf{w}\mathbf{w}^\top = 1}{\text{minimize}}\ \ (\mathbf{w}^\top (\mathbf{B} - \gamma \mathbf{C}) \mathbf{w}), \tag{4.31}$$

where $\mathbf{B}$ and $\mathbf{C}$ are the following $T \times T$ matrices:

$$\mathbf{B} = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{i,j \in C_k} (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^\top, \tag{4.32}$$

$$\mathbf{C} = \sum_{k=1}^{K} \frac{1}{\mathcal{D}(C_k)} \sum_{\substack{i \in C_k \\ j \notin C_k}} (\mathbf{S}_{(i)} - \mathbf{S}_{(j)})(\mathbf{S}_{(i)} - \mathbf{S}_{(j)})^{\top}. \tag{4.33}$$

The solution to the optimization problem is the eigenvector of $\mathbf{B} - \gamma\mathbf{C}$ corresponding to its smallest eigenvalue. Once the optimal weights $\mathbf{w}$ are obtained, the multi-scale features can be updated as $\tilde{\mathbf{F}} = \left(w_1 \mathbf{\Phi}_1^{\top} + \sum_{s=2}^{T} w_s \mathbf{\Psi}_s^{\top} \right) \mathbf{F}$.

4.5 Computational Complexity

The computational complexity of the algorithms is mostly due to the eigendecompositions and inverse operations at each iteration. There is only one full eigendecomposition at the beginning of the algorithm for the normalized Laplacian of the graph. Eigendecompositions of a $N \times N$ matrix, in general, have complexity on the order $O(N^3)$. However, it is important to note that this is the worst-case scenario. In practice, there are algorithms that approximate the spectral decomposition of graphs [61] reducing the computational complexity to $O(N^2)$. The work in [61] complexity bounds for spectral decomposition by restricting the analysis to graph families with good recursive separators. These graphs can be partitioned into roughly equal subgraphs with a small separator, allowing for efficient recursive spectral decomposition. We conducted an empirical analysis of the graphs used in this chapter using the Matlab Mesh Partitioning and Graph Separator Toolbox. Specifically, we applied the recursive spectral partitioning function and found that our graphs can be divided into multiple smaller subgraphs of nearly equal size, consistent with the properties required in [61]. Thus, the computational complexity of finding the spectral decomposition can be reduced to $O(N^2)$. At each iteration, we find the clusters by computing the eigenvectors corresponding to the K smallest eigenvalues of $\tilde{\mathbf{W}}_n - \alpha\mathbf{A}_n$. The computational complexity of finding the K eigenvectors corresponding to the smallest eigenvalues is $O(N^2K)$. The filter coefficients $\mathbf{h}$ for FIR and $\mathbf{a}$ for ARMA and the optimal weights for MSGWC are found by computing the eigenvector corresponding to the smallest eigenvalue of $T \times T$ matrices, with computational complexity $O(T^2)$. For finding $\mathbf{b}$ and $\mathbf{c}$, we have an inverse operation, and the standard matrix inversion algorithm has a time complexity of $O((Q-1)^3)$ and $O((M)^3)$ for finding $\mathbf{b}$ and $\mathbf{c}$, respectively. It is important

to note that $K, T, Q, M << N$, so the total complexity of both steps is dominated by $O(N^2K)$.

Table 4.1: Computational complexities of the optimization steps.

Clustering Step	$O(N^2K)$		
Filter Design Step	FIR	ARMA	MSGWC
	h: $O(T^2)$	**a**: $O(T^2)$	**w**: $O(T^2)$
		b: $O((Q-1)^3)$	
		c: $O((M)^3)$	
Total	$O(N^2K)$		

4.6 Experimental Results on Real Networks

4.6.1 Datasets

We evaluate the proposed graph filter learning method for both FIR and ARMA filters on five attributed networks. The first three, Cora, Citeseer, and PubMed [223], are citation networks where the nodes correspond to publications, and the edges correspond to citations among the papers. Cora has 2,708 machine learning papers classified into seven classes: case-based reasoning, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning, and theory. Citeeser has 3,327 machine learning publications classified into 6 classes: agents, artificial intelligence, database, information retrieval, machine learning, and human-computer interaction. PubMed has 19,717 papers classified into one of three classes: Diabetes Mellitus -Experimental, Diabetes Mellitus type 1, and Diabetes Mellitus type 2. Wiki [272] is a webpage network where the nodes are webpages and the edges are the links between them. It contains 2,405 long text documents classified into 17 classes. Sinanet [125] is a microblog users' relationship network where the edges between the users represent the followers/followees relationships. It contains 3,490 users from 10 major forums, including finance and economics, literature and arts, fashion and vogue, current events and politics, sports, science and technology, entertainment, parenting and education, public welfare, and normal life. The nodes in Cora and Citeseer are associated with binary word vectors indicating the presence or absence of some words, and the nodes in PubMed and Wiki are associated with tf-idf weighted word vectors. Sinanet is associated with a tf-idf weighted vector indicating the users' interest distribution in each forum. Table 4.2 summarizes the details for each dataset.

Table 4.2: Datasets statistics.

Dataset	Cora	Citeseer	Sinanet	Wiki	PubMed
Nodes	2708	3327	3490	2405	19717
Links	5429	4732	30282	17981	44338
Features	1433	3703	10	4973	500
Classes	7	6	10	17	3

4.6.2 Baseline Methods and Metrics

In order to evaluate the performance of our methods and the importance of taking into account both the topology and the attributes of a graph, we compare our method with three classes of methods.

- The first class of methods, K-means, spectral clustering, and CGFKM [81] only use the node attributes. CGFKM is a k-means based method that learns a Chebyshev polynomial approximation graph filter similar to our graph learning stage. However, this method does not use the connectivity of the network.
- The second class of methods only use the graph topology. Spectral clustering on the graph, SC-G, uses eigendecomposition on the graph Laplacian, Louvain [28] is a well-known community detection method, and Multi-Scale Community Detection (MS-CD) [248], uses graph spectral wavelets to compute a similarity metric between nodes to find communities at different scales.
- Finally, the third class of methods include AGC [286], GAE and VGAE [139], EGAE [284], ARGE and ARVGE [194], which use both node attributes and graph structure. AGC is based on spectral graph filtering, GAE and VGAE are benchmark autoencoder-based methods, while ARGE and ARVGE are benchmark adversarial GAE methods. EGAE is a GAE based model designed specifically for graph clustering.

The clustering performance of the different methods is quantified using Normalized Mutual Information (NMI) [65] and Adjusted Rand Index (ARI) [117]. NMI provides a quantitative measure of the agreement between the true class labels and the labels assigned by a clustering

algorithm, taking into account both precision and recall, offering a balanced perspective on the quality of the clustering solution. It can be computed as:

$$NMI = \frac{2I(Y;C)}{H(Y) + H(C)},$$

where Y is the set of true class labels, C is the set of cluster labels assigned by the algorithm, $I(Y;C)$ is the mutual information between Y and C, and $H(Y)$ and $H(C)$ are the entropies of Y and C, respectively. NMI ranges from 0 to 1, where 0 indicates no mutual information, and 1 implies perfect agreement between the true and predicted labels. ARI is a widely used metric in cluster analysis and machine learning for evaluating the similarity between two clustering solutions. It measures the agreement between the true class labels and the labels assigned by a clustering algorithm while correcting for chance and is given by:

$$ARI = \frac{RI - \text{Expected RI}}{\max(RI) - \text{Expected RI}},$$

where RI is the Rand Index, which measures the proportion of agreements between the true and predicted clusterings. Expected RI is the expected Rand Index under the assumption of random clustering. It represents the expected value of RI when clustering is performed randomly. The max(RI) term in the denominator represents the maximum possible Rand Index, which normalizes the ARI to the range [-1, 1].

4.6.3 Hyperparameter Selection

The input parameters required by our methods are the number of clusters K, the filter orders (T, Q) for GraFiCA, the number os scales T for MSGWC, and the hyperparameters, α for the clustering step, and γ for the filter optimization step in both methods. For a fixed γ, the optimal solution $\mathbf{h}$ in Eq. (4.14) is given by the eigenvector corresponding to the smallest eigenvalue of the matrix $\mathbf{B} - \gamma\mathbf{C}$, as expressed in equations (4.14) and (4.31). Since $\mathbf{B}$ and $\mathbf{C}$ correspond to the within- and between-cluster scatter of the filtered attributes, this formulation is analogous to the scatter difference criterion used in linear discriminant analysis [95]. The solution to this problem is equivalent to that of linear discriminant function when γ is the Lagrange multiplier for the following

optimization:

$$\min_{\mathbf{h}} \frac{\mathbf{h}^\top \mathbf{B}\mathbf{h}}{\mathbf{h}^\top \mathbf{C}\mathbf{h}}.$$

Due to the homogeneity of the Rayleigh quotient, we can normalize $\mathbf{h}^\top \mathbf{C}\mathbf{h} = 1$, and the problem can be rewritten as the following constrained minimization problem:

$$\min_{\mathbf{h}} \ \mathbf{h}^\top \mathbf{B}\mathbf{h} \ \text{ s.t. } \ \mathbf{h}^\top \mathbf{C}\mathbf{h} = 1,$$

which leads to the generalized eigenvalue problem $\mathbf{B}\mathbf{h} = \gamma \mathbf{C}\mathbf{h}$, where γ is the generalized eigenvalue. While the solution to this minimization problem would be the eigenvector corresponding to the smallest eigenvalue, we propose to consider all eigenvalues of the generalized eigenvalue problem as candidate values for γ. This broader selection allows us to explore different trade-offs in the balance between $\mathbf{B}$ and $\mathbf{C}$ scatter terms. To mitigate the effects of noise and ensure optimizing the accuracy, we perform a local grid search for γ between the minimum and maximum eigenvalues of the generalized eigendecomposition problem, $\gamma \in [\lambda_{min}, \lambda_{max}]$.

For selecting the best α , we tested $\alpha \in [0, 1]$, and the results for the filter with the best performance in terms of NMI are reported. The filter orders are determined within a range through exhaustive search. For the FIR filter, we evaluated different values for the filter order T between 3 and 10 to determine the order that gives the best NMI value for each dataset. For ARMA, we assume that $Q > T$ for a pair (T, Q) and $T + Q \leq 10$ [162]. For MSGWC we used $T = 10$, and $\mathbf{\Phi}_1$ is generated using $s_{\min} = 1/\lambda_2$ as the scale, as defined in [248]. The optimal values of the parameters α, γ, for both methods, and the filter orders T for FIR, (T, Q) for ARMA, are listed for each dataset in Table 4.3.

Our experimental results indicate that for sparse graphs, small α values perform the best whereas for fully connected dense graphs, α values close to 1 are optimal. It was also observed that the performance of GraFICA is robust to the filter order. In particular, the performance of ARMA filters is less sensitive to the order of the filter compared to FIR filters. In all of the tested datasets, the number of clusters is known. For the baseline methods, the parameter settings reported in the original papers are used [286, 139, 194].

Table 4.3: The optimal parameters for the different datasets.

	Dataset	Cora	Citeseer	Sinanet	Wiki	PubMed
FIR	α	0.056	0.06	0.044	0.001	0.01
	γ	0.074	0.10	0.022	0.03	0.42
	T	3	3	3	3	3
ARMA	α	0.05	0.058	0.05	0.001	0.01
	γ	0.08	0.101	0.03	0.028	0.40
	(T, Q)	(2,3)	(2,3)	(2,3)	(3,4)	(2,3)
MSGWC	α	0.03	0.110	0.041	0.001	–
	γ	0.072	0.124	0.03	0.032	–

4.6.4 Performance Evaluation

Table 4.4 shows the performance of all the methods, wherein the top three results are highlighted in bold. For all datasets, our method performs better than the state-of-the-art methods in terms of NMI. In particular, GraFiCA and MSGWC outperform all methods that use only the node attributes (first class of methods) or only the graph topology (second class of methods) in all cases our methods consider both the graph topology and node attributes, which complement each other and consequently enhance the clustering algorithm. In addition, GraFiCA and MSGWC outperform conventional GCN-based methods such as GAE, VGAE, ARGE, and ARVGE. This is due to the fact that in GraFiCA the filters are optimized for the clustering task, rather than for reconstruction, and the filter shape is not pre-determined. This leads us to consider larger neighborhoods unlike the traditional GCN-based methods which exploit information from only 2-hop neighborhoods. GraFiCA and MSGWC outperform spectral graph filtering based methods such as AGC in most cases as AGC only considers lowpass filters. For PubMed, AGC performs the best in terms of ARI, but GraFiCA performs the best in terms of NMI. Similarly, GraFiCA and MSGWC outperform EGAE which incorporates relaxed k-means into the decoder of GAE to learn embeddings optimized for graph clustering. This shows that while optimizing the weights in GAE for the clustering task improves the results compared to traditional GCN-based approaches, providing additional flexibility to the graph filter structure as in GraFiCA and MSGWC yields better clustering results. For Wiki, EGAE performs better than GraFiCA using FIR in terms of ARI, but GraFICA gives higher NMI.

This difference in ranking of the different methods using ARI vs. NMI may be due to the fact that Wiki is the most imbalanced data set. In such cases, ARI may be biased towards the larger clusters. For most datasets, the performance of the FIR and ARMA filters are very similar to each other. In the next section, we will discuss the interpretation of these results in terms of the learned filters.

Table 4.4: Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) results. The top three performance metrics in each case are shown by bold. OM indicates cases where the method ran out of memory.

	Cora		Citeseer		Sinanet		Wiki		PubMed	
Algorithms	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
k-means	0.2825	0.1621	0.3597	0.3279	**0.6413**	0.5828	0.3323	0.0578	0.2053	0.1810
SC-F	0.2856	0.1615	0.3244	0.2898	0.5395	0.3802	0.3974	0.0954	0.0164	0.0098
CGFKM [81]	0.1327	0.0522	0.0773	0.0011	0.6415	0.5664	0.0765	0.0025	0.1347	0.0911
SC-G	0.0872	0.0185	0.0573	0.0051	0.1653	0.0444	0.1709	0.0073	0.0075	0.0013
Louvain [28]	0.5044	0.3433	0.3645	0.3457	0.2490	0.2044	0.4105	0.2369	0.2059	0.1099
MS-CD [248]	0.5072	0.3453	0.4064	0.3926	0.4064	0.1997	0.3548	0.2677	0.1773	0.1465
AGC [286]	0.5170	0.3982	0.4086	0.4124	0.5573	0.3892	0.4268	0.1440	0.3158	**0.3105**
GAE [139]	0.4209	0.3160	0.1706	0.1018	0.4346	0.3523	0.1316	0.0781	0.2374	0.1955
VGAE [139]	0.4276	0.3188	0.2112	0.1440	0.4567	0.3881	0.2982	0.1143	0.2403	0.2119
EGAE [284]	0.5401	0.4723	0.4122	**0.4324**	0.3404	0.2773	0.4711	**0.3308**	0.3205	0.2893
ARGE [194]	0.4562	0.3865	0.2967	0.2781	0.4815	0.3781	0.3715	0.1129	0.2359	0.2258
ARVGE [194]	0.4657	0.3895	0.3124	0.3022	0.4854	0.3993	0.3987	0.1084	0.0826	0.0373
GraFiCA$_{\text{FIR}}$	**0.5465**	**0.4743**	**0.4228**	0.4283	**0.6578**	**0.6721**	**0.5125**	0.2771	**0.3279**	**0.2995**
GraFiCA$_{\text{ARMA}}$	**0.5421**	**0.4746**	**0.4261**	**0.4365**	**0.6561**	**0.6736**	**0.5150**	**0.2807**	**0.3265**	0.2915
MSGWC	**0.5662**	**0.4928**	**0.4223**	**0.4356**	0.6107	**0.5919**	**0.5005**	**0.2912**	OM	OM

4.6.5 Interpretation of the Filters Learned by GraFiCA

Figure 4.5a-4.2o show the frequency responses of the optimal FIR and ARMA filters for each dataset. For most datasets, the optimal filters have both a low-pass and high-pass region, thus extracting both smooth and non-smooth features from the node attributes. This is in contrast to GAE-based methods that are limited to first-order low-pass filtering and AGC, which is a higher-order low-pass filter. While the clustering performance of FIR and ARMA filters are similar to each other, the filter shapes obtained by ARMA are smoother as they fit IIR filters to approximate the same frequency response. It is also interesting to note that the optimal filter for Sinanet is an all-pass filter, as the original node attributes carry most of the class information. In this case,

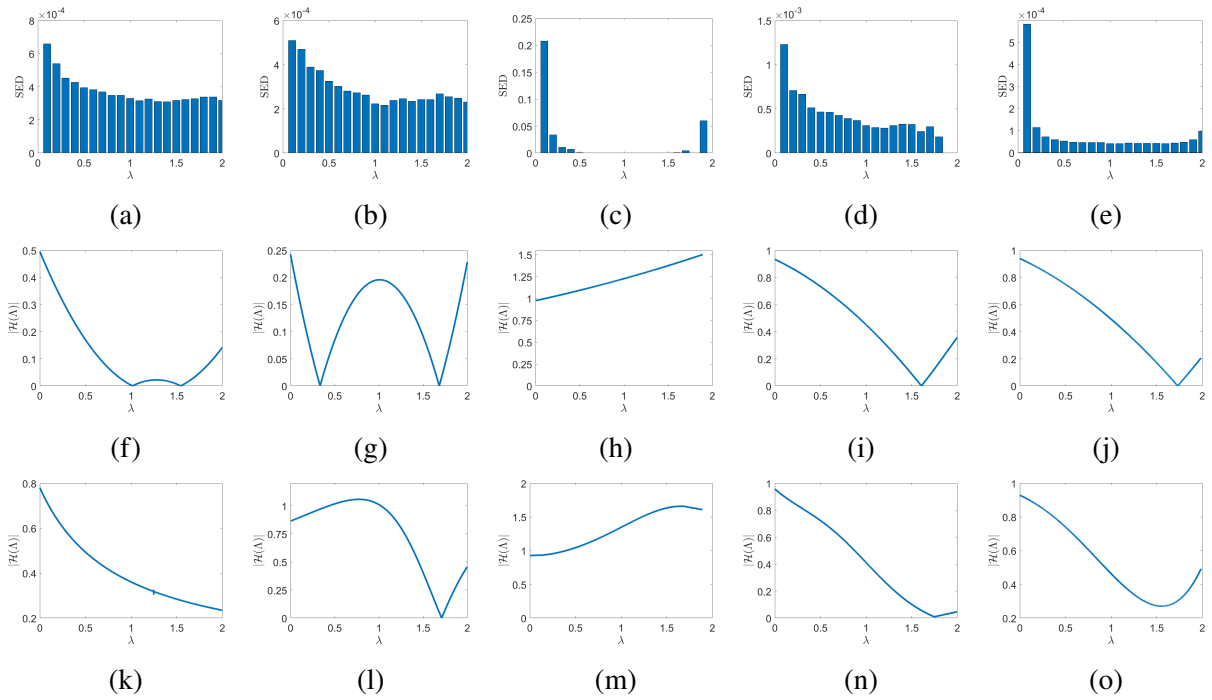


Figure 4.2: (a)-(e) Spectral Energy Distributions (SED) of the graph signal $\mathbf{F}$. (f)-(j) Optimal FIR graph filters for each dataset. (k)-(o) Optimal ARMA graph filters for each dataset. From left to right, Cora, Citeseer, Sinanet, Wiki, and PubMed, respectively.

filtering the attributes may not improve the accuracy of clustering as seen in the second row of Figure 4.3 where the attributes before and after filtering are very similar. This also explains why methods like k-means and CGFKM which only rely on the attributes perform well in clustering this dataset. On the contrary, for Cora, the clusters become better separated after filtering, as seen in the first row of Figure 4.3. This is in line with the poor clustering performance of methods that only use node attributes. These results indicate that our method adapts to the characteristics of the data and yields interpretable filters.

Except for the filters for Sinanet, which are all-pass, for most of the other datasets, the filters are mostly low-pass with content in the middle and high frequencies. The fact that our filters do not entirely suppress high-frequency components suggests a balance between preserving the cohesiveness of clusters and highlighting crucial differences across various regions of the graph.

In order to better understand and interpret the frequency responses of the learned filters, we examine the Spectral Energy Distribution (SED) of the graph signals with respect to the eigenvalues of the graph Laplacian. For a graph signal $\mathbf{F}$ and its graph Fourier transform $\hat{\mathbf{F}} = \mathbf{U}^\top \mathbf{F}$, the spectral

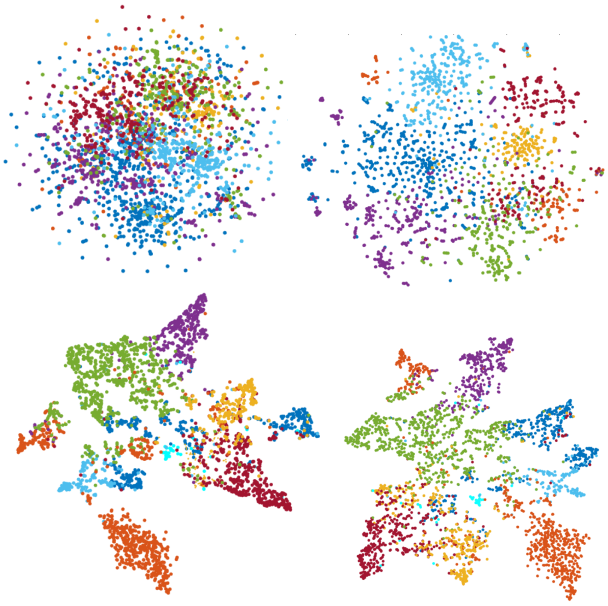


Figure 4.3: t-SNE visualization [254] of the original (left) and filtered (right) attributes for Cora (top row) and Sinanet (bottom row) for GraFiCA with FIR filter. Each color represents a cluster.

energy distribution at λ_i is defined as the average of $\hat{F}_{ip}^2 / \sum_{i=1}^{N} (\hat{F}_{ip})^2$ across the p attributes. Figure 4.2a-4.2e show the spectral energy distribution of the graph signals for each dataset. A concentration of energy in the lower end of the spectrum suggests that the graph signal is smooth and varies slowly across the graph. This can also indicate strong connectivity within certain graph regions or the presence of well-defined communities. Significant energy in the higher frequencies indicates that the graph signal has high variability or rapid changes across edges. This might also suggest that the graph contains regions of sparse connectivity and other regions with higher density.

As we can see in Figure 4.2a and 4.2b, Cora and Citeseer are similar in terms of their SED. They both have SED uniformly distributed across all frequencies with increased SED in the low frequencies. Both datasets represent papers in different machine learning areas, where it is fair to assume that the papers could be easily related to more than just one of those areas and therefore, the clusters are not very well defined. Due to this distribution of SED, the optimal filters for Cora and Citeseer have significant power in the middle frequencies. On the other hand, in Sinanet, the main areas of the 10 different forums range from parenting, history and arts, politics, to science and technology, and the clusters are well separated, hence there is a significant peak at the low frequencies as seen in Figure 4.2c. As for the peak in the high frequencies for SED of Sinanet,

this might be due to the different densities in the clusters. Wiki and Pubmed show similar behavior with respect to their SEDs, with most of the SED concentrated in the lowest frequencies as seen in Figure 4.2. Similar to their SED profiles, the shape of the optimal filters for these two datasets are similar and primarily lowpass.

