

# COMMUNITY DETECTION IN TEMPORAL MULTI-LAYER NETWORKS

By

Esraa Mustafa Al-sharoa

A DISSERTATION

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

Electrical Engineering - Doctor of Philosophy

2019

# **ABSTRACT**

## **COMMUNITY DETECTION IN TEMPORAL MULTI-LAYER NETWORKS**

**By**

**Esraa Mustafa Al-sharova**

Many real world systems and relational data can be modeled as networks or graphs. With the availability of large amounts of network data, it is important to be able to reduce the network's dimensionality and extract useful information from it. A key approach to network data reduction is community detection. The objective of community detection is to summarize the network by a set of modules, where the similarity within the modules is maximized while the similarity between different modules is minimized. Early work in graph based community detection methods focused on static or single layer networks. This type of networks is usually considered as an oversimplification of many real world complex systems, such as social networks where there may be different types of relationships that evolve with time. Consequently, there is a need for a meaningful representation of such complex systems. Recently, multi-layer networks have been used to model complex systems where the objects may interact through different mechanisms. However, there is limited amount of work in community detection methods for dynamic and multi-layer networks.

In this thesis, we focus on detecting and tracking the community structure in dynamic and multi-layer networks. Two particular applications of interest are considered including temporal social networks and dynamic functional connectivity networks (dFCNs) of the brain.

In order to detect the community structure in dynamic single-layer and multi-layer networks, we have developed methods that capture the structure of these complex networks. In Chapter 2, a low-rank + sparse estimation based evolutionary spectral clustering approach is proposed to detect

and track the community structure in temporal networks. The proposed method tries to decompose the network into low-rank and sparse parts and obtain smooth cluster assignments by minimizing the subspace distance between consecutive time points, simultaneously. Effectiveness of the proposed approach is evaluated on several synthetic and real social temporal networks and compared to the existing state-of-the-art algorithms. As the method developed in Chapter 2 is limited to dynamic single-layer networks and can only take limited amount of historic information into account, a tensor-based approach is developed in Chapter 3 to detect the community structure in dynamic single-layer and multi-layer networks. The proposed framework is used to track the change points as well as identify the community structure across time and multiple subjects of dFCNs constructed from resting state functional magnetic resonance imaging (rs-fMRI) data. The dFCNs are summarized into a set of FC states that are consistent over time and subjects. The detected community structures are evaluated using a consistency measure. In Chapter 4, an information-theoretic approach is introduced to aggregate the dynamic networks and identify the time points that are topologically similar to combine them into a tensor. The community structure of the reduced network is then detected using a tensor based approach similar to the one described in Chapter 3. In Chapter 5, a temporal block spectral clustering framework is introduced to detect and track the community structure of multi-layer temporal networks. A set of intra- and inter-adjacency matrices is constructed and combined to create a set of temporal supra-adjacency matrices. In particular, both the connections between nodes of the network within a time window, i.e. intra-layer adjacency, as well as the connections between nodes across different time windows, i.e. inter-layer adjacency are taken into account. The community structure is then detected by applying spectral clustering to these supra-adjacency matrices. The proposed approach is evaluated on dFCNs constructed from rs-fMRI across time and subjects revealing dynamic connectivity patterns between the resting state networks (RSNs).

*This work is dedicated to my beloved parents, Mustafa Al-sharoa and Shama Al-kofahi, who have been the source of inspiration, encouragement and unconditional love during all times. To my soulmate and lovely husband, Mahmood Al-khassaweneh, who has been supporting and encouraging me to make my dreams come true. Thank you for being in my life and believing in me. To my precious children, Ahmad, Ghadaq, Alaya and Tabarak. You are the source of my happiness, strength and success. I love you to the moon and beyond. To my family in law, sisters, brothers and friends. Thank you for your prayers, kind words and support.*

## **ACKNOWLEDGMENTS**

I would like to thank my advisor Prof. Selin Aviyente who without her guidance, valuable comments and inputs this dissertation would not have been possible. Many thanks to Prof. Aviyente for being ready to help in all times. In addition, I would like to thank my committee members Prof. John Deller, Prof. Hayder Radha and Dr. Mark Iwen for their useful comments and guidance.

A very special thanks go to the Schlumberger Foundation, Faculty for the Future Fellowship who supported me through my PhD. I am very grateful to them.

I would like also to thank my husband Mahmood Al-khassaweneh, my children Ahmad, Ghadaq, Alaya and Tabarak. With their encouragement, support and love I was able to finish this dissertation.

Finally, I would like to thank the best role models in my life, my parents Mustafa Al-sharoa and Shama Al-kofahi, who have been showering me with their love and prayers. They inculcated me with the willing to learn and build a bright future. Thanking them will never be enough.

# TABLE OF CONTENTS

|                                                                                                                                                         |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <b>LIST OF TABLES</b> . . . . .                                                                                                                         | <b>ix</b>  |
| <b>LIST OF FIGURES</b> . . . . .                                                                                                                        | <b>x</b>   |
| <b>LIST OF ALGORITHMS</b> . . . . .                                                                                                                     | <b>xiv</b> |
| <b>Chapter 1 Introduction</b> . . . . .                                                                                                                 | <b>1</b>   |
| 1.1 Graph Theory . . . . .                                                                                                                              | 2          |
| 1.2 Community Detection in Static Networks . . . . .                                                                                                    | 4          |
| 1.2.1 Graph Cut problem and Spectral Clustering . . . . .                                                                                               | 5          |
| 1.2.2 Modularity Based Clustering . . . . .                                                                                                             | 6          |
| 1.3 Background on Tensors . . . . .                                                                                                                     | 7          |
| 1.4 Community Detection in Dynamic Networks . . . . .                                                                                                   | 10         |
| 1.5 Community Detection in Dynamic Multi-layer Networks . . . . .                                                                                       | 12         |
| 1.6 Dynamic Functional Connectivity Networks of the Brain . . . . .                                                                                     | 13         |
| 1.7 Organization and Contributions of this Thesis . . . . .                                                                                             | 16         |
| <b>Chapter 2 Detecting and Tracking Community Structure in Temporal Networks: A Low-Rank + Sparse Estimation Based Evolutionary Clustering Approach</b> | <b>19</b>  |
| 2.1 Introduction . . . . .                                                                                                                              | 19         |
| 2.2 Background . . . . .                                                                                                                                | 22         |
| 2.2.1 Spectral Clustering . . . . .                                                                                                                     | 22         |
| 2.2.2 Low-rank + Sparse matrix approximation . . . . .                                                                                                  | 23         |
| 2.2.3 Chordal Subspace Distance Measure . . . . .                                                                                                       | 24         |
| 2.2.4 Variation of Information . . . . .                                                                                                                | 24         |
| 2.3 Methods: low-rank + sparse estimation based evolutionary spectral clustering (LS-ESC) . . . . .                                                     | 25         |
| 2.3.1 Problem Formulation . . . . .                                                                                                                     | 25         |
| 2.3.2 Distance Between Successive Subspaces . . . . .                                                                                                   | 26         |
| 2.3.3 Problem Solution . . . . .                                                                                                                        | 28         |
| 2.3.4 Tracking the community structure over time . . . . .                                                                                              | 30         |
| 2.3.5 Detecting the number of clusters . . . . .                                                                                                        | 30         |
| 2.4 Results . . . . .                                                                                                                                   | 32         |
| 2.4.1 Simulated temporal networks . . . . .                                                                                                             | 32         |
| 2.4.1.1 Simulated binary temporal networks . . . . .                                                                                                    | 32         |
| 2.4.1.2 Simulated weighted temporal networks . . . . .                                                                                                  | 35         |
| 2.4.2 Real social network: MIT Reality Mining Dataset . . . . .                                                                                         | 38         |
| 2.4.3 Real social network: Primary school network . . . . .                                                                                             | 39         |
| 2.4.3.1 Primary school data set description and temporal network construction . . . . .                                                                 | 39         |

|                  |                                                                                                       |           |
|------------------|-------------------------------------------------------------------------------------------------------|-----------|
| 2.4.3.2          | Detecting and tracking the community structure in the primary school temporal network (PSTN)          | 42        |
| 2.5              | Conclusions                                                                                           | 43        |
| <b>Chapter 3</b> | <b>Tensor Based Temporal and Multi-layer Community Detection</b>                                      | <b>47</b> |
| 3.1              | Introduction                                                                                          | 47        |
| 3.2              | Background                                                                                            | 50        |
| 3.2.1            | Quality Metrics to Determine the Number of Clusters                                                   | 50        |
| 3.3              | Methods                                                                                               | 52        |
| 3.3.1            | Spectral Clustering                                                                                   | 52        |
| 3.3.2            | 3D-Windowed Tensor Approach (3D-WTA)                                                                  | 53        |
| 3.3.3            | Running Time Tensor Approach (RTTA)                                                                   | 54        |
| 3.3.4            | 4D-Windowed Tensor Approach (4D-WTA)                                                                  | 56        |
| 3.3.5            | Tracking the Community Structure Over Time                                                            | 57        |
| 3.4              | Results                                                                                               | 60        |
| 3.4.1            | Simulated Temporal Networks                                                                           | 60        |
| 3.4.1.1          | Description of the Data sets                                                                          | 60        |
| 3.4.1.2          | Comparison of different quality metrics                                                               | 63        |
| 3.4.1.3          | Comparison between 3D-WTA and state of the art algorithms                                             | 64        |
| 3.4.1.4          | Comparison between 3D-WTA and 4D-WTA                                                                  | 65        |
| 3.4.1.5          | Flocks of Boids Dataset                                                                               | 69        |
| 3.4.2            | MIT Reality Mining Dataset                                                                            | 69        |
| 3.4.3            | Application to dFCN From EEG Dataset                                                                  | 70        |
| 3.4.4            | Application to dFCN From rs-fMRI Dataset                                                              | 71        |
| 3.4.4.1          | Data Preprocessing and Functional Connectivity Networks Construction                                  | 71        |
| 3.4.5            | Dynamic Functional Connectivity Network Construction                                                  | 72        |
| 3.4.5.1          | Resting State Networks (RSNs)                                                                         | 72        |
| 3.4.5.2          | Detecting the Community Structure in rs-fMRI                                                          | 72        |
| 3.4.5.3          | Community structure of each FC state                                                                  | 76        |
| 3.4.5.4          | State transition profiles                                                                             | 77        |
| 3.4.5.5          | Consistency of the RSNs across time                                                                   | 78        |
| 3.4.5.6          | Comparison Between 3D-WTA and 4D-WTA                                                                  | 79        |
| 3.5              | Conclusions                                                                                           | 79        |
| <b>Chapter 4</b> | <b>Detecting Brain Dynamics During Resting State: A Tensor Based evolutionary Clustering Approach</b> | <b>90</b> |
| 4.1              | Introduction                                                                                          | 90        |
| 4.2              | Materials and Methods                                                                                 | 94        |
| 4.2.1            | Data and Preprocessing                                                                                | 94        |
| 4.2.2            | Dynamic Functional Connectivity Networks (dFCNs)                                                      | 95        |
| 4.2.2.1          | Dynamic Functional Connectivity Network Construction                                                  | 95        |
| 4.2.2.2          | Surrogate Data Generation for Testing Statistical Significance of the dFCN                            | 95        |
| 4.2.3            | Reducibility of Brain dFCNs                                                                           | 96        |

|                     |                                                                                                               |            |
|---------------------|---------------------------------------------------------------------------------------------------------------|------------|
| 4.2.4               | Dynamic Network Clustering . . . . .                                                                          | 99         |
| 4.2.4.1             | HOOI Clustering Based Approach . . . . .                                                                      | 99         |
| 4.2.4.2             | Choosing the Number of Clusters . . . . .                                                                     | 100        |
| 4.3                 | Results . . . . .                                                                                             | 100        |
| 4.3.1               | dFCN Reduction . . . . .                                                                                      | 100        |
| 4.3.2               | FC States Summarization . . . . .                                                                             | 102        |
| 4.4                 | Conclusions . . . . .                                                                                         | 106        |
| <b>Chapter 5</b>    | <b>Temporal Block Spectral Clustering for Multi-layer Temporal Functional Connectivity Networks . . . . .</b> | <b>109</b> |
| 5.1                 | Introduction . . . . .                                                                                        | 109        |
| 5.2                 | Methods . . . . .                                                                                             | 111        |
| 5.2.1               | Construction of multi-layer temporal networks . . . . .                                                       | 111        |
| 5.2.1.1             | Construction of a temporal network from signals . . . . .                                                     | 111        |
| 5.2.1.2             | Construction of multi-layer temporal network . . . . .                                                        | 112        |
| 5.2.2               | Temporal block spectral clustering approach (TBSCA) . . . . .                                                 | 114        |
| 5.3                 | Results . . . . .                                                                                             | 115        |
| 5.3.1               | Simulated networks . . . . .                                                                                  | 115        |
| 5.3.2               | Community Detection in dynamic functional connectivity networks (dFCNs) . . . . .                             | 116        |
| 5.3.2.1             | Data and preprocessing . . . . .                                                                              | 116        |
| 5.3.2.2             | Dynamic Community structure . . . . .                                                                         | 117        |
| 5.3.2.3             | Consistency of the RSNs across time . . . . .                                                                 | 118        |
| 5.4                 | Conclusions . . . . .                                                                                         | 121        |
| <b>Chapter 6</b>    | <b>Conclusions and Future Work . . . . .</b>                                                                  | <b>122</b> |
| 6.1                 | Conclusions . . . . .                                                                                         | 122        |
| 6.2                 | Future Work . . . . .                                                                                         | 125        |
| <b>APPENDIX</b>     | <b>. . . . .</b>                                                                                              | <b>127</b> |
| <b>BIBLIOGRAPHY</b> | <b>. . . . .</b>                                                                                              | <b>140</b> |

## LIST OF TABLES

|            |                                                                                                                                                                                                                                                                                                  |     |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Table 2.1: | Parameters of the simulated binary networks from Girvan and Newman benchmark for six different data sets and the selected values of the regularization parameters. . . . .                                                                                                                       | 33  |
| Table 2.2: | Parameters of the truncated Gaussian distribution used to generate the edges of the simulated weighted temporal networks for three different data sets, the ground truth of the nodes' community membership for the generated networks and the selected values of the regularization parameters. | 37  |
| Table 2.3: | Different classes and number of teachers and students participating in the primary school data from each class. . . . .                                                                                                                                                                          | 41  |
| Table 3.1: | Parameters of the truncated Gaussian distribution used to generate the edges of the simulated networks for four different datasets, and the ground truth of the nodes' community membership for the generated networks. .                                                                        | 62  |
| Table 3.2: | Parameters of the truncated Gaussian distribution used to generate the edges of the simulated networks for the 4D simulated data, and the ground truth of the nodes' community membership for the generated network. . .                                                                         | 68  |
| Table 3.3: | The ROIs belonging to the different subnetworks in resting state. The numbers in the table refer to the ROIs defined in Table 1. . . . .                                                                                                                                                         | 73  |
| Table 4.1: | The ROIs that defines the RSNs. The numbers in the table refer to the ROIs defined in Table 1. . . . .                                                                                                                                                                                           | 95  |
| Table 4.2: | Key Attributes of the Tensors Constructed from the Reduced dFCN. . . .                                                                                                                                                                                                                           | 102 |
| Table 1:   | List of the ROIs in the AAL 90 template. The odd and even indices refer to the left and right brain hemispheres ROIs, respectively. . . . .                                                                                                                                                      | 139 |

## LIST OF FIGURES

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1: | An example of dynamic multi-layer network: The network consists of 6 nodes, 3 time points and 3 layers. Solid lines represent the edges between the nodes in the same layer while dashed lines represent the edges between nodes across layers. . . . .                                                                                                                                                                                                                                                                                                                                                     | 3  |
| Figure 2.1: | VI comparison between LS-ESC, AFFECT and DYNMOGA for simulated binary networks from Girvan and Newman benchmark with the parameters presented in Table 2.1. LS-ESC scored the lowest VI over time compared to the other methods: (a) Data set 1; (b)Data set 2; (c) Data set 3; (d) Data set 4; (e) Data set 5; and (f) Data set 6. . . . .                                                                                                                                                                                                                                                                 | 34 |
| Figure 2.2: | Comparison between LS-ESC and state of the art algorithms for the different weighted networks in Table 3.1: (a)-(c) The cost function calculated by LS-ESC, (d)-(f) The detected number of clusters (DNOC) for the simulated networks using different algorithms. LS-ESC estimated the correct number of clusters over time and the cost function reflected the correct change points; (g)-(i) Comparison of variation of information (VI) between LS-ESC, AFFECT and GenLov for the simulated networks. LS-ESC outperforms other methods and scored the lowest variation of information over time. . . . . | 36 |
| Figure 2.3: | Cost function calculated by LS-ESC for the MIT Reality mining network. The changes in the cost function coincides with six important dates during the school year (red dashed lines). . . . .                                                                                                                                                                                                                                                                                                                                                                                                               | 39 |
| Figure 2.4: | Detected community structure of the MIT reality mining temporal network at different times: (a) Week 5; (b)Week 15; (c) Week 27; and (d) Week 39. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 40 |
| Figure 2.5: | Cost function calculated by LS-ESC for the primary school temporal network for the: (a) first day, (b) second day. Red dashed lines refer to the time during the day. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                               | 44 |
| Figure 2.6: | Detected community structure in the first day of the primary school temporal network at different times: (a) Morning, (b) break, (c) lunch and (d) end of the school day. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                           | 45 |
| Figure 2.7: | Detected community structure in the second day of the primary school temporal network at different times: (a) Morning, (b) break, (c) lunch and (d) end of the school day. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                          | 46 |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 3.1: | Detected number of clusters (DNOC) by 3D-WTA for the data sets in Table 3.1 using different metrics, including: Eigengap (EG), modularity (Q), multi-resolution modularity $Q_\gamma$ and asymptotical surprise (AS). . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 63 |
| Figure 3.2: | Comparison between 3D-WTA and state of the art algorithms for the different data sets in Table 3.1: (a)-(d) The normalized cost function calculated by 3D-WTA, (e)-(h) The detected number of clusters (DNOC) for the simulated networks using different algorithms. 3D-WTA estimated the correct number of clusters over time and the normalized cost function reflected the correct change points represented by the pink dashed lines. (i)-(l) Comparison of variation of information (VI) between 3D-WTA, AFFECT, PCM-NC, PCQ-NC, TC and GenLov for the simulated networks. 3D-WTA scored the lowest variation of information over time compared to the other methods. . . . . | 66 |
| Figure 3.3: | Variation of information (VI) comparison between the detected community structure by 3D-WTA and 4D-WTA and the ground truth for simulated data. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 67 |
| Figure 3.4: | (a) The normalized cost function and the detected number of clusters for Flocks of Boids dataset using different algorithms. 3D-WTA results in the best estimate of the number of clusters especially after the change point, (b) Rand index comparison between 3D-WTA, AFFECT, PCM and PCQ for Flocks of Boids Dataset. . . . .                                                                                                                                                                                                                                                                                                                                                   | 82 |
| Figure 3.5: | Detected Change Points for Reality Mining Data. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 83 |
| Figure 3.6: | Detected Change Points for ERN Data. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 83 |
| Figure 3.7: | Community structure for the ERN networks obtained by RTTA: (a) Pre-ERN, (b) ERN, (c) Post-ERN. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 84 |
| Figure 3.8: | A flowchart of the application of 4D-WTA to rs-fMRI: (a) Time series from each ROI; (b) dFCNs constructed across time and subjects using Pearson correlation coefficient; (c) After random subsampling across time and subjects, 4 dimensional FC tensors are constructed for each sample. Each tensor consists of 30 random time points from each subject; (d) Tucker decomposition of each FC tensor; (e) Community structure found by 4D-WTA for each sample; (f) The detected community structure is summarized into 5 functional connectivity states defined through community detection using k-means. . . . .                                                               | 85 |
| Figure 3.9: | (a) Distribution of weights across time for all samples, (b) Distribution of weights across subjects for all samples. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 86 |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |     |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figure 3.10: | The similarity among the community structures obtained by 4D-WTA across the 200 subsamples across time and subjects. The similarity between the different community structures is quantified by variation of information. Samples with similar community structure are grouped together by k-means. . . . .                                                                                                                                                                                 | 87  |
| Figure 3.11: | State transition profiles for subjects 4, 7, 9 and 12. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                              | 87  |
| Figure 3.12: | Variation in RSNs' consistency across subjects calculated by (a) 3D-WTA, (b) PCQ-NC, (c) GenLov, (d) 4D-WTA. . . . .                                                                                                                                                                                                                                                                                                                                                                        | 88  |
| Figure 3.13: | Full and circular view of the community structure of the FC states found by 4D-WTA: (a) State 1 with 9 clusters; (b) State 2 with 4 clusters; (c) State 3 with 8 clusters; (d) State 4 with 7 clusters and (e) State 5 with 6 clusters. . . . .                                                                                                                                                                                                                                             | 89  |
| Figure 4.1:  | Jensen-Shannon divergence between all pairs of time points in the dFCN. . . . .                                                                                                                                                                                                                                                                                                                                                                                                             | 101 |
| Figure 4.2:  | The dendrogram resulting from hierarchical clustering and the corresponding values of $q$ . The dashed red line identifies the cut point of the dendrogram from the second peak of $q$ . . . . .                                                                                                                                                                                                                                                                                            | 101 |
| Figure 4.3:  | The detected community structure for the FC States (1 – 10). For each state, (left) the optimized weighted average of the adjacency matrices within the state. Each cluster is denoted by the main ROIs detected within the cluster. The clusters shown in the circular graphs (right) refer to the different RSNs defined in literature, and the connections between them are detected by the proposed algorithm. The number of clusters for the FC states varies between 3 and 4. . . . . | 108 |
| Figure 5.1:  | Comparison between the detected community structure by TBSCA, BNLSC and GenLov and the ground truth for the synthetic data set in terms of variation of information. The number of clusters changes from 3 to 2 at time point 21, and from 2 to 3 at time point 41. . . . .                                                                                                                                                                                                                 | 116 |
| Figure 5.2:  | Agreement between the detected community structures across time for subject 1 measured by variation of information (VI). The dashed lines refer to the detected change points. . . . .                                                                                                                                                                                                                                                                                                      | 118 |
| Figure 5.3:  | Detected community structure from selected change points. The ROIs in different networks reconfigured their community membership across time. Visualized by circularGraph toolbox [1]. . . . .                                                                                                                                                                                                                                                                                              | 119 |
| Figure 5.4:  | Agreement between the detected community structures across time for all subjects measured by variation of information (VI). . . . .                                                                                                                                                                                                                                                                                                                                                         | 120 |

Figure 5.5: Consistency of different RSNs across subjects calculated from the detected community structure by TBSCA. . . . . 120

## LIST OF ALGORITHMS

|               |                                                |     |
|---------------|------------------------------------------------|-----|
| Algorithm 2.1 | LS-ESC . . . . .                               | 31  |
| Algorithm 3.2 | 3D-Windowed Tensor Approach (3D-WTA) . . . . . | 54  |
| Algorithm 3.3 | Running Time Tensor Approach (RTTA) . . . . .  | 55  |
| Algorithm 3.4 | <b>TrackU</b> . . . . .                        | 56  |
| Algorithm 3.5 | 4D Windowed Tensor Approach (4D-WTA) . . . . . | 58  |
| Algorithm 4.6 | HOOI clustering based approach. . . . .        | 100 |

# Chapter 1

## Introduction

Many real world systems can be modeled as complex networks. The objects in these systems are defined as the nodes or vertices of the graph, while the interactions between these objects define the edges of the graph. With the growth of network data, it is important to be able to reduce the dimensionality and extract useful information from these networks. Examples of real systems that are naturally structured as networks include social [2], biological [3] and communication systems [4]. One of the most important characteristics of these networks is the community structure, where the nodes are organized into a set of modules. One common way to understand the community structure in these networks is through community detection techniques.

The objective of graph based community detection is to partition a set of nodes into smaller subsets (modules or communities) such that nodes within the same module are similar to each other and are densely connected while they are different from the nodes in the other modules. Since many complex systems can be modeled as graphs or networks, various graph based community detection techniques have been developed. However, most of these methods are limited to static networks, i.e. the communities do not change with time. As many real world systems exhibit variation in their community structure over time, there is a growing need to develop algorithms that detect the community structure and track its evolution across time [5]. Other complex systems might have different types of relationships between their entities and these systems can be modeled as multi-

layer networks [6]. Multi-layer networks consider relationships between nodes within layers, i.e. intra-layer edges, and across layers, i.e. inter-layer edges. In particular, multi-layer networks consider all possible edges between nodes within and across layers. Consequently, it is important to develop community detection algorithms that can detect and track the community structure in multi-layer networks in order to understand these systems correctly.

In this thesis, we focus on developing community detection approaches for two types of networks: dynamic (temporal) single-layer networks and dynamic multi-layer networks. In dynamic single-layer networks, the network topology changes over time as the nodes might come and go or the interactions between them might change across time. For instance, we can model a group of users on Facebook and their interactions by a dynamic network, where the interactions between them might vary across time as shown in the first row of Fig. 1.1. In multi-layer networks, the objects in the network have different types of interactions between them and each layer corresponds to one type of these interactions. For instance, a group of users on Facebook may also have accounts on Twitter and Instagram and interactions between these users can occur through the different accounts which reflect the multiple relationships between them. In dynamic multi-layer networks, the relationships between the objects vary both by the type of the relationship and time. For example, the interactions between the same group of users on Facebook, Twitter and Instagram may change over time as well as across the different layers and this can be represented by a dynamic multi-layer network as shown in Fig. 1.1.

## 1.1 Graph Theory

Let  $\mathcal{G} = \{V, E, \mathbf{A}\}$  be an undirected weighted graph with a set of vertices or nodes set  $V = \{v_1, \dots, v_n\}$ , and a set of edges,  $E$ , that represents the pairwise weights between the nodes. The

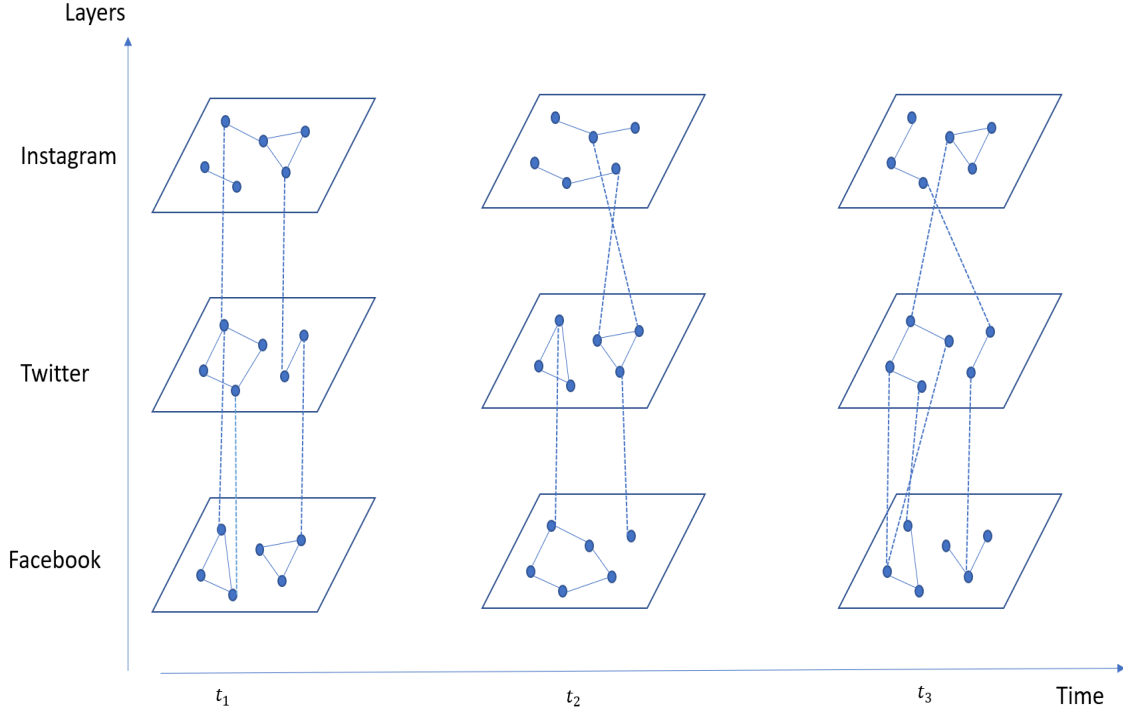


Figure 1.1: An example of dynamic multi-layer network: The network consists of 6 nodes, 3 time points and 3 layers. Solid lines represent the edges between the nodes in the same layer while dashed lines represent the edges between nodes across layers.

weighted adjacency matrix,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , is symmetric and its elements represent the similarity between each pair of nodes. For binary graphs,  $A_{ij} \in \{0, 1\}$  and for weighted graphs  $A_{ij} \in [0, 1]$ . The degree of a vertex,  $v_i \in V$ , is defined as  $d_i = \sum_{j=1}^n A_{ij}$  and the degree matrix,  $D$ , is defined as the diagonal matrix with  $\{d_1, \dots, d_n\}$  on its diagonal [7, 8].

For a dynamic network with  $M$  time points, a set of weighted and undirected graphs are constructed to represent the network over time as  $\{\mathcal{G}^{(t_m)}\}$ , where  $t_m \in \{t_1, t_2, \dots, t_M\}$ . The graph at each time point is represented by the adjacency matrix  $\mathbf{A}^{(t_m)} \in \mathbb{R}^{n \times n}$ .

## 1.2 Community Detection in Static Networks

Early work in community detection has focused on static networks. Different clustering approaches have been proposed for community detection, including hierarchical clustering, partitional [9] [7] and modularity optimization approaches [10, 11]. In hierarchical clustering approaches, a hierarchical structure is constructed using a distance matrix and visualized through a dendrogram. Hierarchical clustering is usually implemented by agglomerative or divisive algorithms. In agglomerative algorithms, clustering starts with individual nodes as separate clusters, then at each step the closest pairs of clusters are merged until all of the nodes form one big cluster. On the other hand, divisive algorithms start with all nodes as one cluster, then at each step it divides the clusters until each cluster contains only one node [12, 13]. Under partitional algorithms, squared-error based algorithms and graph-theoretic algorithms were introduced to assign the objects into  $r$  clusters. Squared-error based algorithms rely on optimizing a criterion function in order to produce the clusters. k-means clustering, the most popular squared-error algorithm, relies on a cost function that minimizes the within-cluster sum of squares (intra-cluster distances) and maximizes the between-cluster sum of squares (inter-clusters distances). Graph-theoretic algorithms partition the graph using a cut cost depending on the spectral properties of the graph, i.e. eigenvalues and eigenvectors of the adjacency matrix [7]. Under modularity optimization methods, the modularity measures the deviation of the within community edges from the expected value of the same quantity in a network with the same community partitions but random connections between the vertices [11]. Even though the aforementioned methods are limited to static networks, recent work has extended these methods to handle time varying networks. More discussion follows in Sec. 1.4.

### 1.2.1 Graph Cut problem and Spectral Clustering

Over the past decades, several methods have been proposed for graph partitioning. One of the simplest ways to partition a graph,  $\mathcal{G} = \{V, E, \mathbf{A}\}$ , is to solve the mincut problem. The mincut problem can be solved by choosing the partition  $C = \{C_1, \dots, C_r\}$  with  $C_l \cap C_{l'} = \emptyset$  where  $l \neq l'$  and  $C_1 \cup \dots \cup C_r = V$  which minimizes the following objective function:

$$cut(C_1, \dots, C_r) = \frac{1}{2} \sum_{l=1}^r A(C_l, \bar{C}_l), \quad (1.1)$$

where  $\bar{C}_l$  is the complement of  $C_l$  and  $A(C_l, \bar{C}_l) = \sum_{i \in C_l, j \in \bar{C}_l} A_{ij}$  for two disjoint sets  $C_l, \bar{C}_l \subset V$ . The mincut problem performs efficiently when  $r = 2$  as discussed in [14]. However, solving this problem might result in unbalanced clusters as it might separate one single vertex from the rest of the graph and consider it as a separate cluster. In order to overcome this problem, normalized versions of the cut function have been proposed. The two most commonly used ones are RatioCut [15] and normalized cut, Ncut [16]. In RatioCut, the size of a cluster in the graph is measured by the number of vertices,  $|C_l|$ , whereas in Ncut the size is measured by the total weights of the within cluster edges,  $vol(C_l)$ . These functions are defined as:

$$RatioCut(C_1, \dots, C_r) = \frac{1}{2} \sum_{l=1}^r \frac{A(C_l, \bar{C}_l)}{|C_l|}, \quad (1.2)$$

$$Ncut(C_1, \dots, C_r) = \frac{1}{2} \sum_{l=1}^r \frac{A(C_l, \bar{C}_l)}{vol(C_l)}. \quad (1.3)$$

Minimizing both of these functions leads to balanced clusters normalized by the number of vertices or edge weights, respectively. However, adding the balancing conditions,  $|C_l|$  and  $vol(C_l)$ , makes the cut problem NP hard to begin with as discussed in [17]. Spectral clustering offers an efficient solution to the relaxed versions of these problems. Relaxing the RatioCut leads to unnormalized spectral clustering and relaxing the Ncut leads to normalized spectral clustering.

Spectral clustering maps the relationship between the nodes in a graph into a lower dimensional subspace. Given a network with adjacency matrix,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , spectral clustering solves the following trace optimization problem [7],

$$\min_{\mathbf{U} \in \mathbb{R}^{n \times K}} \text{Trace}(\mathbf{U}^\dagger \mathbf{\Phi} \mathbf{U}), \quad s.t. \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}, \quad (1.4)$$

where  $\mathbf{\Phi} = \mathbf{I} - \mathbf{A}_N$  is the normalized Laplacian matrix with  $\mathbf{A}_N = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$  the normalized adjacency matrix [18, 19].

The normalized Laplacian matrix is a square, symmetric and positive semi-definite matrix. Consequently, all the eigenvalues,  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ , of  $\mathbf{\Phi}$  are real and positive with 0 as the smallest eigenvalue. The solution to the optimization problem in Eq. (2.1) is found by choosing  $\mathbf{U}$  as the matrix that contains the  $r$  eigenvectors that correspond to the smallest  $r$  eigenvalues of  $\mathbf{\Phi}$ . The community structure is then determined by applying  $k$ -means to the matrix  $\mathbf{U}$  [7]. Since we are concerned about community detection in dynamic networks, one approach is to apply standard spectral clustering at each time point individually. However, this approach is very sensitive to instantaneous noise, and produces inaccurate clustering results for noisy networks.

### 1.2.2 Modularity Based Clustering

Modularity based clustering is considered as one of the most popular optimization based clustering techniques. This approach relies on maximizing the modularity function across all possible partitions [11, 20]. Modularity can be defined as,

$$Q = \frac{1}{2E} \sum_{c \in C} \sum_{i, j \in V_c} [A_{ij} - \frac{D_i D_j}{2E}], \quad (1.5)$$

where  $2E = \sum_{i,j=1}^n A_{ij}$  is the total edge weight,  $A_{ij}$  is the  $ij^{th}$  entry of the adjacency matrix,  $D_i$  is the degree of node  $i$ ,  $C = \{C_1, \dots, C_r\}$  is the set of clusters and  $V_c$  is the set of nodes in the  $c^{th}$  cluster.

Modularity measures the deviation ratio of the within community edges in the network from the expected value of the same quantity in the network with the same community partitions but random connections between the vertices [11]. However, there are several drawbacks with modularity based clustering, such as computational cost, null model selection, preference for bigger clusters and the possibility to converge to a sub-optimal solution [20, 21].

### 1.3 Background on Tensors

Tensors are multi-dimensional arrays that can be used to provide a compact representation for higher-order data and preserve the multi-linear relationships in data. Different approaches have been developed to summarize and extract meaningful structure from high-order data using tensors. The advantages of using tensor analysis over two-dimensional analysis for high-order data have been discussed in great detail in [22–24].

**Tensor matricization:** A  $J$ -order tensor is denoted as  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_J}$  where  $x_{i_1, i_2, \dots, i_J}$  corresponds to the  $(i_1, i_2, \dots, i_J)^{th}$  element in the tensor,  $\mathcal{X}$ . A  $J$ -order tensor can be transformed into a matrix by reordering its elements in a process called matricization or unfolding. The mode- $j$  matricization of a tensor,  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_J}$ , is denoted as  $\mathbf{X}_{(j)}$ . A tensor element  $(i_1, i_2, \dots, i_J)$  is mapped to a matrix element  $(i_j, k)$ , where

$$k = 1 + \sum_{l=1, l \neq j}^J (i_l - 1)K_l \quad \text{with} \quad K_l = \prod_{q=1, q \neq j}^{l-1} I_q. \quad (1.6)$$

**$j$ -mode product:** The  $j$ -mode product of a tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_j \times \dots \times I_J}$  and a matrix  $\mathbf{U} \in \mathbb{R}^{K \times I_j}$  is denoted as  $\mathcal{X} \times_j \mathbf{U}$  and has a size of  $I_1 \times \dots \times I_{j-1} \times K \times I_{j+1} \times \dots \times I_J$  with its elements defined as:

$$(\mathcal{X} \times_j \mathbf{U})_{i_1 i_2 \dots i_{j-1} k i_{j+1} \dots i_J} = \sum_{i_j=1}^{I_j} x_{i_1 \dots i_j} u_{ki_j}, \quad (1.7)$$

which can be expressed in terms of unfolded tensors as:

$$\mathcal{C} = \mathcal{X} \times_j \mathbf{U} \quad \Leftrightarrow \quad \mathbf{C}_{(j)} = \mathbf{U} \mathbf{X}_{(j)} \quad (1.8)$$

**The  $j$ -Rank:** The rank of a  $j$ -mode tensor,  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_J}$ , is denoted as  $\text{rank}_j(\mathcal{X})$  and defined as the column rank of  $\mathbf{X}_{(j)}$ . The  $n$ -rank of  $\mathcal{X}$  is then defined as the collection of ranks of mode matrices  $\mathbf{X}_{(j)}$  and can be expressed as:

$$\text{rank}_j(\mathcal{X}) = \{\text{rank}(\mathbf{X}_{(1)}), \text{rank}(\mathbf{X}_{(2)}), \dots, \text{rank}(\mathbf{X}_{(j)})\}, \quad (1.9)$$

where  $j = 1, \dots, J$ .

**Tensor factorization:** Matrix factorization methods such as singular value decomposition (SVD) and principle component analysis (PCA) have been extended to tensors through Parallel Factor for Cross Products (PARAFAC) and Tucker decomposition [25]. Tucker decomposition provides orthogonal subspace information along each mode of the tensor, where the tensor is decomposed into a core tensor multiplied by an orthogonal component matrix along each mode. A  $J$  mode tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_J}$ , can be written using the Tucker model as follows:

$$\mathcal{X} = \mathcal{Y} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \cdots \times_J \mathbf{U}_{(J)}, \quad (1.10)$$

where  $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_J}$  is the core tensor,  $\mathbf{U}_{(j)} \in \mathbb{R}^{I_j \times I_j}$  for  $j = 1, 2, \dots, J$  is the orthogonal component matrix along the  $j$ th mode.

Tucker decomposition is commonly implemented through high-order SVD (HOSVD) [26] or higher order orthogonal iteration (HOOI) [27]. HOSVD performs SVD on the unfolded matrix,  $\mathbf{X}_{(j)}$ , along each mode of the tensor independently [28]. On the other hand, HOOI performs alternating optimization to find the best projection matrices.

HOOI is an iterative algorithm used to compute the low rank approximation of tensors. For a  $J$  mode tensor  $\mathcal{X}$ , let  $r_1, r_2, \dots, r_J$  be a set of integers satisfying  $r_j \leq I_j$ , for  $j = 1, \dots, J$ . The best rank- $(r_1, r_2, \dots, r_J)$  approximation of the tensor  $\mathcal{X}$  is achieved by finding a set of  $\mathbf{U}_{(j)} \in \mathbb{R}^{I_j \times r_j}$  matrices with orthonormal columns for  $j = 1, 2, \dots, J$ , and a core tensor  $\mathcal{Y} \in \mathbb{R}^{r_1 \times r_2 \times \dots \times r_J}$  that satisfies the following least squares optimization problem,

$$\min_{\mathbf{U}_{(1)}, \mathbf{U}_{(2)}, \dots, \mathbf{U}_{(J)}, \mathcal{Y}} \|\mathcal{X} - \mathcal{Y} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \cdots \times_J \mathbf{U}_{(J)}\|_F^2. \quad (1.11)$$

The optimal solution of Eq. (1.11) is given by,

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}_{(1)}^T \times_2 \mathbf{U}_{(2)}^T \cdots \times_J \mathbf{U}_{(J)}^T, \quad (1.12)$$

with  $\mathbf{U}_{(j)}^T \mathbf{U}_{(j)} = \mathbf{I}$ , where  $\mathbf{I} \in \mathbb{R}^{r_j \times r_j}$  is the identity matrix. The HOOI relies on alternating least squares approach to solve the best rank- $(r_1, r_2, \dots, r_J)$  approximation problem, and solves a set of restricted optimization problems successively as follows,

$$\min_{\mathbf{U}_{(p)}} \|\mathcal{X} - \mathcal{Y} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \cdots \times_J \mathbf{U}_{(J)}\|_F^2, \quad (1.13)$$

where the optimization is performed over the  $j^{th}$  matrix  $\mathbf{U}_{(j)}$  with the latest optimized values of the other  $\mathbf{U}_{(p)}$ 's substituted in (1.11) and (1.12), where  $j \neq p$ . A more detailed description can be found in [27].

## 1.4 Community Detection in Dynamic Networks

The community structure of most real world systems varies across time. For instance, in social networks, the community structure can change with time as people leave or join different communities [29, 30]. As a result, evolutionary clustering approaches have been implemented to detect the community structure in this type of networks.

Evolutionary clustering provides a framework to cluster the network at each time point, such that it preserves the network structure at the current time and the cluster assignments change smoothly across time. Over the past decade, different evolutionary clustering approaches have been developed to detect the community structure of dynamic networks. These approaches include simple extensions of static clustering methods [31, 32], statistical modeling based methods [33], tensor based methods [34] and modularity based methods [35].

Extensions of static clustering methods such as k-means and agglomerative hierarchical clustering approaches have been developed in [31] for dynamic networks. A cost function that considers both the cluster membership at the current time and the deviation from the previous time point was introduced. In k-means evolutionary clustering, this deviation is quantified through a distance measure between centroids of clusters from consecutive points. However, this distance measure is sensitive to variation in the historic centroids, since it depends on matching the current centroid with its peer from recent history. Similarly, in evolutionary hierarchical clustering, the merging step of the tree nodes at the current time is affected by the historic community assignment. In [32], the k-means evolutionary clustering approach was extended using two spectral clustering based frameworks: preserving cluster quality (PCQ) and preserving cluster membership (PCM). In both frameworks, the clustering depends on a cost function to guarantee temporal smoothness. In PCQ, the temporal smoothness is defined as the cost of applying the current clustering results to the

previous data point, consequently the current clustering results are rejected if they disagree with recent history. On the other hand, PCM considers the projection distance between the current and previous clustering results, which is not a proper metric [36], to measure the difference between the current and historic clusterings. Moreover, the suggested temporal cost function requires a *priori* knowledge about the changes in the community structure and depends on the choice of a changing parameter.

Under statistical modeling based methods, authors in [33] assumed a statistical model for the evolution of the adjacency matrix and an adaptive forgetting factor for evolutionary clustering and tracking (AFFECT) was introduced. In AFFECT, static clustering methods were performed after smoothing the proximity between objects over time. However, AFFECT depends on a stochastic block model for the adjacency matrix and has higher computational complexity compared to static clustering to estimate the forgetting factor.

Under modularity based methods, a statistical method based on greedy modularity optimization has been proposed in [35] to identify the changing modular organization in temporal multiplex networks. Yet this approach has some drawbacks as it depends on modularity optimization as discussed in [20].

Under tensor based methods, a non-negative tensor factorization approach was introduced in [34]. In this approach, the community structure of the temporal network is detected using PARAFAC, where all the successive adjacency matrices in the temporal network were represented by a single 3-way tensor. However, this representation of the temporal network may not reflect the community structure of the network at different time points.

## 1.5 Community Detection in Dynamic Multi-layer Networks

Most real world complex systems cannot be simply represented by static networks. In many disciplines, the systems are represented by dynamic networks [5] that evolve over time, and the community structure has to be detected across time. As a result, different evolutionary clustering approaches have been introduced, as discussed in Section 1.4. Moreover, many of these systems can be comprised of multiple types of relationships, and can be modeled as multi-relational or multi-layer networks [6]. In multi-layer networks, the entities or the nodes in the network exhibit different types of relationships, i.e. within layer and across-layers relationships, and these relationships should be taken into account to understand the correct underlying structure of these networks [37]. For instance, social networks can be modeled as multi-layer networks, where the nodes correspond to the same group of individuals across all layers while the edges within each layer correspond to intra-layer relationships and the edges across layers correspond to inter-layer relationships [37–39]. Another example is air transportation network [40], where the nodes correspond to the airports and the edges correspond to the direct flights between them. Each layer then corresponds to the different airlines. More recently, the need to combine the concepts of dynamic networks and multi-layer networks has increased, resulting in a new representation for complex systems as dynamic multi-layer networks. Dynamic multi-layer networks provide a complete representation for systems with various types of relationships that evolve across time. The aforementioned social network example can be remodeled as a dynamic multi-layer network when considering the changes in the relationships between individuals across time.

Over the past years, few methods have been developed to detect the community structure in dynamic multi-layer networks. In [35], the authors introduced a general framework that encompasses time-evolving, multi-scale and multi-layer networks to determine the community structure

in multi-slice networks. These networks include a set of adjacency matrices that represent the variation in connectivity between nodes across time, or variation in the types of relationships across layers, or variation of the same network across multiple scales. The framework relies on a generalization of the modularity optimization problem for detecting the community structure of a single network. Since this framework depends on modularity, it has multiple drawbacks including the computational cost, null model selection and the possibility of the convergence to a non-optimal solution as discussed in [20]. Moreover, this approach only considers the inter-layer edges between each node and itself and ignores the inter-layer edges between different nodes across layers. Another evolutionary clustering approach to mine and track dynamic multi-layer networks is introduced in [41]. The proposed framework relies on a multi-objective optimization representation and the solution is obtained by a genetic algorithm. This approach has a high computational cost and is impractical in the case of detecting the community structure for large networks. Furthermore, this approach does not consider any inter-layer edges between nodes across layers.

In this thesis, we address the problem of community detection in dynamic multi-layer networks. In particular, we focus on analyzing brain functional connectivity by modeling it as a dynamic network across multiple subjects. This representation allows a comprehensive understanding of the dynamic behavior of the brain functional connectivity and its variation across subjects.

## **1.6 Dynamic Functional Connectivity Networks of the Brain**

With the advancement of neuroimaging technology, it is now possible to collect neuronal data from the human brain across different modalities, spatial and temporal resolutions, such as electroencephalography (EEG) and fMRI data [42–44]. One of the major ways of analyzing this multidimensional neurophysiological data has been to construct functional connectivity networks of

the brain. Functional connectivity (FC) is defined as the statistical dependency between spatially distributed brain regions [45–47].

With the development of complex network theory, it is now possible to map human brain activity as a complex network [48–50]. In particular, the nodes of the network correspond to the different brain regions and the edges are quantified through the interactions between these regions. The possibility of modeling the brain as a network opened the door for researchers to investigate brain behavior during task related events and resting state. Early work on functional connectivity network (FCN) analysis focuses on static networks. However, recent studies have shown that FCNs exhibit a dynamic behavior [51–53].

In recent years, understanding the spontaneous activity of the brain during resting state attracted lots of attention. Consequently, different approaches focused on understanding the functional mechanisms of human brain during resting state. This was accomplished by investigating the functional connectivity of the brain networks from fMRI during resting state. FC between different brain regions has been quantified through the computation of Pearson’s correlation between regions of interest (ROIs) [54–56]. Increased interest in studying resting state fMRI (rs-fMRI) has shown that the different brain regions of the motor cortex are active even in the absence of a motor task [57]. Many of the current studies on rs-fMRI reported a set of networks that appear consistently in healthy subjects such as default mode network, visual network, somatomotor network, executive control network and auditory network [58, 59].

Early work on FCN analysis relied on static network representation. However, recent studies revealed evidence for dynamic nature of FCNs [51–53], [60–62]. Different methods have been developed to study the dynamics of the functional connectivity networks (dFCNs). These methods include clustering based methods [55], [63–67], statistical state modeling methods [68, 69] and subspace analysis based methods [56], [70–72], [73].

Under clustering based methods, different algorithms have been developed assuming that the network is at one distinct state at each time point, and the states reoccur across time. In [55], an approach based on k-means clustering of dFCNs across time and subjects is introduced to identify "FC states" from rs-fMRI data. Similarly, the authors in [63–66] used k-means clustering to obtain "synchrostates" from a time-frequency dependent phase difference data. Another clustering based approach is introduced in [67], where a codebook was obtained to facilitate the description of the "FC $\mu$ state" through Neural Gas algorithm. Under statistical state modeling, an approach to identify the network states through hierarchical clustering followed by a Hidden Markov Model (HMM) is introduced in [68]. Similarly, the authors in [69] obtained the network states and the transitions between them through independent vector analysis and Markov modeling. Under subspace modeling methods, an approach based on principal component analysis (PCA) is introduced in [56], where it is assumed that the network states are made up of multiple building blocks, or eigenconnectivities. A similar PCA based approach is adopted in [70] using EEG data to reconstruct the principal resting state dynamics. Most of the current work focuses on extracting FC states and reduces the dimension of the dFCNs by vectorizing the connectivity matrices before extracting the FC states. This procedure may result in the loss of the topological structure of the FCN. These approaches also reduce the dFCNs into a few common basis functions or states without explicitly identifying the topological organization within each state. Moreover, they do not necessarily identify the change points between these states. More recently, approaches that can first detect the change points in dFCN connectivity patterns and then summarize the network structure have been introduced. These include greedy algorithms using graphical approaches [74], Bayesian network modeling based models [75] and tensor based approaches [76], [73].

## 1.7 Organization and Contributions of this Thesis

In this thesis, we present novel approaches to both track and detect the community structure in dynamic multi-layer networks. A particular focus of this thesis is detecting and tracking the community structure in temporal social networks and dynamic multi-layer functional connectivity networks (dFCNs) of the brain.

In Chapter 2, we introduce a low-rank + sparse estimation based evolutionary spectral clustering approach to detect and track the community structure in temporal networks. The proposed method decomposes the network into low-rank and sparse parts and obtains smooth cluster assignments by minimizing the subspace distance between consecutive time points. The method tries to obtain smooth cluster assignments by minimizing the subspace chordal distance between consecutive time points. Furthermore, the proposed algorithm explicitly considers the changes in the dimensions of the consecutive subspaces over time by modifying the chordal distance. In addition, the introduced framework is robust to noise and outliers and can detect the community structure in both binary and weighted temporal networks efficiently. Therefore, unlike existing evolutionary clustering methods, the proposed algorithm does not require any prior knowledge or make assumptions about the community structure and does not assume any particular model for the network. The proposed approach is evaluated by applying it to multiple simulated and real social temporal networks.

In Chapter 3, we propose a tensor based approach using 3-way and 4-way tensors to detect the community structure of dynamic single-layer and multi-layer networks, respectively. In the dynamic single-layer network case, two approaches, windowed tensor approach (3D-WTA) and running time tensor approach (RTTA) are proposed. In 3D-WTA, a fixed length widow is used to construct the tensors that represent the network at each time point, while in RTTA, a variable

length window is used to construct the tensors. The proposed framework takes the history of the networks over time into account and optimizes the contribution of each time point in the network. Moreover, a normalized cost function is introduced to track the changes in the community structure over time. The proposed approach is applied to synthetic and real networks including social networks and dFCNs. In particular, it is applied to EEG data collected during a study of error-related negativity (ERN) in order to understand the dynamic behavior of the FCN during cognitive control. For dynamic multi-layer networks, 3D-WTA is extended for temporal multi-layer networks, namely 4D-WTA, to identify and track the community structure across time and layers, simultaneously. The proposed algorithm is applied to study the temporal evolution of communities in FCNs constructed across different regions of interests (ROIs) in the brain from rs-fMRI. In 4D-WTA, the contribution of each subject and time point within the framework is optimized. Moreover, the dFCNs are summarized into a set of FC states defined through their community structure that repeat over time and subjects. The detected community structures are evaluated through a consistency measure and the dynamic behavior of the FCNs is compared to the results from prior work.

In Chapter 4, we address the issue of community detection in dynamic networks by considering the ability to reduce the network and aggregate the time points that are topologically similar to achieve more compact and efficient community detection. The proposed approach consists of two steps. The first step relies on an information-theoretic function to aggregate the dynamic networks and identify the time points that possess similar topological structure to combine them into a tensor. In the second step, a tensor based spectral clustering approach similar to the one proposed in Chapter 3 is applied to identify the community structure of the constructed tensors. The detected community structure is summarized as a set of connectivity patterns or states that reoccur across time. A particular application of this approach is the analysis of resting state FCNs

of the brain from fMRI signals. A set of FC states is extracted from the FCNs and evaluated by comparing it to the findings of previous studies.

In Chapter 5, we propose a temporal block spectral clustering framework to detect and track the community structure of multi-layer temporal networks. The main contribution of the proposed approach is the construction of data-driven within and across layer adjacency matrices. The proposed approach considers all possible edges between nodes within and across layers to construct intra- and inter- layers adjacency matrices unlike existing methods that ignores inter-layer edges or considers inter-layer edges between the node and itself only, i.e. multiplex networks. In particular, a sliding window correlation approach is used to model the pair-wise temporal relationships between all nodes within and across time windows. A set of intra- and inter-adjacency matrices are constructed and combined to create a set of temporal supra-adjacency matrices. The community structure is then determined by applying spectral clustering to these supra-adjacency matrices. The proposed algorithm is evaluated on simulated networks and dFCNs from rs-fMRI across time and subjects and compared to results reported from previous studies.

Finally, Chapter 6 summarizes the contributions of this thesis and suggests future work to improve current approaches.

## **Chapter 2**

### **Detecting and Tracking Community**

### **Structure in Temporal Networks: A**

### **Low-Rank + Sparse Estimation Based**

### **Evolutionary Clustering Approach**

#### **2.1 Introduction**

Many real world systems and relational data can be modeled as networks or graphs, where the system's objects and the interactions between them are represented by the vertices and edges of a graph, respectively. With the availability of large amounts of network data, it is important to be able to reduce the network's dimensionality and extract useful information from it. A key approach to network data reduction is community detection [77]. Early work in graph based community detection methods has focused on static networks [2] [7]. This type of networks is usually considered as an oversimplification as many real world complex systems exhibit variation in their community structure over time. Consequently, there is a growing need to develop algorithms that detect the

community structure and track its evolution across time [5].

Over the past decade, various methods have been proposed to detect the community structure in temporal networks. Evolutionary clustering approaches provide a framework to cluster the network over time, such that the network structure at each time point is preserved and the cluster assignments change smoothly across time. These approaches include extensions of static clustering methods [31] [32], statistical modeling based methods [33], tensor based methods [34] [78], modularity based methods [35] and multi-objective optimization genetic algorithms [79]. Extensions of static clustering methods to temporal networks include k-means and agglomerative hierarchical clustering approaches [31]. In [32], k-means evolutionary clustering approach was extended to two spectral evolutionary clustering based frameworks: preserving cluster quality (PCQ) and preserving cluster membership (PCM). The clustering in both frameworks depends on a cost function to guarantee temporal smoothness. However, this cost function requires a *priori* knowledge about the community structure of the underlying network. For statistical modeling algorithms, a statistical model for the evolution of the adjacency matrix was assumed [33] and an adaptive forgetting factor for evolutionary clustering and tracking (AFTECT) was introduced. In AFTECT, the proximity between objects was smoothed over time followed by a static clustering method. AFTECT assumes a stochastic block model for the adjacency matrix and has higher computational complexity compared to static clustering due to the estimation of the forgetting factor. Under tensor based methods, non-negative PARAFAC tensor factorization community detection approaches were introduced in [34] and [78] where the temporal network is represented by a single 3-way tensor. In [78], PARAFAC decomposition and k-means are used to learn the network generative models and detect the clusters, respectively. However, this framework requires a *priori* knowledge about the ground truth clusters in the temporal network, including the number of the clusters, the nodes that belong to each cluster and the presence of the clusters across time, to determine the significant

clusters and evaluate them. Modularity based methods are based on greedy modularity optimization to identify the changing modular organization [35]. This approach has some drawbacks such as computational cost, null model selection, preference for bigger clusters and the possibility to converge to a sub-optimal solution as discussed in [20]. Multi-objective optimization genetic algorithms consider a multi-objective approach (DYNMOGA) [79] that consist of two objectives: Snapshot cost and temporal cost. The snapshot cost relies on computing four quality measures, including modularity, conductance, normalized cut and community score. On the other hand, the temporal cost relies on minimizing the normalized mutual information between consecutive partitions. However, DYNMOGA has high computational complexity that increases with the number of generations. Moreover, DYNMOGA performs well in binary networks whereas its performance declines for weighted networks, especially when the networks are fully connected.