We also evaluated the relationship between the cluster quality and the learned filter at each iteration of GraFiCA. Figure 4.4 shows the clustering at each iteration of the algorithm through t-SNE visualization along with the corresponding FIR filter for Cora. From this figure, it can be seen that when the clusters are not well-separated from each other, the learned filter emphasizes the high frequencies. After the first step of the iteration, while the points within clusters are moving closer to each other, the distance between clusters is also small. This results in a filter with significant low and high frequency parts. As the distance between clusters starts growing after iteration 2, the high frequency part of the filter gets attenuated. Thus, the scatter captured by $\mathbf{B}$ and $\mathbf{C}$ determine the low and high-frequency parts of the learned filter, respectively.

4.6.6 Interpretation of the Multi-scale Optimal Graph Filters

Figure 4.5 shows the frequency responses of the optimal filters obtained by MSGWC for Cora, Citeseer, Sinanet, and Wiki, where most of the filter responses are low frequency except for Sinanet where the learned weighting vector results in a wideband bandpass filter, which consistent with the earlier description of Sinanet, where clusters are already well separated.

Figure 4.6 illustrates the application of MSGWC to Cora where both the frequency response of the graph filters at different scales and their corresponding outputs are given. Additionally, the final filter after learning the optimal weight vector $\mathbf{w}$ and the multi-scale features $\tilde{\mathbf{F}}$ are shown. From the t-SNE maps, it can be seen that the clusters are better separated using the optimal multi-scale features. t-SNE maps corresponding to the features from individual scales do not show as much separation as the optimal multi-scale features. This suggests that multi-scale processing of the input attributes across a range of scales provides a more flexible representation and a better separation of clusters.

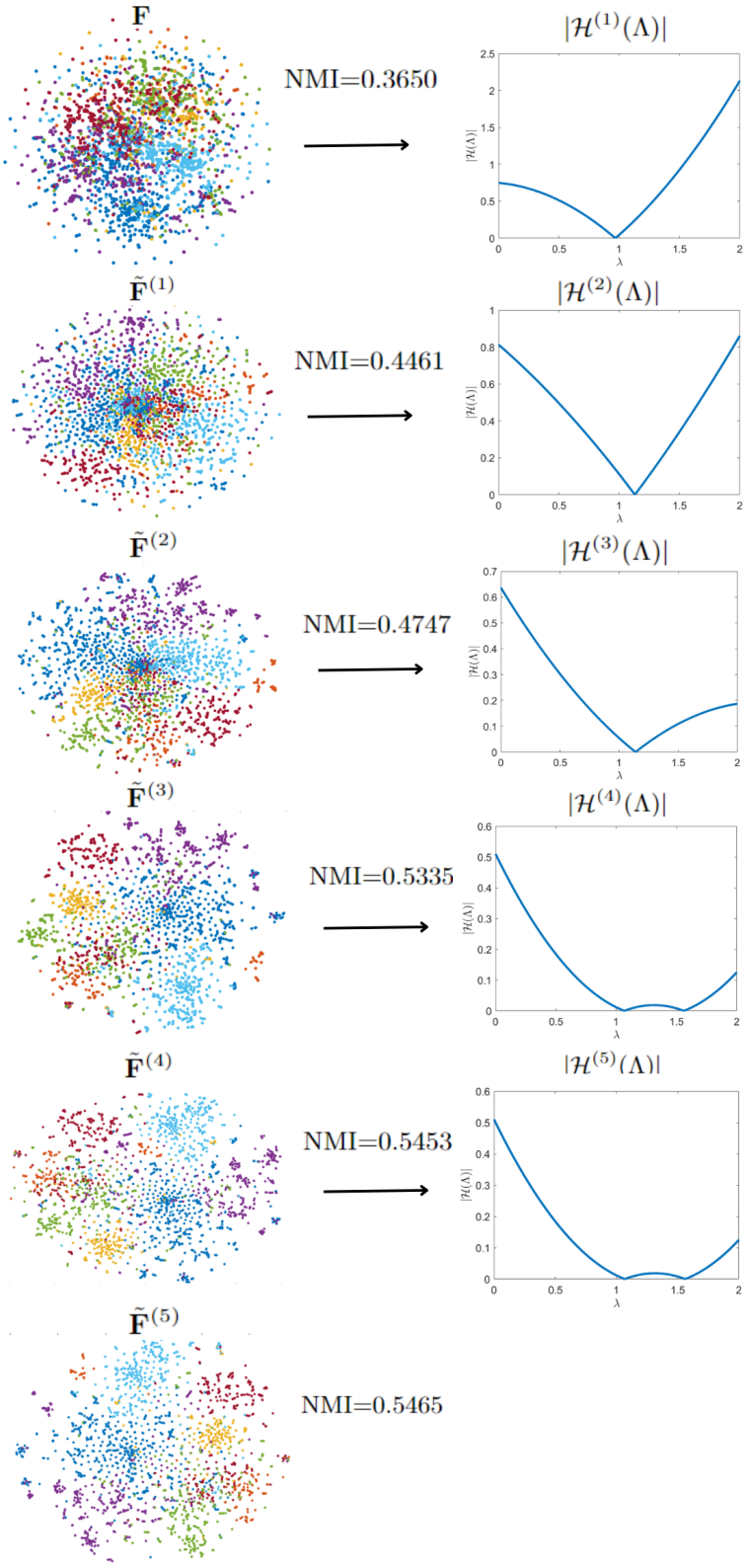
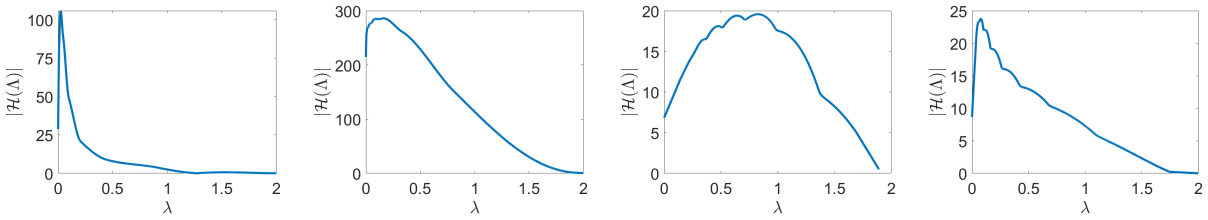


Figure 4.4: t-SNE visualization of the original and filtered signals and the corresponding learned FIR filter at each iteration of GraFiCA for the Cora dataset. Each color in the t-SNE visualization represents a cluster.



(a) (b) (c) (d)

Figure 4.5: Optimal graph filters for each dataset. From left to right, Cora, Citeseer, Sinanet, and Wiki, respectively.

4.6.7 Parameter Sensitivity

We investigated the effect of the two hyperparameters, α and γ, on clustering accuracy for FIR filters for $T = 3$ as seen in Figure 4.7. For sparse networks such as Cora and Citeseer, the connectivity information does not help to improve the performance, thus making the performance invariant to the choice of α. On the other hand, for dense networks such as Wiki, the performance is more sensitive to the choice of α. Finally, for Sinanet where most of the community information is reflected by the attributes, the performance is sensitive to the choice of γ as it quantifies the tradeoff between within and between class association of the filtered attributes.

4.6.8 Convergence Analysis

The optimization problem in (4.1) is solved in an alternating way, i.e., we fix one variable and optimize the other. When $\mathbf{h}$ is fixed, the solution of (4.8), $\bar{\mathbf{Z}}$, obtained by selecting the K eigenvectors corresponding to K smallest eigenvalues of $\tilde{\mathbf{W}}_n - \alpha\mathbf{A}_n$ is the global optimum solution to the problem in (4.8). Similarly, when $\bar{\mathbf{Z}}$ is fixed, i.e., the community structure is known, the solution of (4.14) $\mathbf{h}$ is the eigenvector corresponding to the smallest eigenvalue of $\mathbf{B}_T - \gamma\mathbf{C}_T$ and is a global solution to the problem in (4.14). Although $\bar{\mathbf{Z}}$ is the global optimum solution to (4.8), the partition $\boldsymbol{C}$ is found by applying K-means to $\bar{\mathbf{Z}}$. While k-means is known to converge quickly, it does not guarantee convergence to a global optimum. Thus, while each step of the algorithm converges to a global optimum, the final partition may not be globally optimum.

Figure 4.8 illustrates the empirical convergence behavior, where the value of the cost function as a function of the number of iterations is given for the different datasets. Despite the local nature of k-means convergence, our overall algorithm consistently converges within a few number

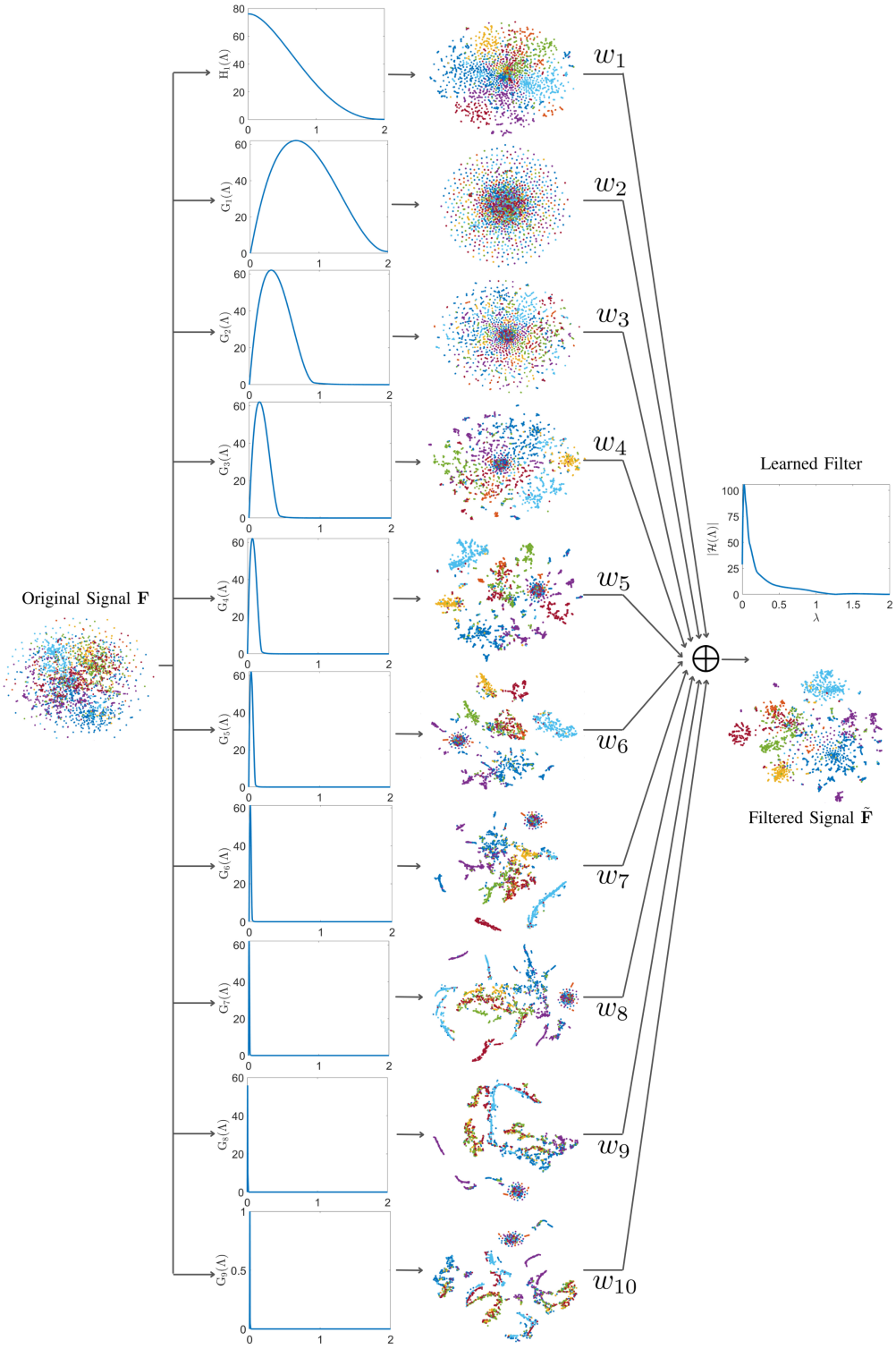


Figure 4.6: Application of the proposed framework on the Cora dataset. From left to right: Original signal $\mathbf{F}$, frequency response of graph filters at various scales with their respective outputs, and the final filter and the multi-scale features $\tilde{\mathbf{F}}$.

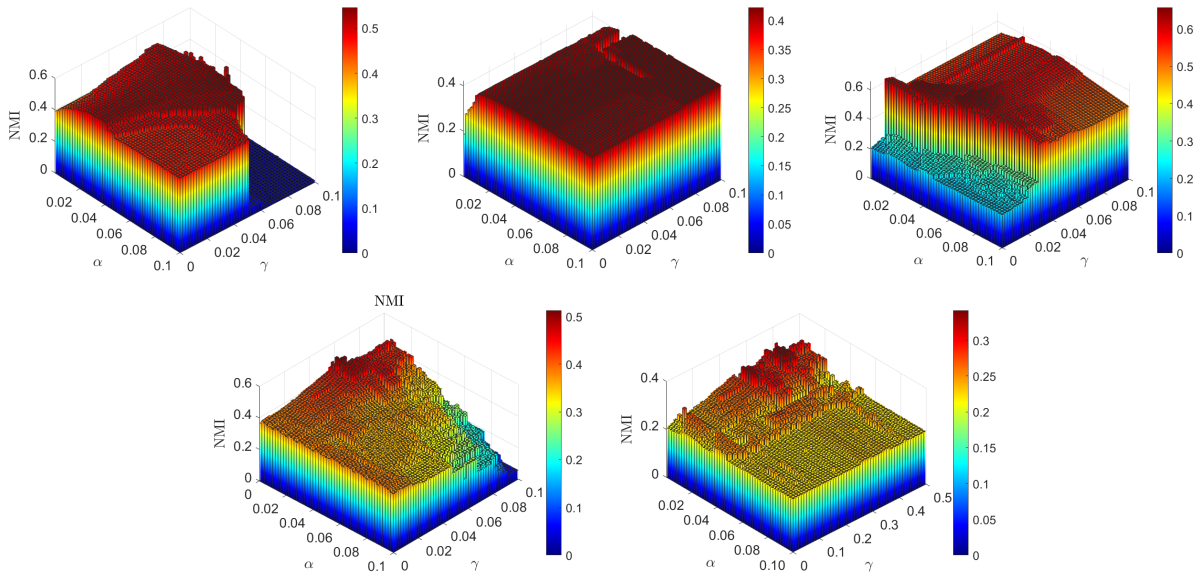


Figure 4.7: Parameter sensitivity for Cora, Citeseer, Sinanet, Wiki, and PubMed with $T = 3$.

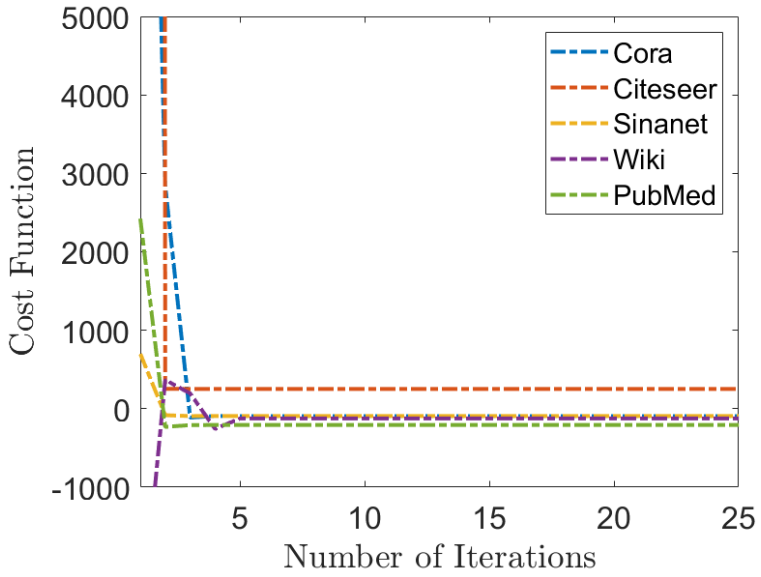


Figure 4.8: Cost function value vs the number of iterations.

of iterations. This analysis was done for GraFiCA with FIR filters but similar arguments can be made for ARMA and MSGWC.

4.6.9 Application to Brain Functional Connectivity Network

We applied GraFiCA to functional connectivity networks of the brain. Electroencephalogram (EEG) data collected from a cognitive control-related error processing study [108], i.e., Flanker task,

was used to construct both the graphs and the graph signals. The EEG was recorded following the international 10/20 system for placement of 64 Ag–AgCl electrodes. The sampling frequency was 512 Hz. After the removal of the trials with artifacts, the Current Source Density (CSD) Toolbox [246] was employed to minimize the volume conduction. In this study, trials corresponding to Error-Related Negativity (ERN) after an error response were used. Each trial was one second long. The total number of trials was 480 in which the total number of error trials in different participants varied from 20 to 61.

As previous studies indicate neural oscillations in the theta-band (4–7 Hz) may be one mechanism that underlies functional communication between networks involving medial prefrontal cortex (mPFC) and lateral prefrontal cortex (lPFC) regions during the ERN (25–75 ms time window) [108, 250, 44, 30], all analysis was performed for this time and frequency range. The average phase synchrony corresponding to theta band and 25-75 ms time window were computed to construct 64 $\times$ 64 connectivity matrices for each subject. The graph signal for subject l, $\mathbf{F}^l \in \mathbb{R}^{64\times 512}$, is defined as the average time series across trials for each electrode. In this chapter, we consider data from 20 participants. The FCNs across subjects can be modeled as a multiplex network with 64 nodes and 20 layers, corresponding to the number of brain regions and subjects, respectively.

GraFiCA is extended to multiplex networks with L layers with adjacency matrix $\mathbf{A}^l \in \mathbb{R}^{N\times N}$ and the corresponding graph signal, $\mathbf{F}^l \in \mathbb{R}^{N\times P}$ for the FIR filter. The consensus community structure across layers can be learned by extending the proposed cost function in (4.1) as

$$\sum_{l=1}^{L}\sum_{k=1}^{K}\frac{1}{\mathcal{D}^l(C_k)}\sum_{i,j\in C_k}||\tilde{F}^l_{i\cdot}-\tilde{F}^l_{j\cdot}||^2-\gamma\sum_{l=1}^{L}\sum_{k=1}^{K}\frac{1}{\mathcal{D}^l(C_k)}\sum_{\substack{i\in C_k\\ j\notin C_k}}||\tilde{F}^l_{i\cdot}-\tilde{F}^l_{j\cdot}||^2. \tag{4.34}$$

The partition $\boldsymbol{C}$ and the FIR filter coefficients $\mathbf{h}$ can be found by extending the corresponding optimization problems derived above as

$$\begin{aligned}\bar{\mathbf{Z}} &:= \underset{\bar{\mathbf{Z}},\bar{\mathbf{Z}}^\top\bar{\mathbf{Z}}=\mathbf{I}}{\operatorname{argmin}}\ \operatorname{tr}(\bar{\mathbf{Z}}^\top\sum_{l=1}^{L}(\tilde{\mathbf{W}}^l_n-\alpha\mathbf{A}^l_n)\bar{\mathbf{Z}}),\\ \mathbf{h} &:= \underset{\mathbf{h}}{\operatorname{argmin}}\ \ (\mathbf{h}^\top\sum_{l=1}^{L}(\mathbf{B}^l-\gamma\mathbf{C}^l)\mathbf{h}).\end{aligned} \tag{4.35}$$

For the selection of the parameters α and γ, we performed a grid search as described in Section 4.6.3 and selected the pair (α, γ) that achieves the highest modularity metric [164], since the ground truth is not available for this dataset and NMI cannot be computed.

Figure 4.9 shows the multiplex community structure for different numbers of clusters, K, from 4 to 8 across 20 subjects, and $\alpha = 1$, $\gamma = 0.1$, and $T = 3$. As shown in Figure 4.9, for each value of K, we consistently obtain a community comprised of the frontal-central nodes corresponding to the medial prefrontal cortex (mPFC), e.g. FCz, FCz, FC2. Frontal-central connectivity in theta oscillations is known to play a critical role in the flexible management of cognitive control [176]. In addition, the community structure reveals distinct communities corresponding to the visual and lateral prefrontal cortex (lPFC) areas.

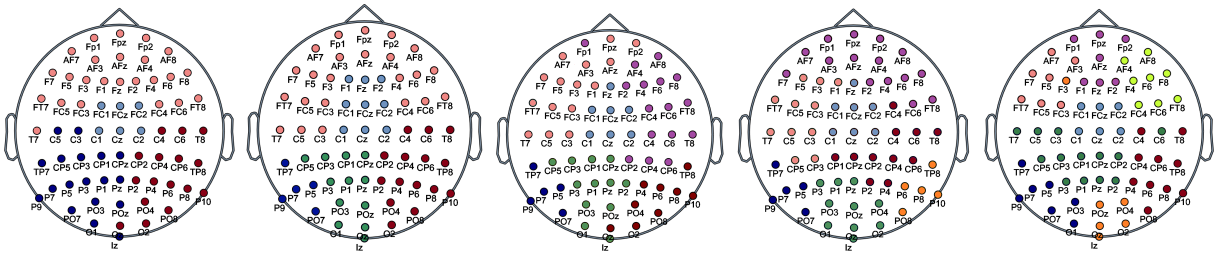


Figure 4.9: Community structure for $K = 4, 5, 6, 7, 8$.

Next, we evaluated the consistency of the community structure obtained from the multiplex network with the community structure of individual subjects. We applied GraFiCA to each layer individually, and the optimal FIR filter with $T = 3$ and the corresponding community structure for each subject was learned with different values of K ranging from 4 to 8. Scaled inclusivity (SI) [234] was employed as a metric to evaluate the consistency of the community structure across subjects. SI is calculated by measuring the overlap of communities across multiple networks while penalizing for the disjunction of communities [234, 181]. The Global Scaled Inclusity (GSI) [234, 181] across these 20 community structures is calculated. Figure 4.10 shows the GSI for $K = 4$ and $K = 6$. In both cases, the central nodes, FC1, FCz, FC2, C1, Cz, and C2, are among the 10 nodes with the highest GSI values. As we can see from Figure 4.9, these 6 nodes are consistently detected in the same community which indicates that the multiplex extension of the algorithm obtained

communities that are consistent with the individual subjects' community structure.

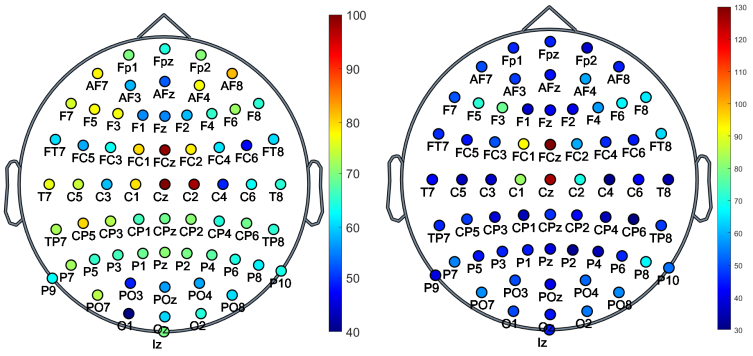


Figure 4.10: Global Scaled Inclusivity.

4.7 Conclusions

As the amount of large-scale network data with node attributes increases, it is important to develop efficient and interpretable graph clustering methods that identify the node labels. In this chapter, we proposed two community detection methods, GraFiCA and MSGWC, for attributed networks. The proposed methods were evaluated on real-world networks with both binary and numerical attributes. The proposed methods make some key contributions to the field. First, GraFiCA and MSGWC, learn the parameters of polynomial graph filters and the optimal linear combination of multi-scale features, respectively, with respect to a loss function that quantifies both the within-cluster dissimilarity and the separation between clusters. Thus, the filter parameters are optimized for the clustering task. Second, the proposed methods do not constrain the filters to be lowpass. Results indicate that the learned filters take into account the useful information in middle and high-frequency bands and the structure of the filter, i.e., whether the filter is lowpass, highpass or bandpass, is determined directly by the data. Third, GraFiCA is formulated for both FIR and IIR filters, providing similar performance across datasets. While FIR filters are computationally less expensive, IIR filters provide a smoother frequency response. Finally, the learned filters are evaluated with respect to spectral energy distribution of the attributed graphs providing interpretability to the proposed design procedure. The interpretability of MSGWC is illustrated through both the distribution of the learned features and the corresponding multi-scale filters. Future work will consider extensions to other nonlinear filters.

CHAPTER 5

GRAPH FILTERING LEARNING FOR STRUCTURE-FUNCTION COUPLING BASED HUB NODE IDENTIFICATION

5.1 Introduction

The recent developments in the field of human connectome research [41, 231] provide us the opportunity to unravel the topological characteristics of brain networks using graph-theoretic approaches. Both the structural and functional brain systems can be characterized using tools from complex network theory such as small-world topology, highly connected hubs and modularity [15, 230]. In this line of research, the brain is modeled as a graph composed of nodes and edges. The nodes represent neurons or brain regions and the edges represent physical connections or statistical associations between regions [27] for structural and functional networks, respectively. Graph-theoretic methods offer measures to depict the features of the network, including modules [115, 278] and hubs [39, 202].

Hubs are defined as densely connected regions in human brain networks and play a crucial role in global brain communication [253] and support a broad range of cognitive tasks, such as working memory [163] and semantic processing [270]. Growing evidence suggests that these highly connected brain hubs are preferentially targeted by many neuropsychiatric disorders [64], providing critical clues for understanding the biological mechanisms of disorders and establishing biomarkers for disease diagnosis and treatment [90].

Traditionally, hubs have been defined as nodes with high degree or high centrality based on functional connectivity networks. Node degree is the simplest and most commonly used means of identifying hubs in graphs. However, it has been shown that this approach is problematic in correlation based networks such as the functional connectivity networks [202]. The influence of community size on degree and the susceptibility of degree to distortion in volume-based brain networks result in biased estimates of hub nodes. For this reason, other centrality metrics based on the combination of degree and path length, e.g., betweenness, eigenvector, and PageRank centralities, have been proposed to characterize hubs [129]. Others have used the node role

approach that relies on the community structure of the functional network. Namely, centrality measures identify hubs and participation coefficients using within-module degree z-score then classify hub type, e.g., provincial vs. connector hubs [115, 199].

While these centrality measures are easy to implement, they rely only on the functional connectivity network without considering the coupling between the brain's anatomical wiring, i.e., structural network, and its dynamic functional properties, e.g., the BOLD signal [93]. In particular, they have been designed for networks represented only by simple graphs and do not take additional information about nodes, i.e., the node attributes, into account. The idea of integrating graph topology with node attributes for hub node identification first originated in social sciences literature [21, 87]. In this line of work, centrality measures are either weighted by the norm of the node attributes or modified by homophily of nodes defined by the attributes. Hub nodes integrate and distribute information over the network through their high number of connections made with a *diverse* set of nodes. Thus, using the attribute information along with the network topology can better reveal hubs as it allows differentiating highly connected nodes based on homophily, i.e., most of the connections are within a group or with nodes with similar attributes, versus heterophily. Specifically, heterophilic ones are of interest as they are not only highly connected but also connected to the nodes from different groups, indicating their importance for global communication of the network.

Although aforementioned methods utilize this benefit of node attributes for hub identification, they can only handle categorical attributes. In this chapter, we extend this idea to the case where the node attributes are continuous, which we consider as graph signals based on GSP literature. In particular, inspired by recent work in graph anomaly detection using graph neural networks (GNNs) [13, 96, 242], we utilize the spectral content of graph signals with respect to network topology to differentiate nodes as homophilic or heterophilic. Recent work shows the relationship between homophily and the smoothness of graph signals, while associating heterophily to non-smoothness [96]. In terms of spectral content, this indicates that homophilic and heterophilic activity concentrates at low-frequency and high-frequency spectral components, respectively. We adopt this prior research to hub node identification by using graph filters to separate homophilic

content from heterophilic content based on the graph signal spectrum and using the heterophilic content through a novel hub scoring to identify hubs.

Based on the previous discussion, in this chapter we introduce a GSP-based framework for hub node identification in brain networks utilizing both the structural connectome and functional BOLD signals. The proposed approach is based on learning the optimal graph filter for detecting hub nodes with the following assumptions: (i) hub nodes are sparse and have high activation patterns simultaneously with a more diverse set of connections (heterophilic), i.e., their activity corresponds to the high-frequency component of the BOLD signal, and (ii) the non-hub nodes' activation patterns are low-frequency/smooth with respect to the structural connectome, thus can be modeled as the output of low-pass polynomial graph filter. These assumptions are incorporated into a general optimization framework where the smoothness and sparsity are quantified by graph total variation and ℓ_1 norm, respectively. Once the optimal graph filter is learned, a hub scoring function based on the local gradient of the nodes is introduced to identify the hub nodes. Participation coefficient is used to further identify the connector hubs. The proposed method is evaluated on both simulated data and rs-fMRI data from HCP. The results are compared to the state-of-the-art hub node identification methods and recently published meta-analysis of hub nodes in rs-fMRI [271].

The main contributions of the proposed work are as follows. First, unlike existing hub node detection methods that rely on only the connectivity graphs, the proposed method, GraFHub, incorporates the structural connectivity and the functional activation signals into the same framework thus taking the structure-function coupling in the brain into account [93]. Second, in addition to introducing a GSP-based learning framework, this chapter also introduces a new smoothness-based metric for hub node scoring, as well as two different methods for identifying the hubs, i.e., thresholding vs. rank ordering. The proposed smoothness metric quantifies the local gradient of the graph signal with respect to the graph and thus quantifies the hub score by taking both the graph signal and the graph structure into account. Moreover, the smoothness metric is in line with the proposed cost function for hub node learning. Finally, by learning the optimal graph filter for separating

hub nodes from non-hub nodes, GraFHub provides interpretability to hub node identification. In particular, we show a strong correlation between the average hub score for a given brain network, the graph spectrum of the functional activation signals and the shape of the learned filters.

5.2 Related Work

5.2.1 Graph Signal Processing for the Brain

Tools from GSP have been adapted to study brain neuroimaging data in order to characterize anatomical, functional and pathological features of brain. A common approach in this line of work is to employ structural and diffusion MRI data to construct a brain graph representing anatomical features such as cortical morphology or white matter fiber architecture. The functional and pathological neuroimaging data is then treated as graph signals defined on the constructed structural brain graph. GSP concepts, such as graph Fourier transform (GFT) and graph filtering, have been utilized to analyze these functional and pathological signals with respect to the spectrum of the underlying structural connectivity. The early work focuses on extracting the graph Fourier modes of the functional brain signals (collected with fMRI [116, 204, 205]; or Electroencephalogram (EEG) [102]) with the structural connectivity graph estimated from diffusion MRI. This analysis reveals low (high) frequency modes that are aligned (disaligned) with respect to the underlying graph structure. For example, in [116], it is shown how eigenmodes of structural connectome constrain spatiotemporal patterns of neural dynamics in humans. Within this setting, [205] quantified the degree of structure-function dependency for each brain region by means of Structural-Decoupling Index (SDI).

Another line of work explored the use of graph filtering for neuroimaging data. Early works employ graph filters derived from the adjacency or Laplacian matrices of brain structural network for decoding brain states [177, 226]. In follow up work, fMRI data was processed by the polynomial approximated graph filters with different kernel functions for computationally efficient filtering on large voxel-based graphs [154, 18]. This line of work was extended to local filter design to concentrate energy onto specific graph nodes or predefined subgraphs. Kernel functions, such as the Slepian basis, was applied to construct graph filters adapted to diffusion MRI to focus energy

on the predefined subnetworks [31]. It was shown that the Slepian kernel and localized GFT that enable spatial and graph spectral localization classify fMRI data tasks better than the traditional GFT.

Finally, GSP-based approaches have been used for feature extraction for subsequent learning tasks. These features include signal or signal energy decomposed into different frequency bands [97], signal smoothness across underlying graph structure [180], and eigenvalues of graph Laplacian matrix. For instance, in [35], projection of resting-state fMRI (rs-fMRI) time series on a structural brain graph was used as features for autism spectrum disorder classification. In [288], a functional graph Laplacian embedding of deep neural networks is used to classify task fMRI time series, in a joint GSP-deep learning framework. Similarly, projection of recorded scalp EEG [132] and MEG data [217] onto the lower and higher graph frequency bands have been used to reduce the data dimensionality and extract features for Brain Computer Interface (BCI) and visual stimuli classification tasks, respectively.

5.2.2 Hub node identification

Current hub node identification methods can be grouped into two categories. The first category of methods determines hubs based on node centrality. These methods sequentially select a set of hub nodes by ranking a nodal centrality metric such as degree [188], clustering coefficient [189], vulnerability [131], betweenness [282], and eigenvector centrality [165]. However, detecting hubs only using high nodal centrality ignores the interdependencies in the networks resulting in the detection of provincial hubs, instead of connector hubs which predominantly connect nodes across different modules. The second group of methods uses module-based methods that identify hub nodes based on the network modularity [253]. These methods detect hub nodes by first identifying the modular organization of the network using a community detection algorithm [91]. Connector hub nodes are then detected based on the diversity of connections associated with the module partition. Although this method initially considers global network properties, the final hub detection uses a sorting-based method. Moreover, the optimality of the detected modules is not guaranteed [91]. In recent years, alternatives to these two categories of methods have been

proposed by combining multiple approaches, such as degree and participation coefficient, to obtain more reliable estimates of hubs [126].

Recently, graph spectral methods have been proposed to detect the hubs such that the removal of the identified hubs results in a network with multiple connected components, or equivalently an increase in the number of 0-eigenvalues in the graph Laplacian spectrum [274, 273]. GFT has also been employed to define a measure of centrality called GFT centrality (GFT-C) [227]. GFT-C first defines an importance signal for each node based on the shortest paths between that node and the other nodes. Hub scores are then determined by the weighted sum of GFT coefficients of the importance signal where the weight function is a pre-determined high-pass filter. Both graph spectral methods and GFT-C still rely on only the connectivity graph, i.e., the structural or functional connectome, without considering the coupling between the two modalities.

5.2.3 Graph Filter Learning

Graph filtering offers an extension of conventional filtering approaches to signals defined in the non-Euclidean domain, e.g., irregular data structures arising in biological, financial, social, economic, sensor networks etc. [78, 121]. Graph filters are information processing architectures tailored to graph-structure data and have been used for many signal processing tasks, such as denoising [54], signal recovery [173, 240], classification [53], and anomaly detection [80]. The design of graph filters to obtain a desired graph frequency response has been studied and analyzed in prior work [162, 220]. More recently, the problem of learning the optimal graph filter for a given task has been addressed. For example, in [210], the problem of blind deconvolution is addressed where the observed signals are modeled as the output of a graph filter and both the filter and the input signal are learned simultaneously. In [143], the problem of random graph signal estimation from a nonlinear observation model is addressed with an estimator that is parameterized in terms of shift-invariant graph filters. In all of these cases, the goal is graph signal recovery or reconstruction, minimizing the mean square error and not to identify the outlying nodes.

Closely related to the problem of hub node identification, spectral graph filtering has recently been employed in node-based anomaly detection. In particular, spectral properties of anomalies

have been analyzed using graph signal processing concepts and graph spectral filters [84, 94, 242, 96]. In [84], graph-based filters are employed to project graph signals onto normal and anomaly subspaces, and a thresholding mechanism is used to label anomalous instances. In [94], a community based anomaly detection approach is proposed using spectral graph filters. However, both of these methods use pre-determined filters such as the ideal low-pass/high-pass filters with no optimization of the filter shapes. More recently, the problem of estimating network centrality from the data observed on the nodes has been addressed [114, 212]. The data supported on the nodes is modeled as the output of a graph filter applied to white noise and centrality rank is learned without inferring the graph structure. This formulation reduces to determining centrality based on the principal components of the observed data's covariance matrix without taking the graph structure into account. Our proposed approach is similar to this line of work in the way it models the non-hub activity as the output of a graph filter. However, in our framework the graph structure is known and the non-hub activity is the output of an unknown graph filter where the input is observed.

5.3 Optimal Graph Filtering for Hub Node Identification

5.3.1 Problem Formulation

Given a graph $\mathcal{G} = (V, E, \mathbf{A})$ with N nodes and P graph signals $\mathbf{F} \in \mathbb{R}^{N \times P}$ defined on $\mathcal{G}$, in this chapter we focus on the detection of hub nodes. First, we model the observed graph signal $\mathbf{F}$ as $\tilde{\mathbf{F}} + \mathbf{F}_h$, where $\tilde{\mathbf{F}}$ and $\mathbf{F}_h$ correspond to the non-hub part and hub part of the observed graph signal, respectively. Note that with this decomposition, each node will have both non-hub and hub activity and whether a node is a hub node or not will be determined by the strength of $\mathbf{F}_h$ at that particular node. We characterize the spectral properties of these two parts of the observed graph signal through the following assumptions.

Assumption 1. $\mathbf{F}_h$ *is high-frequency with respect to* $\mathcal{G}$.

This assumption is inspired by the hypothesis that "hub regions possess the highest level of activity" [107]. This implies that hub nodes have different levels of activity compared to their neighbors, which leads to a 'right-shift' in spectral energy, i.e. the energy of graph signals concentrates less in low frequencies and more in high frequencies [242, 45].

Assumption 2. $\tilde{\mathbf{F}}$ *is the output of a low-pass graph filter defined in terms of the GSO,* $\mathbf{L}_n$.

This is the direct consequence of Assumption 1. As high-frequency part of the graph signals is attributed to $\mathbf{F}_h$, the remaining spectral content is mostly concentrated in low-frequency components. While this implies $\tilde{\mathbf{F}}$'s smoothness, Assumption 2 further models $\tilde{\mathbf{F}}$ as the output of a low-pass graph filter based on prior work in GSP literature. Prior work shows that the observed graph signals in a lot of applications can be viewed as the output of an unknown graph-based filter excited by an input [113, 85, 221]. This modeling provides flexibility and does not assume precise knowledge of graph generative models. In particular, graph-based filters model smoothness of signals, defined on a designated graph. Some examples include diffusion kernels defined on graphs [141] and polynomials of graph Laplacian matrices used to define localized diffusion operators on graphs. The assumption that observed graph signals are filtered by a low pass graph filter is commonly encountered in applications such as economics, social networks, power systems, and brain connectomics [209].

Assumption 3. *The number of hub nodes is much smaller than the non-hub nodes, i.e., hub nodes are sparse.*

This assumption implies $\mathbf{F}_h$ is a sparse matrix and is based on the fact that the graph data generally includes only a small number of hub nodes. We propose an optimization problem based on these three assumptions to learn $\tilde{\mathbf{F}}$, which is later employed to identify hub nodes. In particular, using Assumption 2, $\tilde{\mathbf{F}}$ can be learned by filtering out high-frequency components in observed signals $\mathbf{F}$, i.e., $\tilde{\mathbf{F}} = \mathcal{H}(\mathbf{L}_n)\mathbf{F}$, where $\mathcal{H}(\mathbf{L}_n) = \sum_{t=0}^{T-1} h_t \mathbf{L}_n^t$ is a low-pass graph filter to be learned. Since $\tilde{\mathbf{F}}$ is considered to be smooth in Assumption 2, the coefficients of $\mathcal{H}(\mathbf{L}_n)$ are learned such that the total variation of $\tilde{\mathbf{F}}$ as calculated by (1.1) is minimized. Finally, using Assumption 3, hub node activity, i.e., $\mathbf{F}_h = \mathbf{F} - \tilde{\mathbf{F}}$, needs to be sparse. This leads to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{h},\mathbf{h}^\top\mathbf{h}=1} \ & \alpha||\mathbf{F} - \tilde{\mathbf{F}}||_1 + \mathrm{tr}(\tilde{\mathbf{F}}^\top \mathbf{L}_n \tilde{\mathbf{F}}), \\ \text{s.t.} \quad & \tilde{\mathbf{F}} = \mathcal{H}(\mathbf{L}_n)\mathbf{F} = \sum_{t=0}^{T-1} h_t \mathbf{L}_n^t \mathbf{F}, \end{aligned} \tag{5.1}$$

where $\mathbf{h} = [h_0, h_1, \cdots, h_{T-1}]$ is the vector of filter coefficients with the added constraint that $\mathbf{h}^\top\mathbf{h} = 1$, i.e., the filter coefficients are normalized. The first term enforces the sparsity of the hub nodes (Assumption 3), the second term quantifies the smoothness of the filtered signal (Assumption 2), and α controls the trade-off between these two terms. Expressing the filtered signal as $\tilde{\mathbf{F}} = \mathcal{H}(\mathbf{L}_n)\mathbf{F}$, the problem in (5.1) becomes:

$$\min_{\mathbf{h},\mathbf{h}^\top\mathbf{h}=1} \quad \alpha||\mathbf{F} - \sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\mathbf{F}||_1 + \text{tr}\left(\mathbf{F}^\top\left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{L}_n\left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{F}\right). \tag{5.2}$$

5.3.2 Optimization

In this section, we derive the solution to (5.2) using ADMM [195]. By introducing an auxiliary variable $\mathbf{Z} = \mathbf{F} - \left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{F}$, the optimization problem is rewritten as

$$\min_{\mathbf{h},\mathbf{Z}} \text{tr}\left(\mathbf{F}^\top\left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{L}_n\left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{F}\right) + \alpha||\mathbf{Z}||_1, \text{ s.t } \mathbf{h}^\top\mathbf{h} = 1, \mathbf{Z} = \mathbf{F} - \left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{F}. \tag{5.3}$$

The corresponding scaled augmented Lagrangian is

$$\mathcal{L}(\mathbf{Z}, \mathbf{h}, \mathbf{V}) = \frac{\rho}{2}||\mathbf{Z} - (\mathbf{F} - \left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{F}) + \mathbf{V}||_F^2 + \text{tr}\left(\mathbf{F}^\top\left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{L}_n\left(\sum_{t=0}^{T-1} h_t\mathbf{L}_n^t\right)\mathbf{F}\right) + \alpha||\mathbf{Z}||_1, \tag{5.4}$$

where $\mathbf{V} \in \mathbb{R}^{N\times P}$ is the Lagrange multiplier. The ADMM steps are then as follows.

1. $\mathbf{Z}$ update: The variable $\mathbf{Z}$ can be updated as

$$\begin{aligned}
\mathbf{Z}^{(l+1)} &= \underset{\mathbf{Z}}{\text{argmin}}\ \mathcal{L}(\mathbf{Z}, \mathbf{h}^{(l)}, \mathbf{V}^{(l)}), \\
&= \underset{\mathbf{Z}}{\text{argmin}}\ \alpha||\mathbf{Z}||_1 + \frac{\rho}{2}||\mathbf{Z} - \mathbf{F} + \left(\sum_{t=0}^{T-1} h_t^{(l)}\mathbf{L}_n^t\right)\mathbf{F} + \mathbf{V}^{(l)}||_F^2, \\
&= S_{\frac{\alpha}{\rho}}\left(\mathbf{F} - \left(\sum_{t=0}^{T-1} h_t^{(l)}\mathbf{L}_n^t\right)\mathbf{F} - \mathbf{V}^{(l)}\right),
\end{aligned} \tag{5.5}$$

where $S_{\frac{\alpha}{\rho}}(\cdot)$ is the elementwise thresholding operator, which is the proximal operator of ℓ_1 norm [195].

2. $\mathbf{h}$ update: The filter coefficients $\mathbf{h}$ can be updated using:

$$\begin{aligned}\mathbf{h}^{(l+1)} &= \underset{\mathbf{h},\mathbf{h}^\top\mathbf{h}=1}{\operatorname{argmin}} \mathcal{L}(\mathbf{Z}^{(l+1)}, \mathbf{h}, \mathbf{V}^{(l)}), \\ &= \underset{\mathbf{h},\mathbf{h}^\top\mathbf{h}=1}{\operatorname{argmin}} \frac{\rho}{2}||\mathbf{Z}^{(l+1)} - \mathbf{F} + \left(\sum_{t=0}^{T-1} h_t \mathbf{L}_n^t\right)\mathbf{F} + \mathbf{V}^{(l)}||_F^2 + \operatorname{tr}\left(\mathbf{F}^\top \left(\sum_{t=0}^{T-1} h_t \mathbf{L}_n^t\right)\mathbf{L}_n \left(\sum_{t=0}^{T-1} h_t \mathbf{L}_n^t\right)\mathbf{F}\right).\end{aligned} \tag{5.6}$$

Following the definitions in [220], we define the t times shifted input signal as $\mathbf{S}^{(t)} := \mathbf{U}\Lambda^t\mathbf{U}^\top\mathbf{F}$. We also define $\mathbf{S}_{(i)} \in \mathbb{R}^{T\times P}$ corresponding to node i where each row is the t times shifted input signal at the i-th node, i.e. $[\mathbf{S}_{(i)}]_{t\cdot} := [\mathbf{S}^{(t)}]_{i\cdot}$. Hence, the filtered graph signal at node i is $\tilde{\mathbf{F}}_{i\cdot} = \sum_{t=0}^{T-1} h_t [\mathbf{S}^{(t)}]_{i\cdot} = \mathbf{h}^\top \mathbf{S}_{(i)}$ and we have $\tilde{F}_{ip} = \mathbf{h}^\top \mathbf{s}_i^p$ where $\mathbf{s}_i^p = [\mathbf{S}_{(i)}]_{\cdot p}$. The objective function in (5.6) can then be written element-wise:

$$\mathcal{L}(\mathbf{Z}^{(l+1)}, \mathbf{h}, \mathbf{V}^{(l)}) = \frac{\rho}{2}\sum_{p=1}^{P}\sum_{i=1}^{N}(Z_{ip}^{(l+1)} - F_{ip} + \mathbf{h}^\top\mathbf{s}_i^p + V_{ip}^{(l)})^2 + \sum_{p=1}^{P}\sum_{i,j=1}^{N}(\mathbf{h}^\top\mathbf{s}_i^p)L_{ij}(\mathbf{h}^\top\mathbf{s}_j^p).$$

Taking derivative of $\mathcal{L}(\mathbf{Z}^{(l+1)}, \mathbf{h}, \mathbf{V}^{(l)})$ with respect to $\mathbf{h}$ and equating it to 0 yields $\mathbf{h}^{(l+1)} = -\mathbf{Y}^{-1}\mathbf{b}$, where

$$\mathbf{b} = \rho\sum_{p=1}^{P}\sum_{i=1}^{N}\mathbf{s}_i^p(Z_{ip}^{(l+1)} - F_{ip} + V_{ip}^{(l)}),$$

$$\mathbf{Y} = \sum_{p=1}^{P}(2\sum_{i,j=1}^{N}\mathbf{s}_i^p L_{ij}\mathbf{s}_j^{p^\top} + \rho\sum_{i=1}^{N}\mathbf{s}_i^p\mathbf{s}_i^{p^\top}).$$

Finally, in order to satisfy the constraint $\mathbf{h}^\top\mathbf{h} = 1$, $\mathbf{h}^{(l+1)}$ is projected onto the set defined by $\mathbf{h}^\top\mathbf{h} = 1$.