Although a variety of methods have been developed to detect the community structure in temporal networks, there are still some problems and limitations associated with these methods. First, some of the existing approaches are limited to binary networks [34]. Second, some of the current algorithms require a *priori* knowledge or make assumptions about the community structure or the adjacency matrix [33] [35] [78]. Third, some of the existing algorithms such as [79] have a high computational complexity. Finally, many of the existing algorithms are not robust to outliers which leads to the detection of inaccurate community structure. Consequently, there is a need to develop new approaches to overcome these shortcomings.

In this chapter, a low-rank + sparse estimation based evolutionary spectral clustering approach is proposed to detect and track the community structure in temporal weighted and binary networks. The major advantages of the proposed approach over existing methods are five fold. First, the proposed method can detect and track the community structure in both weighted and binary temporal networks without any adjustments to the optimization problem. Second, the extraction of the low-

rank part of the adjacency matrix at each time point removes the sparse noise and outliers in the network which improves the robustness of the method. Third, minimizing the subspace distance between consecutive time points results in smooth cluster assignments across time, i.e. preserves the community structure at the current time point and guarantees that the structure is evolving smoothly across time. Fourth, a cost function defined through subspace distance measure provides a way to track the changes in the community structure across time. Finally, the proposed approach does not require *a priori* knowledge or make any assumptions about the community structure unlike some existing methods.

## 2.2 Background

### 2.2.1 Spectral Clustering

A popular approach for detecting the community structure in static networks is spectral clustering, which maps the relationship between the nodes in a graph to a lower dimensional subspace. Given a network with adjacency matrix,  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , spectral clustering solves the following trace minimization problem [7],

$$\min_{\mathbf{U} \in \mathbb{R}^{n \times r}} Tr(\mathbf{U}^\dagger \mathbf{\Phi} \mathbf{U}), \quad s.t. \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}_r, \quad (2.1)$$

where  $\dagger$  is the transpose operator and  $\mathbf{\Phi} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$  is the normalized Laplacian matrix with  $\mathbf{D}$  as the diagonal degree matrix [19].  $\mathbf{\Phi}$  is always a positive semi-definite matrix for undirected graphs. The solution to the optimization problem is the matrix  $\mathbf{U}$  containing the  $r$  eigenvectors that correspond to the smallest  $r$  eigenvalues of  $\mathbf{\Phi}$ . The community structure is then determined by applying  $k$ -means to the matrix  $\mathbf{U}$  [7]. The number of the eigenvectors,  $r$ , refers to the number of

clusters in the network [7].

### 2.2.2 Low-rank + Sparse matrix approximation

Different methods have been proposed to find a low-rank approximation for a noise corrupted matrix  $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ . For example, principal component analysis (PCA) finds a low rank approximation by solving the following optimization problem [80], [81], [82]:

$$\min_{\mathbf{L} \in \mathbb{R}^{n_1 \times n_2}} \|\mathbf{A} - \mathbf{L}\|_F^2, \quad s.t. \quad \text{rank}(\mathbf{L}) \leq r, \quad (2.2)$$

where  $\mathbf{L} \in \mathbb{R}^{n_1 \times n_2}$  is the low rank approximation of  $\mathbf{A}$ .

The solution to this problem is given in terms of singular value decomposition (SVD) of the matrix  $\mathbf{A}$ . Let  $\mathbf{A} = \mathbf{Q}_1 \mathbf{\Sigma} \mathbf{Q}_2^\dagger$  be the SVD of  $\mathbf{A}$ , where  $\mathbf{Q}_1 \in \mathbb{R}^{n_1 \times n_1}$  and  $\mathbf{Q}_2 \in \mathbb{R}^{n_2 \times n_2}$  are the left and right singular vectors matrices, respectively and  $\mathbf{\Sigma} \in \mathbb{R}^{n_1 \times n_2}$  is a diagonal matrix with the singular values on its diagonal. The rank- $r$  matrix  $\mathbf{L}$  is given as  $\mathbf{L} = \tilde{\mathbf{Q}}_1 \tilde{\mathbf{\Sigma}} \tilde{\mathbf{Q}}_2^\dagger$ , where  $\tilde{\mathbf{Q}}_1 \in \mathbb{R}^{n_1 \times r}$ ,  $\tilde{\mathbf{Q}}_2 \in \mathbb{R}^{n_2 \times r}$  and  $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{r \times r}$  are the top  $r$  singular vectors and singular values matrices, respectively. Even though PCA is the most commonly used technique for identifying a low-rank approximation of a corrupted observations matrix, it is known that its performance declines when the corruptions are non-Gaussian.

In order to address this issue, robust PCA (RPCA) has been introduced to recover the low-rank approximation from data corrupted by non-Gaussian noise [83]. RPCA decomposes the matrix  $\mathbf{A}$  into a low-rank component,  $\mathbf{L}$ , and a sparse component,  $\mathbf{S}$ , by solving the following optimization problem:

$$\min_{\mathbf{L} \in \mathbb{R}^{n_1 \times n_2}, \mathbf{S} \in \mathbb{R}^{n_1 \times n_2}} \|\mathbf{L}\|_* + \mu \|\mathbf{S}\|_1, \quad s.t. \quad \mathbf{L} + \mathbf{S} = \mathbf{A}, \quad (2.3)$$

where  $\|\mathbf{L}\|_*$  is the nuclear norm of  $\mathbf{L}$ ,  $\|\mathbf{S}\|_1$  is the  $l_1$ -norm of  $\mathbf{S}$  and  $\mu > 0$  is the regularization

parameter. The unconstrained version of the problem in Eq. (2.3) is given by:

$$\min_{\mathbf{L} \in \mathbb{R}^{n_1 \times n_2}, \mathbf{S} \in \mathbb{R}^{n_1 \times n_2}} \|\mathbf{L}\|_* + \mu \|\mathbf{S}\|_1 + \frac{\gamma}{2} \|\mathbf{A} - \mathbf{L} - \mathbf{S}\|_F^2, \quad (2.4)$$

which can be solved efficiently by iterative SVD soft-thresholding algorithm [83].

### 2.2.3 Chordal Subspace Distance Measure

Let  $B$  be a subspace of dimension  $r_1$  in  $\mathbb{R}^n$ , and  $C$  another subspace of dimension  $r_2$  in  $\mathbb{R}^n$ , where  $r_1, r_2 \leq n$ . The chordal distance, which defines a metric on subspaces of different dimensions, is defined as [84]:

$$d_{Ch}^2(B, C) = |r_2 - r_1| + \sum_{i=1}^{\min\{r_1, r_2\}} (\sin \theta_i)^2, \quad (2.5)$$

where  $\theta_i$  represents the  $i^{th}$  principal angle between  $B$  and  $C$  which quantifies how close the two subspaces are geometrically.

### 2.2.4 Variation of Information

In order to evaluate the performance of a community detection algorithm, the degree of agreement between different partitions of the same network should be determined. Throughout this chapter, the variation of information metric introduced in [85] is used for this purpose. Variation of information (VI) between two partitions  $\bar{C} = (C_1, C_2, \dots, C_r)$  and  $\bar{C}' = (C'_1, C'_2, \dots, C'_{r'})$  with  $r$  and  $r'$  clusters, respectively is defined as:

$$VI(\bar{C}, \bar{C}') = H(\bar{C}|\bar{C}') + H(\bar{C}'|\bar{C}), \quad (2.6)$$

where  $H(\bar{C}|\bar{C}')$  and  $H(\bar{C}'|\bar{C})$  are conditional entropies. VI ranges between  $0 \leq VI \leq \log n$ , where  $n$  is the number of nodes in the network. A value of 0 means the partitions are exactly the same and as the VI value increases, the degree of agreement between partitions decreases.

## 2.3 Methods: low-rank + sparse estimation based evolutionary spectral clustering (LS-ESC)

### 2.3.1 Problem Formulation

Given noisy adjacency matrix observations,  $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$ , of a low-rank temporal network,  $\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}$ , the objective is to obtain temporally smooth community detection assignments from these noisy adjacency matrix observations. To accomplish this objective, we propose to decompose the corrupted adjacency matrix into low-rank,  $\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}$ , and sparse,  $\mathbf{S}^{(t)} \in \mathbb{R}^{n \times n}$ , components. The estimated low-rank adjacency matrix at each time point is then used to extract the corresponding subspace, i.e. cluster assignment, with the additional constraint that the communities evolve slowly over time. This results in the following optimization problem:

$$\begin{aligned} \min_{\substack{\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}, \mathbf{S}^{(t)} \in \mathbb{R}^{n \times n}, \\ \mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}}} \quad & \|\mathbf{L}^{(t)}\|_* + \mu \|\mathbf{S}^{(t)}\|_1 + \lambda_1 \text{Tr}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)} \mathbf{U}^{(t)}) + \lambda_2 d_{ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}), \\ \text{s.t.} \quad & \mathbf{A}^{(t)} = \mathbf{L}^{(t)} + \mathbf{S}^{(t)}, \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \mathbf{L}^{(t)} = \mathbf{L}^{(t)\dagger}, L^{(t)}_{ij} \geq 0, L^{(t)}_{ii} = 0. \end{aligned} \quad (2.7)$$

The different terms in problem in Eq. (2.7) achieve the following goals:

- The first two terms represent the RPCA problem. The constraints  $\mathbf{L}^{(t)} = \mathbf{L}^{(t)\dagger}$ ,  $L^{(t)}_{ij} \geq 0$ ,  $L^{(t)}_{ii} = 0$  are added to ensure the symmetry and nonnegativity of the low-rank adjacency matrix with zeros on the diagonal.

- The third term,  $Tr(\mathbf{U}^{(t)\dagger} \mathbf{\Phi}^{(t)} \mathbf{U}^{(t)})$ , with the constraint  $\mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}$ , represents the spectral clustering problem and is introduced to detect the community structure of the underlying low-rank network at time point  $t$ . The normalized Laplacian matrix,  $\mathbf{\Phi}^{(t)}$ , is defined in terms of the low-rank component as  $\mathbf{\Phi}^{(t)} = \mathbf{I} - \mathbf{D}^{(t)-0.5} \mathbf{L}^{(t)} \mathbf{D}^{(t)-0.5}$ .
- The last term,  $d_{Ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)})$ , is introduced to minimize the distance between the consecutive networks' subspaces at time points  $t$  and  $t - 1$ . By adding this term to the optimization problem in Eq. (2.7), the community structure of the temporal network is guaranteed to be evolving smoothly over time.
- $\mu, \lambda_1$  and  $\lambda_2 > 0$  are regularization parameters.

### 2.3.2 Distance Between Successive Subspaces

In order to define the distance between consecutive subspaces, let  $\mathbf{U}^{(t)}$  be the matrix whose columns form a basis for the  $r$ -dimensional subspace in  $\mathbb{R}^n$ , defined by the  $r$  eigenvectors corresponding to the smallest eigenvalues of the graph Laplacian at time point  $t$ . Similarly,  $\mathbf{U}^{(t-1)}$  can be defined as the basis spanning the  $r_1$ -dimensional subspace in  $\mathbb{R}^n$ , that contains the  $r_1$  eigenvectors corresponding to the smallest eigenvalues of the graph Laplacian at time point  $(t - 1)$ , i.e.  $\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}$  and  $\mathbf{U}^{(t-1)} \in \mathbb{R}^{n \times r_1}$ . One of our main objectives is to find the subspace spanned by  $\mathbf{U}^{(t)}$  that is close to the subspace spanned by  $\mathbf{U}^{(t-1)}$ , and at the same time conserves the nodes' connectivity at time point  $t$ . Using the definition of chordal distance in Eq. (2.5) and the fact that the columns of  $\mathbf{U}^{(t)}$  are orthonormal, we have three different cases.

In the first case, the dimensions of the subspaces corresponding to the consecutive time points are equal to each other, i.e.  $r = r_1$ . In the second case, the dimension of  $\mathbf{U}^{(t)}$  is smaller than the dimension of  $\mathbf{U}^{(t-1)}$ , i.e.  $r \leq r_1$ . In the third case, the dimension of  $\mathbf{U}^{(t)}$  is larger than the dimension

of  $\mathbf{U}^{(t-1)}$ , i.e.  $r > r_1$ . The chordal distance between the successive subspaces for the three cases can be simplified as follows.

For the first case:

$$\begin{aligned}
d_{Ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}) &= \sum_{i=1}^{r_1} (\sin \theta_i)^2 \\
&= \sum_{i=1}^{r_1} (1 - (\cos \theta_i)^2) \\
&= r_1 - \sum_{i=1}^{r_1} (\cos \theta_i)^2 \\
&= r_1 - \text{Trace}(\mathbf{U}^{(t)} \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}),
\end{aligned} \tag{2.8}$$

For the second case:

$$\begin{aligned}
d_{Ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}) &= r_1 - r + \sum_{i=1}^r (\sin \theta_i)^2 \\
&= r_1 - r + \sum_{i=1}^r (1 - (\cos \theta_i)^2) \\
&= r_1 - r + r - \sum_{i=1}^r (1 - (\cos \theta_i)^2) \\
&= r_1 - \sum_{i=1}^r (1 - (\cos \theta_i)^2) \\
&= r_1 - \text{Trace}(\mathbf{U}^{(t)} \mathbf{U}^{(t)\dagger} \hat{\mathbf{U}}^{(t-1)} \hat{\mathbf{U}}^{(t-1)\dagger}),
\end{aligned} \tag{2.9}$$

where  $\mathbf{U}^{(t-1)} = \begin{bmatrix} \hat{\mathbf{U}}^{(t-1)} & \bar{\mathbf{U}}^{(t-1)} \end{bmatrix}$ ,  $\hat{\mathbf{U}}^{(t-1)} \in \mathbb{R}^{n \times r}$  is the matrix that contains the first  $r$  eigenvectors of the Laplacian matrix at the previous time point, and  $\bar{\mathbf{U}}^{(t-1)} \in \mathbb{R}^{n \times (r_1 - r)}$  is the matrix that contains the last  $r_1 - r$  eigenvectors of the Laplacian matrix at the previous time point.

For the third case:

$$\begin{aligned}
d_{Ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}) &= r - r_1 + \sum_{i=1}^{r_1} (\sin \theta_i)^2 \\
&= r - r_1 + \sum_{i=1}^{r_1} (1 - (\cos \theta_i)^2) \\
&= r - r_1 + r_1 - \sum_{i=1}^{r_1} (1 - (\cos \theta_i)^2) \\
&= r - \sum_{i=1}^{r_1} (1 - (\cos \theta_i)^2) \\
&= r - \text{Trace}(\hat{\mathbf{U}}^{(t)} \hat{\mathbf{U}}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}),
\end{aligned} \tag{2.10}$$

where  $\mathbf{U}^{(t)} = [\hat{\mathbf{U}}^{(t)} \quad \bar{\mathbf{U}}^{(t)}]$ ,  $\hat{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times r_1}$  is the matrix that contains the first  $r_1$  eigenvectors of the Laplacian matrix at the current time point, and  $\bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}$  is the matrix that contains the last  $r - r_1$  eigenvectors of the Laplacian matrix at the current time point.

### 2.3.3 Problem Solution

The solution to the optimization problem in Eq. (2.7) can be obtained by alternating direction method of multipliers (ADMM) [86] combined with proximal algorithms [87] [88] due to the non-differentiability of the nuclear and  $l_1$  - norms. First, the problem in Eq. (2.7) is split by adding an auxiliary variable  $\mathbf{W}^{(t)}$  as follows:

$$\begin{aligned}
&\min_{\substack{\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}, \mathbf{S}^{(t)} \in \mathbb{R}^{n \times n}, \\ \mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}, \mathbf{W}^{(t)} \in \mathbb{R}^{n \times n}}} \|\mathbf{L}^{(t)}\|_* + \mu \|\mathbf{S}^{(t)}\|_1 + \lambda_1 \text{Tr}(\mathbf{U}^{(t)\dagger} (\mathbf{I} - \mathbf{D}^{(t)-0.5} \mathbf{W}^{(t)} \mathbf{D}^{(t)-0.5}) \mathbf{U}^{(t)}) \\
&\quad + \lambda_2 d_{Ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}), \\
&s.t \ \mathbf{A}^{(t)} = \mathbf{L}^{(t)} + \mathbf{S}^{(t)}, \mathbf{W}^{(t)} = \mathbf{L}^{(t)}, \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \mathbf{W}^{(t)} = \mathbf{W}^{(t)\dagger}, W^{(t)}_{ij} \geq 0, W^{(t)}_{ii} = 0.
\end{aligned} \tag{2.11}$$

Second, the augmented Lagrange multipliers (ALM) function is rewritten by adding additional quadratic terms to guarantee the attainment of the first two equality constraints:

$$\begin{aligned}
& \min_{\substack{\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}, \mathbf{S}^{(t)} \in \mathbb{R}^{n \times n}, \\ \mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}, \mathbf{W}^{(t)} \in \mathbb{R}^{n \times n}}} \|\mathbf{L}^{(t)}\|_* + \mu \|\mathbf{S}^{(t)}\|_1 + \lambda_1 \text{Tr}(\mathbf{U}^{(t)\dagger} (\mathbf{I} - \mathbf{D}^{(t)-0.5} \mathbf{W}^{(t)} \mathbf{D}^{(t)-0.5}) \mathbf{U}^{(t)}) \\
& + \lambda_2 d_{ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}) + \langle \mathbf{X}_1^k, \mathbf{A}^{(t)} - \mathbf{L}^{(t)} - \mathbf{S}^{(t)} \rangle \\
& + \frac{\gamma_1}{2} \|\mathbf{A}^{(t)} - \mathbf{L}^{(t)} - \mathbf{S}^{(t)}\|_F^2 + \langle \mathbf{X}_2^k, \mathbf{W}^{(t)} - \mathbf{L}^{(t)} \rangle + \frac{\gamma_2}{2} \|\mathbf{W}^{(t)} - \mathbf{L}^{(t)}\|_F^2, \\
& s.t. \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \mathbf{W}^{(t)} = \mathbf{W}^{(t)\dagger}, W^{(t)}_{ij} \geq 0, W^{(t)}_{ii} = 0,
\end{aligned} \tag{2.12}$$

with Lagrange multipliers (dual variables) at the  $k^{th}$  iteration defined as,

$$\mathbf{X}_1^{k+1} = \mathbf{X}_1^k + \gamma_1 (\mathbf{A}^{(t)} - \mathbf{L}^{(t)k+1} - \mathbf{S}^{(t)k+1}), \tag{2.13}$$

$$\mathbf{X}_2^{k+1} = \mathbf{X}_2^k + \gamma_2 (\mathbf{W}^{(t)k+1} - \mathbf{L}^{(t)k+1}). \tag{2.14}$$

The problem in Eq. (2.12) can be solved at each time point,  $t$ , by following the steps in Algorithm 2.1 where the primal variables  $\mathbf{L}^{(t)}, \mathbf{S}^{(t)}, \mathbf{W}^{(t)}$  and  $\mathbf{U}^{(t)}$  are updated iteratively as follows:

$$\mathbf{L}^{(t)k+1} = \text{prox}_{\frac{1}{\gamma_1 + \gamma_2} \|\mathbf{L}^{(t)}\|_*} \left( \frac{\gamma_1 \mathbf{Y}_1^k + \gamma_2 \mathbf{Y}_2^k}{\gamma_1 + \gamma_2} \right), \tag{2.15}$$

where  $\mathbf{Y}_1^k = \mathbf{A}^{(t)} - \mathbf{S}^{(t)k} + \frac{\mathbf{X}_1^k}{\gamma_1}$ ,  $\mathbf{Y}_2^k = \mathbf{W}^{(t)k} + \frac{\mathbf{X}_2^k}{\gamma_2}$  and  $\text{prox}_f(\mathbf{Y}) = \arg\min_{\mathbf{L} \in \mathbb{R}^{n \times n}} f(\mathbf{L}) + \frac{1}{2} \|\mathbf{L} - \mathbf{Y}\|_F^2$  is the proximity operator of the convex function  $f$  where  $f(\mathbf{L})$  in this case is defined as  $f(\mathbf{L}) = \|\mathbf{L}^{(t)}\|_*$ . Details are included in the Appendix.

Similarly,

$$\mathbf{S}^{(t)k+1} = \text{prox}_{\frac{\mu}{\gamma_1} \|\mathbf{S}^{(t)}\|_1} \left( \mathbf{A}^{(t)} - \mathbf{L}^{(t)k+1} + \frac{\mathbf{X}_1^k}{\gamma_1} \right). \tag{2.16}$$

with  $f(\mathbf{S})$  defined as  $f(\mathbf{S}) = \|\mathbf{S}^{(t)}\|_1$  as explained in the Appendix.

The update of  $\mathbf{W}^{(t)}$  is performed by computing the closed form solution as defined in Eq. (2.17) followed by the spectral projected gradient (SPG) [89] method as explained in the Appendix to fulfill all the constraints:

$$\mathbf{W}^{(t)k+1} = \mathbf{L}^{(t)k+1} - \frac{\mathbf{X}_2^k}{\gamma_2} + \frac{\lambda_1}{\gamma_2} (\mathbf{D}^{(t)k})^{-0.5} \mathbf{U}^{(t)k} \mathbf{U}^{(t)k\dagger} (\mathbf{D}^{(t)k})^{-0.5} \tag{2.17}$$

Finally, the update of  $\mathbf{U}^{(t)}$  requires the solution of multiple optimization subproblems depending on the dimension of the consecutive subspaces. For the first two cases defined in Section 2.3.2,  $\mathbf{U}^{(t)}$  is updated by solving:

$$\min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \lambda_1 \text{Tr}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}_{mod}^{(t)k+1} \mathbf{U}^{(t)}), \quad s.t. \quad \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \quad (2.18)$$

where  $\boldsymbol{\Phi}_{mod}^{(t)k+1} = \boldsymbol{\Phi}^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \hat{\mathbf{U}}^{(t-1)} \hat{\mathbf{U}}^{(t-1)\dagger}$ . For the last case,  $\mathbf{U}^{(t)}$  is updated by solving:

$$\begin{aligned} \min_{\hat{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times r_1}} \lambda_1 \text{Tr}(\hat{\mathbf{U}}^{(t)\dagger} \boldsymbol{\Phi}_{mod}^{(t)k+1} \hat{\mathbf{U}}^{(t)}) \quad s.t. \quad \hat{\mathbf{U}}^{(t)\dagger} \hat{\mathbf{U}}^{(t)} = \mathbf{I}, \\ \min_{\bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}} \lambda_1 \text{Tr}(\bar{\mathbf{U}}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)}), \\ s.t. \quad \bar{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} = \mathbf{I}, \quad \hat{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} = \mathbf{0}, \end{aligned} \quad (2.19)$$

where  $\boldsymbol{\Phi}_{mod}^{(t)k+1} = \boldsymbol{\Phi}^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}$  and  $\mathbf{U}^{(t)} = \begin{bmatrix} \hat{\mathbf{U}}^{(t)} & \bar{\mathbf{U}}^{(t)} \end{bmatrix}$ .

The derivations are included in the Appendix.

### 2.3.4 Tracking the community structure over time

In order to track changes in the community structure over time, a cost function is defined using the distance between consecutive subspaces as follows:

$$CostFunction = d_{ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}), \quad (2.20)$$

where this cost represents the minimum distance achieved between consecutive subspaces. The changes in the cost function reflect the changes in the community structure.

### 2.3.5 Detecting the number of clusters

During the application of LS-ESC, the number of clusters at each time point is determined using the asymptotical surprise (AS) metric [90]. After the iterations start at Step 10 in Algorithm 2.1,  $\mathbf{U}^{(t)k+1}$  is calculated using a range of eigenvectors, e.g. (2-20), followed by k-means clustering.

---

**Algorithm 2.1** LS-ESC

---

**Input:**  $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{U}^{(t-1)} \in \mathbb{R}^{n \times r_1}$ ,  $\mu, \lambda_1, \lambda_2, \varepsilon$ .

**Output:**  $\mathbf{L}^{(t)}$ ,  $\mathbf{S}^{(t)}$ ,  $\mathbf{U}^{(t)}$ , Clustering labels.

- 1:  $k \leftarrow 0$
  - 2: Initialize  $\mathbf{L}^{(t)} \leftarrow \mathbf{A}^{(t)}$ ,  $\mathbf{S}^{(t)} \leftarrow \text{zeros}(n, n)$ ,  $\mathbf{W}^{(t)} \leftarrow \mathbf{L}^{(t)}$ .
  - 3:  $\gamma_1 \leftarrow 1$ ,  $\gamma_2 \leftarrow 1$ .
  - 4:  $\mathbf{X}_1^k \leftarrow \mathbf{A}^{(t)} - \mathbf{L}^{(t)k} - \mathbf{S}^{(t)k}$ ,  $\mathbf{X}_2^k \leftarrow \mathbf{W}^{(t)k} - \mathbf{L}^{(t)k}$ . % Define dual variables
  - 5:  $\mathbf{P}_1^k = \|\mathbf{L}^{(t)k}\|_*$ ,  $\mathbf{P}_2^k = \mu \|\mathbf{S}^{(t)k}\|_1$ ,  $\mathbf{P}_3^k = \lambda_1 \text{Tr}(\mathbf{U}^{(t)k \dagger} \mathbf{\Phi}^{(t)k} \mathbf{U}^{(t)k})$ ,  $\mathbf{P}_4^k = \lambda_2 d_{Ch}^2(\mathbf{U}^{(t)k}, \mathbf{U}^{(t-1)})$ .  
% Define primal objectives
  - 6: **while**  $\frac{\|\mathbf{P}_1^{k+1} - \mathbf{P}_1^k\|_F^2}{\|\mathbf{P}_1^k\|_F^2} > \varepsilon$  and  $\frac{\|\mathbf{P}_2^{k+1} - \mathbf{P}_2^k\|_F^2}{\|\mathbf{P}_2^k\|_F^2} > \varepsilon$  and  $\frac{\|\mathbf{P}_3^{k+1} - \mathbf{P}_3^k\|_F^2}{\|\mathbf{P}_3^k\|_F^2} > \varepsilon$  and  $\frac{\|\mathbf{P}_4^{k+1} - \mathbf{P}_4^k\|_F^2}{\|\mathbf{P}_4^k\|_F^2} > \varepsilon$  and  $\frac{\|\mathbf{X}_1^{k+1} - \mathbf{X}_1^k\|_F^2}{\|\mathbf{X}_1^k\|_F^2} > \varepsilon$  and  $\frac{\|\mathbf{X}_2^{k+1} - \mathbf{X}_2^k\|_F^2}{\|\mathbf{X}_2^k\|_F^2} > \varepsilon$  **do**
    - 7: Update  $\mathbf{L}^{(t)k+1}$  using Eq. (3).
    - 8: Update  $\mathbf{S}^{(t)k+1}$  using Eq. (5).
    - 9: Update  $\mathbf{W}^{(t)k+1}$  using Eq. (9) and Algorithm ??.
    - 10: Update  $\mathbf{U}^{(t)k+1}$  by solving Eq. (14) or Eq. (24).
    - 11: **if**  $k = 2$  **then**
      - 12: Determine the number of clusters using Eq. (2.21).
    - 13: **end if**
    - 14: Update  $\mathbf{X}_1^{k+1}$  using Eq. (2.13).
    - 15: Update  $\mathbf{X}_2^{k+1}$  using Eq. (2.14).
    - 16: Update  $\mathbf{P}_1^{k+1}$ ,  $\mathbf{P}_2^{k+1}$ ,  $\mathbf{P}_3^{k+1}$  and  $\mathbf{P}_4^{k+1}$  using Step 5.
    - 17:  $k \leftarrow k + 1$ .
  - 18: **end while**
  - 19: Apply k-means on  $\mathbf{U}^{(t)}$  to obtain clustering labels.
-

The correct number of clusters at time point  $t$  is determined as the number that maximizes the AS metric [90] with respect to the underlying network and is defined as:

$$r : \underset{r}{\operatorname{argmax}} AS(\mathbf{A}^{(t)}, \textit{Clustering labels}), \quad (2.21)$$

where *Clustering labels* :  $\{C_1, C_2, \dots, C_r\}$ . The number of clusters is then fixed for time point  $t$  until the algorithm converges. This process is repeated at each time point to update the number of clusters across time.