3. $\mathbf{V}$ update: The Lagrangian multiplier can be updated using:

$$\mathbf{V}^{(l+1)} = \mathbf{V}^{(l)} + \rho(\mathbf{Z}^{(l+1)} - \mathbf{F} + \left(\sum_{t=0}^{T-1} h_t^{(l+1)}\mathbf{L}_n^t\right)\mathbf{F}). \tag{5.7}$$

These three variables are updated until convergence as described in Algorithm 5.1. Since our problem is a non-smooth convex optimization over a non-convex manifold $\mathbf{h}^\top\mathbf{h} = 1$, when applying ADMM to the proposed optimization problem, there are no formal global convergence guarantees. However, ADMM is known to perform well on non-convex problems, often converging to locally optimal solutions [33, 263]. Recent works also empirically show the convergence of ADMM for non-smooth problems over non-convex manifolds [142].

Algorithm 5.1: GraFHub.

Input: Adjacency matrix $\mathbf{A}$, graph signal $\mathbf{F}$, parameters α, ρ, and filter order T.
Output: $\tilde{\mathbf{F}}$, graph filter $\mathcal{H}(\mathbf{\Lambda})$.

$\mathbf{L}_n \leftarrow \mathbf{I} - \mathbf{A}_n$
$[\mathbf{U}, \mathbf{\Lambda}] \leftarrow \text{EVD}(\mathbf{L}_n)$
$\mathbf{S}^{(t)} \leftarrow \mathbf{U}\mathbf{\Lambda}^t\mathbf{U}^\top\mathbf{F}, t \in \{0, 1, \cdots, T-1\}$
$[\mathbf{S}_{(i)}]_{t\cdot} := [\mathbf{S}^{(t)}]_{i\cdot}$, for each node $i \in V$
Initialize $\mathbf{h} = \text{rand}(T, 1)$, $\mathbf{V} = \text{rand}(N, P)$
while $||\mathbf{h}^{(l+1)} - \mathbf{h}^{(l)}||^2 > 10^{-3}$ **do**
    update $\mathbf{Z}^{(l+1)}$ according to Eq. (5.5)
    update $\mathbf{h}^{(l+1)}$ according to Eq. (5.6)
    update $\mathbf{V}^{(l+1)}$ according to Eq. (5.7)
end while
$\tilde{\mathbf{F}} \leftarrow \mathbf{U}\left(\sum_{t=0}^{T-1} h_t^{(l+1)}\mathbf{\Lambda}^t\mathbf{U}^\top\right)\mathbf{F}$

5.3.3 Hub Scoring

Once the filtered graph signal $\tilde{\mathbf{F}}$ is obtained, hubs are scored using the graph signal's local smoothness, node gradient [228], before and after filtering:

$$\text{scores}(i) = E(i) - \tilde{E}(i),$$

where $E(i) = \sum_{j=1}^{N} A_{ij}||\mathbf{F}_{i\cdot} - \mathbf{F}_{j\cdot}||^2$ and $\tilde{E}(i) = \sum_{j=1}^{N} A_{ij}||\tilde{\mathbf{F}}_{i\cdot} - \tilde{\mathbf{F}}_{j\cdot}||^2$ are the node gradients at node i for the original and filtered signals, respectively. This metric quantifies the difference in the similarity of a node's value to its neighbors before and after filtering. It is hypothesized that for hub nodes, this difference will be larger as the hub nodes' activity tends to be dissimilar to its neighbors (Assumption 1).

Once the scores for each node $i \in V$ are computed, hubs are detected in two different ways: *(i)* thresholding and *(ii)* top K-hubs. For thresholding, we use the z-score approach, i.e., nodes whose z-score is larger than 3 are denoted as hubs. For the top K-hub approach, we consider the top-K nodes with the highest scores as hubs [273]. In our experiments, K is chosen as the point where there is a significant drop in the hub scoring metric, similar to the elbow criterion [183], as detailed in Section 5.4.

5.4 Experiments on simulated data

5.4.1 Benchmark Methods

In this section, we evaluate the performance of our method, GraFHub, on simulated data. We compare the accuracy of GraFHub to three groups of commonly used hub node detection methods. The first group of methods relies only on the graph topology. This group includes graph-theoretic centrality measures such as the degree, eigenvector, and betweeness centrality [253], Graph Fourier Transform Centrality (GFT-C) [227], a GSP based method that uses the GFT coefficients of an importance signal derived from the shortest path with respect to a particular node, and joint hub identification (JHI) method [273] which uses the spectrum of the graph to detect connector hubs. The second group of methods only relies on graph signals and does not consider graph connectivity. This class includes clustering methods, such as Isolation Forest [159]. These methods learn an anomaly region and classify the nodes based on whether the node resides within the region or not. Unlike our method, none of these methods utilizes both the graph topology and the graph signals.

Finally, we compare our learning based method to a fixed graph filtering based method, graph high-pass filtering (GHF) [291]. GHF utilizes the connectivity information and the graph signal to detect the hub nodes. GHF solves the following optimization problem to obtain the graph signals, $\tilde{\mathbf{F}}$, corresponding to the non-hub activity:

$$\min_{\tilde{\mathbf{F}}} \left\| (\mathbf{I} + \beta \mathbf{L}_n)^{1/2} \left(\tilde{\mathbf{F}} - \mathbf{F} \right) \right\|_F^2 + \frac{\xi}{2} \mathrm{tr} \left(\tilde{\mathbf{F}}^T \mathbf{L}_n \tilde{\mathbf{F}} \right), \tag{5.8}$$

where β is a parameter that controls the cutoff frequency of the high-pass filter, and ξ is the regularization parameter. Similar to GraFHub, GHF learns $\tilde{\mathbf{F}}$ that is smooth with respect to the underlying graph. Unlike GraFHub, this method does not learn the filter shape and does not enforce sparsity on the hub nodes.

5.4.2 Simulated Data

The simulated data are generated by first constructing a graph G from either Erdös-Rényi (ER) or Barabási–Albert (BA) models. ER model creates a random graph where each edge is generated independently with probability p, resulting in a graph with no inherent structure, where edges are

uniformly distributed. In our simulations, an ER random graph with N nodes and edge density probability $p = 0.1$ is generated. BA model follows a preferential attachment mechanism with growth parameter m, producing a scale-free graph. In particular, a graph with N nodes is generated iteratively starting from an initial graph with $m + 1$ nodes. At each iteration, a new node is added to the graph and the new node has m edges, which preferentially attach to the higher-degree nodes. The generated graph has a few highly connected nodes and many nodes with fewer connections, mimicking real-world networks like social and biological networks. In the following, BA parameter m is set to 3.

Once $\mathcal{G}$ is constructed, P smooth graph signals, $\mathbf{X} = [\mathbf{x}_1|\cdots|\mathbf{x}_P]$, are generated using Tikhonov filtering, i.e., $\mathbf{x}_p = (\gamma\mathbf{L}_n + \mathbf{I})^{-1}\mathbf{x_0} \in \mathbf{R}^N$, $p \in \{1, \ldots, P\}$, as the non-hub nodes' activity where $\mathbf{x_0} \sim \mathcal{N}(0, \mathbf{I})$ and γ is the degree of smoothness. We then add synthetic hub nodes to $\mathbf{X}$ by selecting a fixed percentage of the nodes as hubs. For the ER graph, these hub nodes are selected randomly, while for the BA graphs, they are selected in two different ways: *(i)* hub nodes are selected as the nodes with the highest degree in the generated graph (BAdegree); *(ii)* half of the hub nodes are selected randomly and the other half are selected from nodes with the highest degree (BAmixed). The signal values of selected hub nodes are perturbed by adding uniform noise in the interval $[-u\sigma, u\sigma]$ where σ is the standard deviation of the norm of $\mathbf{X}_{\cdot i}$, and u is the strength of the hubs. Unless noted otherwise, $N = 1000$, $P = 100$, $\gamma = 30$, $u = 1$, and the hub nodes represent 10% of N.

For all experiments, we report the performance of the aforementioned methods and our proposed method, GraFHub The best α and T values for GraFHub are determined from $\alpha \in [0.001, 0.01, 0.1, 1, 10, 50, 100, 1000, 2000]$ and T between 2 and 6. The results with the best AUC performance are reported. For GHF, $\xi/2 = 1/\alpha$, and $\beta = 1$. For GFT-C, the weights of the weighting function are computed as described in [227]. For JHI, $\rho = 2$ and μ is selected from the same set of values as α above, and the results with the best AUC values are reported. For signal-based anomaly detection methods, i.e., Isolation Forest, the default parameters are used. For Isolation Forest, number of estimators is set to 100. The results with the best performance are reported. The performance is quantified using Area Under the Receiver Operating Characteristic

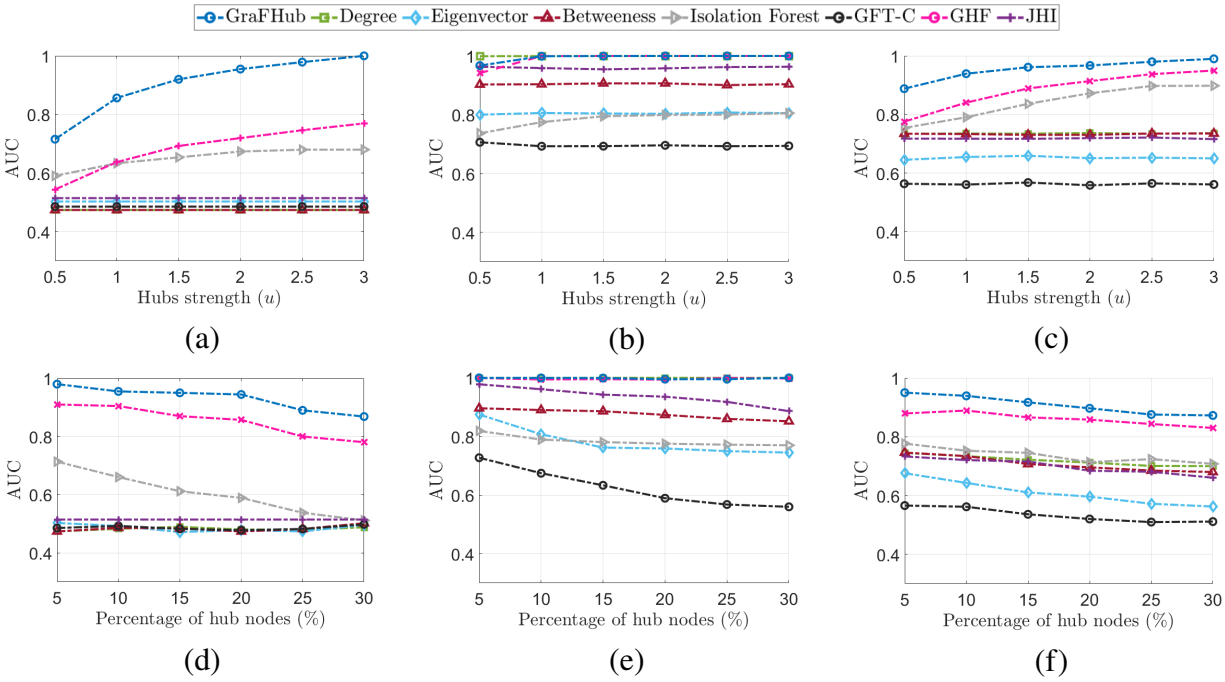


Figure 5.1: Performance of GraFHub on synthetic attributed graphs. AUC vs. Hub Signal Strength (u) (first row) and AUC vs. Percentage of nodes that are hubs (second row). From left to right, ER, BAdegree, and BAmixed models.

curve (AUC-ROC) where the hub scores returned by the methods are used without any thresholding or ranking. The average AUC-ROC over 50 runs is reported.

Experiment 1: Strength of hubs

In the first experiment, we vary the strength of the hubs, u. The top row of Figure 5.1 shows the results for ER (Figure 5.1 (a)), BAdegree (Figure 5.1 (b))and BAmixed (Figure 5.1 (c)) as u is increased from 0.5 to 3. It can be seen that AUC-ROC score improves as u increases for most of the methods that utilize graph signals for hub identification since the hubs become better separated. Methods that rely only on network connectivity alone, such as centrality measures and JHI, show no change in performance as hub strength increases. These methods are inherently limited because they do not take the graph signal into account. In ER and BAmixed models, GraFHub outperforms all of the other methods even when the strength of the hub signal is weak. GraFHub effectively captures the differences between hub and non-hub nodes by learning an optimal graph filter, allowing for better hub identification even when the hub nodes' strength is weak. For BAdegree case, the performances of GraFHub, GHF and degree centrality are close to each other as shown

in Figure 5.1 (b). This is expected since hubs in BAdegree are defined by high connectivity, which aligns with the definition of hubs in degree centrality. GHF performs similarly to our method for BAdegree; however, GraFHub outperforms GHF for ER and BAmixed cases as it learns the filter from the graph and the observed node activity unlike GHF which uses a pre-determined filter.

Experiment 2: Percentage of hub nodes

In the second experiment, we evaluate the performance of the methods as the percentage of hub nodes increases. The bottom row of Figure 5.1 shows AUC-ROC for ER (Figure 5.1 (d)), BAdegree (Figure 5.1 (e)), and BAmixed (Figure 5.1 (f)), where the percentage of hub nodes is increased from 5% to 30%. The performance of most methods decreases as the percentage of hub nodes increases. In the case of GraFHub, this drop in performance can be attributed to one of the core assumptions of our method, i.e., the sparsity of hub nodes. The methods that rely only on connectivity information or node signal information are not affected by the number of hubs. However, they still perform worse than methods that use both connectivity and graph signals. In general, GraFHub has higher AUC-ROC scores than the other methods. For BAdegree, GraFHub, and degree centrality show similar performance since hub node definition aligns with degree centrality as discussed above. For ER and BAmixed, GraFHub performs the best followed by GHF. In particular, when the number of hub nodes is small, the performance gain by GraFHub is apparent thanks our sparsity assumption.

5.5 Application to Resting State fMRI Data

5.5.1 HCP Data

The proposed method is applied on structural and functional neuroimaging data from 56 subjects collected as part of the Human Connectome Project (HCP)[1]. The subjects are selected from HCP's healthy young adult study (HCP 900) and include 34 females and 22 males, in the age range 26-35. This subject group was selected in a previous study [205] as the data is complete and does not have any missing sessions. The consent forms, including consent to share de-identified data, were collected for all subjects (within the HCP) and approved by the Washington University institutional review board. All methods were carried out in accordance with relevant guidelines and regulations.

[1] `db.humanconnectome.org`

Data acquisition is performed using a Siemens 3T Skyra with a 32-channel head coil [255]. The scanning protocol includes high-resolution T1-weighted scans (256 slices, 0.7 mm^3 isotropic resolution, TE = 2.14 ms, TR = 2400 ms, TI = 1000 ms, flip angle = 8°, FOV = 224×224 mm^2, BW = 210 Hz/px, iPAT = 2) [100]. Diffusion data is collected with Spin-echo EPI, TR = 5520 ms, TE = 89.5 ms, flip angle = 78°, refocusing flip angle = 160°, FOV = 210×180 mm^2 (RO × PE), matrix = 168×144 (RO × PE), 111 slices with thickness of 1.25 mm and 1.25 mm isotropic voxel size, multiband factor 3, echo spacing = 0.78 ms, BW = 1488 Hz/px, phase partial Fourier = $\frac{6}{8}$, b-values = $1000, 2000$, and 3000 s/mm^2. Functional scans were collected using a multi-band sequence with MB factor 8, isotropic 2 mm^3 voxels, TE = 33 ms, TR = 720 ms, flip angle = 52°, FOV = 208×180 mm^2 (RO × PE), 72 slices, 2.0 mm isotropic resolution, BW = 290 Hz/px, echo spacing = 0.58 ms [100]. One hour of resting state data was acquired per subject in 15-minute intervals over two separate sessions with eyes open and fixation on a crosshair. Within each session, oblique axial acquisitions alternated between phase encoding in a right-to-left (RL) direction in one run and left-to-right (LR) in the other run. Minimally ICA-FIX cleaned resting-state fMRI (rsfMRI) and diffusion-weighted preprocessed images from HCP are used in the following analysis.

5.5.2 Preprocessing

The images in the HCP dataset were minimally preprocessed as described in [100]. Briefly, each image was corrected for gradient distortion and motion and aligned to a corresponding T1-weighted (T1w) image with one spline interpolation step. This was further corrected for intensity bias and normalized to a mean of 10,000 and projected to a $32k_f s_L R$ mesh, excluding outliers, and aligned to a common space using a multi-modal surface registration [211].

Building on this preprocessing framework, diffusion-weighted scans are analyzed using MRtrix3[2] to construct the structural connectomes. The following operations are employed: multi-shell multi-tissue response function estimation, Glasser's multimodal cortical atlas parcellation, constrained spherical deconvolution, and tractogram generation with 10^6 output streamlines. The volume is split into $N = 360$ regions in two hemispheres (180 areas on the right and 180 areas on

[2] https://www.mrtrix.org/

the left). In each hemisphere, Glasser divides 180 "areas" into 22 separate "regions", which are referred to as the 22 larger partition cortices. Each of the 180 regions occupies one of 22 cortices which are displayed in a separate atlas. The number of fibers connecting two regions divided by the atlas regions' volume is used to quantify the structural connectivity.

As an additional preprocessing step, resting-state fMRI data is cleaned of structured noise using ICA-FIX, a method that combines independent component analysis with the FSL tool FIX to automatically remove artifactual or "bad" components [104]. Following this denoising step, functional volumes are spatially smoothed with a Gaussian kernel (5 mm full-width at half-maximum). The first 10 volumes are discarded so that the fMRI signal achieves steady-state magnetization, resulting in $P = 1190$ time points. Voxel fMRI time courses are detrended and band-pass filtered [0.01 - 0.15] Hz to improve the signal-to-noise ratio for typical resting-state fluctuations. Finally, Glasser's multimodal parcellation (the same used for the structural connectome) resliced to fMRI resolution is used to parcellate fMRI volumes and compute regionally averaged fMRI signals. These were z-scored and stored in an $N \times P$ matrix. The functional connectivity network for each subject is also constructed by computing the pairwise Pearson correlation between the time-series data corresponding to each region. For baseline comparison with respect to functional connectivity based hub node detection, node strengths, i.e., degrees, of the functional connectome are computed as the sum of absolute correlation values.

5.5.3 Hub Node Detection

GraFHub is applied to HCP data from two sessions, where the structural networks correspond to the graphs and the BOLD signals to the graph signals. The hub nodes for all subjects and sessions are detected separately based on thresholding or top K-hub node methods as described in Section 5.3.3. For the thresholding method, hubs are nodes with z-score greater than 3. For the top K-hub node method, the value of K is determined by calculating the average hub score of each node across subjects, and K is set to the value where there is a significant drop in the average hub score, as shown in Figure 5.2. Based on Figure 5.2, we select $K = 8$.

The detected hub nodes are further filtered through the participation coefficient to identify the

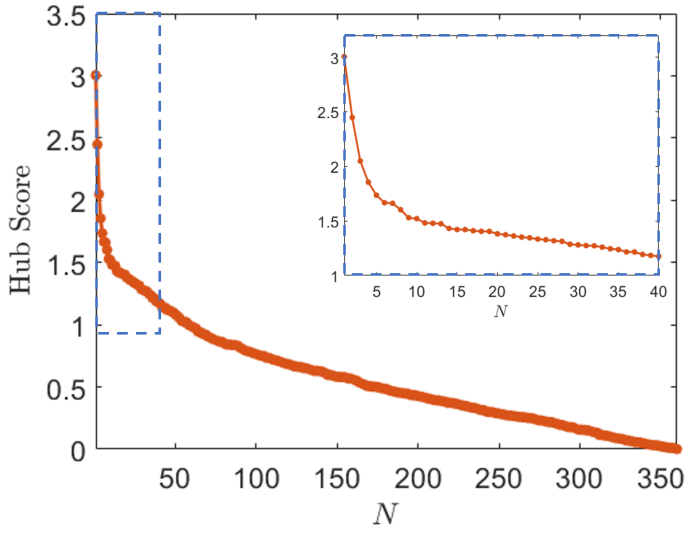


Figure 5.2: Average hub scores across subjects are sorted in decreasing order.

connector hubs. For this purpose, the Louvain algorithm [28] is applied to each subject's structural connectivity graph to detect the community structure. Each node's participation coefficient, which quantifies how evenly a node's connections are distributed with respect to community structure, is computed as [106]:

$$P_i = 1 - \sum_{s=1}^{N_M} \left(\frac{k_{is}}{k_i}\right)^2, \tag{5.9}$$

where N_M is number of identified modules (communities), k_i is degree of node i and k_{is} is the total strength of the connections node i makes with nodes in module s. The hub nodes with participation coefficient $0.35 < P < 0.72$ are considered as connector hubs [106].

Since hub node detection is an unsupervised task, we determined the optimal values for the filter order, T, and the hyperparameter, α, in (5.1) based on the consistency of hubs across subjects inspired by [271]. For each (T, α) pair, a 56×22 matrix is constructed where each row indicates the number of hub nodes in a brain cortex (based on the Glasser parcellation) for a given subject across the two sessions. Next, the consistency of the detected hub nodes across subjects is quantified by computing the 56×56 correlation matrix, $\mathbf{C}$. $\|\mathbf{C}\|_F$ is computed to quantify the average correlation of detected hub nodes in each region across subjects. The (T, α) pair that yields the highest norm is selected. In Figure 5.3, we show the effect of T and α on the consistency of the detected hubs

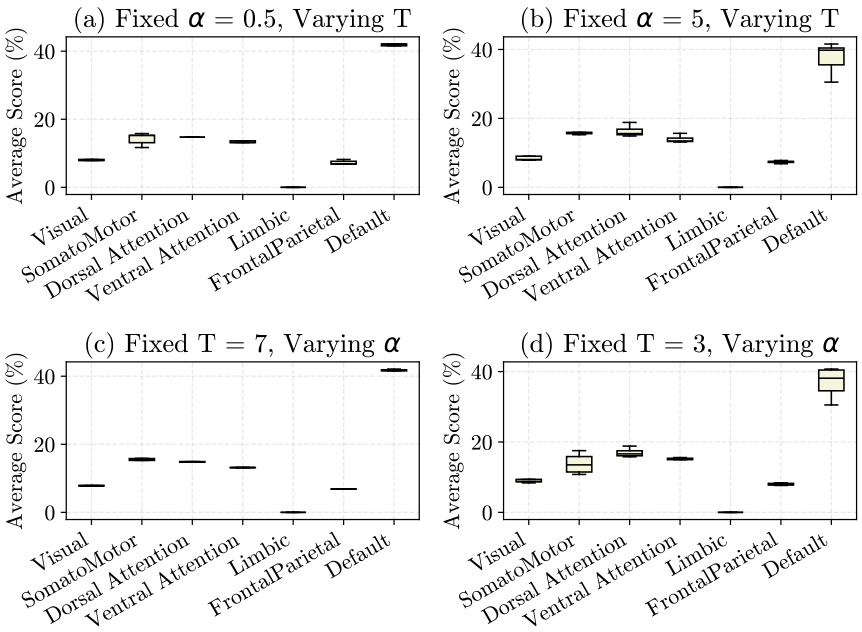


Figure 5.3: Robustness of GraFHub to the choice of the hyperparameters α and T: Top row shows the variation of hub nodes in each brain region with respect to T for fixed α. Bottom row shows the variation of hub nodes in each brain region with respect to α for fixed T.

across subjects in different brain networks. In the top row, we show the effect of varying the filter order for $\alpha = 0.5$ and $\alpha = 5$. When α is small, i.e., the importance of error is minimized, it can be seen that the number of hub nodes in each brain region does not vary much. On the other hand, when α is large, i.e., the importance of sparsity is high, the variation of hub nodes increases for Default Mode Network. In the bottom row, we show the effect of varying α for $T = 7$ and $T = 3$. When $T = 7$, varying α does not change the number of hub nodes in each brain region. When $T = 3$, the variation increases for somatomotor and default mode networks. This result aligns with our understanding of graph filters and brain networks. $T = 3$ only captures local neighborhoods and may not be sufficient for correctly identifying the hub nodes. Moreover, somatomotor and default mode networks play an important role in rs-fMRI, thus they are known to have more hub nodes. Thus, a change to the filter order T or the sparsity parameter α will affect these regions more than others. For the thresholding method, the optimal parameters are found as $T = 6$ and $\alpha = 5$. For the top K-hub method, the optimal parameters are $T = 6$ and $\alpha = 0.2$.

Figure 5.4 illustrates the top-K connector hubs detected by GraFHub across all subjects for one

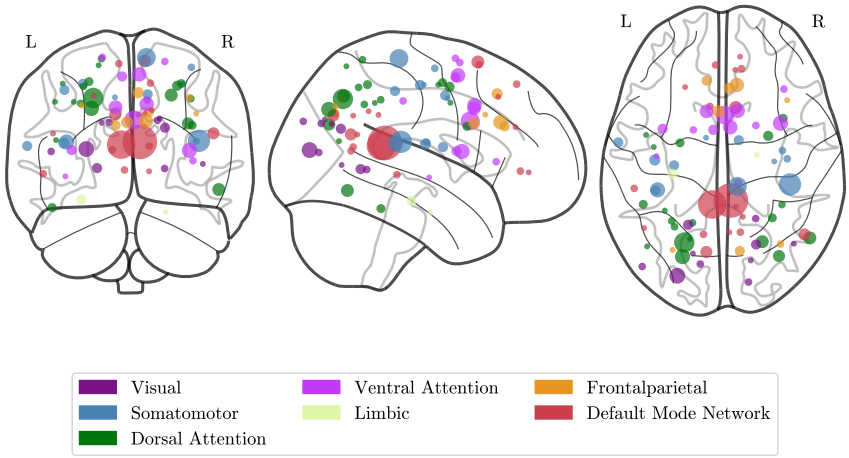


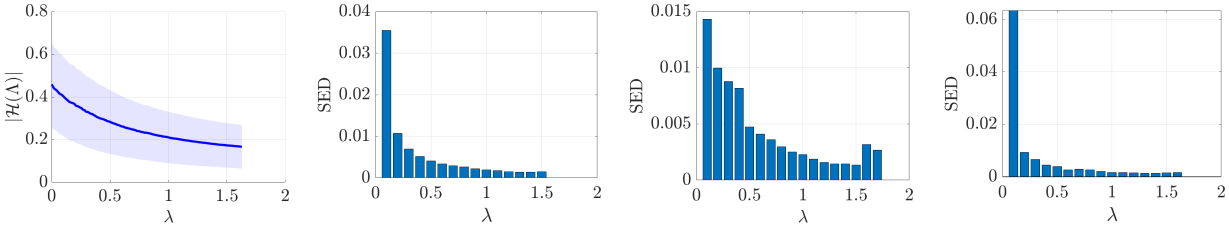
Figure 5.4: Consistency of the top-K hubs detected by GraFHub across all subjects plotted over the brain topomap. The size and the color of the nodes correspond to the number of times across 56 subjects a particular node has been detected as a hub and the brain network (Yeo's parcellation networks) to which the node belongs, respectively.

run. In particular, the size of the hub denotes how many times a particular node has been detected as a hub across 56 subjects. The color of the hub denotes the resting state brain network, determined by Yeo's parcellation [278], the node belongs to. From this figure, it can be seen that nodes in the default mode network (DMN), frontal parietal and dorsal attention networks are consistently selected as hubs across subjects.

For both detection methods for GraFHub, the percentage of hubs within a brain network across two sessions and all subjects are calculated similarly to a recently published meta-analysis of hub nodes in rs-fMRI [271]. While the hub detection is performed on the Glasser parcellation with 360 regions, the percentage of hubs is reported for larger brain networks determined by Yeo's parcellation with 7 networks (Table 5.1). We compare the different implementations of GraFHub, denoted GraFHub$_{\text{Z}}$ and GraFHub$_{\text{K}}$ for the thresholding and top K-hubs approaches, respectively, with the methods discussed in Section 5.4. For centrality-based methods, i.e., degree, eigenvector, and betweenness, nodes with top-10 hub scores are selected as hubs. For methods that are only based on graph connectivity, i.e., centrality-based measures, GHFC, GFT-C and JHI, hubs are further filtered using participation coefficient to find connector hubs similar to GraFHub. For Isolation Forest, all detected hubs are used since they only employ graph signals.

Table 5.1: Percentage of hub nodes detected in brain networks defined by Yeo's parcellation [278].

Networks (area%)	Degree	Eigenvector	Betweenness	Isolation Forest	GHFC	GFT-C	JHI	Meta Analysis [271]	GraFHub_Z	GraFHub_K
Visual (14.8%)	38.2	40.4	7.5	3.9	4.8	14.6	35.3	9.9	8.0	6.3
SomatoMotor (20.2%)	26.7	26.5	8.2	3.6	15.9	10.7	23.0	14.4	16.0	18.7
Dorsal Attention (11.4%)	13.1	12.3	6.8	5.7	15.5	22.9	12.0	16.5	14.9	17.7
Ventral Attention (12.1%)	15.4	15.2	6.4	4.8	18.8	9.1	10.1	15.6	11.2	12.4
Limbic (7.8%)	0.1	0.0	39.4	58.9	0.5	9.9	0 .0	0.2	0.0	0.5
FrontalParietal (12.9%)	3.0	2.4	8.2	4.5	12.6	9.1	6.9	15.9	9.6	10.9
Default (20.8%)	3.7	3.3	23.5	18.8	31.9	23.7	12.6	27.5	40.4	33.4



(a) (b) (c) (d)

Figure 5.5: (a) Average filter response across subjects. (b) Average Spectral Energy Distribution (SED) of the graph signal $\mathbf{F}$ across all subjects. (c) SED of the graph signal $\mathbf{F}$ of subject 13 (Average Hub Score = 3.7354). (d) SED of the graph signal $\mathbf{F}$ of subject 36 (Average Hub Score = 2.7851).

5.5.4 Frequency Response of the Learned Filters

In Figure 5.5a, the frequency response of the learned filter, $\mathcal{H}(\mathbf{\Lambda})$, averaged across subjects, is shown. As expected, the frequency response is low-pass since $\tilde{\mathbf{F}}$ corresponds to the non-hub or smooth activity on the graph (Assumption 2). The standard deviation of the filter across subjects shows that while there's subject variation in the magnitude response of the filter, the overall shape across subjects does not change.

In order to better understand and interpret the frequency response of the learned filters, we examine the Spectral Energy Distribution (SED) of the graph signals with respect to the eigenvalues of the graph normalized Laplacian. For a graph signal $\mathbf{F}$ and its GFT $\widehat{\mathbf{F}} = \mathbf{U}^\top \mathbf{F}$, the spectral energy distribution at i-th eigenvalue λ_i is defined as the average of $\widehat{F}_{ip}^2 / \sum_{i=1}^{N} (\hat{F}_{ip})^2$ across the P graph signals. Figure 5.5b shows the average SED of the graph signals across all subjects. As it can be seen, the average of the spectral energy distribution across all subjects is mostly localized in the low-frequency range, similar to the learned filter in Figure 5.5a. This alignment of the filter shape with the SED profile is expected as the filter is learned to capture the non-hub activity, i.e.,

low-frequency content.

In order to further investigate the relationship between the underlying signal's graph spectrum and the learned hub nodes, we compute the ratio of the total SED in the high-frequency band to the total SED in the low-frequency band as $\sum_{\lambda_i \geq 1} \text{SED}(\lambda_i) / \sum_{\lambda_i < 1} \text{SED}(\lambda_i)$ for each subject. In addition, the hub activity for each subject is quantified by the average score of the top-K hub nodes. Figure 5.5c shows the SED of subject 13, which is the subject with the highest hub activity and the highest SED ratio. On the other hand, Figure 5.5d shows the SED of subject 36, which is the subject with the lowest hub scores and the lowest SED ratio. From these figures, it can be seen that subject 13 has significant high-frequency activity compared to subject 36, whose SED is mostly concentrated in the low frequencies. The average hub activity z-scores for subject 13 is 3.7354, compared to the average hub activity z-scores across all subjects 3.0959. On the other hand, the hub activity z-scores for subject 36 is 2.7851. These results validate Assumption 1 as the subjects with high hub activity have more high-frequency content.

5.5.5 Inter-Subject Variability

The consistency of the hubs detected by GraFHub across subjects is quantified using normalized entropy. For each cortex r, we construct a vector $\mathbf{x}^r \in \mathbb{R}^{56 \times 1}$ whose entries correspond to the number of hub nodes within cortex r for a particular subject. After normalizing this vector, $p_i^r = \frac{x_i^r}{\sum_{i=1}^{56} x_i^r(i)}$, we calculate the normalized entropy for cortex r as:

$$S^r = -\frac{\sum_{i=1}^{56} p_i^r \log_2(p_i^r)}{\log_2(56)}. \tag{5.10}$$

The higher the entropy, the more consistent the number of hubs is in that cortex across subjects.

In order to quantify the significance of the normalized entropy estimates, we utilize bias-corrected and accelerated (BCa) bootstrapping with 9,999 samples and calculate the normalized entropy of each sample. We report the cortices with significant normalized entropy values at the 95% confidence interval in Table 5.2.

Cortices with high normalized entropy correspond to regions with higher consistency across subjects. For example, posterior cingulate cortex (PCC) has the highest normalized entropy for

GraFHub and is part of the DMN which has the highest percentage of hubs according to Table 5.1. Similarly, anterior cingulate and medial prefrontal cortices have high normalized entropy and correspond to frontoparietal and ventral attention networks. Thus, the statistical significance of the brain networks with high percentage of hub nodes in Table 5.1 is established through normalized entropy as cortices with significantly high normalized entropy correspond to these networks.

Table 5.2: Normalized entropy of the hub nodes determined by GraFHub for each cortex. – refers to cortices where no hub nodes are detected for any of the subjects. $*$ refers to cortices that have statistically significant normalized entropy values.

Cortices	$\text{GraFHub}_{\text{Z}}$	$\text{GraFHub}_{\text{K}}$
Anterior Cingulate and Medial Prefrontal	0.64*	0.77*
Auditory Association	0.00	0.00
Dorsal Stream Visual	0.46	0.58
Dorsolateral Prefrontal	0.17	0.30
Early Auditory	0.50	0.67*
Early Visual	–	–
Inferior Frontal	–	0.17
Inferior Parietal	0.34	0.53
Insular and Frontal Opercular	0.00	0.49
Lateral Temporal	0.00	0.16
MT+ Complex and Neighboring Visual Areas	0.33	0.47
Medial Temporal	–	0.00
Orbital and Polar Frontal	–	–
Paracentral Lobular and Mid Cingulate	0.17	0.60
Posterior Cingulate	0.80*	0.90*
Posterior Opercular	0.16	0.34
Premotor	0.00	0.39
Primary Visual	–	–
Somatosensory and Motor	0.50	0.61
Superior Parietal	0.56*	0.70*
Temporo Parieto Occipital Junction	–	0.00
Ventral Stream Visual	–	–

5.5.6 Verification of Hubs Through Global Efficiency

Global Efficiency (GE) is a metric that characterizes the efficiency of a parallel working system, where all the nodes in the network exchange information simultaneously [146, 3] and is, therefore, a measure of integration and global communication efficiency. Given a graph, GE is defined as the average inverse shortest path length in the network, which is inversely related to the characteristic

path length [216, 215]:

$$\mathrm{GE} = \frac{1}{N(N-1)} \sum_{i,j \in V, i \neq j} \frac{1}{d_{ij}}, \tag{5.11}$$

where d_{ij} is the shortest path between nodes i and j. A small-world network will have GE greater than a regular lattice but GE less than a random network.

In order to verify that the detected hub nodes are important for the overall information processing in the brain network, we calculate the change in GE when a hub node is removed from the network. In particular, we remove each node and its connections from the network and calculate the GE of the resulting network. We then compute the difference in the global efficiency of the network before and after node removal. We repeat this procedure for each node. A larger difference in GE implies that the removed node is important for information processing, i.e., it has "hub-like" characteristics. For each subject, we calculate the average difference in global efficiency of the network before and after removing non-hub and hub nodes as shown in the last two columns of Table 5.3.

Table 5.3: Average difference in GE for hub nodes detected by GraFHub compared to hub nodes removed by other methods.

Method	ΔGE (10^{-4})
$\mathrm{GraFHub}_{\text{Non-Hubs}}$	(0.59 ± 0.13)
$\mathrm{GraFHub}_{\text{Hubs}}$	(7.40 ± 4.46)
Degree	(1.02 ± 0.6)
Eigenvector	(1.06 ± 0.6)

As expected, the difference in global efficiency for removing hub nodes is larger than removing non-hub nodes for every subject. The average difference is about 12.1 times the global efficiency loss when non-hub nodes are removed. This result shows that the detected hub nodes are indeed more important for information processing in the brain and contribute more to the small-world characteristics.

In addition, we compared the change in GE when hub nodes detected by GraFHub are removed versus the change for hub nodes detected by two other centrality measures. The loss in global efficiency was greater for GraFHub compared to the loss when hub nodes detected by eigenvector and degree centrality are removed. This indicates that our method detects hub nodes that contribute

more to the network organization and information transfer compared to traditional methods (see the first three columns of Table 5.3.

5.6 Discussion

From Table 5.1, it can be seen that graph-theoretic methods such as degree and eigenvector centrality detect hub nodes that are concentrated in the visual and somatomotor networks. These networks comprise the primary sensory-motor cortices and have been shown to have high global connection in prior studies [58]. This high connectivity may be either due to the relatively large size of the network or reflect the privileged placement of visual processing in the human brain [251]. Similarly, JHI detects hub nodes consistent with degree centrality as it relies directly on the graph spectrum of the connectivity graph without using the BOLD signal. On the other hand, betweenness centrality measure detect hub nodes in the limbic and default mode networks. While the DMN is known to be a critical network during resting state [70], less is known about the limbic network. Similarly, methods that only utilize the functional BOLD signals primarily detect hub nodes in the limbic network. This network consists of regions outside the cerebral cortex and is important for emotion, reward, and other valence processing [192]. Unfortunately, the areas that form this resting state network are usually poorly visualized with fMRI due to nearby portions of the skull creating susceptibility artifact [222]. Thus, the BOLD signal values in this network may be very different from those in other regions, causing methods that only utilize the signals to detect these as hub nodes.

As there is no ground truth for hub nodes, in this chapter, we compare our results to a recent harmonized meta-connectomic analysis [271] of resting-state functional MRI data of 5212 healthy young adults across 61 independent cohorts. The majority of the hub nodes detected by GrafHub are in the DMN followed by somatomotor and dorsal attention networks similar to the ordering in [271]. These results are also consistent with prior studies which report components of the DMN as hubs [58, 247, 70]. DMN has been noted to be active primarily in studies of resting state activity [208] and is engaged by mind wandering [175], prospective and retrospective self-reflection [63], and memory retrieval [40], suggesting that the ‘default mode’ involves ongoing processing of

information for relevance to the self. In prior studies, DMN has been shown to have the highest global brain connectivity which may reflect connections necessary to implement the wide variety of cognitive functions the network is involved in. In conjunction with DMN, another large-scale network implementing a variety of cognitive functions, the cognitive control network (CCN), is also among the highest connectivity networks [58]. While our results indicate that the majority of the hub nodes are in DMN, this is followed by somatomotor, dorsal and ventral attention networks which are part of CCN.

The proposed graph filter learning framework provides additional interpretability to the importance of structure-function coupling in hub node identification. In particular, the frequency response of the optimal graph filter and the spectral energy distribution of the BOLD signal are shown to be closely related to the average hub score for a given subject. Thus, subjects whose BOLD signals have higher graph frequency content, i.e., reduced structure-function coupling, tend to have more hub node activity.

In addition to introducing a learning framework, the proposed approach also introduces a new hub scoring metric and hub node detection methods. Comparing the thresholding and top-K approaches for in Table 5.1, we can see that the top-K approach distributes the hub nodes more evenly across resting state networks. This is due to the fact that the same number of hub nodes, K, is selected across subjects, treating each subject equally, while with the thresholding method, one may detect more hubs for one subject vs. another detecting only the highest activity regions, such as DMN and neglecting other important networks.

5.7 Conclusions

In this chapter, we introduced a graph signal processing based framework for identifying the hub nodes in the brain. The proposed framework relies on the assumption that hub nodes are highly connected and have high activity levels with respect to their neighbors. From the perspective of GSP, this assumption results in modeling the hub nodes' activity as high-frequency with respect to the underlying graph, while the non-hub nodes have low-frequency or smooth activity. This model is implemented through an optimization problem that learns the optimal graph filter for

detecting hub nodes. The proposed framework, GraFHub, is applied to both simulated and real brain network data. It is shown that GraFHub performs better than existing connectivity-based hub node identification methods for both simulated and real brain networks as it takes the coupling between the graph topology and the graph signals defined on the graph. Moreover, the learned graph filters are low-pass and the filter response is highly correlated with the spectral energy density of the signals. Thus, learning the optimal filter provides interpretability to the spectrum of the underlying graph signal and can be used as a predictor for the number of hubs in a given brain network.

CHAPTER 6

CONCLUSIONS

In this thesis, methods for community detection and hub node identification problems in complex networks using graph-based learning techniques are presented. The contributions of this work span multiple aspects of network analysis, including community detection in multiplex networks, discriminative subgraph identification between different multiplex networks, and graph filtering for clustering attributed graphs and hub node identification.

In Chapter 2, we presented an algorithm for community detection in multiplex networks that identifies both common and private communities. We also proposed an algorithm for determining the number of communities, which is an input parameter in most community detection methods. The experiment results indicate that our method is superior to existing multiplex community detection methods as it does not enforce a consensus community structure. A proof of convergence is provided, along with an in-depth analysis of the algorithm, including studies of overfitting and ablation, recovery guarantees and consistency. The proposed algorithm is evaluated on synthetic and real multiplex networks, as well as for multiview clustering applications, and compared to state-of-the-art techniques. MX-ONMTF consistently outperformed established approaches across a range of synthetic scenarios with varying complexity, noise levels, and inter-layer dependencies. Additionally, when applied to real-world multiplex networks—including social networks and biological data—our method identified meaningful community structures aligned closely with known metadata. In addition, the application of MX-ONMTF to an fMRI dataset where the nodes are subjects and the layers represent different functional areas of the brain, reveal subgroups of subjects that exhibit significant differences in key functional areas, such as the default mode network (DMN) and anterior prefrontal cortex (antPFC), as well as in their corresponding clinical scores.

In Chapter 3, we introduced a spectral clustering-based discriminative community detection framework designed to identify communities that distinguish structural differences between distinct groups or conditions. Unlike traditional methods that focus on finding shared community structures, our framework explicitly focuses on capturing discriminative network substructures

across different multiplex networks. We presented three methods: the first, MX-DSC, identifies discriminative subspaces between two static multiplex networks; the second, MX-DCSC, extends this by simultaneously learning consensus, discriminative, and layer-specific subspaces; and the third, TMX-DiSG, further adapts this discriminative framework to temporal multiplex networks, finding discriminative communities between two groups across time. These methods are evaluated on synthetic networks involving extensive experiments under various conditions, including changes in the noise level, variability across layers and time, and the number of shared communities. Real-world applications to EEG and dynamic fMRI brain networks demonstrated that our framework effectively identifies task-specific discriminative subgraphs.

In Chapter 4, we proposed two graph signal processing-based methods for clustering attributed networks, GraFiCA and MSGWC. GraFiCA learns finite impulse response (FIR) and infinite impulse response (IIR) graph filters, whereas MSGWC optimally combines multi-scale features derived from graph wavelet transforms. Both approaches optimize filter parameters specifically for clustering by minimizing a novel loss function that simultaneously quantifies within-cluster and between-cluster dissimilarity of the filtered attributes. Thus, the filter parameters are optimized for the clustering task. Experiments on various real-world datasets—including EEG brain networks, citation networks, and social graphs—revealed that both methods significantly outperform state-of-the-art techniques, yielding more accurate and interpretable clusters and learning filters that adapt to the characteristics of the datasets.

In Chapter 5, we introduced a graph signal processing based framework for identifying the hub nodes in the human brain. The proposed framework relies on the assumption that hub nodes are highly connected and have high activity levels with respect to their neighbors. From the perspective of GSP, this assumption results in modeling the hub nodes' activity as high-frequency with respect to the underlying graph, while the non-hub nodes have low-frequency or smooth activity. This model is implemented through an optimization problem that learns the optimal graph filter for detecting hub nodes. The proposed framework, GraFHub, is applied to both simulated and real brain network data. It is shown that GraFHub performs better than existing connectivity-based

hub node identification methods for both simulated and real brain networks as it takes the coupling between the graph topology and the graph signals defined on the graph. Moreover, the learned graph filters are low-pass and the filter response is highly correlated with the spectral energy density of the signals. Thus, learning the optimal filter provides interpretability to the spectrum of the underlying graph signal and can be used as a predictor for the number of hubs in a given brain network.

6.1 Future Work

The work in this thesis suggests new research directions. In this section, we summarize some potential areas of future work for each chapter.