## 2.4 Results

In order to validate the performance of the proposed algorithm in detecting and tracking the community structure of temporal networks, the algorithm is evaluated on a set of simulated and real binary and weighted temporal networks. The performance of the proposed LS-ESC is compared to the state-of-the-art graph-based community detection algorithms including; AFFECT [33], DYN-MOGA [79] and generalized Louvain for multi-layer modularity maximization (GenLov) [35]. The comparison is conducted in terms of detected number of clusters (DNOC) and accuracy of the detected community structure across time using VI metric.

### 2.4.1 Simulated temporal networks

#### 2.4.1.1 Simulated binary temporal networks

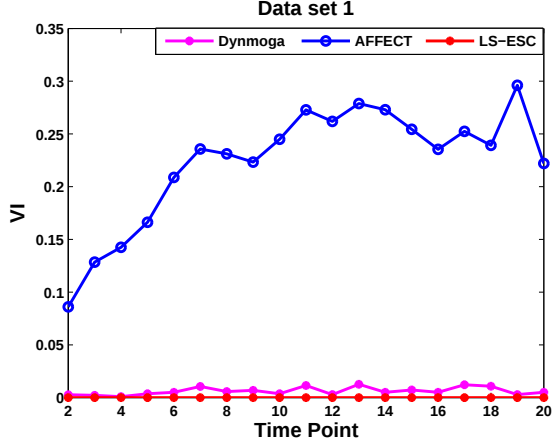
To evaluate the performance of the proposed algorithm in detecting the community structure of binary temporal networks, the benchmark networks proposed by Girvan and Newman in [91] and implemented in [92] are adopted. In this data set, the temporal network consists of 20 time points with 128 nodes divided into 4 clusters. Each node has a fixed average degree ( $AD$ ) and shares a number ( $z$ ) of edges with nodes from other communities. As  $AD$  increases, the network becomes

| Data set                                      | 1                | 2                | 3                | 4                | 5                | 6                |
|-----------------------------------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| $AD$                                          | 32               | 32               | 20               | 20               | 16               | 16               |
| $z$                                           | 3                | 5                | 2                | 1                | 3                | 3                |
| $NC\%$                                        | 10               | 10               | 20               | 30               | 20               | 30               |
| $\mu, \lambda_1, \frac{\lambda_2}{\lambda_1}$ | 0.3, 0.1,<br>0.3 | 0.4, 0.1,<br>0.3 | 0.5, 0.1,<br>0.1 | 0.5, 0.1,<br>0.1 | 0.5, 0.1,<br>0.1 | 0.6, 0.1,<br>0.1 |

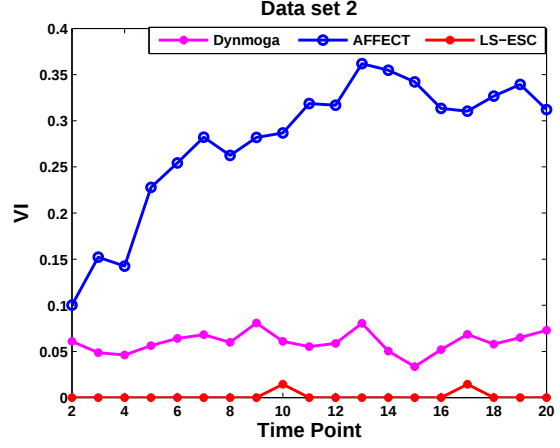
Table 2.1: Parameters of the simulated binary networks from Girvan and Newman benchmark for six different data sets and the selected values of the regularization parameters.

more dense and as  $z$  decreases, the clusters become more distinct. In order to impose nonstationarity to the community structure over time, a percentage of nodes ( $NC\%$ ) are moved randomly from their communities and assigned to other communities at each time step. A set of temporal networks are generated with different parameters to obtain different community structures as summarized in Table 2.1. The regularization parameters are determined empirically and reported in Table 2.1.

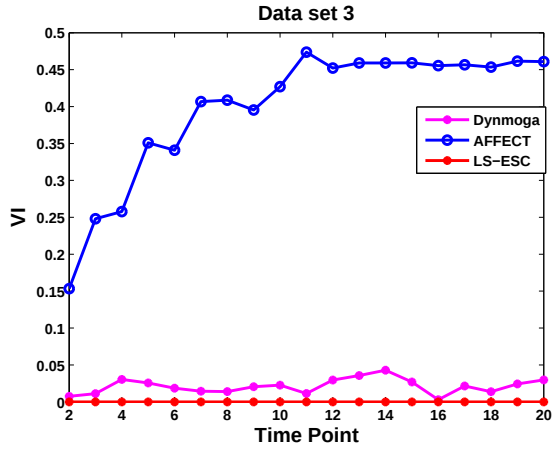
The performance of the proposed approach is compared to the state-of-the-art algorithms using VI metric in Fig. 2.1. The simulations are repeated 100 times and the VI is averaged over all simulations. As it can be seen in Fig. 2.1, the proposed algorithm performs better than AFFECT and DYNMOGA in detecting the community structure over time for the different networks. The performance of DYNMOGA is found to be comparable to the proposed algorithm for data sets 1 and 2 in Fig. 2.1a and Fig. 2.1d, respectively. This is due to the fact that In data set 1,  $AD$  is high and  $NC\%$  is low, i.e. the networks are densely connected and more stationary. On the other hand, in data set 4,  $z$  is low, i.e. the clusters are more distinct. However, the performance of DYNMOGA degrades as  $AD$  decreases and  $z$  and  $NC\%$  increases, i.e. in cases where the network is sparse, non-stationary and is less modular such as data set 5 and 6. It is found that AFFECT does not perform well in binary networks.



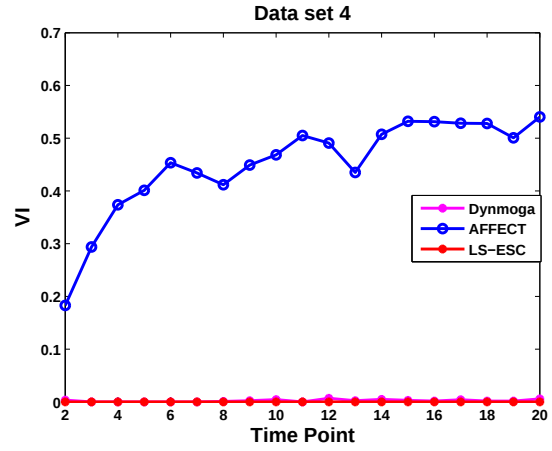
(a)



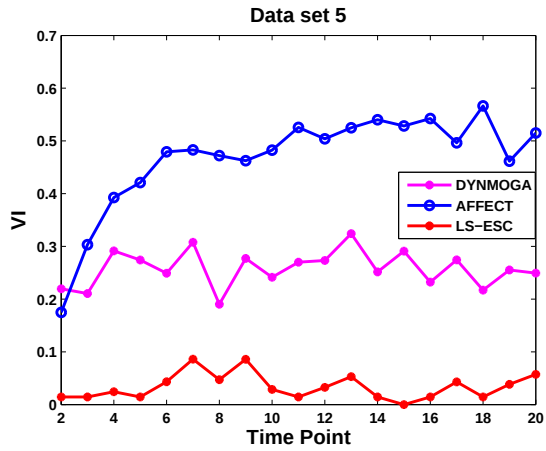
(b)



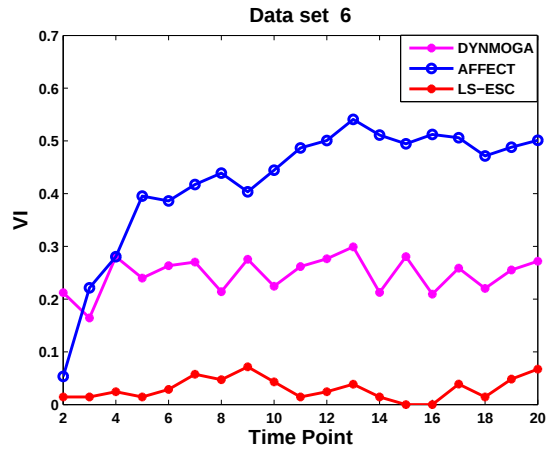
(c)



(d)



(e)



(f)

Figure 2.1: VI comparison between LS-ESC, AFFECT and DYNMOGA for simulated binary networks from Girvan and Newman benchmark with the parameters presented in Table 2.1. LS-ESC scored the lowest VI over time compared to the other methods: (a) Data set 1; (b) Data set 2; (c) Data set 3; (d) Data set 4; (e) Data set 5; and (f) Data set 6.

#### 2.4.1.2 Simulated weighted temporal networks

As many of the real-world networks are weighted, a set of weighted temporal networks are generated to evaluate the performance of the proposed approach in detecting and tracking the community structure over time. Temporal networks are generated for 60 time points with 100 nodes. Intra and inter-cluster edges are selected from a truncated Gaussian distribution in the range of  $[0, 1]$ . Different levels of sparse noise ( $SN\%$ ) are added to the inter-cluster edges. The parameters and the details of the community structure for these data sets are given in Table 3.1. The values of the regularization parameters are chosen empirically and reported in Table 3.1.

The performance of LS-ESC and the state-of-the-art algorithms in detecting the community structure for the weighted temporal networks in Table 3.1 is compared in terms of the DNOC and VI over time in Fig. 2.2. DNOC at each time point for LS-ESC and AFFECT is determined as the number that maximizes the asymptotical surprise metric. The simulations are repeated for 100 times and the cost function, DNOC and VI are averaged over all simulations.

As it can be seen in Fig. 2.2(a)-(c), the cost function can track the change points in the community structure over time and detects changes at time points 20 and 40. Fig. 2.2(d)-(f) show that the proposed approach performs better than the other methods in detecting the correct number of clusters over time. Finally, Fig. 2.2(g)-(i) present a comparison between the different methods using the VI metric. LS-ESC outperforms the other methods in terms of detecting and tracking the correct community structure over time. Moreover, LS-ESC has better performance around the change points and detects the correct community structure faster than the other methods. The performances of AFFECT and GenLov decline as the  $SN\%$  increases and the size of clusters decreases. In particular, both methods show preference to large clusters and tend to merge small clusters together as can be seen during the time intervals (40-60), (20-40) and (20-60) in data sets

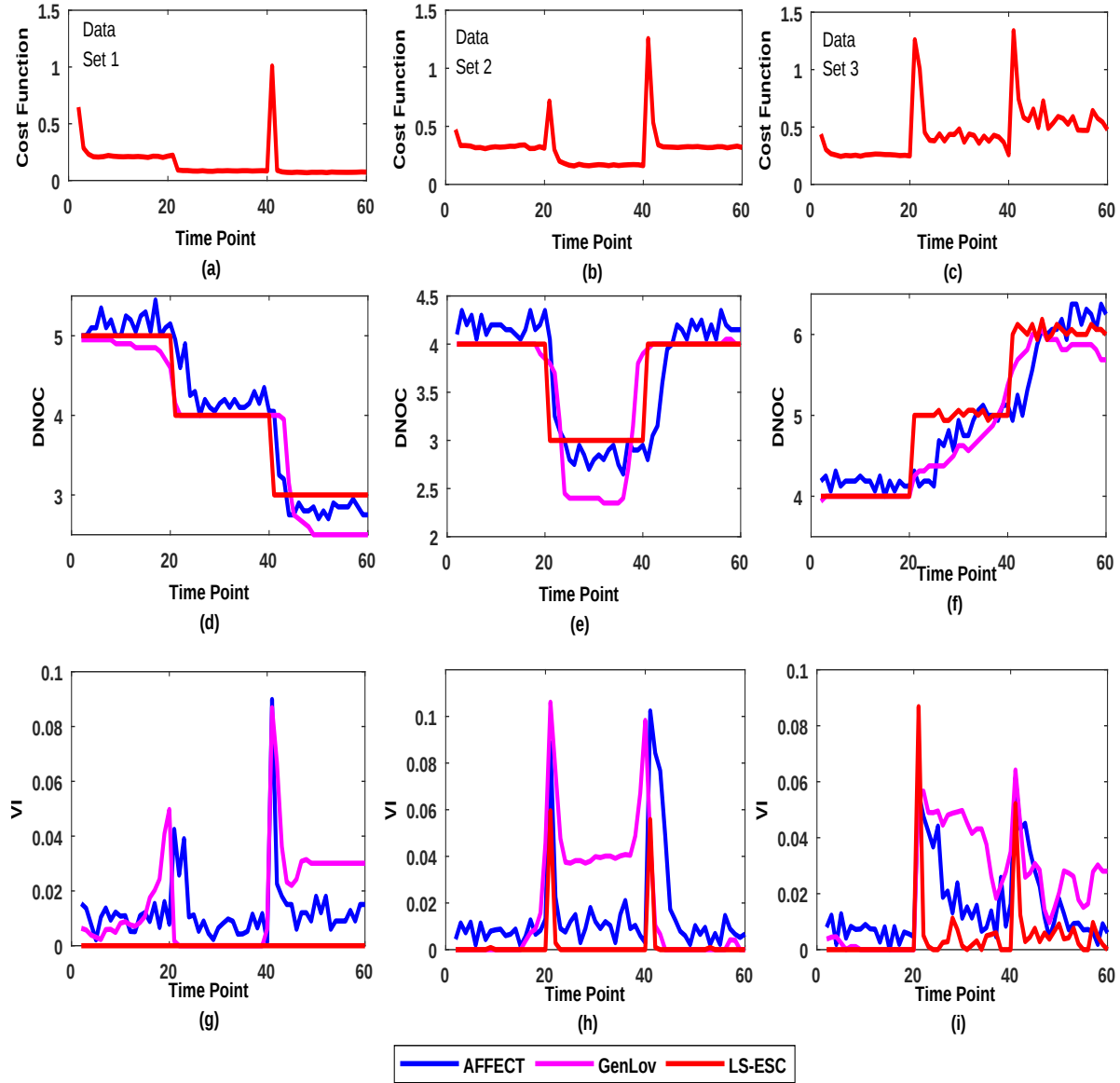


Figure 2.2: Comparison between LS-ESC and state of the art algorithms for the different weighted networks in Table 3.1: (a)-(c) The cost function calculated by LS-ESC, (d)-(f) The detected number of clusters (DNOC) for the simulated networks using different algorithms. LS-ESC estimated the correct number of clusters over time and the cost function reflected the correct change points; (g)-(i) Comparison of variation of information (VI) between LS-ESC, AFFECT and GenLov for the simulated networks. LS-ESC outperforms other methods and scored the lowest variation of information over time.

1, 2 and 3, respectively. The results for DYNMOGA are not presented in these figures as it does not perform well on weighted networks.

| Data Sets |                                                           | Time points 1-20                                                      | Time points 21-40                                                     | Time points 41-60                                                                 |
|-----------|-----------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 1         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.5, 0.3, 0.2, 0.1                                                    | 0.5, 0.2, 0.1, 0.1                                                    | 0.5, 0.3, 0.2, 0.1                                                                |
|           | Nodes in each cluster C                                   | $C_1(1-30), C_2(31-45),$<br>$C_3(46-60), C_4(61-80)$<br>$C_5(81-100)$ | $C_1(1-30), C_2(31-60),$<br>$C_3(61-80), C_4(81-100)$                 | $C_1(1-60), C_2(61-80),$<br>$C_3(81-100)$                                         |
|           | Number of clusters                                        | 5                                                                     | 4                                                                     | 3                                                                                 |
|           | $SN\%, \mu, \lambda_1, \frac{\lambda_2}{\lambda_1}$       | 10%, 0.3, 0.1, 0.1                                                    | 10%, 0.3, 0.1, 0.1                                                    | 10%, 0.3, 0.1, 0.1                                                                |
| 2         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.7, 0.3, 0.2, 0.1                                                    | 0.7, 0.2, 0.3, 0.2                                                    | 0.7, 0.3, 0.2, 0.1                                                                |
|           | Nodes in each cluster C                                   | $C_1(1-30), C_2(31-60),$<br>$C_3(61-80), C_4(81-100)$                 | $C_1(1-60), C_2(61-80),$<br>$C_3(81-100)$                             | $C_1(1-30), C_2(31-60),$<br>$C_3(61-80), C_4(81-100)$                             |
|           | Number of clusters                                        | 4                                                                     | 3                                                                     | 4                                                                                 |
|           | $SN\%, \mu, \lambda_1, \frac{\lambda_2}{\lambda_1}$       | 20%, 0.5, 0.1, 0.1                                                    | 20%, 0.5, 0.1, 0.1                                                    | 20%, 0.5, 0.1, 0.1                                                                |
| 3         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.8, 0.2, 0.3, 0.2                                                    | 0.6, 0.1, 0.2, 0.1                                                    | 0.8, 0.2, 0.3, 0.2                                                                |
|           | Nodes in each cluster C                                   | $C_1(1-30), C_2(31-60),$<br>$C_3(61-80), C_4(81-100)$                 | $C_1(1-30), C_2(31-45),$<br>$C_3(46-60), C_4(61-80)$<br>$C_5(81-100)$ | $C_1(1-15), C_2(16-30),$<br>$C_3(31-45), C_4(46-60)$<br>$C_5(61-80), C_6(81-100)$ |
|           | Number of clusters                                        | 4                                                                     | 5                                                                     | 6                                                                                 |
|           | $SN\%, \mu, \lambda_1, \frac{\lambda_2}{\lambda_1}$       | 30%, 0.25, 0.1, 0.1                                                   | 30%, 0.2, 0.1, 0.1,                                                   | 30%, 0.2, 0.1, 0.1                                                                |

Table 2.2: Parameters of the truncated Gaussian distribution used to generate the edges of the simulated weighted temporal networks for three different data sets, the ground truth of the nodes' community membership for the generated networks and the selected values of the regularization parameters.

### 2.4.2 Real social network: MIT Reality Mining Dataset

This data set was collected between August 2004 and June 2005, by recording the cell phone activity for 94 students and staff at MIT [93]. This data set can be represented as a binary dynamic network across 46 time points. Each point refers to a week during the school year. Two dominant clusters are assumed at each time point. LS-ESC approach is applied to this data set to detect the community structure across time. The cost function is used to track the changes in the community structure over time as shown in Fig. 2.3. The changes in the cost function coincides with six important dates during the school year, namely the beginning and end of the Fall and Spring semesters, Thanksgiving and spring breaks. The detected community structures for selected time points are shown in Fig. 2.4. These time points are selected to present the detected community structures in the beginning of the Fall semester, Thanksgiving break, 2 weeks after the beginning of the Spring semester and end of the Spring semester. The black frames represent the expected ground truth and the different colors represent the communities detected by LS-ESC. As it can be seen from Fig. 2.4a, small clusters are detected at the beginning of the Fall semester indicating less interactions between students during this time. On the other hand, bigger clusters are detected during Thanksgiving break and Spring semester indicating more interactions between participants during these times as shown in Fig. 2.4b and Fig. 2.4c. Finally, two big clusters are detected at the end of the Spring semester as shown in Fig. 2.4d. However, these clusters are less dense than the ones detected during the semester which indicates that the interactions between participants decreases toward the end of the semester.

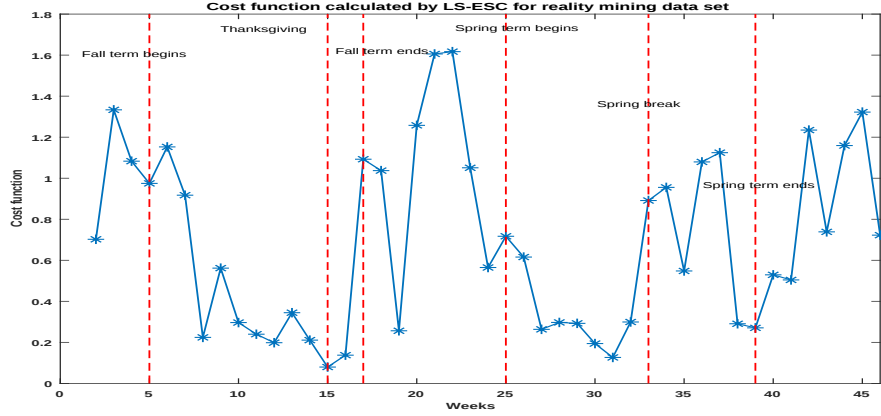


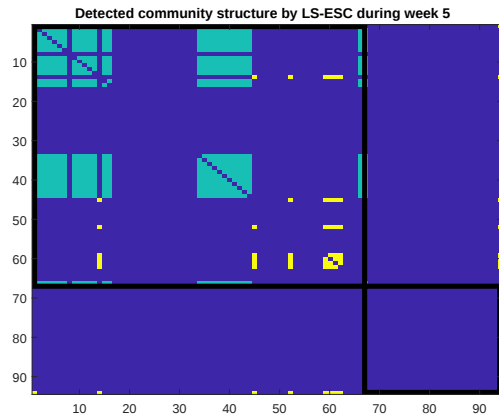
Figure 2.3: Cost function calculated by LS-ESC for the MIT Reality mining network. The changes in the cost function coincides with six important dates during the school year (red dashed lines).

### 2.4.3 Real social network: Primary school network

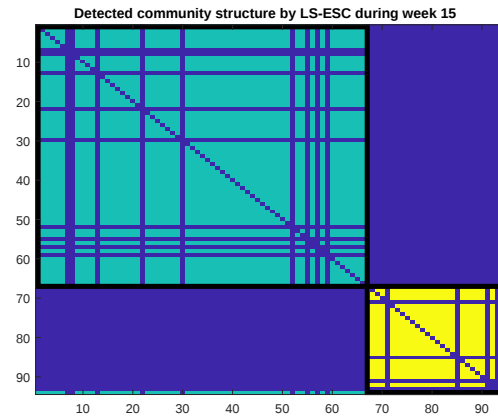
In order to validate the performance of the proposed approach in detecting and tracking the community structure in real weighted temporal networks, a high-resolution data set that describes a temporal social network in a primary school is used [94]. This data set studies the relation between contact patterns among children in a primary school and the propagation of diseases [95].

#### 2.4.3.1 Primary school data set description and temporal network construction

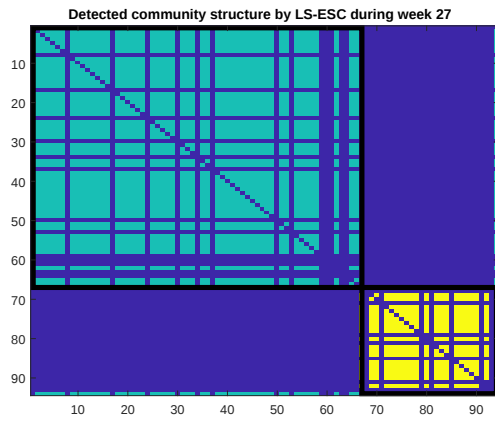
This data set is collected by the SocioPatterns collaboration (<http://www.sociopatterns.org>) during two consecutive days in October 2009: Thursday, October 1<sup>st</sup> from 8:45 am to 5:20 pm and Friday, October 2<sup>nd</sup> from 8:30 am to 5:05 pm. The data set is collected using wearable sensors that sense face-to-face proximity interactions (about 1 to 1.5 meters) between individuals. The sensing system has a temporal resolution of 20 seconds and the proximity interactions are detected over consecutive 20 seconds long time intervals. The school consists of five grades, where each grade is divided into two sections. The total number of participants in the data set is 242 including 232 students and 10 teachers. Details about the students in the different grades are presented in Table



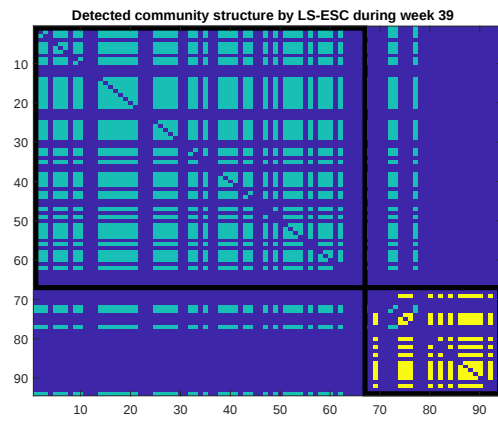
(a)



(b)



(c)



(d)

Figure 2.4: Detected community structure of the MIT reality mining temporal network at different times: (a) Week 5; (b) Week 15; (c) Week 27; and (d) Week 39.

| Class name             | 1A | 1B | 2A | 2B | 3A | 3B | 4A | 4B | 5A | 5B | Teachers |
|------------------------|----|----|----|----|----|----|----|----|----|----|----------|
| Number of participants | 23 | 25 | 23 | 26 | 23 | 22 | 21 | 23 | 22 | 24 | 10       |

Table 2.3: Different classes and number of teachers and students participating in the primary school data from each class.

2.3. The school day runs from 8:30 am 4:30 pm, with lunch break from 12:00 pm to 2:00 pm and two breaks of 20 – 25 minutes around 10:30 am and 3:30 pm. Only two or three classes have breaks at the same time. Lunch is served in a common canteen in two consecutive time periods and a common playground located outside the building is used during the other two breaks. A more detailed description of the data set can be found in [94].

In order to analyze the primary school data set, two temporal networks are constructed to present the temporal interactions between students and teachers for two consecutive days. In each network, the raw sensor data is averaged over time intervals of approximately 13 minutes, i.e. each time step of the constructed temporal network represents the average interactions between individuals over 13 minutes. Each time step is represented by an undirected weighted adjacency matrix  $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$  with  $n = 242$  nodes. This results in two temporal networks with  $T = 40$  time points representing each day.

An important advantage of this network is that since it reflects the interactions in a primary school, a set of communities are expected to represent the different classes in the school. Moreover, the classes, lunch and break times are known which provides an expected ground truth about the community structure and how it changes over time. For instance, face-to-face interactions are expected to be higher between children in the same class during class times while the interactions between children from different classes are expected to increase during lunch or break times.

### **2.4.3.2 Detecting and tracking the community structure in the primary school temporal network (PSTN)**

The proposed LS-ESC approach is applied to PSTN for both days to detect and track the community structure over time. In both days, the detected number of communities in the network varies between 4 and 15 communities.

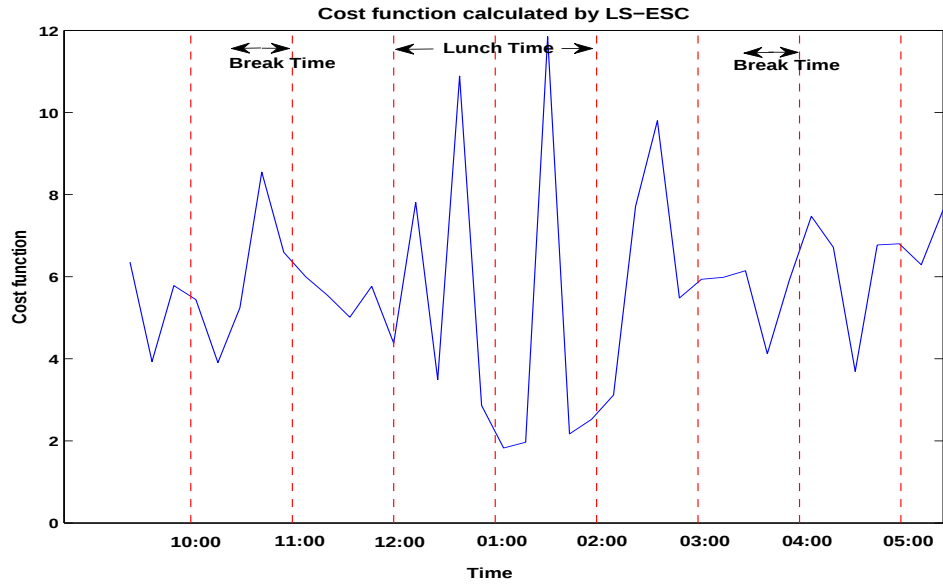
The proposed cost function is calculated for both days and presented in Fig. 2.5(a) and (b). As it can be seen in Fig. 2.5, both cost functions follow a similar pattern during the day where the changes in the cost function reflect the changes in the community structure corresponding to breaks and lunch times. In particular, the biggest variation in the cost functions corresponds to lunch times where face-to-face proximity interactions between students from different classes increase.