6.1.1 Multiplex Community Detection

In Chapter 2, we introduced a multiplex community detection method that identifies both common and layer-specific communities in multiplex networks. While our method effectively captures the heterogeneous structure of different network layers, an important future direction is extending this model to handle signed graphs, where edges can take positive or negative values. Signed graphs naturally arise in many real-world applications. In brain network analysis, for instance, functional connectivity graphs are often signed, representing both positive (correlated) and negative (anti-correlated) interactions between brain regions. Standard community detection methods often assume only positive links, which limits their applicability to neuroscientific datasets where functional interactions are inherently signed. Furthermore, signed multiplex networks extend beyond neuroscience and have applications in social network analysis, where friendships and antagonisms co-exist, as well as financial networks, where assets may exhibit both positive and negative correlations [150]. By developing a signed multiplex community detection framework, we can enhance our method's applicability across diverse domains. Thus, future work will focus on adapting the Multiplex Orthogonal Nonnegative Matrix Tri-Factorization framework to integrate signed adjacency matrices using one of the existing Semi-NMF variants [156, 76], ensuring that both positive and negative interactions are appropriately handled during the community detection process.

6.1.2 Discriminative Sugbraph Identification between Multiple Groups

In Chapter 3, we proposed a framework for finding discriminative subgraphs between two static and dynamic multiplex networks. Future work will focus on extending this approach to multiple datasets where the goal would be to differentiate between multiple groups of networks at the subgraph level. There are two main approaches for extending our framework: (1) identifying a discriminative subspace that uniquely characterizes each group in relation to the rest of the multiplex networks collectively, and (2) performing pairwise discrimination among multiple groups, systematically identifying subgraph features that differentiate between pair of groups.

Beyond identifying discriminative subgraphs, this framework can also be adapted into a supervised classification framework. Currently, our method is fully unsupervised, focusing on learning discriminative and consensus subspaces between and within two multiplex networks, respectively. A key future direction is to leverage the obtained discriminative subspaces to develop a classification model capable of predicting the group affiliation of new, unseen brain connectivity networks. By training on networks from distinct groups (e.g., healthy versus diseased individuals, or patients with different neurological conditions), a supervised learning extension could enable more precise classification. This adaptation holds significant potential for clinical applications, particularly in early detection of neurodegenerative diseases and cognitive impairments.

6.1.3 Graph Filtering for Clustering Attributed Graphs

In Chapter 4, we introduced two methods for clustering in attributed graphs, where the parameters of FIR and IIR graph filters, along with the optimal linear weights for combining multi-scale wavelet transforms, were learned by minimizing a cost function specifically designed for the clustering task. Future work will focus on developing a family of cost functions for clustering in attributed graphs. Our proposed cost function in Chapter 4, in its general form, has two components: (i) $\mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; \beta))$ that quantifies the quality of the partition based on the filtered attributes, $\tilde{\mathbf{F}} = \mathbf{U}\mathcal{H}(\boldsymbol{\Lambda})\mathbf{U}^\top\mathbf{F}$; (ii) $\mathcal{R}(\boldsymbol{C}, \mathbf{A}_n)$ that quantifies the alignment between the community structure $\boldsymbol{C}$ and the input connectivity matrix $\mathbf{A}_n$, thus explicitly taking the connectivity information into account. Moreover, $\mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; \beta))$ may be decomposed into two parts $\ell_{internal}(\boldsymbol{C}, \tilde{\mathbf{F}})$ which quan-

tifies the cohesiveness within communities and $\ell_{external}(\boldsymbol{C}, \tilde{\mathbf{F}})$ which quantifies the separability between communities, resulting in the following form

$$\underbrace{f(\ell_{internal}(\boldsymbol{C}, \tilde{\mathbf{F}}), \ell_{external}(\boldsymbol{C}, \tilde{\mathbf{F}}))}_{\mathcal{L}(\boldsymbol{C}, \mathcal{H}(\boldsymbol{\Lambda}; \beta))} + \alpha \mathcal{R}(\boldsymbol{C}, \mathbf{A}_n), \tag{6.1}$$

where f is a mapping that combines the internal and external cluster quality functions.

In the cost function proposed in Chapter 4, the quality of the clusters is determined by Euclidean distance of filtered attributes within and between clusters. As future work, we will consider different quality functions to quantify $\ell_{internal}$ and $\ell_{external}$ as well as different mappings, f, to combine them as well as different regularization functions, $\mathcal{R}(\boldsymbol{C}, \mathbf{A}_n)$ for incorporating the connectivity information.

For instance, two quality metrics we can consider for quantifying the "goodness" of a community structure are the sum of squared errors (SSE), commonly used as the cost function for k-means, and modularity. The regularization function, $\mathcal{R}(\boldsymbol{C}, \mathbf{A}_n)$ in (6.1) that quantifies the quality of the partition with respect to the observed connectivity matrix will be chosen to match the chosen quality function. For example, for k-means quality function, we will employ Euclidean distance between $\mathbf{A}_n$ and the community membership matrix. Similarly, for modularity-based clustering, the regularization function will be based on the spectral clustering of the modularity matrix, $\mathbf{B}$.

6.1.4 Graph Filtering for Hub Node Identification

In Chapter 5, we presented a graph signal processing based framework for identifying hub nodes in brain networks. Future work will consider several extensions of the proposed framework. First, we will consider the dynamic change in hub nodes across time. It is well-known that rs-fMRI is a dynamic process, thus the hub nodes may be changing across time similar to network connectivity states [10]. In our current framework, the graph signal $\mathbf{F} \in \mathbb{R}^{N \times P}$ was constructed using the BOLD signals, where N represents the number of nodes (brain regions) and P represents the number of time points. However, extending this approach to a dynamic setting requires a fundamental change in the construction of $\mathbf{F}$. Instead of a single $\mathbf{F}$ representing all time points, we need to construct a separate graph signal $\mathbf{F}^t$ for each time point $t = 1, 2 \ldots, P$. To achieve this, we can redefine our

graph signals using subjects as independent samples. This would result in a set of P graph signal matrices, $\mathbf{F}^t$ for $t = 1, 2 \ldots, P$, where each $\mathbf{F}^t \in \mathbb{R}^{N \times S}$ represents a graph signal for a given time point across S subjects.

Second, we will more closely study the relationship between hub nodes and the graph frequency spectrum. For example, the contribution of different hub nodes to the SED in different frequency bands can be quantified and used as predictors for hub node identification. Third, we will explore alternative hub scoring metrics to complement our current approach. A comparative analysis of these metrics could refine our identification strategy and improve the robustness of hub detection across different datasets and conditions.

BIBLIOGRAPHY

[1] Emmanuel Abbe. "Community detection and stochastic block models: recent developments". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6446–6531.

[2] Abubakar Abid et al. "Contrastive principal component analysis". In: *arXiv preprint arXiv:1709.06716* (2017).

[3] Sophie Achard and Ed Bullmore. "Efficiency and cost of economical brain functional networks". In: *PLoS computational biology* 3.2 (2007), e17.

[4] Tulay Adali, Matthew Anderson, and Geng-Shen Fu. "Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging". In: *IEEE Signal Process. Mag* 31.3 (2014), pp. 18–33.

[5] Tülay Adali, MABS Akhonda, and Vince D Calhoun. "ICA and IVA for data fusion: An overview and a new approach based on disjoint subspaces". In: *IEEE sensors letters* 3.1 (2018), pp. 1–4.

[6] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. "Link communities reveal multiscale complexity in networks". In: *nature* 466.7307 (2010), pp. 761–764.

[7] Alberto Aleta, Sandro Meloni, and Yamir Moreno. "A multilayer perspective for the analysis of urban transportation systems". In: *Scientific reports* 7.1 (2017), pp. 1–9.

[8] Hafiz Tiomoko Ali et al. "Latent heterogeneous multilayer community detection". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 8142–8146.

[9] Esmaeil Alinezhad et al. "Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches". In: *Neural Computing and Applications* 32.8 (2020), pp. 3203–3220.

[10] Elena A Allen et al. "Tracking whole-brain connectivity dynamics in the resting state". In: *Cerebral cortex* 24.3 (2014), pp. 663–676.

[11] Alessia Amelio and Clara Pizzuti. "Community detection in multidimensional networks". In: *2014 IEEE 26th International Conference on Tools with Artificial Intelligence*. IEEE. 2014, pp. 352–359.

[12] Aamir Anis, Akshay Gadde, and Antonio Ortega. "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies". In: *IEEE Transactions on Signal Processing* 64.14 (2016), pp. 3775–3789.

[13] Muhammet Balcilar et al. "Analyzing the expressive power of graph neural networks in a spectral perspective". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2021.

[14] Albert-László Barabási. "Network science". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987 (2013), p. 20120375.

[15] Danielle S Bassett and Edward T Bullmore. "Small-world brain networks revisited". In: *The Neuroscientist* 23.5 (2017), pp. 499–516.

[16] Marya Bazzi et al. "Generative benchmark models for mesoscale structure in multilayer networks". In: *arXiv preprint arXiv:1608.06196* (2016), p. 20.

[17] Christian F Beckmann and Stephen M Smith. "Tensorial extensions of independent component analysis for multisubject FMRI analysis". In: *Neuroimage* 25.1 (2005), pp. 294–311.

[18] Hamid Behjat and Martin Larsson. "Spectral characterization of functional MRI data on voxel-resolution cortical graphs". In: *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2020, pp. 558–562.

[19] Berrabah Bendoukha and Hafida Bendahmane. "Inequalities between the sum of powers and the exponential of sum of positive and commuting selfadjoint operators". In: *Archivum Mathematicum* 47.4 (2011), pp. 257–262.

[20] Yoav Benjamini and Daniel Yekutieli. "False discovery rate–adjusted multiple confidence intervals for selected parameters". In: *JASA* 100.469 (2005), pp. 71–81.

[21] Oualid Benyahia and Christine Largeron. "Centrality for graphs with numerical attributes". In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 2015, pp. 1348–1353.

[22] Dimitris Berberidis and Georgios B Giannakis. "Adaptive-similarity node embedding for scalable learning over graphs". In: *arXiv preprint arXiv:1811.10797* (2018).

[23] Michele Berlingerio, Michele Coscia, and Fosca Giannotti. "Finding and characterizing communities in multidimensional networks". In: *2011 International Conference on advances in social networks analysis and mining*. IEEE. 2011, pp. 490–494.

[24] Michele Berlingerio, Fabio Pinelli, and Francesco Calabrese. "Abacus: frequent pattern mining-based community discovery in multidimensional networks". In: *Data Mining and Knowledge Discovery* 27.3 (2013), pp. 294–320.

[25] Sharmodeep Bhattacharyya and Shirshendu Chatterjee. "Spectral clustering for multiple sparse networks: I". in: *arXiv preprint arXiv:1805.10594* (2018).

[26] Filippo Maria Bianchi et al. "Graph neural networks with convolutional arma filters". In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3496–3507.

[27] Bharat Biswal et al. "Functional connectivity in the motor cortex of resting human brain using echo-planar MRI". in: *Magnetic resonance in medicine* 34.4 (1995), pp. 537–541.

[28] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[29] Deyu Bo et al. "A survey on spectral graph neural networks". In: *arXiv preprint arXiv:2302.05631* (2023).

[30] Marcos Bolanos et al. “A weighted small world network measure for assessing functional connectivity”. In: *Journal of neuroscience methods* 212.1 (2013), pp. 133–142.

[31] Thomas AW Bolton et al. “Graph slepians to strike a balance between local and global network interactions: Application to functional brain imaging”. In: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE. 2018, pp. 1239–1243.

[32] Oualid Boutemine and Mohamed Bouguessa. “Mining community structures in multidimensional networks”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.4 (2017), pp. 1–36.

[33] S. Boyd et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: 3.1 (Jan. 2011), pp. 1–122.

[34] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[35] Abdelbasset Brahim and Nicolas Farrugia. “Graph Fourier transform of fMRI temporal signals based on an averaged structural connectome for the classification of neuroimaging”. In: *Artificial Intelligence in Medicine* 106 (2020), p. 101870.

[36] Guillaume Braun, Hemant Tyagi, and Christophe Biernacki. “Clustering multilayer graphs with missing nodes”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2260–2268.

[37] Maria Brbić and Ivica Kopriva. “Multi-view low-rank sparse subspace clustering”. In: *Pattern Recognition* 73 (2018), pp. 247–258.

[38] Randy L Buckner, Jessica R Andrews-Hanna, and Daniel L Schacter. “The brain’s default network: anatomy, function, and relevance to disease”. In: *Annals of the new York Academy of Sciences* 1124.1 (2008), pp. 1–38.

[39] Randy L Buckner et al. “Cortical hubs revealed by intrinsic functional connectivity: mapping, assessment of stability, and relation to Alzheimer’s disease”. In: *Journal of neuroscience* 29.6 (2009), pp. 1860–1873.

[40] Randy L Buckner et al. “Molecular, structural, and functional characterization of Alzheimer’s disease: evidence for a relationship between default activity, amyloid, and memory”. In: *Journal of neuroscience* 25.34 (2005), pp. 7709–7717.

[41] Ed Bullmore and Olaf Sporns. “Complex brain networks: graph theoretical analysis of structural and functional systems”. In: *Nature reviews neuroscience* 10.3 (2009), pp. 186–198.

[42] Hongyun Cai, Vincent W Zheng, and Kevin Chang. “A comprehensive survey of graph embedding: problems, techniques and applications”. In: (2018).

[43] Vince D Calhoun, Jingyu Liu, and Tülay Adalı. “A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data”. In: *Neuroimage* 45.1 (2009), S163–S172.

[44] James F Cavanagh and Michael J Frank. "Frontal theta as a mechanism for cognitive control". In: *Trends in cognitive sciences* 18.8 (2014), pp. 414–421.

[45] Ziwei Chai et al. "Can Abnormality be Detected by Graph Neural Networks?" In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI), Vienna, Austria*. 2022, pp. 23–29.

[46] Guoqing Chao, Shiliang Sun, and Jinbo Bi. "A survey on multiview clustering". In: *IEEE transactions on artificial intelligence* 2.2 (2021), pp. 146–168.

[47] Kamalika Chaudhuri et al. "Multi-view clustering via canonical correlation analysis". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 129–136.

[48] Beth L Chen, David H Hall, and Dmitri B Chklovskii. "Wiring optimization can relate neuronal structure and function". In: *Proceedings of the National Academy of Sciences* 103.12 (2006), pp. 4723–4728.

[49] Chuan Chen, Michael K Ng, and Shuqin Zhang. "Block spectral clustering methods for multiple graphs". In: *Numerical Linear Algebra with Applications* 24.1 (2017), e2075.

[50] Jia Chen, Gang Wang, and Georgios B Giannakis. "Nonlinear dimensionality reduction for discriminative analytics of multiple datasets". In: *IEEE Transactions on Signal Processing* 67.3 (2018), pp. 740–752.

[51] Mingming Chen, Konstantin Kuzmin, and Boleslaw K Szymanski. "Community detection via maximization of modularity and its variants". In: *IEEE Transactions on Computational Social Systems* 1.1 (2014), pp. 46–65.

[52] Pin-Yu Chen and Alfred O Hero. "Multilayer spectral graph clustering via convex layer aggregation: Theory and algorithms". In: *IEEE Transactions on Signal and Information Processing over Networks* 3.3 (2017), pp. 553–567.

[53] Siheng Chen et al. "Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring". In: *IEEE Transactions on Signal Processing* 62.11 (2014), pp. 2879–2893.

[54] Siheng Chen et al. "Signal denoising on graphs via graph filtering". In: *2014 ieee global conference on signal and information processing (globalsip)*. IEEE. 2014, pp. 872–876.

[55] Siheng Chen et al. "Signal recovery on graphs: Variation minimization". In: *IEEE Transactions on Signal Processing* 63.17 (2015), pp. 4609–4624.

[56] Fan Chung and Mary Radcliffe. "On the spectra of general random graphs". In: *the electronic journal of combinatorics* (2011), P215–P215.

[57] FRK Chung. "Spectral Graph Theory, Providence, RI: Amer". In: *Math. Soc* (1997).

[58] Michael W Cole, Sudhir Pathak, and Walter Schneider. "Identifying the brain's most globally connected regions". In: *Neuroimage* 49.4 (2010), pp. 3132–3148.

[59] Andrej Čopar, Blaž Zupan, and Marinka Zitnik. "Fast optimization of non-negative matrix

tri-factorization". In: *PloS one* 14.6 (2019), e0217994.

[60] Michael Costanzo et al. "The genetic landscape of a cell". In: *science* 327.5964 (2010), pp. 425–431.

[61] Mario Coutino et al. "Fast spectral approximation of structured graphs with applications to graph filtering". In: *Algorithms* 13.9 (2020), p. 214.

[62] Emanuele Cozzo et al. "Structure of triadic relations in multiplex networks". In: *New Journal of Physics* 17.7 (2015), p. 073029.

[63] Arnaud D'Argembeau et al. "Self-reflection across time: cortical midline structures differentiate between present and past selves". In: *Social cognitive and affective neuroscience* 3.3 (2008), pp. 244–252.

[64] Zhengjia Dai et al. "Identifying and mapping connectivity patterns of brain network hubs in Alzheimer's disease". In: *Cerebral cortex* 25.10 (2015), pp. 3723–3742.

[65] Leon Danon et al. "Comparing community structure identification". In: *Journal of statistical mechanics: Theory and experiment* 2005.09 (2005), P09008.

[66] Caterina De Bacco et al. "Community detection, link prediction, and layer interdependence in multilayer networks". In: *Physical Review E* 95.4 (2017), p. 042317.

[67] Manlio De Domenico. "Multilayer modeling and analysis of human brain networks". In: *Giga Science* 6.5 (2017), gix004.

[68] Manlio De Domenico, Mason A Porter, and Alex Arenas. "MuxViz: a tool for multilayer analysis and visualization of networks". In: *Journal of Complex Networks* 3.2 (2015), pp. 159–176.

[69] Manlio De Domenico et al. "Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems". In: *Physical Review X* 5.1 (2015), p. 011027.

[70] Francesco De Pasquale et al. "The connectivity of functional cores reveals different degrees of segregation and integration in the brain at rest". In: *Neuroimage* 69 (2013), pp. 51–61.

[71] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering". In: *Advances in neural information processing systems* 29 (2016).

[72] Shay Deutsch, Antonio Ortega, and Gérard Medioni. "Robust denoising of piece-wise smooth manifolds". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 2786–2790.

[73] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. *A unified view of kernel k-means, spectral clustering and graph cuts*. Citeseer, 2004.

[74] Chris Ding, Xiaofeng He, and Horst D Simon. "On the equivalence of nonnegative matrix factorization and spectral clustering". In: *Proceedings of the 2005 SIAM international conference on data mining*. 2005, pp. 606–610.

[75] Chris Ding et al. “Orthogonal nonnegative matrix t-factorizations for clustering”. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, pp. 126–135.

[76] Chris HQ Ding, Tao Li, and Michael I Jordan. “Convex and semi-nonnegative matrix factorizations”. In: *IEEE transactions on pattern analysis and machine intelligence* 32.1 (2008), pp. 45–55.

[77] Xiaowen Dong et al. “Clustering on multi-layer graphs via subspace analysis on Grassmann manifolds”. In: *IEEE Transactions on signal processing* 62.4 (2013), pp. 905–918.

[78] Xiaowen Dong et al. “Graph signal processing for machine learning: A review and new perspectives”. In: *IEEE Signal processing magazine* 37.6 (2020), pp. 117–127.

[79] Claire Donnat et al. “Learning Structural Node Embeddings via Diffusion Wavelets”. In: 2018.

[80] Elisabeth Drayer and Tirza Routtenberg. “Detection of false data injection attacks in smart grids based on graph signal processing”. In: *IEEE Systems Journal* 14.2 (2019), pp. 1886–1896.

[81] Liang Du et al. “K-Means Clustering Based on Chebyshev Polynomial Graph Filtering”. In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2024, pp. 7175–7179.

[82] Yuhui Du et al. “Evidence of shared and distinct functional and structural brain signatures in schizophrenia and autism spectrum disorder”. In: *Communications biology* 4.1 (2021), pp. 1–16.

[83] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[84] Hilmi E Egilmez and Antonio Ortega. “Spectral anomaly detection using graph-based filtering for wireless sensor networks”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 1085–1089.

[85] Hilmi E Egilmez, Eduardo Pavez, and Antonio Ortega. “Graph learning from filtered signals: Graph system and diffusion kernel identification”. In: *IEEE Transactions on Signal and Information Processing over Networks* 5.2 (2018), pp. 360–374.

[86] Justine Eustace, Xingyuan Wang, and Yaozu Cui. “Overlapping community detection using neighborhood ratio matrix”. In: *Physica A: Statistical Mechanics and its Applications* 421 (2015), pp. 510–521.

[87] Martin G Everett and Stephen P Borgatti. “Categorical attribute based centrality: E–I and G–F centrality”. In: *Social Networks* 34.4 (2012), pp. 562–569.

[88] Jie Fan, Cihan Tepedelenlioglu, and Andreas Spanias. “Graph-based classification with multiple shift matrices”. In: *IEEE Transactions on Signal and Information Processing over Networks* 8 (2022), pp. 160–172.

[89] Xing Fan et al. “ALMA: alternating minimization algorithm for clustering mixture multi-

layer network". In: *The Journal of Machine Learning Research* 23.1 (2022), pp. 14855–14900.

[90] Alex Fornito, Andrew Zalesky, and Michael Breakspear. "The connectomics of brain disorders". In: *Nature Reviews Neuroscience* 16.3 (2015), pp. 159–172.

[91] Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3-5 (2010), pp. 75–174.

[92] Santo Fortunato and Darko Hric. "Community detection in networks: A user guide". In: *Physics reports* 659 (2016), pp. 1–44.

[93] Panagiotis Fotiadis et al. "Structure–function coupling in macroscale human brain networks". In: *Nature Reviews Neuroscience* (2024), pp. 1–17.

[94] Rodrigo Francisquini, Ana Carolina Lorena, and Mariá CV Nascimento. "Community-based anomaly detection using spectral graph filtering". In: *Applied Soft Computing* 118 (2022), p. 108489.

[95] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

[96] Yuan Gao et al. "Addressing heterophily in graph anomaly detection: A perspective of graph spectrum". In: *Proceedings of the ACM Web Conference 2023*. 2023, pp. 1528–1538.

[97] K Georgiadis et al. "Connectivity steered graph Fourier transform for motor imagery BCI decoding". In: *Journal of neural engineering* 16.5 (2019), p. 056021.

[98] Amir Ghasemian, Homa Hosseinmardi, and Aaron Clauset. "Evaluating overfit and underfit in models of network community structure". In: *IEEE Transactions on Knowledge and Data Engineering* 32.9 (2019), pp. 1722–1735.

[99] Michelle Girvan and Mark EJ Newman. "Community structure in social and biological networks". In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826.

[100] Matthew F Glasser et al. "The minimal preprocessing pipelines for the Human Connectome Project". In: *Neuroimage* 80 (2013), pp. 105–124.

[101] Vladimir Gligorijević, Yannis Panagakis, and Stefanos Zafeiriou. "Non-negative matrix factorizations for multiplex network analysis". In: *IEEE transactions on pattern analysis and machine intelligence* 41.4 (2018), pp. 928–940.

[102] Katharina Glomb et al. "Connectome spectral analysis to track EEG task dynamics on a subsecond scale". In: *NeuroImage* 221 (2020), p. 117137.

[103] Evan M Gordon et al. "Precision functional mapping of individual human brains". In: *Neuron* 95.4 (2017), pp. 791–807.

[104] Ludovica Griffanti et al. "ICA-based artefact removal and accelerated fMRI acquisition for improved resting state network imaging". In: *Neuroimage* 95 (2014), pp. 232–247.

[105] Adrian R Groves et al. "Linked independent component analysis for multimodal data

fusion". In: *Neuroimage* 54.3 (2011), pp. 2198–2217.

[106] Roger Guimera and Luís A Nunes Amaral. "Cartography of complex networks: modules and universal roles". In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.02 (2005), P02001.

[107] W de Haan et al. "Activity Dependent Degeneration Explains Hub Vulnerability in Alzheimer's Disease". In: *PLoS computational biology* 8.8 (2012), e1002582.

[108] Jason R Hall, Edward M Bernat, and Christopher J Patrick. "Externalizing psychopathology and the error-related negativity". In: *Psychological science* 18.4 (2007), pp. 326–333.

[109] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. "Wavelets on graphs via spectral graph theory". In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[110] Qiuyi Han, Kevin S Xu, and Edoardo M Airoldi. "Consistent estimation of dynamic and multi-layer networks". In: *arXiv preprint arXiv:1410.8597* (2014).

[111] Chaobo He et al. "A survey of community detection in complex networks using nonnegative matrix factorization". In: *IEEE Transactions on Computational Social Systems* 9.2 (2021), pp. 440–457.

[112] Mingguo He, Zhewei Wei, Hongteng Xu, et al. "Bernnet: Learning arbitrary graph spectral filters via bernstein approximation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 14239–14251.

[113] Yiran He and Hoi-To Wai. "Detecting central nodes from low-rank excited graph signals via structured factor analysis". In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 2416–2430.

[114] Yiran He and Hoi-To Wai. "Estimating centrality blindly from low-pass filtered graph signals". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 5330–5334.

[115] Yong He et al. "Uncovering intrinsic modular organization of spontaneous brain activity in humans". In: *PloS one* 4.4 (2009), e5226.

[116] Weiyu Huang et al. "A graph signal processing perspective on functional brain imaging". In: *Proceedings of the IEEE* 106.5 (2018), pp. 868–885.

[117] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pp. 193–218.

[118] Muhammad U Ilyas et al. "A distributed algorithm for identifying information hubs in social networks". In: *IEEE Journal on Selected Areas in Communications* 31.9 (2013), pp. 629–640.

[119] Elvin Isufi et al. "Autoregressive moving average graph filtering". In: *IEEE Transactions on Signal Processing* 65.2 (2016), pp. 274–288.

[120] Elvin Isufi et al. "Graph filters for signal processing and machine learning on graphs". In:

arXiv preprint arXiv:2211.08854 (2022).

[121] Elvin Isufi et al. "Graph filters for signal processing and machine learning on graphs". In: *IEEE Transactions on Signal Processing* (2024).

[122] Assen Jablensky. "The diagnostic concept of schizophrenia: its history, evolution, and future prospects". In: *Dialogues Clin. Neurosci.* 12.3 (2010), p. 271.

[123] Anil K Jain. "Data clustering: 50 years beyond K-means". In: *Pattern recognition letters* 31.8 (2010), pp. 651–666.

[124] LGS Jeub and M Bazzi. *A generative model for mesoscale structure in multilayer networks implemented in MATLAB*. 2016. URL: https://github.com/MultilayerGM/MultilayerGM-MATLAB.

[125] Caiyan Jia et al. "Node attribute-enhanced community detection in complex networks". In: *Scientific reports* 7.1 (2017), pp. 1–15.

[126] Zhuqing Jiao et al. "Hub recognition for brain functional networks by using multiple-feature combination". In: *Computers & Electrical Engineering* 69 (2018), pp. 740–752.

[127] Rui Jin et al. "Dictionary learning-based fMRI data analysis for capturing common and individual neural activation maps". In: *IEEE Journal of Selected Topics in Signal Processing* 14.6 (2020), pp. 1265–1279.

[128] Bing-Yi Jing et al. "Community detection on mixture multilayer networks via regularized tensor decomposition". In: *The Annals of Statistics* 49.6 (2021), pp. 3181–3205.

[129] Karen E Joyce et al. "A new measure of centrality for brain networks". In: *PloS one* 5.8 (2010), e12200.

[130] Inderjit S Jutla, Lucas GS Jeub, Peter J Mucha, et al. "A generalized Louvain method for community detection implemented in MATLAB". in: (). URL: https://github.com/GenLouvain/GenLouvain%20(2011-2019).

[131] Marcus Kaiser and Claus C Hilgetag. "Edge vulnerability in neural and metabolic networks". In: *Biological cybernetics* 90.5 (2004), pp. 311–317.

[132] Golnar Kalantar and Arash Mohammadi. "Graph-based dimensionality reduction of EEG signals via functional clustering and total variation measure for BCI systems". In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 4603–4606.

[133] Zhao Kang et al. "Fine-grained attributed graph clustering". In: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM. 2022, pp. 370–378.

[134] Esin Karahan et al. "Tensor analysis and fusion of multimodal brain images". In: *Proceedings of the IEEE* 103.9 (2015), pp. 1531–1559.

[135] Fatemeh Karimi, Shahriar Lotfi, and Habib Izadkhah. "Multiplex community detection in complex networks using an evolutionary approach". In: *Expert Systems with Applications* 146 (2020), p. 113184.

[136] Brian Karrer and Mark EJ Newman. "Stochastic blockmodels and community structure in networks". In: *Physical review E* 83.1 (2011), p. 016107.

[137] Stanley R Kay, Abraham Fiszbein, and Lewis A Opler. "The positive and negative syndrome scale (PANSS) for schizophrenia". In: *Schizophr. Bull.* 13.2 (1987), pp. 261–276.

[138] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: Toulon, France, Apr. 2017.

[139] Thomas N Kipf and Max Welling. "Variational Graph Auto-Encoders". In: *NIPS Workshop on Bayesian Deep Learning* (2016).

[140] Mikko Kivelä et al. "Multilayer networks". In: *Journal of complex networks* 2.3 (2014), pp. 203–271.

[141] R. I. Kondor and J. Lafferty. "Diffusion kernels on graphs and other discrete structures". In: Sydney, Australia, July 2002, pp. 315–322.

[142] Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein. "MADMM: a generic algorithm for non-smooth optimization on manifolds". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer. 2016, pp. 680–696.

[143] Ariel Kroizer, Tirza Routtenberg, and Yonina C Eldar. "Bayesian estimation of graph signals". In: *IEEE transactions on signal processing* 70 (2022), pp. 2207–2223.

[144] Abhishek Kumar, Piyush Rai, and Hal Daume. "Co-regularized multi-view spectral clustering". In: *Advances in neural information processing systems* 24 (2011).

[145] Zhana Kuncheva and Giovanni Montana. "Community detection in multiplex networks using locally adaptive random walks". In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 2015, pp. 1308–1315.

[146] Vito Latora and Massimo Marchiori. "Economic small-world behavior in weighted networks". In: *The European Physical Journal B-Condensed Matter and Complex Systems* 32 (2003), pp. 249–263.

[147] Emmanuel Lazega et al. *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*. Oxford University Press on Demand, 2001.

[148] Daniel Lee and Hyunjune Seung. "Algorithms for Non-negative Matrix Factorization". In: *Adv. Neural Inform. Process. Syst.* 13 (2001), pp. 535–541.

[149] Jing Lei, Kehui Chen, and Brian Lynch. "Consistent community detection in multi-layer network data". In: *Biometrika* 107.1 (2020), pp. 61–73.

[150] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. "Predicting positive and negative links in online social networks". In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 641–650.

[151] Jure Leskovec and Julian Mcauley. "Learning to discover social circles in ego networks".

In: *Advances in neural information processing systems* 25 (2012).

[152] Qimai Li et al. "Label efficient semi-supervised learning via graph filtering". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9582–9591.

[153] Tao Li and Cha-charis Ding. "Nonnegative matrix factorizations for clustering: A survey". In: *Data Clustering*. Chapman and Hall/CRC, 2018, pp. 149–176.

[154] Yang Li and Gonzalo Mateos. "Identifying structural brain networks from functional connectivity: A network deconvolution approach". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 1135–1139.

[155] Yixuan Li et al. "Local spectral clustering for overlapping community detection". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12.2 (2018), pp. 1–27.

[156] Zhen Li et al. "Community detection based on regularized semi-nonnegative matrix trifactorization in signed networks". In: *Mobile Networks and Applications* 23 (2018), pp. 71–79.

[157] Zhenping Li et al. "Quantitative function for community detection". In: *Physical review E* 77.3 (2008), p. 036109.

[158] Li F. F., Andreeto M., Ranzato M., and Perona P. *Caltech 101 (1.0) [Data set]. CaltechDATA.*. 2022. URL: https://doi.org/10.22002/D1.20086.

[159] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest". In: *2008 eighth ieee international conference on data mining*. IEEE. 2008, pp. 413–422.

[160] Fuchen Liu et al. "Global spectral clustering in dynamic networks". In: *Proceedings of the National Academy of Sciences* 115.5 (2018), pp. 927–932.

[161] Jialu Liu et al. "Multi-view clustering via joint nonnegative matrix factorization". In: *Proceedings of the 2013 SIAM international conference on data mining*. SIAM. 2013, pp. 252–260.

[162] Jiani Liu, Elvin Isufi, and Geert Leus. "Filter design for autoregressive moving average graph filters". In: *IEEE Transactions on Signal and Information Processing over Networks* 5.1 (2018), pp. 47–60.

[163] Jin Liu et al. "Intrinsic brain hub connectivity underlies individual differences in spatial working memory". In: *Cerebral cortex* 27.12 (2017), pp. 5496–5508.

[164] Jin Xia Liu et al. "Quantitative function for community detection". In: *Advanced Materials Research* 433 (2012), pp. 6441–6446.

[165] Gabriele Lohmann et al. "Eigenvector centrality mapping for analyzing connectivity patterns in fMRI data of the human brain". In: *PloS one* 5.4 (2010), e10232.

[166] Dan-Dan Lu et al. "Community detection combining topology and attribute information". In: *Knowledge and Information Systems* (2022), pp. 1–22.

[167] Hong Lu et al. “Community detection algorithm based on nonnegative matrix factorization and pairwise constraints”. In: *Physica A: Statistical Mechanics and its Applications* 545 (2020), p. 123491.

[168] Xin Luo et al. “Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.3 (2021), pp. 1203–1215.

[169] Xiaoke Ma, Di Dong, and Quan Wang. “Community detection in multi-layer networks using joint nonnegative matrix factorization”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.2 (2018), pp. 273–286.

[170] Matteo Magnani et al. “Community detection in multiplex networks”. In: *ACM Computing Surveys (CSUR)* 54.3 (2021), pp. 1–35.

[171] Deepanshu Malhotra and Anuradha Chug. “A modified label propagation algorithm for community detection in attributed networks”. In: *International Journal of Information Management Data Insights* 1.2 (2021), p. 100030.

[172] Shawn Mankad and George Michailidis. “Structural and functional discovery in dynamic networks with non-negative matrix factorization”. In: *Physical Review E* 88.4 (2013), p. 042812.

[173] Antonio G Marques et al. “Sampling of graph signals with successive local aggregations”. In: *IEEE Transactions on Signal Processing* 64.7 (2015), pp. 1832–1843.

[174] Sohir Maskey et al. “A fractional graph laplacian approach to oversmoothing”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[175] Malia F Mason et al. “Wandering minds: the default network and stimulus-independent thought”. In: *science* 315.5810 (2007), pp. 393–395.

[176] Gráinne McLoughlin et al. “Midfrontal theta activity in psychiatric illness: an index of cognitive vulnerabilities across disorders”. In: *Biological Psychiatry* 91.2 (2022), pp. 173–182.

[177] John D Medaglia et al. “Functional alignment with anatomical networks is associated with cognitive flexibility”. In: *Nature human behaviour* 2.2 (2018), pp. 156–164.

[178] Marek-Marsel Mesulam and Norman Geschwind. “On the possible role of neocortex and its limbic connections in the process of attention and schizophrenia: clinical cases of inattention in man and experimental anatomy in monkey.” In: *Journal of psychiatric research* (1978).

[179] Gianluca Mingoia et al. “Default mode network activity in schizophrenia studied at resting state using probabilistic ICA”. in: *Schizophrenia research* 138.2-3 (2012), pp. 143–149.

[180] Sepehr Mortaheb et al. “A graph signal processing approach to study high density EEG signals in patients with disorders of consciousness”. In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2019, pp. 4549–4553.

[181] Malaak N Moussa et al. “Consistency of network modules in resting-state FMRI connectome data”. In: (2012).

[182] Peter J Mucha et al. “Community structure in time-dependent, multiscale, and multiplex networks”. In: *science* 328.5980 (2010), pp. 876–878.

[183] Rena Nainggolan et al. “Improved the performance of the K-means cluster using the sum of squared error (SSE) optimized by using the Elbow method”. In: *Journal of Physics: Conference Series*. Vol. 1361. 1. IOP Publishing. 2019, p. 012015.

[184] Sunil K Narang, Akshay Gadde, and Antonio Ortega. “Signal processing techniques for interpolation in graph structured data”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 5445–5449.

[185] Sunil K Narang and Antonio Ortega. “Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs”. In: *IEEE transactions on signal processing* 61.19 (2013), pp. 4673–4685.

[186] Sunil K Narang and Antonio Ortega. “Perfect reconstruction two-channel wavelet filter banks for graph structured data”. In: *IEEE Transactions on Signal Processing* 60.6 (2012), pp. 2786–2799.

[187] Hung T Nguyen, Thang N Dinh, and Tam Vu. “Community detection in multiplex social networks”. In: *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2015, pp. 654–659.

[188] Emil HJ Nijhuis, Anne-Marie van Cappellen van Walsum, and David G Norris. “Topographic hub maps of the human structural neocortical network”. In: *PloS one* 8.6 (2013), e65511.

[189] Jukka-Pekka Onnela et al. “Intensity and coherence of motifs in weighted complex networks”. In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 71.6 (2005), p. 065103.

[190] Masaki Onuki et al. “Graph signal denoising via trilateral filter on graph spectral domain”. In: *IEEE Transactions on Signal and Information Processing over Networks* 2.2 (2016), pp. 137–148.

[191] Alp Ozdemir et al. “Hierarchical spectral consensus clustering for group analysis of functional brain networks”. In: *IEEE Transactions on Biomedical Engineering* 62.9 (2015), pp. 2158–2169.

[192] Camillo Padoa-Schioppa and John A Assad. “Neurons in the orbitofrontal cortex encode economic value”. In: *Nature* 441.7090 (2006), pp. 223–226.

[193] A Roxana Pamfil et al. “Relating modularity maximization and stochastic block models in multilayer networks”. In: *SIAM Journal on Mathematics of Data Science* 1.4 (2019), pp. 667–698.

[194] Shirui Pan et al. “Adversarially Regularized Graph Autoencoder for Graph Embedding.” In: *IJCAI*. 2018, pp. 2609–2615.

[195] Neal Parikh, Stephen Boyd, et al. “Proximal algorithms”. In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239.

[196] Sohee Park and Philip S Holzman. “Schizophrenics show spatial working memory deficits”. In: *Arch. Gen. Psychiatry* 49.12 (1992), pp. 975–982.

[197] Subhadeep Paul and Yuguo Chen. “Spectral and matrix factorization methods for consistent community detection in multi-layer networks”. In: *The Annals of Statistics* 48.1 (2020), pp. 230–250.

[198] Martin P Paulus et al. “Parietal dysfunction is associated with increased outcome-related decision-making in schizophrenia patients”. In: *Biological Psychiatry* 51.12 (2002), pp. 995–1004.

[199] Mangor Pedersen et al. “Reducing the influence of intramodular connectivity in participation coefficient”. In: *Network Neuroscience* 4.2 (2020), pp. 416–431.

[200] JB Pochon et al. “The neural system that bridges reward and cognition in humans: an fMRI study”. In: *Proc. Natl. Acad. Sci.* 99.8 (2002), pp. 5669–5674.

[201] Filippo Pompili et al. “Two algorithms for orthogonal nonnegative matrix factorization with application to clustering”. In: *Neurocomputing* 141 (2014), pp. 15–25.

[202] Jonathan D Power et al. “Evidence for hubs in human functional brain networks”. In: *Neuron* 79.4 (2013), pp. 798–813.

[203] Soumajit Pramanik et al. “Discovering community structure in multilayer networks”. In: *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2017, pp. 611–620.

[204] Maria Giulia Preti, Thomas AW Bolton, and Dimitri Van De Ville. “The dynamic functional connectome: State-of-the-art and perspectives”. In: *Neuroimage* 160 (2017), pp. 41–54.

[205] Maria Giulia Preti and Dimitri Van De Ville. “Decoupling of brain function from structure reveals regional behavioral specialization in humans”. In: *Nature communications* 10.1 (2019), p. 4747.

[206] Ioannis Psorakis et al. “Overlapping community detection using Bayesian non-negative matrix factorization”. In: *Physical Review E* 83.6 (2011), p. 066114.

[207] Guo-Jun Qi et al. “Exploring context and content links in social media: A latent space method”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (2011), pp. 850–862.

[208] Marcus E Raichle et al. “A default mode of brain function”. In: *Proceedings of the national academy of sciences* 98.2 (2001), pp. 676–682.

[209] Raksha Ramakrishna, Hoi-To Wai, and Anna Scaglione. “A user guide to low-pass graph signal processing and its applications: Tools and applications”. In: *IEEE Signal Processing Magazine* 37.6 (2020), pp. 74–85.

[210] David Ramírez, Antonio G Marques, and Santiago Segarra. “Graph-signal reconstruc-

tion and blind deconvolution for structured inputs". In: *Signal Processing* 188 (2021), p. 108180.

[211] Emma C Robinson et al. "MSM: a new flexible framework for multimodal surface matching". In: *Neuroimage* 100 (2014), pp. 414–426.

[212] T Mitchell Roddenberry and Santiago Segarra. "Blind inference of eigenvector centrality rankings". In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 3935–3946.

[213] Martin Rosvall and Carl T Bergstrom. "Maps of random walks on complex networks reveal community structure". In: *Proceedings of the national academy of sciences* 105.4 (2008), pp. 1118–1123.

[214] Tirza Routtenberg. "Non-Bayesian estimation framework for signal recovery on graphs". In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 1169–1184.

[215] Mikail Rubinov and Olaf Sporns. "Complex network measures of brain connectivity: uses and interpretations". In: *Neuroimage* 52.3 (2010), pp. 1059–1069.

[216] Mikail Rubinov and Olaf Sporns. "Weight-conserving characterization of complex functional brain networks". In: *Neuroimage* 56.4 (2011), pp. 2068–2079.

[217] Liu Rui, Hossein Nejati, and Ngai-Man Cheung. "Dimensionality reduction of brain imaging data using graph signal processing". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 1329–1333.

[218] Pilar Salgado-Pineda et al. "Sustained attention impairment correlates to gray matter decreases in first episode neuroleptic-naive schizophrenic patients". In: *Neuroimage* 19.2 (2003), pp. 365–375.

[219] Aliaksei Sandryhaila and Jose MF Moura. "Classification via regularization on graphs". In: *2013 IEEE global conference on signal and information processing*. IEEE. 2013, pp. 495–498.

[220] Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro. "Optimal graph-filter design and applications to distributed linear network operators". In: *IEEE Transactions on Signal Processing* 65.15 (2017), pp. 4117–4131.

[221] Santiago Segarra et al. "Network topology inference from spectral templates". In: *IEEE Transactions on Signal and Information Processing over Networks* 3.3 (2017), pp. 467–483.

[222] Benjamin A Seitzman et al. "The state of resting state networks". In: *Topics in Magnetic Resonance Imaging* 28.4 (2019), pp. 189–196.

[223] Prithviraj Sen et al. "Collective classification in network data". In: *AI magazine* 29.3 (2008), p. 93.

[224] D. I. Shuman et al. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In: 30.3 (May 2013), pp. 83–98. ISSN: 1053-5888.

[225] David I Shuman et al. "Distributed signal processing via Chebyshev polynomial approxi-

mation". In: *IEEE Transactions on Signal and Information Processing over Networks* 4.4 (2018), pp. 736–751.

[226] Saurabh Sihag et al. "Multimodal dynamic brain connectivity analysis based on graph signal processing for former athletes with history of multiple concussions". In: *IEEE Transactions on Signal and Information Processing over Networks* 6 (2020), pp. 284–299.

[227] Rahul Singh, Abhishek Chakraborty, and BS Manoj. "GFT centrality: A new node importance measure for complex networks". In: *Physica A: Statistical Mechanics and its Applications* 487 (2017), pp. 185–195.

[228] Keith Smith et al. "Locating temporal functional dynamics of visual short-term memory binding using graph modular dirichlet energy". In: *Scientific reports* 7.1 (2017), p. 42013.

[229] Sandra E Smith-Aguilar et al. "Using multiplex networks to capture the multidimensional nature of social structure". In: *Primates* 60.3 (2019), pp. 277–295.

[230] Olaf Sporns and Richard F Betzel. "Modular brain networks". In: *Annual review of psychology* 67.1 (2016), pp. 613–640.

[231] Olaf Sporns, Christopher J Honey, and Rolf Kötter. "Identification and classification of hubs in brain networks". In: *PloS one* 2.10 (2007), e1049.

[232] R Nathan Spreng, Raymond A Mar, and Alice SN Kim. "The common neural basis of autobiographical memory, prospection, navigation, theory of mind, and the default mode: a quantitative meta-analysis". In: *Journal of cognitive neuroscience* 21.3 (2009), pp. 489–510.

[233] Natalie Stanley et al. "Clustering network layers with the strata multilayer stochastic block model". In: *IEEE transactions on network science and engineering* 3.2 (2016), pp. 95–105.

[234] Matthew Steen et al. "Assessing the consistency of community structure in complex networks". In: *Physical Review E* 84.1 (2011), p. 016111.

[235] Gregory P Strauss, James A Waltz, and James M Gold. "A review of reward processing and motivational impairment in schizophrenia". In: *Schizophrenia bulletin* 40.Suppl_2 (2014), S107–S116.

[236] Jing Sui et al. "An ICA-based method for the identification of optimal FMRI features and components using combined group-discriminative techniques". In: *Neuroimage* 46.1 (2009), pp. 73–86.

[237] Bing-Jie Sun et al. "A non-negative symmetric encoder-decoder approach for community detection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 597–606.

[238] Carol A Tamminga et al. "Bipolar and schizophrenia network for intermediate phenotypes: outcomes across the psychosis continuum". In: *Schizophr. Bull.* 40.Suppl_2 (2014), S131–S137.

[239] Carol A Tamminga et al. "Clinical phenotypes of psychosis in the Bipolar-Schizophrenia Network on Intermediate Phenotypes (B-SNIP)". in: *Am. J. Psychiatry* 170.11 (2013),

pp. 1263–1274.

[240] Yuichi Tanaka et al. “Sampling signals on graphs: From theory to applications”. In: *IEEE Signal Processing Magazine* 37.6 (2020), pp. 14–30.