The detected community structures at selected time points are shown in Fig.2.6 and Fig. 2.7. These time points are selected to present the detected community structures in the morning, break, lunch and end of the day. Teachers are represented by the last ten nodes in red, the black frames represent the different classes and the different colors represent the communities detected by LS-ESC. As it can be seen from Fig.2.6 and Fig. 2.7, the detected community structures in the morning and at the end of the school day, i.e. Fig. 2.6(a) and (d) and Fig. 2.7 (a) and (d), represent the different classes in the school where the face-to-face proximity interactions are high between children from the same classes. Also, teachers are members of these communities during these times. These results agree with the activity pattern of the school since these times represent class times. On the other hand, bigger clusters that consist of more than one class are detected during break and lunch times as shown in Fig. 2.6 and Fig. 2.7(b) and (c), respectively, which reflects the true activity pattern during these times.

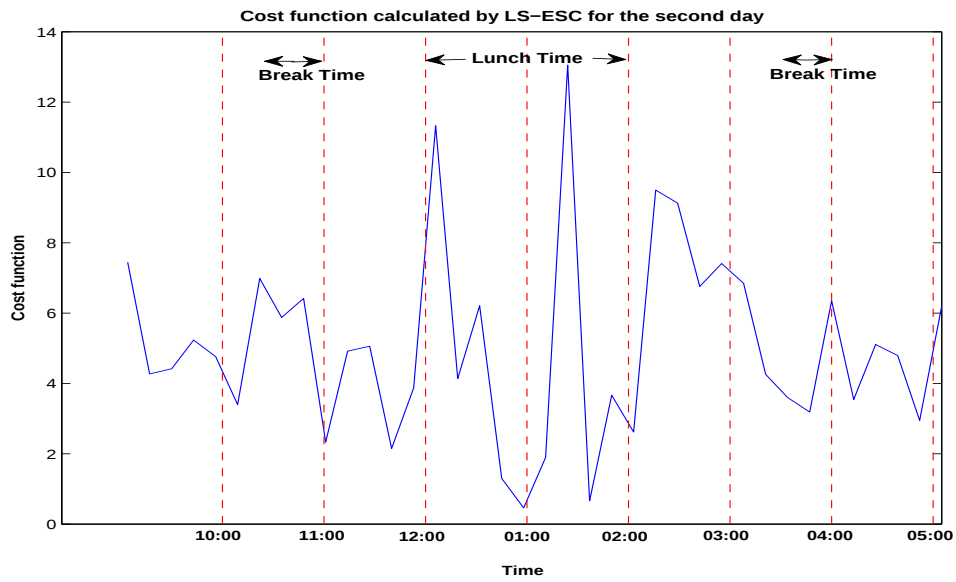
## 2.5 Conclusions

In this chapter, a new low-rank + sparse estimation based evolutionary spectral clustering approach is proposed to detect and track the community structure in temporal networks. In particular, the proposed method decomposes the network into low-rank and sparse parts and obtains smooth cluster assignments by minimizing the subspace distance between consecutive time points, simultaneously. The estimated low-rank adjacency matrix is used for clustering and the subspaces are defined through spectral embedding. Furthermore, a cost function to track changes in the community structure across time is introduced through the subspace distance measure. The algorithm is validated on multiple simulated and real weighted and binary temporal networks and compared to the existing state-of-the-art evolutionary clustering algorithms. The simulation results show that the proposed algorithm can adapt to the changing community structure and detect the correct community structure over time.

The proposed approach detects and tracks the community structure in both weighted and binary temporal networks using the same optimization problem efficiently unlike some existing methods that are limited to either binary or weighted networks. Moreover, LS-ESC approach is robust to noise and outliers, non-stationarity and abrupt changes in the structure of the temporal network due to the estimation of the low-rank part of the adjacency matrix at each time point.



(a)



(b)

Figure 2.5: Cost function calculated by LS-ESC for the primary school temporal network for the: (a) first day, (b) second day. Red dashed lines refer to the time during the day.

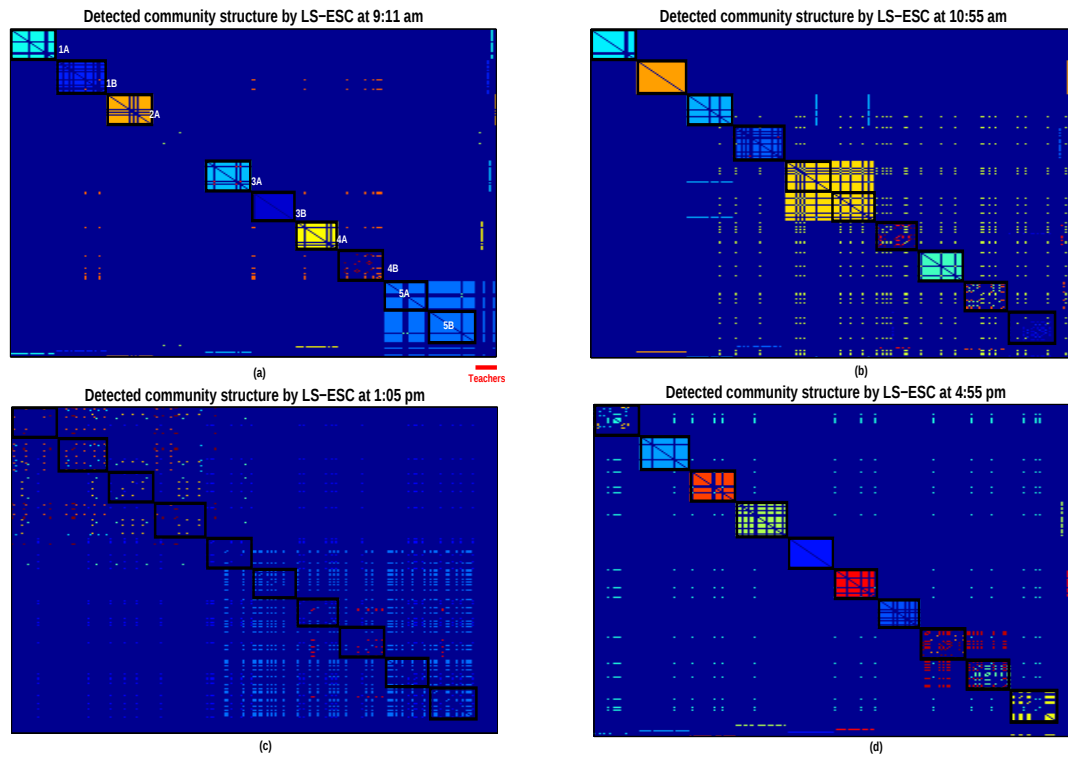


Figure 2.6: Detected community structure in the first day of the primary school temporal network at different times: (a) Morning, (b) break, (c) lunch and (d) end of the school day.

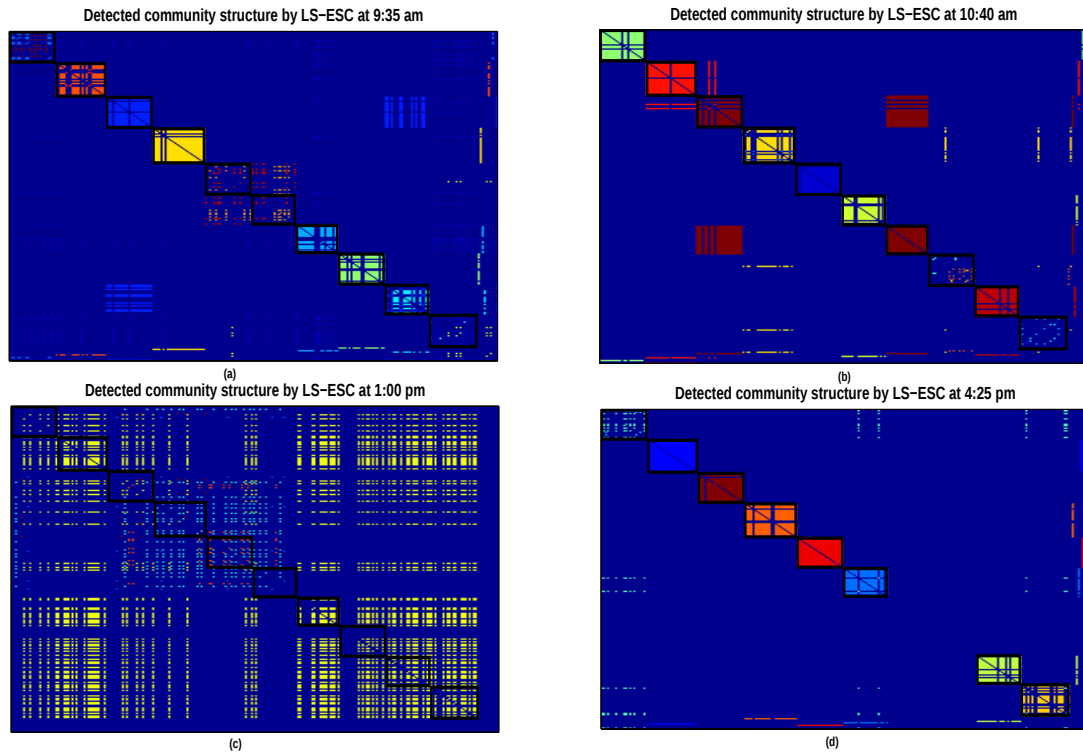


Figure 2.7: Detected community structure in the second day of the primary school temporal network at different times: (a) Morning, (b) break, (c) lunch and (d) end of the school day.

# Chapter 3

## Tensor Based Temporal and Multi-layer Community Detection

### 3.1 Introduction

Networks are commonly used to model the relationships and interactions between people or objects such as in social, biological, and communication networks. A lot of research has been done in detecting the community structure of networks, especially for static networks [96]. However, in many applications, the community structure changes over time. In these applications, the system is represented by a dynamic network [5]. The topology of dynamic networks change over time, since nodes and edges might come and go. In other applications, the nodes of each graph might have different types of relationships that evolve with time, and this variation should be taken into account to understand the correct community structure of these networks [37]. In this case, the system is represented by a dynamic multi-layer network [41]. Over the past decade, multiple approaches have been proposed to investigate the community structure in dynamic networks. Yet, community detection algorithms for dynamic multi-layer networks are very limited. Standard spectral clustering can be used to analyze these networks at each time point or layer separately, but this approach is very sensitive to instantaneous noise.

In recent years, different approaches have been developed to detect the community structure of dynamic networks [31–34]. These approaches include simple extensions of static clustering methods, statistical modeling based methods and tensor based methods. Extensions of static clustering methods such as k-means and agglomerative hierarchical clustering approaches were developed in [31] for dynamic networks. A cost function that considers both the cluster membership at the current time and the deviation from the previous time point was introduced. In [32], two spectral clustering based frameworks were introduced: preserving cluster quality (PCQ) and preserving cluster membership (PCM). In both frameworks, the clustering depends on a cost function to guarantee temporal smoothness. However, this cost function requires a *priori* knowledge about the number of clusters and depends on the choice of a changing parameter. Under statistical modeling based methods, [33] assumed a statistical model for the evolution of the adjacency matrix and an adaptive forgetting factor for evolutionary clustering and tracking (AFPECT) was introduced. In AFPECT, static clustering methods were used after smoothing the proximity between objects over time. Under tensor based methods, a non-negative tensor factorization approach was introduced in [34]. This approach used PARAFAC to detect the community structure in temporal networks, where a single 3-way tensor was constructed to represent all the successive adjacency matrices in the temporal network.

Over the past years, few methods have been developed to detect the community structure in dynamic multi-layer networks. In [35], the authors introduced a general framework that encompasses time-evolving, multi-scale and multi-layer networks to determine the community structure in multi-slice networks. These networks include a set of adjacency matrices that represents the variation in connectivity between nodes across time, or variation in the types of relationships across layers, or variation of the same network across multiple scales. The framework relies on a generalization of the modularity optimization problem for detecting the community structure of a single

network. Since this framework depends on modularity, it has multiple drawbacks including the computational cost, null model selection and the possibility of the convergence to a non-optimal solution as discussed in [20]. Another evolutionary clustering approach to mine and track dynamic multi-layer networks is introduced in [41]. The proposed framework relies on a multi-objective optimization representation and the solution is obtained by a genetic algorithm. This approach has a high computational cost and impractical in the case of detecting the community structure for large networks.

In this chapter, we propose a tensor based approach using 3-way and 4-way tensors to detect the community structure of dynamic single-layer and dynamic multi-layer networks, respectively. In the case of dynamic networks, two approaches, 3-dimensional windowed tensor approach (3D-WTA) and running time tensor approach (RTTA) are proposed. In 3D-WTA, a fixed length sliding window is used to construct the tensors, while a variable length sliding widow is used to construct the tensors in RTTA. In the case of dynamic multi-layer networks, we extend 3D-WTA to 4-dimensional windowed tensor approach (4D-WTA) to optimize the contribution of the multiple layers and time points to the community structure, simultaneously. The main contributions of the proposed work are five-fold. First, a tensor based approach that extends spectral clustering to temporal and multi-layer networks is introduced. The proposed approach simultaneously takes the past history across time or time and multiple layers into account to obtain a common community structure for the network. Moreover, unlike simple averaging based methods, the proposed approach optimizes the contribution of each layer and time point within the framework. Second, the proposed approach preserves the topological structure of the networks by using tensors to represent the network across time and layers. Third, the proposed framework offers a way to track the changes in the community structure over time. In particular, a normalized cost function is introduced to track the changes in the community structure over time. Fourth, the proposed framework,

unlike AFPECT, is model-free and it tracks and identifies the dynamic community structure across time and layers without making any assumptions about the changes in the community structure like PCM and PCQ the temporal clustering (TC) framework in [78]. Fifth, the proposed work can be applied to investigate different types of networks, such as social and biological networks. An application of interest in our work is studying and analyzing the dynamic functional connectivity networks of the brain during resting state and cognitive control, with a major focus on resting state fMRI (rs-fMRI). In particular, the dFCNs are summarized into a set of functional connectivity patterns or states that repeat over time. For each subject, each time point in the dFCN is represented by a FC state. Unlike previous work [55], [56] and [97], the FC states are defined through a common community structure. Furthermore, transition profiles between different states are obtained for each subject to show the variance in the dynamic brain activity across subjects during rest. A consistency measure is employed to quantify the dynamics of the different subnetworks in rs-fMRI.

## 3.2 Background

### 3.2.1 Quality Metrics to Determine the Number of Clusters

Different methods and quality metrics have been utilized to determine the number of clusters and to quantify the quality of a community structure when the ground truth is unknown [98,99]. Some of the most commonly used metrics include eigengap criterion [7], [100], modularity [11] and asymptotical surprise [90].

In the eigengap criterion, the number of clusters,  $r$ , for a given normalized similarity matrix,  $\mathbf{A}_N$ , with eigenvalues  $\lambda_1 \geq \lambda_2 \dots \geq \lambda_n$  is determined by  $\min\{r : \lambda_i - \lambda_{i+1} \leq \delta, \text{ for all } i > r\}$ . The threshold  $\delta$  is determined as the 0.95 quantile of the largest eigengap obtained from the eigenvalues

of a null model generated from random networks with the same degree distribution and weights as the original network [100].

Another commonly used metric is modularity [11], which has been used as a quality metric in community detection methods [30, 33, 92]. Modularity is defined as  $Q = \frac{1}{2E} \sum_{c \in \bar{C}} \sum_{i, j \in V_c} [A_{ij} - \frac{D_i D_j}{2E}]$  where  $2E = \sum_{i, j=1}^n A_{ij}$  is the total edge weight,  $D_i$  is the degree of node  $i$ ,  $\bar{C} = \{C_1, \dots, C_r\}$  is the set of clusters and  $V_c$  is the set of nodes in the  $c^{th}$  cluster. This metric measures the deviation fraction of the within community edges in the network from the expected value of the same quantity in the network with the same community partitions but random connections between the vertices [11]. However, modularity has been shown to suffer from the resolution and null model selection problems [101]. The addition of a tunable parameter namely a resolution parameter  $\gamma$  is suggested in [102] to address this issue and the definition of the modularity is modified as  $Q_\gamma = \frac{1}{2E} \sum_{c \in \bar{C}} \sum_{i, j \in V_c} [A_{ij} - \gamma \frac{D_i D_j}{2E}]$ , where  $Q_\gamma$  is the multi-resolution modularity metric. An approach to calculate the resolution parameter is suggested in [103]. Despite the fact that the addition of  $\gamma$  improves the performance of modularity, it still does not provide a reliable solution with the inappropriate choice of the null model [20].

In order to overcome the limitations of modularity, asymptotical surprise is proposed in [90]. Asymptotical surprise is defined as  $S_a = 2E D_{KL}(q || \langle q \rangle)$ , where  $2E$  is the sum of edge weights in the adjacency matrix,  $q = \frac{E_{int}}{2E}$  is the observed ratio of the intra-cluster edge weights to the total edge weights and  $\langle q \rangle = \frac{p_{int}}{p}$  is the expected ratio of intra-cluster edge weights with  $p_{int}$  as the total sum of possible intra-cluster edge weights.  $D_{KL}$  is the Kullback-Leibler divergence [104]. This quality metric compares the distribution of the nodes and edges in the detected communities in a certain network with respect to a null model. The community structure with the highest score is then selected. As discussed in [90], [105] and [98], this metric is not affected by the resolution limit unlike modularity.

## 3.3 Methods

### 3.3.1 Spectral Clustering

In graph theory, a common method to detect the community structure in static networks is spectral clustering, as it represents the relationship between the nodes in the graph in a lower dimensional subspace. Spectral clustering solves the following trace optimization problem [7, 8],

$$\min_{\mathbf{U} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^\dagger \boldsymbol{\Phi} \mathbf{U} \right), \quad s.t. \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}, \quad (3.1)$$

where,  $\boldsymbol{\Phi} = \mathbf{I} - \mathbf{A}_N$  is the normalized Laplacian matrix.  $\mathbf{A}_N = \mathbf{D}^{-\frac{1}{2}} \mathbf{A}_N \mathbf{D}^{-\frac{1}{2}}$  is the normalized adjacency matrix, where  $\mathbf{A}_N \in \mathbb{R}^{n \times n}$ ,  $n$  is the number of nodes and  $\mathbf{D}$  is the diagonal degree matrix which is defined as  $D_{ii} = \sum_{j=1}^n a_{ij}$ . Spectral clustering problem can also be written as,

$$\max_{\mathbf{U} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^\dagger \mathbf{A}_N \mathbf{U} \right), \quad s.t. \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}, \quad (3.2)$$

or since  $\mathbf{A}_N$  is positive semi-definite, (3.2) can be written as,

$$\max_{\mathbf{U} \in \mathbb{R}^{n \times r}} \left\| \mathbf{U}^\dagger \mathbf{A}_N \mathbf{U} \right\|_F^2, \quad s.t. \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I}. \quad (3.3)$$

The solution to the optimization problems in (3.1), (3.2) and (3.3) is found by choosing  $\mathbf{U}$  as the matrix that contains the  $r$  eigenvectors that correspond to the smallest eigenvalues of  $\boldsymbol{\Phi}$  or the largest eigenvalues of  $\mathbf{A}_N$ . The community structure is then determined by applying  $k$ -means to the matrix  $\mathbf{U}$  [7].

### 3.3.2 3D-Windowed Tensor Approach (3D-WTA)

In clustering dynamic or temporal networks at time  $t$ , it is important to take the recent history of the network into account so that the resulting community structure is smooth across time. One way to accomplish this is to employ a weighted average of the adjacency matrices within a time window of length  $L$  and then optimize the conventional spectral clustering cost function over this weighted average. This results in an optimization over both the subspace vectors matrix,  $\mathbf{U}^{(t)}$ , and the weight vector,  $W^{(t)}$ . The optimization problem in (3.3) can then be rewritten as,

$$\max_{\mathbf{U}^{(t)}, W^{(t)}} \left\| \mathbf{U}^{(t)\dagger} \left( \sum_{l=1}^L w_l^{(t)} \mathbf{A}_N^{(l)} \right) \mathbf{U}^{(t)} \right\|_F^2, \text{ s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, W^{(t)} \geq 0, \|W^{(t)}\|_2 = 1, \quad (3.4)$$

where  $w_l$  are the weights for the adjacency matrices within a time window of length  $L$ .

In the proposed approach, 3D-WTA, we build a third order tensor,  $\mathcal{X}^{(t)} \in \mathbb{R}^{n \times n \times L}$   $= [\mathbf{A}_N^{(t-L+1)}, \dots, \mathbf{A}_N^{(t)}]$ , where the  $l^{th}$  frontal slice of the tensor  $\mathcal{X}^{(t)}$  is the normalized adjacency matrix  $\mathbf{A}_N^{(l)}$ . A sliding fixed size tensor is constructed across time and (3.4) can be reformulated in terms of Tucker decomposition as follows,

$$\max_{\mathbf{U}^{(t)}, W^{(t)}} \left\| \mathcal{X}^{(t)} \times_1 \mathbf{U}^{(t)\dagger} \times_2 \mathbf{U}^{(t)\dagger} \times_3 W^{(t)\dagger} \right\|_F^2, \text{ s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \|W^{(t)}\|_2 = 1, \quad (3.5)$$

where  $\mathbf{U}^{(t)}$  is the basis along the first and second modes and  $W^{(t)}$  corresponds to the weighting vector across time. Since the networks are undirected, the components along the first two modes are the same. The optimization in (3.5) can be solved by higher-order orthogonal iteration (HOOI), where in each iteration either the matrix  $\mathbf{U}^{(t)}$  or the vector  $W^{(t)}$  is fixed to optimize the other one.

This approach is summarized in the pseudo code in Algorithm 3.2. In this approach, we ini-

tialize the eigenvectors matrix using HOSVD for the initial tensor and then update this estimate using HOOI. For the remaining time points, in order to reduce the computational cost, the eigenvectors matrix at time point  $t$  is initialized using the estimated optimal eigenvectors matrix from the preceding time point  $(t - 1)$ ,  $\mathbf{U}^{(t-1)}$ .

---

**Algorithm 3.2** 3D-Windowed Tensor Approach (3D-WTA)

---

**Input:**  $\mathbf{A}_N^{(1)}, \dots, \mathbf{A}_N^{(M)}, L$ .

**Output:** Clustering Labels, Change points.

- 1: **for**  $t = L : M$  **do**
  - 2:   Construct a similarity tensor  
 $\mathcal{X}^{(t)} = [\mathbf{A}_N^{(t-L+1)}, \dots, \mathbf{A}_N^{(t)}]$ .
  - 3:   Unfold the tensor along mode 3 to get  $\mathbf{X}_{(3)}^{(t)}$ .
  - 4:   **if**  $t = L$  **then**
  - 5:     Initialize  $\mathbf{U}_0^{(t)}$  by HOSVD of  $\mathcal{X}^{(t)}$ .
  - 6:   **else**
  - 7:     Initialize  $\mathbf{U}_0^{(t)}$  by  $\mathbf{U}^{t-1}$ .
  - 8:   **end if**
  - 9:    $i = 0$
  - 10:   Calculate  $W_{i+1}^{(t)}$  as the dominant left singular vector of  $\mathbf{X}_{(3)}^{(t)} (\mathbf{U}_i^{(t)} \otimes \mathbf{U}_i^{(t)})$ .
  - 11:   Compute  $\bar{\mathbf{A}} = \sum_{l=t-L+1}^t (W_{i+1}^{(t)})_l \mathbf{A}_N^{(l)}$ .
  - 12:   Obtain  $\mathbf{U}_{i+1}^{(t)}$  by eigenvalue decomposition of  $\bar{\mathbf{A}}$ .
  - 13:   **if**  $\|\mathbf{U}_{i+1}^{(t)} - \mathbf{U}_i^{(t)}\|_F^2 > \varepsilon$  **then**
  - 14:      $i = i + 1$ .
  - 15:     **goto** line 10.
  - 16:   **end if**
  - 17:   Normalize the columns of  $\mathbf{U}^{(t)}$ .
  - 18:   Apply  $k$ -means on  $\mathbf{U}^{(t)}$  to obtain clustering labels.
  - 19: **end for**
  - 20: Track the change points using (3.11).
- 

### 3.3.3 Running Time Tensor Approach (RTTA)

In this approach, the tensor at time point  $t$  is built by taking all history into account, where  $\mathcal{X}^{(t)} = [\mathbf{A}_N^{(1)} \cdots \mathbf{A}_N^{(t)}]$ . This approach estimates the optimal eigenspace that represents the network at time

$t$  in a similar fashion to 3D-WTA. The main differences are in the tensor size and initialization of the matrix  $\mathbf{U}^{(t)}$ . This approach uses a window size that grows with time. Consequently, the tensor constructed at each time point has a different size. Moreover, at each time point the initial eigenvectors matrix is an updated version of the previous one. This update was motivated by the fact that the tensor at time  $t$  is the same as the tensor at time  $(t - 1)$  except for one time slice. Therefore, the matrix  $\mathbf{U}^{(t)}$  is adaptively updated based on the new adjacency matrix. We used **TrackU** algorithm [106, 107] to update the eigenvectors matrix. The RTTA is described in Algorithm 3.3, and the eigenvectors matrix update is described in Algorithm 3.4.

---

**Algorithm 3.3** Running Time Tensor Approach (RTTA)

---

**Input:**  $\mathbf{A}_N^{(1)} \dots \mathbf{A}_N^{(M)}$ , *starttime*, *forgettingfactor* =  $\beta$ .

**Output:** Clustering Labels, Change points.

- 1: **for**  $t = \text{starttime} : M$  **do**
  - 2:   Construct a similarity tensor  $\mathcal{X}^{(t)} = [\mathbf{A}_N^{(1)} \dots \mathbf{A}_N^{(t)}]$ .
  - 3:   Unfold the tensor along mode 3 to get  $\mathbf{X}_{(3)}^{(t)}$ .
  - 4:   **if**  $t = \text{starttime}$  **then**
  - 5:     Initialize  $\mathbf{U}_0^{(t)}$  by HOSVD of  $\mathcal{X}_1^{(t)}$ .
  - 6:   **else**
  - 7:     Initialize  $\mathbf{U}_0^{(t)} = \text{TrackU}(\mathbf{U}^{(t-1)}, \mathbf{S}^{(t-1)}, \mathbf{A}_N^{(t)}, \beta)$ .
  - 8:   **end if**
  - 9:    $i = 0$
  - 10:   Calculate  $W_{i+1}^{(t)}$  as the dominant left singular vector of  $\mathbf{X}_{(3)}^{(t)} (\mathbf{U}_i^{(t)} \otimes \mathbf{U}_i^{(t)})$ .
  - 11:   Compute  $\bar{\mathbf{A}} = \sum_{l=1}^t (W_{i+1}^{(t)})_l \mathbf{A}_N^{(l)}$ .
  - 12:   Obtain  $\mathbf{U}_{i+1}^{(t)}$  and  $\mathbf{S}_{i+1}^{(t)}$  by eigenvalue decomposition of  $\bar{\mathbf{A}}$  as,  $\bar{\mathbf{A}} = \mathbf{U}_{i+1}^{(t)} \mathbf{S}_{i+1}^{(t)} (\mathbf{U}_{i+1}^{(t)})^\dagger$ .
  - 13:   **if**  $\|\mathbf{U}_{i+1}^{(t)} - \mathbf{U}_i^{(t)}\|_F^2 > \varepsilon$  **then**
  - 14:      $i = i + 1$ .
  - 15:     **goto** Line 10.
  - 16:   **end if**
  - 17:   Normalize the columns of  $\mathbf{U}^{(t)}$ .
  - 18:   Apply k-means on  $\mathbf{U}^{(t)}$  to obtain clustering labels.
  - 19: **end for**
  - 20: Track the change points using the normalized cost function given in (3.11).
-

---

**Algorithm 3.4 TrackU**

---

**Input:** Eigenvectors matrix:  $\mathbf{U}$ , Eigenvalues matrix:  $\mathbf{S}$ , Adjacency matrix at time point  $t$ :  $\mathbf{A}_N^{(t)}$ , forgetting factor:  $\beta$ .