[241] Fengqin Tang, Xuejing Zhao, and Cuixia Li. “Community Detection in Multilayer Networks Based on Matrix Factorization and Spectral Embedding Method”. In: *Mathematics* 11.7 (2023), p. 1573.

[242] Jianheng Tang et al. “Rethinking graph neural networks for anomaly detection”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 21076–21089.

[243] Lei Tang, Xufei Wang, and Huan Liu. “Community detection via heterogeneous interaction analysis”. In: *Data mining and knowledge discovery* 25.1 (2012), pp. 1–33.

[244] Wei Tang, Zhengdong Lu, and Inderjit S Dhillon. “Clustering with multiple graphs”. In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE. 2009, pp. 1016–1021.

[245] Dane Taylor, Rajmonda S Caceres, and Peter J Mucha. “Super-resolution community detection for layer-aggregated multilayer networks”. In: *Physical Review X* 7.3 (2017), p. 031056.

[246] Craig E Tenke and Jürgen Kayser. “Generator localization by current source density (CSD): implications of volume conduction and field closure at intracranial and scalp resolutions”. In: *Clinical neurophysiology* 123.12 (2012), pp. 2328–2345.

[247] Dardo Tomasi and Nora D Volkow. “Association between functional connectivity hubs and brain networks”. In: *Cerebral cortex* 21.9 (2011), pp. 2003–2013.

[248] Nicolas Tremblay and Pierre Borgnat. “Graph wavelets for multiscale community mining”. In: *IEEE Transactions on Signal Processing* 62.20 (2014), pp. 5227–5239.

[249] Nicolas Tremblay et al. “Compressive spectral clustering”. In: *International conference on machine learning*. PMLR. 2016, pp. 1002–1011.

[250] Logan T Trujillo and John JB Allen. “Theta EEG dynamics of the error-related negativity”. In: *Clinical Neurophysiology* 118.3 (2007), pp. 645–668.

[251] Leslie G Ungerleider and James V Haxby. “‘What’and ‘where’in the human brain”. In: *Current opinion in neurobiology* 4.2 (1994), pp. 157–165.

[252] Ravishankar R Vallabhajosyula et al. “Identifying hubs in protein interaction networks”. In: *PloS one* 4.4 (2009), e5344.

[253] Martijn P Van den Heuvel and Olaf Sporns. “Network hubs in the human brain”. In: *Trends in cognitive sciences* 17.12 (2013), pp. 683–696.

[254] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.”. In: *Journal of machine learning research* 9.11 (2008).

[255] David C Van Essen et al. “The WU-Minn human connectome project: an overview”. In: *Neuroimage* 80 (2013), pp. 62–79.

[256] Yves Van Gennip et al. “Community detection using spectral clustering on sparse geosocial data”. In: *SIAM Journal on Applied Mathematics* 73.1 (2013), pp. 67–83.

[257] América Vera-Montecinos et al. “Analysis of networks in the dorsolateral prefrontal cortex in chronic schizophrenia: Relevance of altered immune response”. In: *Frontiers in Pharmacology* 14 (2023), p. 1003557.

[258] Ulrike Von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[259] Chun Wang et al. “Mgae: Marginalized graph autoencoder for graph clustering”. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 889–898.

[260] Hua Wang et al. “Nonnegative matrix tri-factorization based high-order co-clustering and its fast implementation”. In: *2011 IEEE 11th international conference on data mining*. IEEE. 2011, pp. 774–783.

[261] Huaning Wang et al. “Evidence of a dissociation pattern in default mode subnetwork functional connectivity in schizophrenia”. In: *Scientific reports* 5.1 (2015), p. 14655.

[262] Shuai Wang et al. “Clustering by orthogonal NMF model and non-convex penalty optimization”. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 5273–5288.

[263] Yu Wang, Wotao Yin, and Jinshan Zeng. “Global convergence of ADMM in nonconvex nonsmooth optimization”. In: *Journal of Scientific Computing* 78 (2019), pp. 29–63.

[264] Wenhui Wu et al. “Nonnegative matrix factorization with mixed hypergraph regularization for community detection”. In: *Information Sciences* 435 (2018), pp. 263–281.

[265] Zonghan Wu et al. “Beyond low-pass filtering: Graph convolutional networks with automatic filtering”. In: *IEEE Transactions on Knowledge and Data Engineering* (2022).

[266] Tian Xie, Bin Wang, and C-C Jay Kuo. “Graphhop: An enhanced label propagation method for node classification”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[267] Bingbing Xu et al. “Graph convolutional networks using heat kernel for semi-supervised learning”. In: *arXiv preprint arXiv:2007.16002* (2020).

[268] Bingbing Xu et al. “Graph wavelet neural network”. In: *arXiv preprint arXiv:1904.07785* (2019).

[269] Luming Xu et al. “ARMA Graph Filter Design by Least Squares Method Using Reciprocal Polynomial and Second-order Factorization”. In: *2023 16th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE. 2023, pp. 1–5.

[270] Yangwen Xu et al. “Intrinsic functional network architecture of human semantic processing: Modules and hubs”. In: *Neuroimage* 132 (2016), pp. 542–555.

[271] Zhilei Xu et al. “Meta-connectomic analysis maps consistent, reproducible, and transcrip-

tionally relevant functional connectome hubs in the human brain". In: *Communications Biology* 5.1 (2022), p. 1056.

[272] Cheng Yang et al. "Network representation learning with rich text information". In: *Twenty-fourth international joint conference on artificial intelligence*. 2015.

[273] Defu Yang et al. "Joint hub identification for brain networks by multivariate graph inference". In: *Medical image analysis* 73 (2021), p. 102162.

[274] Defu Yang et al. "Joint identification of network hub nodes by multivariate graph inference". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*. Springer. 2019, pp. 590–598.

[275] H Yang et al. "Constrained Independent Component Analysis Based on Entropy Bound Minimization for Subgroup Identification from Multi-subject fMRI Data". In: *ICASSP 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2023, pp. 1–5.

[276] Hanlu Yang et al. "Identification of Homogeneous Subgroups from Resting-State fMRI Data". In: *Sensors* 23.6 (2023), p. 3264.

[277] Jaewon Yang, Julian McAuley, and Jure Leskovec. "Community detection in networks with node attributes". In: *2013 IEEE 13th international conference on data mining*. IEEE. 2013, pp. 1151–1156.

[278] BT Thomas Yeo et al. "The organization of the human cerebral cortex estimated by intrinsic functional connectivity". In: *Journal of neurophysiology* (2011).

[279] Wutao Yin, Longhai Li, and Fang-Xiang Wu. "Deep learning for brain disorder diagnosis based on fMRI images". In: *Neurocomputing* 469 (2022), pp. 332–345.

[280] Jiho Yoo and Seungjin Choi. "Orthogonal nonnegative matrix tri-factorization for co-clustering: Multiplicative updates on stiefel manifolds". In: *Information processing & management* 46.5 (2010), pp. 559–570.

[281] Liu Yue et al. "A survey of deep graph clustering: Taxonomy, challenge, and application". In: *arXiv preprint arXiv:2211.12875* (2022).

[282] Andrew Zalesky, Alex Fornito, and Edward T Bullmore. "Network-based statistic: identifying differences in brain networks". In: *Neuroimage* 53.4 (2010), pp. 1197–1207.

[283] Fan Zhang and Edwin R Hancock. "Graph spectral image smoothing using the heat kernel". In: *Pattern Recognition* 41.11 (2008), pp. 3328–3342.

[284] Hongyuan Zhang et al. "Embedding graph auto-encoder for graph clustering". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[285] John X Zhang, Hoi-Chung Leung, and Marcia K Johnson. "Frontal activations associated with accessing and evaluating information in working memory: an fMRI study". In: *Neuroimage* 20.3 (2003), pp. 1531–1539.

[286] Xiaotong Zhang et al. “Attributed graph clustering via adaptive graph convolution”. In: *arXiv preprint arXiv:1906.01210* (2019).

[287] Yipu Zhang et al. “Multi-paradigm fMRI fusion via sparse tensor decomposition in brain functional connectivity study”. In: *IEEE journal of biomedical and health informatics* 25.5 (2020), pp. 1712–1723.

[288] Yu Zhang et al. “Functional annotation of human cognitive states using deep graph convolution”. In: *NeuroImage* 231 (2021), p. 117847.

[289] Peng Zhou and Liang Du. “Learnable graph filter for multi-view clustering”. In: *Proceedings of the 31st ACM International Conference on Multimedia*. 2023, pp. 3089–3098.

[290] Guangyao Zhu and Kan Li. “A unified model for community detection of multiplex networks”. In: *International Conference on Web Information Systems Engineering*. Springer. 2014, pp. 31–46.

[291] Meiqi Zhu et al. “Interpreting and unifying graph neural networks with an optimization framework”. In: *Proceedings of the Web Conference 2021*. 2021, pp. 1215–1226.

APPENDIX A: AUXILIARY FUNCTION PROOF

Proposition 1. *The following function,* $Z(h, h_{ij}^t)$*,*

$$Z(h, h_{ij}^t) = \mathcal{L}(h_{ij}^t) + 3\mathcal{L}'(h_{ij}^t)(h - h_{ij}^t) + \frac{3}{2}\frac{\sum_{l=1}^{L}(4\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l + 4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t}(h - h_{ij}^t)^2$$

is an auxiliary function of $\mathcal{L}(H)$*,*

$$\mathcal{L}(h) = \mathcal{L}(h_{ij}^t) + \mathcal{L}'(h_{ij}^t)(h - h_{ij}^t) + \frac{1}{2}\mathcal{L}''(h_{ij}^t)(h - h_{ij}^t)^2.$$

Proof. First, when $h = h_{ij}^t$ the equality $Z(h, h) = \mathcal{L}(h)$ holds. Now, we need to show that $Z(h, h_{ij}^t) \geq \mathcal{L}(h)$.

It can be seen that the first and second terms of $Z(h, h_{ij}^t)$ are greater than the first and second terms in $\mathcal{L}(h)$. Therefore, it suffices to show that $\frac{3\sum_{l=1}^{L}(4\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l{}^\top\mathbf{H}^t\mathbf{S}_l+4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} \geq \mathcal{L}''(h_{ij}^t)$.

It can be shown that

$$\frac{(\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} = \frac{\sum_{p,q}(\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top)_{ip}h_{pq}^{qj}}{h_{ij}^t} \geq (\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top)_{ii}\mathbf{S}_{jj},$$

$$\frac{(\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} = \frac{\sum_p h_{ip}^t(\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{pj}}{h_{ij}^t} \geq (\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{jj},$$

$$\frac{(\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} = \frac{\sum_{p,q} h_{ip}^t h_{qp}^t(\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{qj}}{h_{ij}^t} \geq h_{ij}^t(\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij},$$

$$\frac{(\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} = \frac{\sum_{m,b}(\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l)_{im}h_{mb}^{bj}}{h_{ij}^t} \geq (\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l)_{ii}\mathbf{S}_{l_{jj}}.$$

Therefore,

$$3\frac{(\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} \geq (\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij} + h_{ij}^t(\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij} + (\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l)_{ii}\mathbf{S}_{l_{jj}},$$

and thus $\frac{3\sum_{l=1}^{L}(4\mathbf{H}_l\mathbf{G}_l\mathbf{H}_l^\top\mathbf{H}^t\mathbf{S}_l+4\mathbf{H}^t\mathbf{H}^{t\top}\mathbf{A}_l\mathbf{H}^t\mathbf{S}_l)_{ij}}{h_{ij}^t} \geq \mathcal{L}''(h_{ij}^t)$. Therefore, Eq. (2.10) is an auxiliary function of $\mathcal{L}(h)$. □

APPENDIX B: CONSISTENCY PROOF

The proof for Theorem 2 consists of three steps. The first step was addressed in Lemma 2 , where it was shown that true community labels can be recovered from the solution of the objective function applied to the population adjacency tensor $\mathscr{A}$. The rest of the proof is addressed below.

Proof. We will refer to the objective function in (2.11) in the manuscript as $F(\mathbb{A}, (\mathbf{H}'_1, \cdots, \mathbf{H}'_L))$. For any feasible solution $(\mathbf{H}'_1, \cdots, \mathbf{H}'_L)$, we have

$$|F(\mathscr{A}, (\mathbf{H}'_1, \cdots, \mathbf{H}'_L)) - F(\mathbb{A}, (\mathbf{H}'_1, \cdots, \mathbf{H}'_L))| = |\sum_{l=1}^{L}(||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F - ||\mathbf{H}_l'^{\top}\mathbf{A}_l\mathbf{H}'_l||_F)|$$

$$= \sum_{l=1}^{L}\{(||\mathbf{H}_l'^{\top}\mathbf{A}_l\mathbf{H}'_l||_F - ||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F)^2 + |2(||\mathbf{H}_l'^{\top}\mathbf{A}_l\mathbf{H}'_l||_F - ||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F)||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F|\}.$$

For each layer l, $||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F$ term is upper bounded as,

$$||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F \leq \sqrt{k_l}||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_2 \leq \sqrt{k_l}||\mathbf{H}'_l||_2^2||\mathcal{A}_l||_2 \leq \sqrt{k_l}\Delta_l.$$

The inequality in the first line is due to the relationship between the Frobenious norm and spectral norm since. The inequalities in lines 2 and 3 are due to the property of spectral norm $||\mathbf{AB}||_2 \leq ||\mathbf{A}||_2||\mathbf{B}||_2$ and the inequality $||\mathcal{A}_l|| \leq \Delta_l$, respectively.

Since $||\mathbf{A}||_F - ||\mathbf{B}||_F \leq ||\mathbf{A} - \mathbf{B}||_F$, for each layer l, we have

$$\begin{aligned}
&|(||\mathbf{H}_l'^{\top}\mathbf{A}_l\mathbf{H}'_l||_F - ||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F)||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}'_l||_F| \\
&\leq \sqrt{k_l}\Delta_l||\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}_l||_F \\
&= \sqrt{k_l}\Delta_l\sqrt{\text{tr}(\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}'_l\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}'_l)} \\
&\leq \sqrt{k_l}\Delta_l\sqrt{||\mathbf{H}'_l\mathbf{H}_l'^{\top}||_2\text{tr}((\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}'_l\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l))} \\
&= \sqrt{k_l}\Delta_l\sqrt{\text{tr}((\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}'_l\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l))} \\
&\leq \sqrt{k_l}\Delta_l\sqrt{k_l||\mathbf{H}'_l||_2^2||\mathbf{A}_l - \mathcal{A}_l||_2} \\
&\leq \sqrt{k_l}\Delta_l\sqrt{k_l(4\Delta_l log(2n/\epsilon))^{1/2}} \\
&= \sqrt{2}k_l\Delta_l^{5/4}(log(2n/\epsilon))^{1/4}
\end{aligned}$$

with probability at least $1-\epsilon$. Inequality in line 4 is due to the inequality on the trace of the product of a positive semi-definite matrix $\text{tr}((\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}'_l\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)$ with a Hermitian matrix $\mathbf{H}'_l\mathbf{H}_l'^{\top}$ [197].

The inequality in line 6 is due to the relation $\text{tr}(AB) \leq k||A||_2||B||_2$. And the inequality in line 7 comes from results presented in [56] for single graphs, where $||\mathbf{A}_l - \mathcal{A}_l||_2 \leq \sqrt{4\Delta_l log(2N/\epsilon))^{1/2}}$

Similarly,

$$(||\mathbf{H}_l'^{\top}\mathbf{A}_l\mathbf{H}_l'||_F - ||\mathbf{H}_l'^{\top}\mathcal{A}_l\mathbf{H}_l'||_F)^2 \leq ||\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}_l||_F^2 \leq \text{tr}(\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}_l'\mathbf{H}_l'^{\top}(\mathbf{A}_l - \mathcal{A}_l)\mathbf{H}_l')$$
$$\leq 2k_l\Delta_l^{1/2}(log(2N/\epsilon))^{1/2}.$$

Combining the above results, we have

$$\begin{aligned}
&|F(\mathscr{A}, (\mathbf{H}_1', \cdots, \mathbf{H}_L')) - F(\mathbb{A}, (\mathbf{H}_1', \cdots, \mathbf{H}_L'))| \\
&\leq \sum_{l=1}^{L}\{2k_l\Delta_l^{1/2}(log(2N/\epsilon))^{1/2} + 2\sqrt{2}k_l\Delta_l^{5/4}(log(2N/\epsilon))^{1/4}\} \\
&\leq \sum_{l=1}^{L} 6k_l\Delta_l^{5/4}(log(2N/\epsilon))^{1/2} \leq 6k_{max}(log(2N/\epsilon))^{1/2}\sum_{l=1}^{L}\Delta_l^{5/4} \\
&\leq 6k_{max}(log(2N/\epsilon))^{1/2}(\sum_{l=1}^{L}\Delta_l)^{5/4} \leq 6k_{max}(log(2N/\epsilon))^{1/2}(L\bar{\Delta})^{5/4}.
\end{aligned}$$

The fourth inequality follows from the relation $\sum_{i=1}^{n}(x_i)^p \leq (\sum_{i=1}^{n} x_i)^p$, for $x_i > 0$ and real p with $p \geq 1$ proved in [19]. The last inequality is due to $\bar{\Delta} = \frac{1}{L}\sum_{l=1}\Delta_l$.

Finally, let $(\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L')$ be the solution of the optimization problem in (11). Further let $(\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L')$ maximize the population version of the objective function $F(\mathscr{A}, (\mathbf{H}_1', \cdots, \mathbf{H}_L'))$. Then $F(\mathbb{A}, (\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L')) \geq F(\mathbb{A}, (\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L'))$ and $F(\mathscr{A}, (\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L')) \geq F(\mathscr{A}, (\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L'))$. Therefore, we have with probability at least $1 - \epsilon$,

$$\begin{aligned}
&F(\mathscr{A}, (\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L')) - F(\mathscr{A}, (\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L')) \\
&\leq F(\mathscr{A}, (\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L')) - F(\mathscr{A}, (\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L')) + F(\mathbb{A}, (\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L')) - F(\mathbb{A}, (\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L')) \\
&\leq 12k_{max}(log(2N/\epsilon))^{1/2}(L\bar{\Delta})^{5/4}.
\end{aligned}$$

From [197], we have

$$F(\mathscr{A}, (\bar{\mathbf{H}}_1', \cdots, \bar{\mathbf{H}}_L')) - F(\mathscr{A}, (\hat{\mathbf{H}}_1', \cdots, \hat{\mathbf{H}}_L')) \geq \frac{Nr_{MX}}{8N_{max}}\sum_{l=1}^{L}(\lambda_l)^2.$$

Hence, with probability at least $1 - \epsilon$

$$r_{MX} \leq \frac{96N_{max}k_{max}L^{1/4}\bar{\Delta}^{5/4}(log(2N/\epsilon))^{1/2}}{N\frac{1}{L}\sum_{l=1}^{L}(\lambda_l)^2}.$$

□

APPENDIX C: INVERTIBILITY PROOF

The structure of $\mathbf{Y}_1$ involves a Vandermonde-like matrix, which depends on the eigenvalues of the normalized Laplacian, $\mathbf{L}_n$. Specifically, the matrix $\mathbf{Y}_1 \in \mathbb{R}^{Q-1\times Q-1}$ is defined as:

$$\mathbf{Y}_1 = \bar{\mathbf{\Psi}}_Q^\top \operatorname{diag}(\mathbf{\Psi}_M \mathbf{c}) \operatorname{diag}(\mathbf{\Psi}_M \mathbf{c}) \bar{\mathbf{\Psi}}_Q,$$

where,

$$\bar{\mathbf{\Psi}}_Q = \begin{bmatrix} \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{Q-1} \\ \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{Q-1} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_N & \lambda_N^2 & \cdots & \lambda_N^{Q-1} \end{bmatrix}, \mathbf{\Psi}_M = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \cdots & \lambda_1^{M-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & \cdots & \lambda_N^{M-1} \end{bmatrix},$$

are $N \times Q-1$ and $N \times M$ Vandermonde-like matrices, respectively based on the eigenvalues $\{\lambda_i\}_{i=1}^N$ of $\mathbf{L}_n$. Let $\mathbf{C} = \operatorname{diag}(\mathbf{\Psi}_M \mathbf{c}) \operatorname{diag}(\mathbf{\Psi}_M \mathbf{c})$, with entries $C_{ii} = (\sum_{m=0}^{M-1} c_m \lambda_i^m)^2 > 0$.

Proposition 2. *For a symmetric and positive definite matrix* $\mathbf{C} \in \mathbb{R}^{N\times N}$, $\mathbf{Y}_1 = \bar{\mathbf{\Psi}}_Q^\top \mathbf{C} \bar{\mathbf{\Psi}}_Q \in \mathbb{R}^{Q-1\times Q-1}$ *is invertible if and only if* $\bar{\mathbf{\Psi}}_Q$ *has full rank.*

Proof. Let $\mathbf{x} \in \mathbb{R}^{Q-1} \setminus \{0\}$ be an arbitrary vector. If $\bar{\mathbf{\Psi}}_Q$ has full (column) rank, then $\bar{\mathbf{\Psi}}_Q^\top$ has full (row) rank as well and is injective. Hence, define $\mathbf{y} := \bar{\mathbf{\Psi}}_Q \mathbf{x} \in \mathbb{R}^N \setminus \{0\}$. The positive definiteness of $\mathbf{C}$ yields

$$\mathbf{x}^T (\bar{\mathbf{\Psi}}_Q^\top \mathbf{C} \bar{\mathbf{\Psi}}_Q) \mathbf{x} = (\bar{\mathbf{\Psi}}_Q \mathbf{x})^T \mathbf{C} (\bar{\mathbf{\Psi}}_Q \mathbf{x}) = \mathbf{y}^T \mathbf{C} \mathbf{y} > 0,$$

i.e., $\bar{\mathbf{\Psi}}_Q^\top \mathbf{C} \bar{\mathbf{\Psi}}_Q$ is (symmetric and) positive definite and thus invertible. Conversely, if $\bar{\mathbf{\Psi}}_Q$ does not have full rank, it is not injective, and there exists a vector $\mathbf{x} \in \mathbb{R}^{Q-1} \setminus \{0\}$ such that $\bar{\mathbf{\Psi}}_Q \mathbf{x} = 0$. Hence, $\bar{\mathbf{\Psi}}_Q^T \mathbf{C} \bar{\mathbf{\Psi}}_Q \mathbf{x} = 0$, $\bar{\mathbf{\Psi}}_Q^T \mathbf{C} \bar{\mathbf{\Psi}}_Q$ would not be injective and thus not invertible.

Therefore, for $\mathbf{Y}_1$ to be invertible, the Vandermonde matrix must have full rank.

□

In our case, $Q - 1 < N$. Thus, the Vandermonde matrix will be full rank if $\mathbf{L}_n$ has at least $Q - 1$ distinct eigenvalues. This is a reasonable assumption given that $Q - 1 \ll N$.

Similarly, $\mathbf{Y}_2$ depends on $\mathbf{\Psi}_M$. Since the same Vandermonde-like structure is present, the same reasoning applies. As long as $M \ll N$, it suffices to have M distinct eigenvalues of $\mathbf{L}_n$ for $\mathbf{Y}_2$ to be invertible. These conditions are typically satisfied in practical scenarios where $Q, M \ll N$.