**Output:** Updated eigenvectors matrix.

```
1:  $C_A = \text{size}(\mathbf{A}_N, 2)$ ,  $C_U = \text{size}(\mathbf{U}, 2)$ ,  $s = \text{diag}(\mathbf{S})$ .
2: for  $i = 1 : C_A$  do
3:    $\bar{b} = \mathbf{A}_N(:, i)$ .
4:   for  $j = 1 : C_U$  do
5:      $y_j = u_j^\dagger \bar{b}$ .
6:      $s_j^{(t)} = \beta s_j + y_j^2$ .
7:     Compute the error:  $e_j = \bar{b} - y_j u_j$ .
8:      $u_j = u_j + \frac{1}{s_j} y_j e_j$ .
9:      $\bar{b}_{j+1} = \bar{b} - y_j u_j$ .
10:  end for
11: end for
12: Orthogonalize  $\mathbf{U}$  using Gram-Schmidt.
```

---

### 3.3.4 4D-Windowed Tensor Approach (4D-WTA)

In the case of four dimensional networks, where graphs are built over time and across multi-layer or multi-subjects, we want to track the changes in the community structure at time  $t$  across all layers based on the past history. In a similar fashion to the 3D case, we can build a weighted average of the adjacency matrices within a time window of length  $L$  across all layers and time, then optimize the conventional spectral clustering cost function over this weighted average. This results in an optimization over the subspace vectors matrix,  $\mathbf{U}^{(t)}$ , the weighting vector across layers,  $\mathbf{W}_s^{(t)}$ , and the weighting vector over time points within a window,  $\mathbf{W}_w^{(t)}$ . The optimization problem in Eq. (3.3) can then be rewritten as,

$$\begin{aligned} \max_{\mathbf{U}^{(t)}, \mathbf{W}_s^{(t)}, \mathbf{W}_w^{(t)}} \quad & \left\| \mathbf{U}^{(t)\dagger} \left( \sum_{k=1}^K w_{sk}^{(t)} \sum_{l=1}^L w_{wl}^{(t)} \mathbf{A}_N^{(l,k)} \right) \mathbf{U}^{(t)} \right\|_F^2, \\ \text{s.t.} \quad & \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \mathbf{W}_s^{(t)}, \mathbf{W}_w^{(t)} \geq 0, \|\mathbf{W}_s^{(t)}\|_2 = 1, \|\mathbf{W}_w^{(t)}\|_2 = 1, \end{aligned} \quad (3.6)$$

where  $w_{wl}^{(t)}$  are the weights for the adjacency matrices within a time window of length  $L$ ,  $w_{sk}^{(t)}$  are

the weights across layers and  $K$  is the total number of layers. In the proposed 4D-WTA, we build a fixed size fourth order tensor,  $\mathcal{X}^{(t)} \in \mathbb{R}^{n \times n \times L \times K} = [\mathbf{A}_N^{(t-L+1, 1:K)}, \dots, \mathbf{A}_N^{(t, 1:K)}]$ , where the  $(l, k)^{th}$  slice of the tensor  $\mathcal{X}^{(t)}$  is the adjacency matrix  $\mathbf{A}_N^{(l, k)}$  at the time point  $l$  for subject  $k$ . (3.6) can be reformulated in terms of Tucker decomposition as follows,

$$\begin{aligned} \max_{\mathbf{U}^{(t)}, \mathbf{W}_s^{(t)}, \mathbf{W}_w^{(t)}} \quad & \| \mathcal{X}^{(t)} \times_1 \mathbf{U}^{(t)\dagger} \times_2 \mathbf{U}^{(t)\dagger} \times_3 \mathbf{W}_w^{(t)\dagger} \times_4 \mathbf{W}_s^{(t)\dagger} \|_F^2, \\ \text{s.t.} \quad & \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \quad \| \mathbf{W}_w^{(t)} \|_2 = 1, \quad \| \mathbf{W}_s^{(t)} \|_2 = 1, \end{aligned} \quad (3.7)$$

where  $\mathbf{U}^{(t)}$  is the basis along the first and second modes,  $\mathbf{W}_w^{(t)}$  corresponds to the weighting vector across time and  $\mathbf{W}_s^{(t)}$  is the weighting vector across layers. The components along the first two modes are the same, since the networks are undirected. The problem in (3.7) can be solved by HOOI.

This approach is summarized in the pseudo code in Algorithm 3.5. In this approach, the projection matrices for the first tensor are initialized using HOSVD and then updated using HOOI. For the remaining time points, the eigenvectors matrix,  $\mathbf{U}^{(t)}$ , and the weighting vector over time,  $\mathbf{W}_w^{(t)}$ , are initialized from the estimated optimal eigenvectors matrix and time weighting vector at the preceding time point  $(t-1)$ ,  $\mathbf{U}^{(t-1)}$  and  $\mathbf{W}_w^{(t-1)}$ , respectively.

### 3.3.5 Tracking the Community Structure Over Time

To track the changes in the community structure using the proposed approaches, we define a cost function that quantifies the quality of the cluster assignment. For 4D-WTA, this cost function is defined as  $\| \mathcal{X}^{(t)} \times_1 \mathbf{U}^\dagger \times_2 \mathbf{U}^{(t)\dagger} \times_3 \mathbf{W}_w^{(t)\dagger} \times_4 \mathbf{W}_s^{(t)\dagger} \|_F$  and satisfies the following inequality,

$$\begin{aligned} Cost &= \| \mathcal{X}^{(t)} \times_1 \mathbf{U}^{(t)\dagger} \times_2 \mathbf{U}^{(t)\dagger} \times_3 \mathbf{W}_w^{(t)\dagger} \times_4 \mathbf{W}_s^{(t)\dagger} \|_F \\ &\leq \| \mathcal{X}^{(t)} \|_F \| \mathbf{U}^{(t)} \|_2^2 \| \mathbf{W}_w^{(t)} \|_2 \| \mathbf{W}_s^{(t)} \|_2, \end{aligned} \quad (3.8)$$

---

**Algorithm 3.5** 4D Windowed Tensor Approach (4D-WTA)
 

---

**Input:**  $\mathbf{A}_N^{(1,1:K)}, \dots, \mathbf{A}_N^{(M,1:K)}, L$ .

**Output:** Clustering Labels, Change points.

- 1: **for**  $t = L : M$  **do**
  - 2:   Construct a similarity tensor  
 $\mathcal{X}^{(t)} = [\mathbf{A}_N^{(t-L+1,1:K)}, \dots, \mathbf{A}_N^{(t,1:K)}]$ .
  - 3:   Unfold the tensor along mode 3 and mode 4 to get  $\mathbf{X}_{(3)}^{(t)}$  and  $\mathbf{X}_{(4)}^{(t)}$ , respectively.
  - 4:   **if**  $t = L$  **then**
  - 5:     Initialize  $\mathbf{U}_0^{(t)}$  and  $W_{w,0}^{(t)}$  by HOSVD of  $\mathcal{X}^{(t)}$ .
  - 6:   **else**
  - 7:     Initialize  $\mathbf{U}_0^{(t)}$  by  $\mathbf{U}^{(t-1)}$  and  $W_{w,0}^{(t)}$  by  $W_w^{(t-1)}$ .
  - 8:   **end if**
  - 9:    $i = 0$
  - 10:   Calculate  $W_{s,i+1}^{(t)}$  as the dominant left singular vector of  $\mathbf{X}_{(4)}^{(t)} (\mathbf{U}_i^{(t)} \otimes \mathbf{U}_i^{(t)} \otimes W_{w,i}^{(t)})$ .
  - 11:   Calculate  $W_{w,i+1}^{(t)}$  as the dominant left singular vector of  $\mathbf{X}_{(3)}^{(t)} (W_{s,i+1}^{(t)} \otimes \mathbf{U}_i^{(t)} \otimes \mathbf{U}_i^{(t)})$ .
  - 12:   Compute  

$$\bar{\mathbf{A}} = \sum_{k=1}^K (W_{s,i+1}^{(t)})_k \sum_{l=t-L+1}^t (W_{w,i+1}^{(t)})_l \mathbf{A}_N^{(l,k)}.$$
  - 13:   Obtain  $\mathbf{U}_{i+1}^{(t)}$  by eigenvalue decomposition of  $\bar{\mathbf{A}}$ .
  - 14:   **if**  $\|\mathbf{U}_{i+1}^{(t)} - \mathbf{U}_i^{(t)}\|_F^2 > \varepsilon$  **then**
  - 15:      $i = i + 1$ .
  - 16:   **goto** line 10.
  - 17:   **end if**
  - 18:   Normalize the columns of  $\mathbf{U}^{(t)}$ .
  - 19:   Apply  $k$ -means on  $\mathbf{U}^{(t)}$  to obtain clustering labels.
  - 20: **end for**
  - 21: Track the change points using (3.10).
-

which can be further simplified to,

$$\begin{aligned} Cost &= \| \mathcal{X}^{(t)} \times_1 \mathbf{U}^{(t)\dagger} \times_2 \mathbf{U}^{(t)\dagger} \times_3 \mathbf{W}_w^{(t)\dagger} \times_4 \mathbf{W}_s^{(t)\dagger} \|_F \\ &\leq \| \mathcal{X}^{(t)} \|_F, \end{aligned} \quad (3.9)$$

since  $\| \mathbf{W}_w^{(t)} \|_2 = 1$ ,  $\| \mathbf{W}_s^{(t)} \|_2 = 1$  and  $\| \mathbf{U}^{(t)} \|_2^2 = \lambda_{\max}(\mathbf{U}^{(t)\dagger} \mathbf{U})$ , where  $\lambda_{\max}$  is the largest eigenvalue of  $\mathbf{U}^{(t)\dagger} \mathbf{U}$ , which equals to 1 since  $\mathbf{U}^{(t)\dagger} \mathbf{U} = \mathbf{I}$ . Therefore, by dividing both sides in (3.9) by  $\| \mathcal{X}^{(t)} \|_F$  we get a normalized cost function which is always between 0 and 1,

$$Cost4D_{Normalized} = \frac{\| \mathcal{X}^{(t)} \times_1 \mathbf{U}^{(t)\dagger} \times_2 \mathbf{U}^{(t)\dagger} \times_3 \mathbf{W}_w^{(t)\dagger} \times_4 \mathbf{W}_s^{(t)\dagger} \|_F}{\| \mathcal{X}^{(t)} \|_F}. \quad (3.10)$$

Similarly, we define the normalized cost function to detect the changes in community structure with 3D-WTA and RTTA as,

$$Cost3D_{Normalized} = \frac{\| \mathcal{X}^{(t)} \times_1 \mathbf{U}^{(t)\dagger} \times_2 \mathbf{U}^{(t)\dagger} \times_3 \mathbf{W}^{(t)\dagger} \|_F}{\| \mathcal{X}^{(t)} \|_F}. \quad (3.11)$$

Changes in the normalized cost function in Eq. (3.10) and (3.11) reflect changes in the community structure. Both cost functions are computed based on the detected community structure at each time point and quantify the agreement between the detected community structure and the underlying network structure. The change points are then determined using a change detection algorithm [108], [109]. In this algorithm, the normalized cost function is partitioned by choosing random change points. The total deviation of all points within the partition from the empirical mean is calculated. A total residual error is then calculated by summing up these deviations across all partitions. These steps are repeated by varying the location of the change points until the residual error is minimized.

In 3D-WTA and RTTA, the number of clusters at each time point  $t$  is determined by repeating

the algorithm for different number of clusters, usually for 2-10 clusters and the number of clusters that maximizes the asymptotical surprise metric is considered as the best in representing the underlying network. Similarly, in 4D-WTA the algorithm is repeated over a different number of clusters. The asymptotical surprise of the adjacency matrix for each layer and time point is then computed and averaged across all layers. Finally, the number of the clusters that maximizes the average asymptotical surprise is selected as the number of clusters for all layers at that time point.

## 3.4 Results

### 3.4.1 Simulated Temporal Networks

#### 3.4.1.1 Description of the Data sets

Four simulated data sets with different community structures and different noise levels are generated to validate the performance of the proposed 3D-WTA. For each data set, a dynamic network is generated for 60 time points with 100 nodes. Intra and inter-cluster edges are selected from a truncated Gaussian distribution in the range of  $[0, 1]$ . The parameters and the details of the community structures for these data sets are given in Table 3.1.

The SNR of each network is defined as the ratio of the total power of intra-cluster edges to the total power of inter-cluster edges. The SNR can be calculated as:

$$SNR = 10 \log_{10} \frac{\frac{1}{(t_m - t_{M'} + 1)} \sum_{t=t_m}^{t_{M'}} \sum_{i,j \in intra} A_{ij}^{(t)^2}}{\frac{1}{(t_m - t_{M'} + 1)} \sum_{t=t_m}^{t_{M'}} \sum_{i,j \in inter} A_{ij}^{(t)^2}}, \quad (3.12)$$

where  $t_m$  and  $t_{M'}$  refer to the start and end time points in the network, respectively. The term  $(t_m - t_{M'} + 1)$  refers to the total number of time points in the network.

In order to introduce variability across the different data sets, we have further varied the structure of the networks for data set 1 and 2 during the intervals 21-40 and 1-20, respectively. For these networks and time intervals, the large cluster with 60 nodes is generated such that the nodes close to the diagonal of the cluster have stronger connections than the nodes on the off-diagonal. In particular, all the edges close to the diagonal of the cluster are randomly selected from a truncated Gaussian distribution with  $\mu_{inter}$  and  $\sigma_{inter}$  while 60% of the edges on the off-diagonal of the cluster are selected from a truncated Gaussian distribution with  $\mu_{inter}$  and  $\sigma_{inter}$  and the remaining edges are selected from the same distribution with  $\mu_{intra}$  and  $\sigma_{intra}$ .

| Data Sets |                                                           | Time points 1-20                                 | Time points 21-40                                                                                | Time points 41-60                                                        |
|-----------|-----------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| 1         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.7, 0.4, 0.4, 0.2                               | 0.5, 0.3, 0.2, 0.2                                                                               | 0.7, 0.4, 0.4, 0.2                                                       |
|           | SNR(dB)                                                   | -1.69                                            | 3.32                                                                                             | -1.77                                                                    |
|           | Number of clusters                                        | 4                                                | 3                                                                                                | 4                                                                        |
|           | Nodes in each cluster C                                   | $C_1(1-30), C_2(31-60), C_3(61-80), C_4(81-100)$ | $C_1(1-60), C_2(61-80), C_3(81-100)$                                                             | $C_1(1-30), C_2(31-60), C_3(61-80), C_4(81-100)$                         |
| 2         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.5, 0.4, 0.3, 0.2                               | 0.5, 0.2, 0.2, 0.2                                                                               | 0.5, 0.4, 0.3, 0.2                                                       |
|           | SNR(dB)                                                   | 1.839                                            | -1.289                                                                                           | -0.643                                                                   |
|           | Number of clusters                                        | 3                                                | 5                                                                                                | 4                                                                        |
|           | Nodes in each cluster C                                   | $C_1(1-60), C_2(61-80), C_3(81-100)$             | $C_1(1-20), C_2(21-40), C_3(41-60), C_4(61-80), C_5(81-100)$                                     | $C_1(1-40), C_2(41-60), C_3(61-80), C_4(81-100)$                         |
| 3         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.6, 0.4, 0.2, 0.2                               | 0.4, 0.2, 0.2, 0.1                                                                               | 0.6, 0.4, 0.2, 0.2                                                       |
|           | SNR(dB)                                                   | 1.233                                            | 0.1628                                                                                           | -1.191                                                                   |
|           | Number of clusters                                        | 4                                                | 5                                                                                                | 6                                                                        |
|           | Nodes in each cluster C                                   | $C_1(1-30), C_2(31-60), C_3(61-80), C_4(81-100)$ | $C_1(1-30), C_2(31-45), C_3(46-60), C_4(61-80), C_5(81-100)$                                     | $C_1(1-15), C_2(16-30), C_3(31-45), C_4(46-60), C_5(61-80), C_6(81-100)$ |
| 4         | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.8, 0.5, 0.3, 0.2                               | 0.6, 0.2, 0.2, 0.3                                                                               | 0.8, 0.5, 0.3, 0.2                                                       |
|           | SNR(dB)                                                   | 0.0339                                           | -4.541                                                                                           | 0.0388                                                                   |
|           | Number of clusters                                        | 4                                                | 8                                                                                                | 4                                                                        |
|           | Nodes in each cluster C                                   | $C_1(1-30), C_2(31-60), C_3(61-80), C_4(81-100)$ | $C_1(1-15), C_2(16-30), C_3(31-45), C_4(46-60), C_5(61-70), C_6(71-80), C_7(81-90), C_8(91-100)$ | $C_1(1-30), C_2(31-60), C_3(61-80), C_4(81-100)$                         |

Table 3.1: Parameters of the truncated Gaussian distribution used to generate the edges of the simulated networks for four different datasets, and the ground truth of the nodes' community membership for the generated networks.

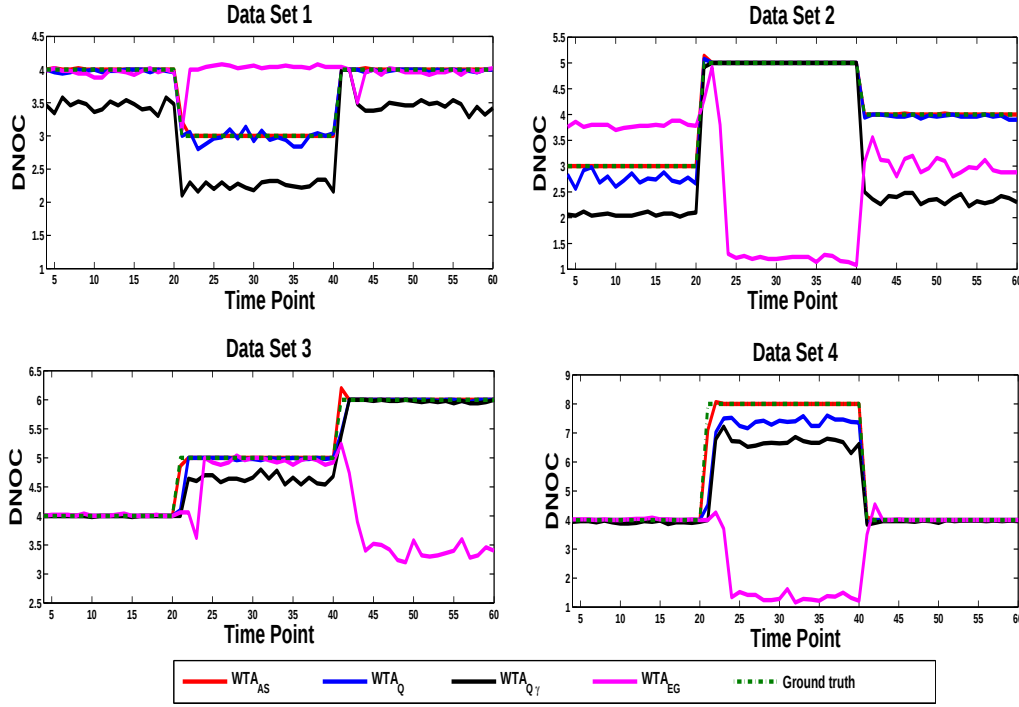


Figure 3.1: Detected number of clusters (DNOC) by 3D-WTA for the data sets in Table 3.1 using different metrics, including: Eigengap (EG), modularity (Q), multi-resolution modularity  $Q_\gamma$  and asymptotical surprise (AS).

### 3.4.1.2 Comparison of different quality metrics

For 3D-WTA, a 3-mode tensor,  $\mathcal{X}^{(t)} \in \mathbb{R}^{64 \times 64 \times 4}$ , is constructed at each time point, and 3D-WTA is applied with a window length of 4. For each data set, the networks are simulated 100 times. In order to choose the best quality metric for determining the number of clusters, the different metrics discussed in Section 3.2.1 are employed. The detected number of clusters (DNOC) for the four different data sets by 3D-WTA using the eigengap criterion (EG) with  $\delta$  calculated as suggested in [100], modularity (Q) [11], multi-resolution modularity ( $Q_\gamma$ ) with  $\gamma$  calculated as suggested in [103] and asymptotical surprise (AS) [90] are presented in Fig. 3.1.

As can be seen in Fig. 3.1, the best quality metric in determining the number of clusters over time is asymptotical surprise. With the EG criterion, the number of clusters is detected more

accurately when the networks have a high SNR and large clusters with all the nodes strongly connected within these clusters. On the other hand, the modularity metric performs better than the multi-resolution modularity for the different data sets. However, both fail to detect the number of clusters correctly when the community structure is noisy with small clusters. Consequently, asymptotical surprise is selected to quantify the quality of the detected community structure and to determine the number of clusters in the rest of this chapter.

### 3.4.1.3 Comparison between 3D-WTA and state of the art algorithms

The performance of the proposed approach is compared to other state of the art community detection algorithms including AFFECT [33], normalized cut PCM (PCM-NC), normalized cut PCQ (PCQ-NC) [32], temporal clustering (TC) [78] and generalized Louvain (GenLov) for multi-layer modularity maximization with the value of  $\gamma$  calculated as described in [103] using variation of information metric. The TC framework is implemented for the different data sets with different number of generative models and the number that detects the best community structure is presented (5, 7, 8 and 7 are the number of generative models that performed the best for data sets 1, 2, 3 and 4, respectively). The GenLov algorithm detected the correct community structure best with  $\gamma = 1$ . A range of coupling coefficients has been tested with GenLov and the best performance is achieved with coupling coefficient,  $\omega = 1$  [35], [110]. The number of clusters at each time point for 3D-WTA, AFFECT, PCM-NC and PCQ-NC is determined as the number that maximizes the asymptotical surprise metric. On the other hand, the ground truth community structures are given to the TC algorithm *a priori* to evaluate the community structure over time.

As it can be seen from Fig. 3.2 (a)-(d), the normalized cost function reflects the changes in the networks, where change points are detected around time points 20 and 40. The proposed algorithm is able to detect the correct number of clusters over time better than other methods as

shown in Fig.3.2 (e)-(h). The variation of information scores achieved by the different methods are given in Fig.3.2 (i)-(l). The results indicate that the proposed algorithm outperforms the other algorithms in terms of detecting the correct community structure over time and can correct itself faster to capture the true community structure after the change points. As seen in Fig.3.2 (e)-(l), the performance of AFFECT, PCM, PCQ and GenLov declines with increasing noise level and decreasing cluster size.

#### 3.4.1.4 Comparison between 3D-WTA and 4D-WTA

In order to compare 3D-WTA with 4D-WTA, a simulated dynamic network for 60 time points with 64 nodes across 20 layers is generated. The edges of the graphs of the simulated dynamic network are taken from a truncated Gaussian distribution in the range of  $[0,1]$ , with the specifications shown in Table 3.2.

For evaluating 3D-WTA, the networks at each time point are averaged across layers. A 3-mode tensor,  $\mathcal{X}^{(t)} \in \mathbb{R}^{64 \times 64 \times 4}$ , is constructed at each time point, with a window length of 4. Moreover, 4D-WTA is applied to the same dynamic network where a 4-mode tensor,  $\mathcal{X}^{(t)} \in \mathbb{R}^{64 \times 64 \times 4 \times 20}$ , is constructed at each time point, with a window length of 4. This experiment is repeated 100 times. A comparison between the detected community structure by 3D-WTA and 4D-WTA and the network's ground truth in terms of variation of information is shown in Fig. 3.3. The variation of information is calculated by comparing the clustering results with the ground truth for each layer separately and then averaged across all layers.

The results show that 4D-WTA performs better than 3D-WTA in detecting the community structure over time. This indicates that the 4D-WTA captures the true community structure better than 3D-WTA since it captures the variations across layers, whereas 3D-WTA loses this variability due to averaging the networks across layers.

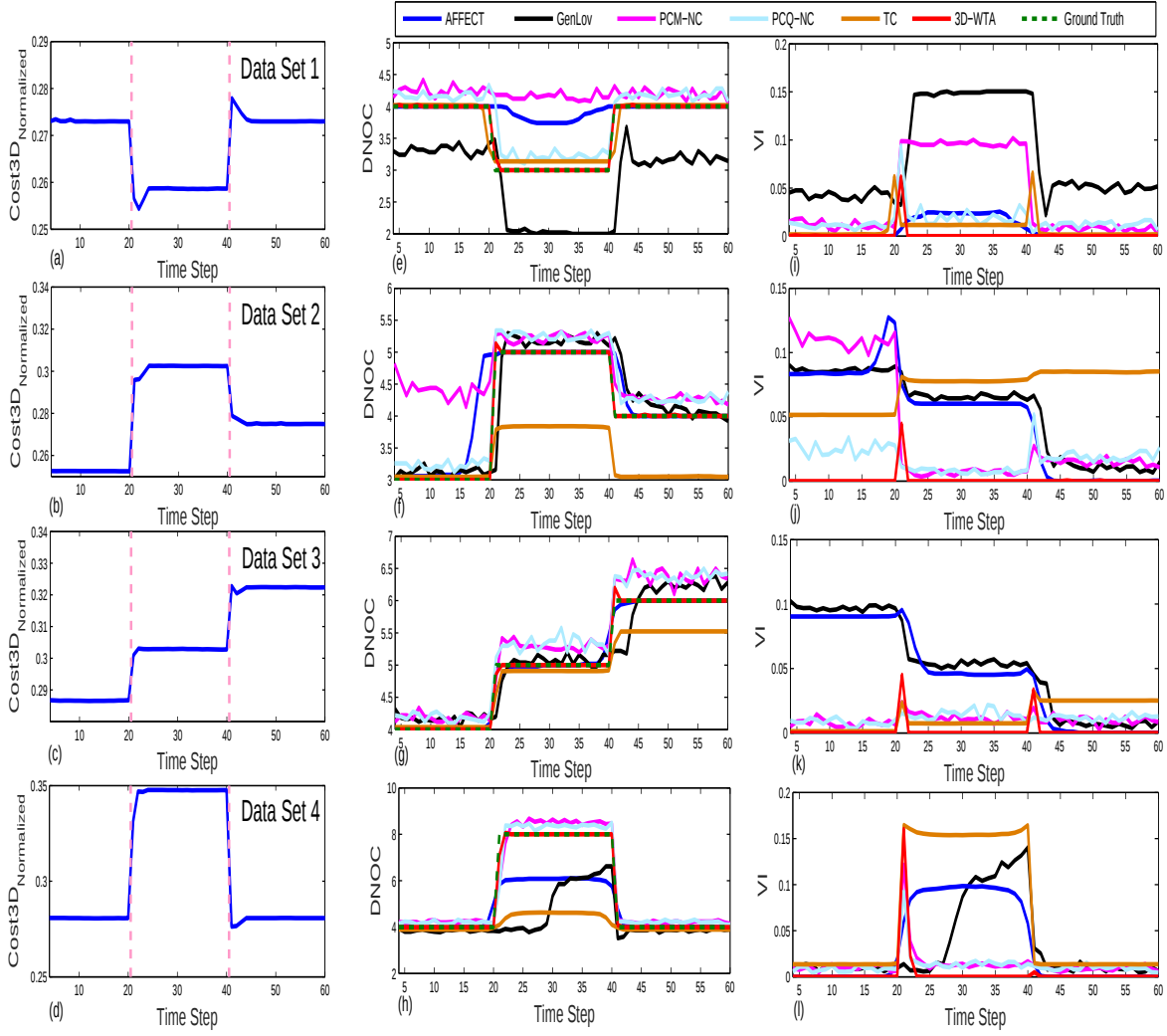


Figure 3.2: Comparison between 3D-WTA and state of the art algorithms for the different data sets in Table 3.1: (a)-(d) The normalized cost function calculated by 3D-WTA, (e)-(h) The detected number of clusters (DNOC) for the simulated networks using different algorithms. 3D-WTA estimated the correct number of clusters over time and the normalized cost function reflected the correct change points represented by the pink dashed lines. (i)-(l) Comparison of variation of information (VI) between 3D-WTA, AFFECT, PCM-NC, PCQ-NC, TC and GenLov for the simulated networks. 3D-WTA scored the lowest variation of information over time compared to the other methods.

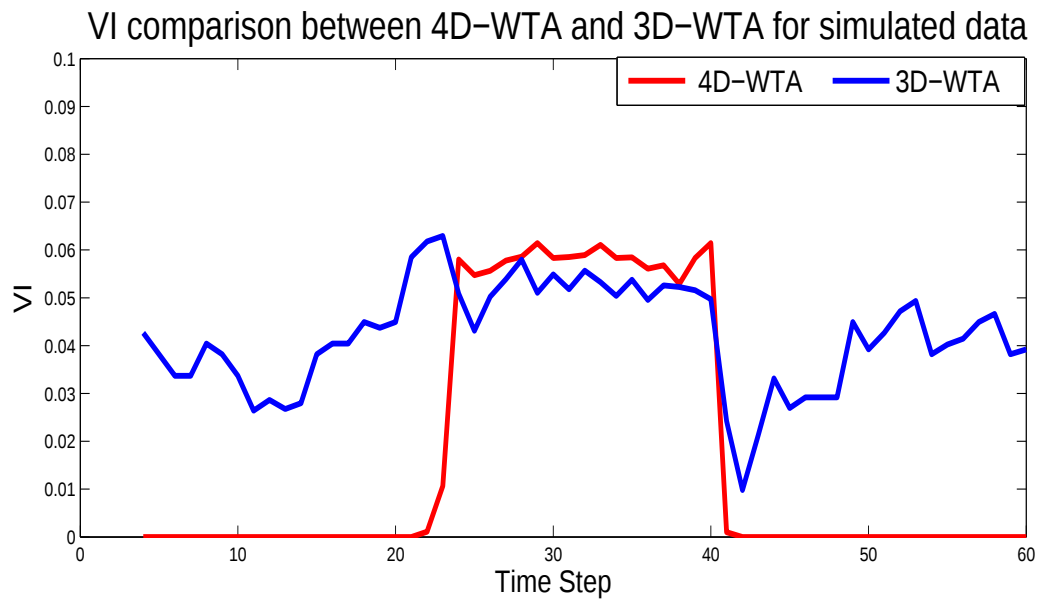


Figure 3.3: Variation of information (VI) comparison between the detected community structure by 3D-WTA and 4D-WTA and the ground truth for simulated data.

| Layers  |                                                           | Time points 1-20                          | Time points 21-40           | Time points 41-60                         |
|---------|-----------------------------------------------------------|-------------------------------------------|-----------------------------|-------------------------------------------|
| 1 – 5   | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.2, 0.2, 0.1, 0.1                        | 0.2, 0.1, 0.1, 0.1          | 0.2, 0.2, 0.1, 0.1                        |
|         | Nodes in each cluster C                                   | $C_1(1 - 22), C_2(23 - 44), C_3(45 - 64)$ | $C_1(1 - 22), C_2(23 - 64)$ | $C_1(1 - 22), C_2(23 - 44), C_3(45 - 64)$ |
| 6 – 15  | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.2, 0.1, 0.1, 0.1                        | 0.1, 0.1, 0.1, 0.1          | 0.2, 0.1, 0.1, 0.1                        |
|         | Nodes in each cluster C                                   | $C_1(1 - 22), C_2(23 - 44), C_3(45 - 64)$ | $C_1(1 - 22), C_2(23 - 64)$ | $C_1(1 - 22), C_2(23 - 44), C_3(45 - 64)$ |
| 16 – 20 | $\mu_{intra}, \sigma_{intra}, \mu_{iner}, \sigma_{inter}$ | 0.4, 0.1, 0.1, 0.1                        | 0.2, 0.1, 0.1, 0.1          | 0.4, 0.1, 0.1, 0.1                        |
|         | Nodes in each cluster C                                   | $C_1(1 - 25), C_2(26 - 49), C_3(50 - 64)$ | $C_1(1 - 25), C_2(26 - 64)$ | $C_1(1 - 25), C_2(26 - 49), C_3(50 - 64)$ |

Table 3.2: Parameters of the truncated Gaussian distribution used to generate the edges of the simulated networks for the 4D simulated data, and the ground truth of the nodes' community membership for the generated network.

#### 3.4.1.5 Flocks of Boids Dataset

This data set represents the natural flocking behavior of birds [33] [111]. The data set is represented by a dynamic network across 40 time points. The network at each time point consists of 100 boids and the community structure changes over time by changing the flock membership of one boid at each time step. The adjacency matrices across a window at time point  $t$  were represented by a 3-mode tensor. The proposed 3D-WTA algorithm was applied to this dataset to detect the community structure and track the changes over time. The number of clusters at each time point was detected using modularity, and the normalized cost function in (3.11) was used to track the changes in the community structure over time. The boids are arranged into 4 clusters until time point 18, then they rearrange into 2 clusters until the end. The change points and the number of clusters shown in Fig. 3.4(a) were detected using 3D-WTA with  $L = 8$ . In Fig. 3.4(b), the Rand index of the proposed algorithm and other state of the art evolutionary clustering methods are compared. The results show that our algorithm performs almost the same as AFFECT and better than PCQ and PCM in terms of Rand index. On the other hand, the proposed algorithm performs better than the existing methods in terms of detecting the changes in community structure and the number of clusters over time. Moreover, the proposed algorithm is robust over multiple runs with respect to tracking changes and number of clusters over time.

#### 3.4.2 MIT Reality Mining Dataset

This data set represents the cell phone activity for 94 students and staff in MIT Media Laboratory. It was collected between August/2004 and June/2005 [93]. The data set is represented by a dynamic network across 46 time points. Each point refers to a week during the school year. Each time point was represented by a 3-mode tensor. Both algorithms were applied to this data to track the changes

over time. We assumed that there were two dominant clusters at each time point, and we used the normalized cost function in (3.10) to track the changes in the community structure over time. The detected change points corresponds to end of the Fall semester, start of the Spring semester, spring break and the end of the Spring semester. The change points shown in Fig.3.5 were detected by the WTA with  $\alpha = 11$ .

### 3.4.3 Application to dFCN From EEG Dataset

The proposed tensor tracking approach is applied to a set of connectivity graphs constructed from EEG data containing the error-related negativity (ERN). The ERN is a brain potential response that occurs following performance errors in a speeded reaction time task usually 25-75 ms after the response. Previous work [112] indicates that there is increased coordination between the lateral prefrontal cortex (LPFC) and medial prefrontal cortex (mPFC) within the theta frequency band (4-8 Hz) and ERN time window. EEG data from 63-channels was collected in accordance with the 10/20 system on a Neuroscan Synamps2 system (Neuroscan, Inc.) sampled at 128 Hz from 91 subjects. The task was a common speeded response letter (H/S) flanker, where error response locked trials from each subject were utilized. The EEG data are pre-processed by the spherical spline current source density (CSD) waveforms to sharpen event-related potential (ERP) scalp topographies and reduce volume conduction [113].

We constructed average connectivity networks at each time point across subjects to represent the adjacency matrix of the connectivity graphs. Individual connectivity networks were constructed by using a previously introduced phase synchrony measure [114]. This results in a dynamic network with 63 nodes and 256 time points. We applied the proposed RTTA to detect the community structure over time. Starting from time point 30, a 3-mode tensor was built at each time point. The change points were detected by the normalized cost function in (3.10) as shown in Fig.3.6. As it

can be seen in Fig.3.6, there is an increase in the cost function around  $-78$  ms right before the response time. Similarly, there is a decrease in the cost function around 400 ms corresponding well with the end of the ERN waveform. The number of clusters was determined by eigengap criterion and the detected community structure corresponding to these three time periods are given in Fig.3.7. While the community structures for pre- and post- ERN intervals are similar, there is increased segregation during ERN especially in the frontal central brain regions. This is aligned with the increased coordination between IPFC and mPFC during this time.

### **3.4.4 Application to dFCN From rs-fMRI Dataset**

#### **3.4.4.1 Data Preprocessing and Functional Connectivity Networks Construction**

The rs-fMRI dataset used in this chapter is from Bangor rs-fMRI data. This dataset is part of 1000 Functional Connectomes Project [115]. The dataset includes 20 healthy subjects (males, aged between 19 – 38). The rs-fMRI data was collected using a 3T Magnet scanner from all subjects while their eyes were open. TR equals to 2s yielding 34 slices and 265 time points. The pre-processing was done using SPM12 [116] and CONN [117] functional connectivity toolboxes. The structural images were registered to the mean functional image and spatially normalized to MNI space for each subject. Slice timing and motion correction were performed on every functional image. The functional images were regionally parcellated by automated anatomical labeling (AAL) atlas [118] and smoothed by spatial convolution with 4-mm FWHM Gaussian kernel. Confounds such as motion parameters obtained from realignment and BOLD signals obtained from white matter and CSF masks were regressed out and band-pass (0.0084-0.09 Hz) temporal filtering was applied to functional images of each subject.

### 3.4.5 Dynamic Functional Connectivity Network Construction

The dynamic functional connectivity networks are constructed for each subject by computing the Pearson correlation coefficient,  $\rho$ , between different ROIs (90 ROIs: with 45 ROIs in the left and the other 45 in the right hemispheres, listed in Table1) using a sliding window approach with a rectangular window of length equal to 60 TRs and an overlap of 1 TR [60], [55], [56].

For each subject, a sequence of weighted and undirected graphs  $\{\mathbf{A}^{(t)}\}_{t_1, \dots, t_M}$  are constructed with  $A_{ij}^{(t)} = \rho_{ij}^t$ , where  $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$  is the adjacency matrix representing the network at time step  $t$ ,  $n$  is the total number of nodes (ROIs) and  $M$  is the total number of time points.

#### 3.4.5.1 Resting State Networks (RSNs)

In order to study the brain dynamics over time, our analysis focuses on the temporal dynamics of functional connectivity communities. The detected communities will be evaluated with respect to well-known resting state networks including the default mode network (DMN), auditory network (AN), visual network (VN), somatomotor network (SMN), bilateral limbic network (BiN), subcortical network (SCN) and cognitive control network (CCN) as summarized in Table 3.3 [55]. Each subnetwork is defined by a group of ROIs that are commonly categorized to be part of these networks. The purpose of applying the proposed approach to the dFCN is to determine the functional organization of the brain and evaluate the obtained functional communities with respect to these commonly defined resting state networks.

#### 3.4.5.2 Detecting the Community Structure in rs-fMRI

The proposed community detection approaches were applied to dFCNs constructed from rs-fMRI dataset described in Section 4.2.2.1. 90 ROIs were extracted after the preprocessing procedure for each subject. Networks built using Pearson correlation are positive semi-definite and consequently

| RSN | ROIs                                       |
|-----|--------------------------------------------|
| VN  | (43 – 56)                                  |
| AN  | (29, 30, 79 – 82)                          |
| BiN | (21, 22, 27, 28, 37 – 42, 83, 84, 87, 88)  |
| DMN | (3 – 6, 23 – 26, 31 – 36, 65 – 68, 85, 85) |
| SMN | (1, 2, 17 – 20, 57 – 60, 63, 64, 69, 70)   |
| SCN | (71 – 78)                                  |
| CCN | (61, 62, 89, 90, 7 – 16)                   |

Table 3.3: The ROIs belonging to the different subnetworks in resting state. The numbers in the table refer to the ROIs defined in Table 1.

can be used as  $\mathbf{A}_N$  in (3.2) and (3.3).

3D-WTA is applied to each subject separately to track and detect the community structure over time and to analyze variability across subjects. At each time point, a 3-mode tensor is constructed using a sliding window with a length of 30,  $\mathcal{X}^{(t)} \in \mathbb{R}^{90 \times 90 \times 30}$ . For each subject, a total of 177 tensors are constructed. The constructed tensors are approximated by Tucker decomposition, as  $\mathcal{X}^{(t)} \approx \mathcal{C} \times_1 \mathbf{U} \times_2 \mathbf{U} \times_3 W$ , where  $\mathcal{C} \in \mathbb{R}^{r \times r \times 1}$  is the core tensor,  $\mathbf{U} \in \mathbb{R}^{90 \times r}$  is the orthogonal low rank component matrix along the connectivity mode and  $W \in \mathbb{R}^{30 \times 1}$  is the weighting vector across time mode.  $r$  is the rank which corresponds to the number of clusters. The number of clusters at each time point is determined using asymptotical surprise metric and varied between 2 and 10 clusters.

For the application of 4D-WTA to rs-fMRI, the fact that not all subjects have the same community structure within the same time window is taken into account. Recent studies in dynamic rs-fMRI literature have revealed groups of reproducible patterns of dFCNs named as FC states [55], [119]. As a result, the possibility that dFCNs across subjects may cycle through a few number of distinct FC states over time has been hypothesized and investigated [56], [71], [120], and [121].

Following what have been proposed in literature, 4D-WTA is applied to rs-fMRI data to extract common network connectivity patterns, or connectivity states, across time and subjects. The main difference between our approach and what is proposed in the literature is that the states are identified through their community structures and not just co-activation patterns. It is also important to note that the goal of 4D-WTA is to achieve data reduction rather than track the dynamics of FC within each subject unlike 3D-WTA. Following the preliminary steps shown in Fig. 3.8, the application of 4D-WTA to the rs-fMRI can be summarized as:

1. Subsample 30 random time points across all subjects, similar to [55], and form a 4D tensor,  $\mathcal{X}^{(sample)} \in \mathbb{R}^{90 \times 90 \times 30 \times 20}$ . This subsampling process was repeated 200 times.
2. Apply 4D-WTA to detect the community structure of the tensor constructed from this random sampling across subjects and time. The tensor corresponding to each random subsampling is approximated by Tucker decomposition following lines 4-16 from Algorithm 3.5.
3. Determine the number of clusters for each subsampling using asymptotic surprise.
4. Use  $\mathbf{U}$  from Eq. (3.7) to obtain the common community structure of the subsampling.
5. Determine the similarity between community structures from different subsamples using variation of information metric and construct a  $200 \times 200$  similarity matrix.
6. Apply k-means clustering to the similarity matrix from Step (5) to identify the samples with similar community structure, i.e. the network states.

After performing Step 2, the optimized weights for the different time points within a time window are found to be very close to each other with a mean of 0.1824 and a standard deviation of 0.0074. However, the optimized weights across subjects show more variation with a mean of 0.2155 and a standard deviation of 0.0597. This is consistent with the variations of functional

connectivity networks across subjects. The distributions of the weights across time and subjects for all samples are shown in Fig. 3.9(a) and (b), respectively. From Fig. 3.9, it can be noticed that the distributions are highly concentrated, indicating that all time points and subjects contributed almost equally to the clustering results.

In the application of 4D-WTA to the rs-fMRI, the detected number of clusters for the different subsamplings is varied between 4 and 10. After applying k-means clustering to identify the samples with similar community structure, 5 clusters that correspond to the FC states shown in Fig. 3.10 are obtained. The optimal number of states is determined using the elbow criterion. The community structure that corresponds to each FC state is determined as the community structure of the centroid of each cluster found by k-means.

The community structure corresponding to the detected FC states is compared to the dynamic functional connectivity network of each subject. For each subject and at each time point, the state that best represents each time point in terms of asymptotical surprise metric is assigned as the community structure at that time point. Finally, a state transition profile that tracks the community structure is obtained for each subject. The obtained state transition profiles for the different subjects followed different patterns, which reflects the variation in the brain activity across subjects during resting state as can be seen in Fig. 3.11. In the next part of this subsection, the community structure corresponding to each FC state and state transition profiles for the different subjects are discussed. Moreover, the dynamic behavior of the different RSNs is quantified through the consistency measure. Finally, a comparison between 3D-WTA and 4D-WTA in terms of tracking brain dynamics is presented.

### 3.4.5.3 Community structure of each FC state

After applying 4D-WTA to the rs-fMRI, the community structure of FC states that are persistent across time and subjects are extracted as shown in Fig. 3.13.

First, the detected FC states are evaluated based on their frequency of occurrence across time and subjects. This is achieved by calculating the average occurrence rate (AR%) across time and subjects, as shown on the top of each state in Fig. 3.13. For each subject, the occurrence rate of each state is calculated as the number of times this state occurred divided by the total number of time points. The occurrence rate of each state across subjects is then averaged to get AR%.

As shown in Fig. 3.13(b), State 2 has the highest AR% at 25%. This state consists of one large cluster consisting of DMN and BiN. Furthermore, there is another large cluster that reflects the connectivity between other RSNs, such as SMN, AN and SCN. A similar state was observed previously in [55]. Other states including States 1 and 3 – 5 have AR% between 13% to 22%.

A major network of interest during resting state is the DMN, as this network is known to be activated during rest and deactivated during task related events [122], [123]. Across the different states, we observe that different ROIs of the DMN are clustered together in some states and clustered with other RSNs in other states. For example, in States 4 and 5, the DMN module includes the main ROIs of the DMN, namely the posterior cingulate gyrus (PCG), precuneus (PCUN), anterior cingulate gyrus (ACG), angular gyrus (AG) and superior frontal gyrus including the dorsolateral (SFGdor), medial (SFGmed) and medial orbital (ORBsupmed). On the other hand, in State 2, the DMN is assigned to the same community with ROIs from the BiN, mainly the hippocampus (HIP) and parahippocampal gyrus (PHG). In fact, HIP and PHG are known to play an important role in long-term memory [124, 125], and previous work considered them as part of the DMN [126], [55]. On the other hand, in States 1 and 3, PCUN, PCG, ACG, SFGmed and middle frontal gyrus (MFG)

from CCN form a cluster, while the remaining ROIs of the DMN are clustered with other RSNs. This variation in the community membership between the different ROIs of the DMN reflects the dynamic behavior of this network and confirms its important role during resting state.

Across the different states, the AN and the VN show a relatively stationary behavior. The ROIs of the AN are clustered together and joined the SMN across all states. Similar behavior is observed for the VN, where the ROIs of the VN form a cluster that appears across all the states. Even though the VN show a relatively stationary behavior, it is strongly connected to ROIs from other RSNs in some states. As can be seen in Fig. 3.13(b), a cluster consisting of the visual network ROIs and the PCUN is observed. This is in line with previously reported results in [55] and [127], where a network state that included PCUN and the ROIs of the VN has been identified. Furthermore, the VN cluster includes the ROIs of the VN and the superior parietal gyrus (SPG) across States 1, 2 and 5, and this agrees with the results reported in [128].

Another observation across the states is related to the dynamics of the SCN. As can be seen in Fig. 3.13(b-d), the ROIs of the SCN exhibit variation in their community membership. In State 1, a complete separation between subcortical and cortical ROIs is observed. Moreover, during this state the ROIs of the DMN join other ROIs from different RSNs to form small clusters. This observation suggests that State 1 represents a light sleep state, which is commonly observed in rs-fMRI as in [55].

#### **3.4.5.4 State transition profiles**

In order to understand the dynamics of the FCNs, we studied the state transition profiles for the different subjects over time. These profiles reflected two important phenomena. First, the state transition profiles followed relatively different patterns over time suggesting variation in the brain activity across subjects during resting state. Moreover, for some subjects the brain may fluctuate

between a few number of states and not necessarily occupy all the detected FC states. Second, we observed that the community structure did not fluctuate quickly between states over time. State transition profiles for selected subjects are shown in Fig. 3.11.

### **3.4.5.5 Consistency of the RSNs across time**

In order to calculate the consistency of a subnetwork, we first calculated the recruitment coefficient defined in [129] for each ROI within the subnetwork. The ROI's recruitment coefficient is defined as the probability that this ROI is in the same cluster as other ROIs of its own subnetwork at each time point. The subnetwork consistency is then calculated as the average of the recruitment coefficients of all ROIs that correspond to the subnetwork.

After the application of 3D-WTA and 4D-WTA to the rs-fMRI FCNs, the consistency of the RSNs across time and subjects was computed to quantify their dynamic behavior. The variation in consistency across the different RSNs is shown in Fig.3.12(a) for 3D-WTA and Fig.3.12(d) for 4D-WTA.

VN, SCN, SMN and AN have relatively higher consistency across subjects implying that the ROIs within each network are strongly connected. In other words, these networks experience less variability over time. On the other hand, DMN, CCN and BiN are less consistent over time, indicating that these RSNs are more dynamic over time. By comparing the variation in consistency calculated by 4D-WTA and 3D-WTA, we observe that the consistency results by 3D-WTA are less representative, since the variation in consistency across different RSNs is very similar. Thus, it is hard to reach a definitive conclusion about the dynamics of these RSNs using 3D-WTA.

In order to compare the results of 3D-WTA and 4D-WTA with other state of the art algorithms, the algorithms that performed the best among others in the simulated data are selected. These methods are PCQ-NC and Genlov. The variation in consistency across all subjects for the different

RSNs is shown in Fig.3.12(b) for PCQ-NC and Fig.3.12(c) for GenLov. From these figures, it can be seen that the consistency results from both methods follow similar patterns that align with the results from both 3D-WTA and 4D-WTA. However, 3D-WTA and 4D-WTA show less variation in the consistency of the different RSNs across subjects, indicating the stability of our methods compared to existing algorithms.

#### **3.4.5.6 Comparison Between 3D-WTA and 4D-WTA**

In order to study the degree of agreement between the community structure detected by 3D-WTA and 4D-WTA for the different subjects, the variation of information is calculated between the clustering results found by 4D-WTA and the clustering results found by 3D-WTA for each subject separately. The average VI across time and subjects has a mean of 0.46 and a standard deviation of 0.0507. These results indicate that in general, there is a high agreement between group and individual community structures.

### **3.5 Conclusions**

In this chapter, we introduced a new tensor based framework to detect and track community structure in temporal and temporal multi-layer networks using 3-way and 4-way tensors, respectively. The framework is implemented by two approaches to detect the community structure in temporal networks: 3D-WTA and RTTA. In 3D-WTA, a fixed length sliding window is used to construct the tensors, while a variable length sliding widow is used to construct the tensors in RTTA. On the other hand, we extended the 3D-WTA to 4D-WTA to detect the community structure in temporal multi-layer networks. In the proposed framework, either a 3-way or 4-way tensor is constructed at each time point by taking the past history across time or time and layers into account. The

community structure is detected by finding a common subspace through Tucker decomposition of the constructed tensors. Moreover, the proposed algorithms optimize the contribution of each time point and layer within the framework. Furthermore, a normalized cost function is introduced to track the changes in the community structure over time. The algorithms were tested on multiple simulated and real data sets and compared to the existing state of the art algorithms. The application of the proposed framework to simulated and real social networks shows that the proposed algorithms are able to detect the correct community structure and changes in the number of clusters over time even in noisy networks.

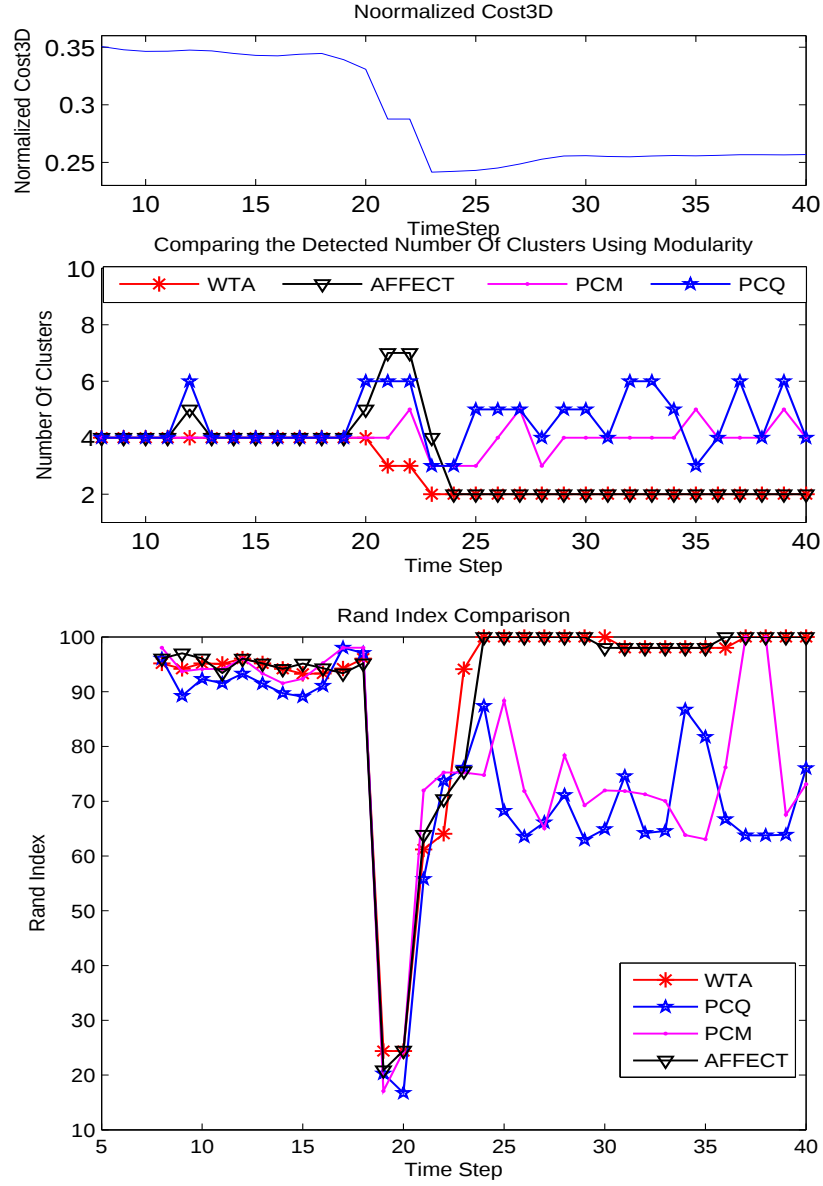
A particular application of interest of the proposed work is to detect and track the community structure dynamics in FCN of the brain during cognitive control or resting state, with the major focus on FCN from rs-fMRI. The application to functional connectivity brain network from EEG data revealed changes in the community structure across time that aligned well with the experimental setting and prior work on cognitive control.

In the application of the proposed framework to the dFCN from rs-fMRI, the dFCNs are summarized into a set of FC states that repeat over time and subjects. The detected FC states are defined through their community structure. Moreover, state transition profiles which reflect the variance in the dynamic brain behavior across subjects during resting state are obtained. The detected community structures were evaluated through consistency measure. From this measure, it is clear that VN, AN, SCN and SMN are the most stationary RSNs, whereas DMN, CCN and BiN reconfigure themselves more frequently across time. In particular, the DMN showed the highest dynamic activity over time as expected.

In addition, the results of 3D-WTA and 4-WTA showed high agreement in the community structure in terms of variation of information. However, the community structure detected by 4D-WTA reflected the RSNs' dynamics and evolution across time better than 3D-WTA. This is due

to the fact that 4D-WTA is using more data from both time and subjects, and it preserves the topological structure of the FCNs.

Future work will consider the extension of this framework to higher order tensor type data clustering across different modes. Moreover, we will consider reducing the computational cost of the algorithms by reducing the size of the constructed tensors. This can be achieved by adding extra terms or constraints to the objective functions. The major goal of this modification will be to select the time points or layers that can be included in the framework. A possible selection criterion could be dependent on selecting the time points or layers that are sparser than others, as this will guarantee the selection of the time points or layers with less noisy community structure.



(b)

Figure 3.4: (a) The normalized cost function and the detected number of clusters for Flocks of Boids dataset using different algorithms. 3D-WTA results in the best estimate of the number of clusters especially after the change point, (b) Rand index comparison between 3D-WTA, AFFECT, PCM and PCQ for Flocks of Boids Dataset.

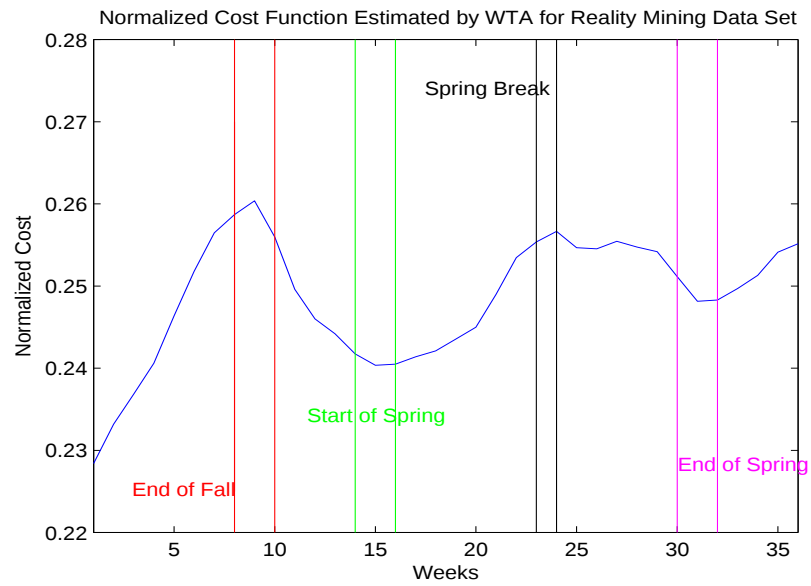


Figure 3.5: Detected Change Points for Reality Mining Data.

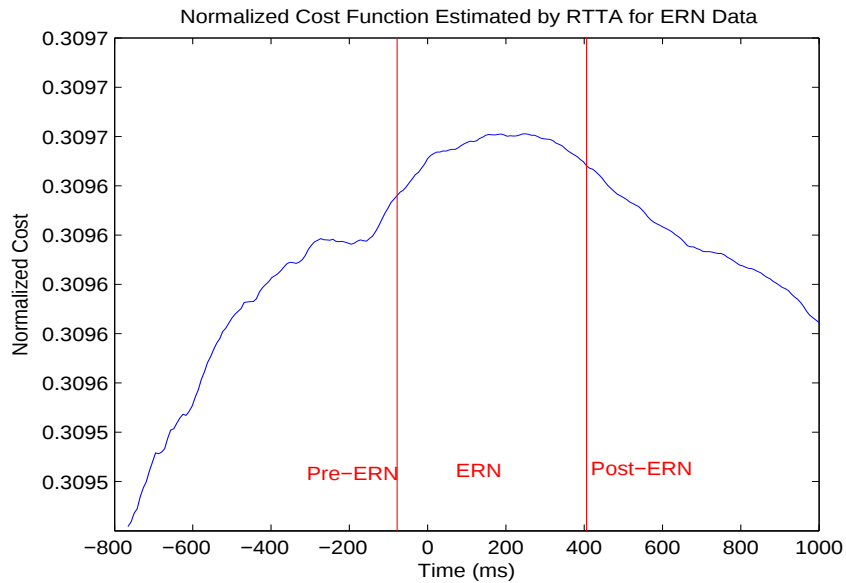


Figure 3.6: Detected Change Points for ERN Data.

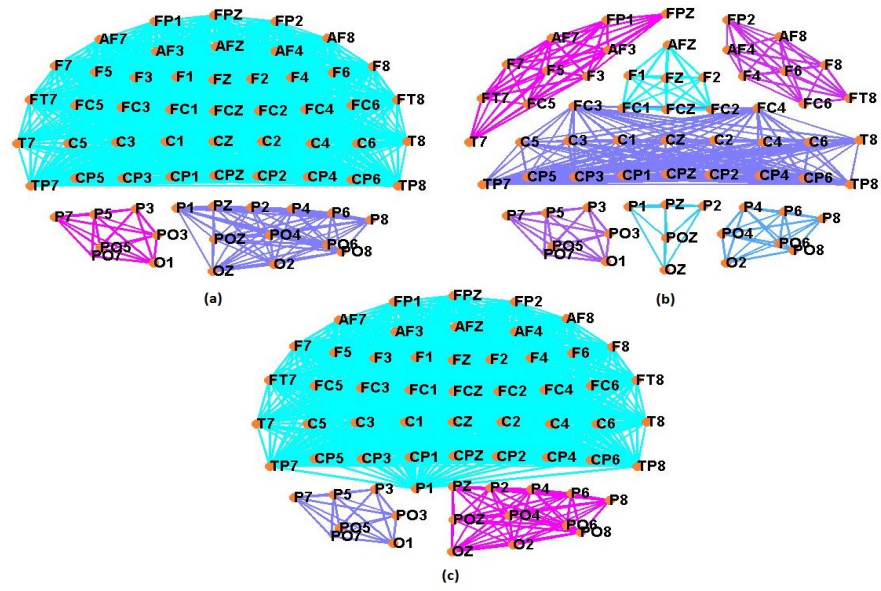


Figure 3.7: Community structure for the ERN networks obtained by RTTA: (a) Pre-ERN, (b) ERN, (c) Post-ERN.

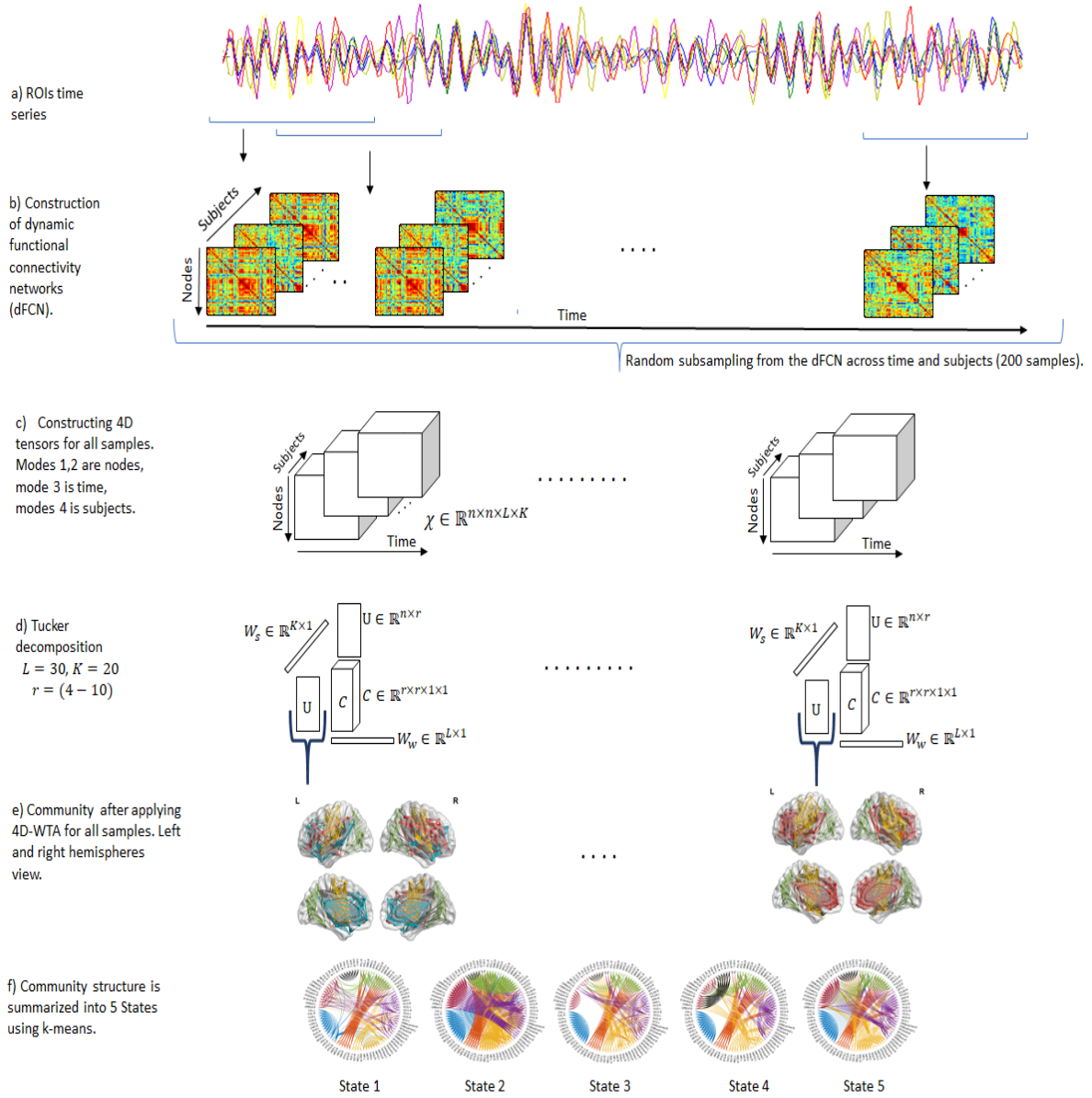
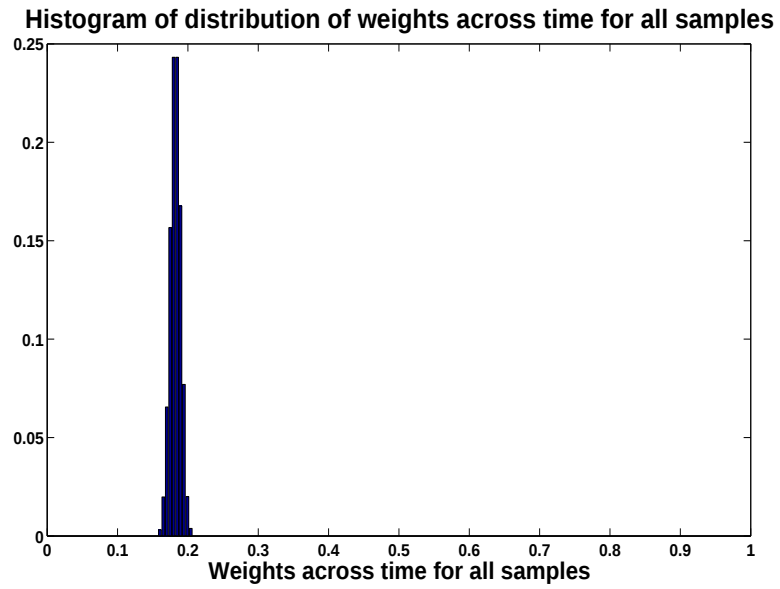
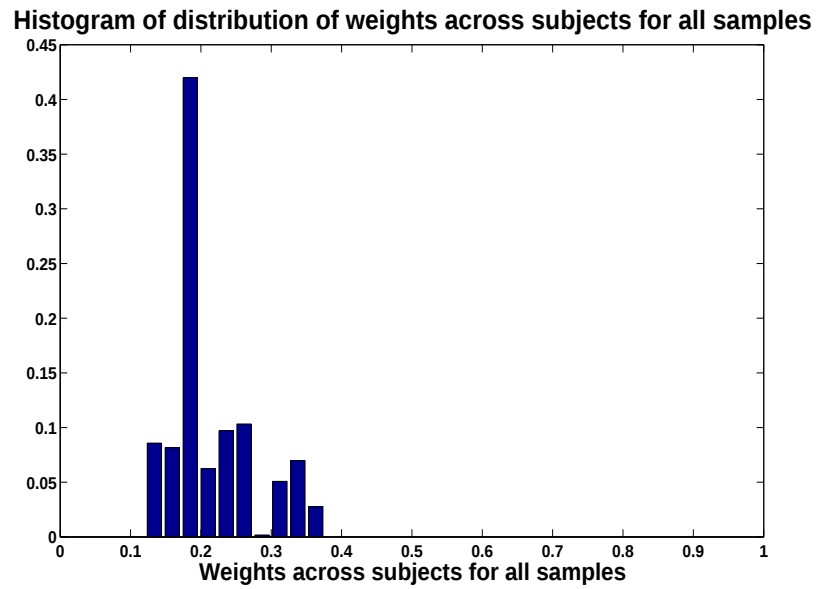


Figure 3.8: A flowchart of the application of 4D-WTA to rs-fMRI: (a) Time series from each ROI; (b) dFCNs constructed across time and subjects using Pearson correlation coefficient; (c) After random subsampling across time and subjects, 4 dimensional FC tensors are constructed for each sample. Each tensor consists of 30 random time points from each subject; (d) Tucker decomposition of each FC tensor; (e) Community structure found by 4D-WTA for each sample; (f) The detected community structure is summarized into 5 functional connectivity states defined through community detection using k-means.



(a)



(b)

Figure 3.9: (a) Distribution of weights across time for all samples, (b) Distribution of weights across subjects for all samples.

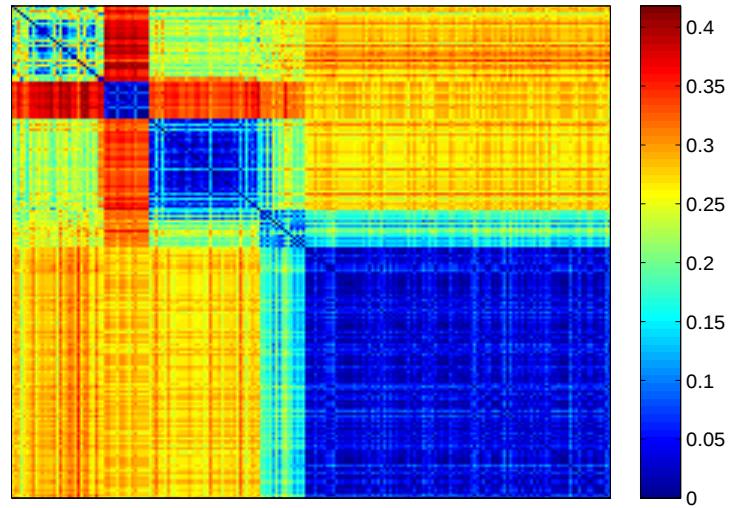


Figure 3.10: The similarity among the community structures obtained by 4D-WTA across the 200 subsamples across time and subjects. The similarity between the different community structures is quantified by variation of information. Samples with similar community structure are grouped together by k-means.

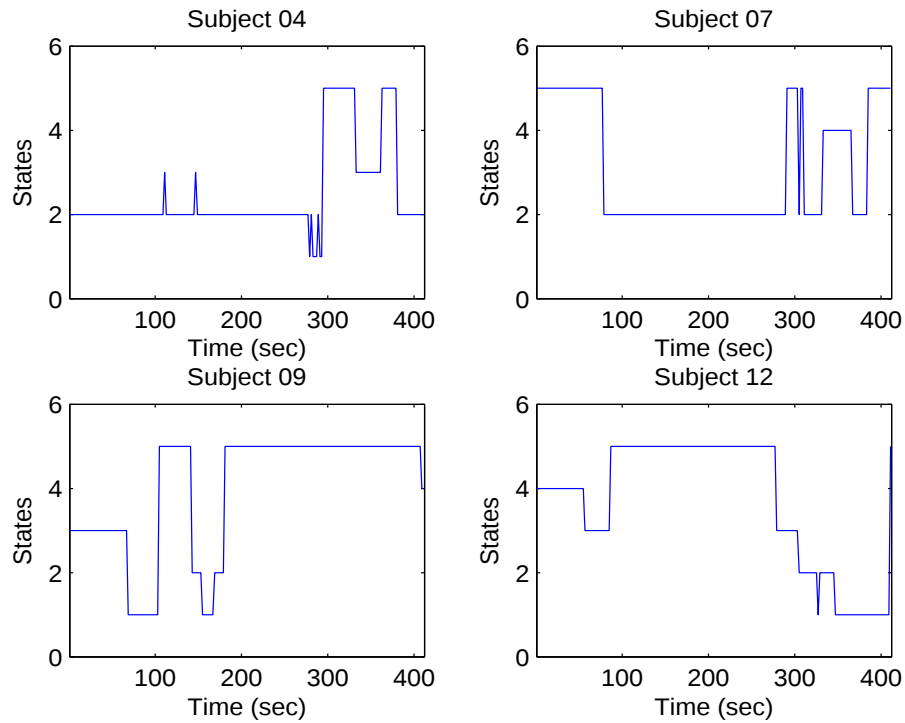
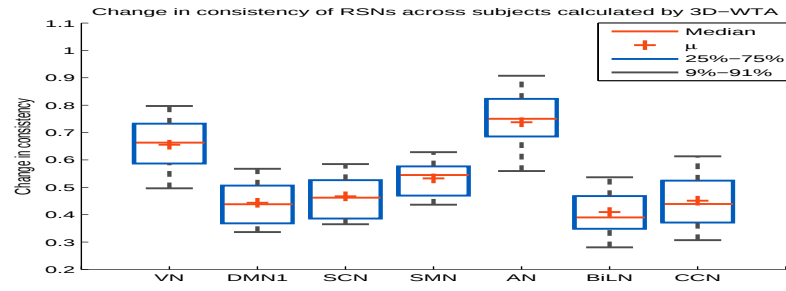
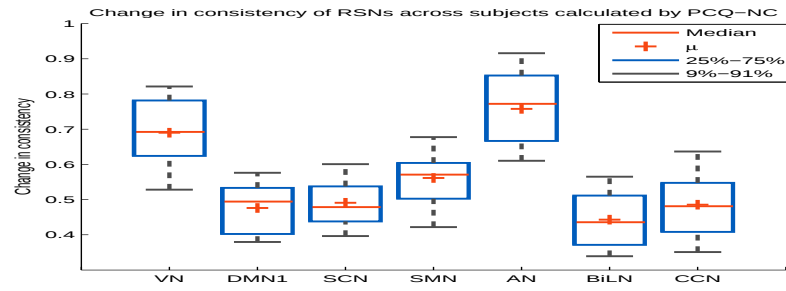


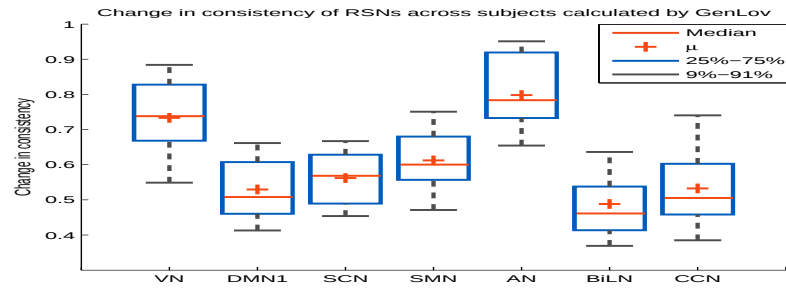
Figure 3.11: State transition profiles for subjects 4, 7, 9 and 12.



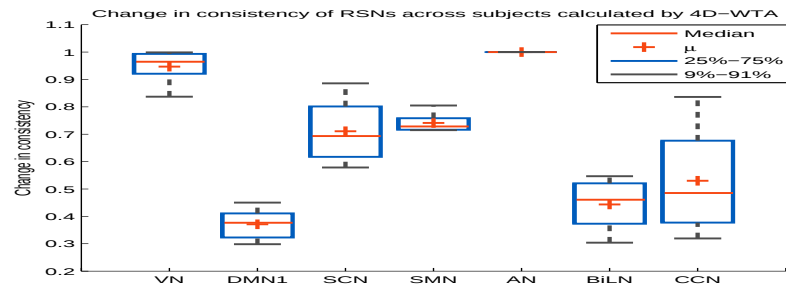
(a)



(b)



(c)



(d)

Figure 3.12: Variation in RSNs' consistency across subjects calculated by (a) 3D-WTA, (b) PCQ-NC, (c) GenLov, (d) 4D-WTA.



# **Chapter 4**

## **Detecting Brain Dynamics During Resting**

### **State: A Tensor Based evolutionary**

### **Clustering Approach**

#### **4.1 Introduction**

With the advancement of neuroimaging technology, researchers are able to collect digital data for the brain across different modalities, spatial and temporal resolutions. With the growth in neuroimaging data, it is now possible to map human brain activity as a complex network. In particular, the nodes of the network correspond to the different regions of the brain and the edges are quantified through the interactions between these regions.

In recent years, different approaches focused on understanding the functional mechanisms of human brain. A particular field of interest is understanding the spontaneous activity of the brain during resting state. This was accomplished by investigating the functional connectivity of the brain networks from functional magnetic resonance imaging (fMRI) during resting state. fMRI is a noninvasive neuroimaging modality that measures the changes of blood oxygenation level-

dependent (BOLD) signal. FC is defined as the statistical dependency between different brain regions and has been quantified through the computation of Pearson's correlation between regions of interest (ROIs) [54–56]. Increased interest in studying resting state fMRI (rs-fMRI) has shown that the different brain regions of the motor cortex are active even in the absence of a motor task [57]. Since that time, rs-fMRI has been widely used as one of the major tools to investigate and explore FC [58, 59, 130].

Many of the current studies on rs-fMRI reported a set of functional connectivity networks that appears consistently in healthy subjects such as default mode network, visual network, somatomotor network, executive control network and auditory network [58, 59]. A major network of interest during resting state is the default mode network (DMN), which consists mainly of posterior cingulate gyrus, precuneus, anterior cingulate gyrus, angular gyrus and medial frontal gyrus. DMN has attracted a lot of attention since it reflects the default neural activity and is deactivated during task related events [122, 123, 131].

Early work on rs-fMRI assumed stationarity of functional connectivity, however, recent studies revealed evidence for dynamic nature of FC during resting state [53, 60]. Different methods have been developed to study the dynamics of the functional connectivity networks (FCNs). These methods include clustering based methods [55], statistical state modeling methods [68] and subspace analysis based methods [56]. Most of the current work focuses on extracting FC states and reduces the dimension of the dynamic FCNs (dFCNs) by vectorizing the connectivity matrices before extracting the FC states. This procedure may result in the the loss of the topological structure of the FCN. These approaches also reduce the dFCNs into a few common basis functions or states without explicitly identifying the topological organization within each state. A well-known method to study the network organization is community detection [132]. Community detection algorithms have attracted lots of attention as the detected structure reflects the topological relationships within

the network [50, 133]. However, most of the current work on community detection focuses on detecting the community structure of static networks, such as standard spectral clustering [7].

Over the past decade, different evolutionary clustering approaches have been developed to detect the community structure of dynamic networks. These approaches include simple extensions of static clustering methods [31, 32], statistical modeling based methods [33], tensor based methods [34, 134] and modularity based methods [35]. Extensions of static clustering methods such as k-means and agglomerative hierarchical clustering approaches were developed in [31] for dynamic networks. A cost function that considers both the cluster membership at the current time and the deviation from the previous time point was introduced. In [32], two spectral clustering based frameworks were introduced: preserving cluster quality (PCQ) and preserving cluster membership (PCM). In both frameworks, the clustering depends on a cost function to guarantee temporal smoothness. However, this cost function requires a *priori* knowledge about the changes in the community structure and depends on the choice of a changing parameter. Under statistical modeling based methods, authors in [33] assumed a statistical model for the evolution of the adjacency matrix and an adaptive forgetting factor for evolutionary clustering and tracking (AFPECT) was introduced. In AFPECT, static clustering methods were performed after smoothing the proximity between objects over time. However, the AFPECT depends on a stochastic block model for the adjacency matrix and requires higher computational cost compared to static clustering to estimate the forgetting factor. Under tensor based methods, a non-negative tensor factorization approach was introduced in [34]. In this approach, the community structure of the temporal network is detected using PARAFAC, where all the successive adjacency matrices in the temporal network were represented by a single 3-way tensor. However, this representation of the temporal network may not preserve the community structure of the different time points. Under modularity based methods, a statistical method based on greedy modularity optimization has been proposed in [35]

to identify the changing modular organization in temporal multiplex networks, yet this approach has some drawbacks since it depends on modularity optimization as discussed in [20].

In this Chapter, we introduce a two step dynamic community detection algorithm to identify the network community structure of dFCN of the brain. First, we introduce an information-theoretic function to reduce the dFCN across time and identify the time points that carry similar topological information to combine them as a multi-way array or a tensor. In the second step of the algorithm, a tensor based approach is developed to identify the community structure of the reduced dFCN over time and summarize it in a set of "FC patterns" or "FC states". The contributions of the proposed work are four-fold. First, the proposed approach relies on quantum theory principle to reduce the whole dFCN into a few distinct network states that have similar topological information. This approach identifies the time points with similar network organization, unlike simple averaging methods, the proposed approach optimizes the contribution of each time point within each state through an optimized weighted average where the common subspace and the weights across time are optimized, simultaneously. Second, the proposed approach preserves the topological structure of the time points in each state, where the FC matrices from the original dFCN are used to identify the community structure within the state, unlike most recent studies that vectorize the network before clustering. Third, the proposed approach detects and tracks the community structure of the dFCN across time without any assumptions about the network structure or the number of clusters or community membership across time. Finally, the proposed approach provides evidence for the dynamic behavior of the RSNs and reveals the fluctuations in the community membership for the different ROIs.

## 4.2 Materials and Methods

### 4.2.1 Data and Preprocessing

The data used in this chapter is from Bangor rs-fMRI dataset. This dataset is part of 1000 Functional Connectomes Project [115]. The dataset includes 20 healthy subjects (males, aged between 19 – 38). The rs-fMRI data was collected using a 3T Magnet scanner from all subjects while their eyes were open. TR equals to 2s yielding 34 slices and 265 time points. The pre-processing was done using SPM12 [116] and CONN [117] functional connectivity toolboxes. The structural images were registered to the mean functional image and spatially normalized to MNI space for each subject. Slice timing and motion correction were performed on every functional image. The functional images were regionally parcellated by automated anatomical labeling (AAL) atlas [118] and smoothed by spatial convolution with 4-mm FWHM Gaussian kernel. Confounds such as motion parameters obtained from realignment and BOLD signals obtained from white matter and CSF masks were regressed out and band-pass (0.008-0.09 Hz) temporal filtering was applied to functional images of each subject.

In order to study the brain dynamics over time, we focus our analysis on some well-known resting state networks (RSNs) including the default mode network (DMN), auditory network (AN), visual network (VN), somatomotor network (SMN), bilateral limbic network (BiN), subcortical network (SCN) and cognitive control network (CCN). These networks are commonly defined by a set of ROIs. In this study, the set of ROIs that are categorized within these RSNs are summarized in Table 4.1. [55, 56]

|     |                                            |
|-----|--------------------------------------------|
| VN  | (43 – 56)                                  |
| AN  | (79 – 82)                                  |
| BiN | (21, 22, 27, 28, 37 – 42, 83, 84, 87, 88)  |
| DMN | (3 – 6, 23 – 26, 31 – 36, 65 – 68, 85, 85) |
| SMN | (1, 2, 17 – 20, 57 – 60, 63, 64, 69, 70)   |
| SCN | (71 – 78)                                  |
| CCN | (61, 62, 89, 90, 7 – 16, 29, 30)           |

Table 4.1: The ROIs that defines the RSNs. The numbers in the table refer to the ROIs defined in Table 1.

## 4.2.2 Dynamic Functional Connectivity Networks (dFCNs)

### 4.2.2.1 Dynamic Functional Connectivity Network Construction

The dynamic functional connectivity networks are constructed for each subject by computing Pearson correlation coefficient,  $\rho$ , between different ROIs (90 ROIs: with 45 ROIs in the left and 45 ROIs in the right hemispheres, listed in Table 1) using a sliding window approach with a rectangular window of length equal to 60 TRs and an overlap of 1 TR [55, 56, 60].

For each subject, a sequence of weighted and undirected graphs  $\{\mathbf{A}_k^{(t)}\}_{t_1, \dots, t_M}$  were constructed with  $A_k^{(t)}(i, j) = \rho_k^{(t)}(i, j)$ , where  $\mathbf{A}_k^{(t)} \in \mathbb{R}^{n \times n}$  is the adjacency matrix representing the network at time step  $t$  for subject  $k$ ,  $n$  is the total number of nodes (ROIs) and  $M$  is the total number of time points. The final dFCN at each time point is averaged across all subjects [135–137] to obtain  $\{\mathbf{A}^{(t)}\}_{t_1, \dots, t_M}$ , where  $A^{(t)}(i, j) = \frac{1}{K} \sum_{k=1}^K \rho_k^{(t)}(i, j)$  and  $\mathbf{A}^{(t)} \in \mathbb{R}^{n \times n}$ .

### 4.2.2.2 Surrogate Data Generation for Testing Statistical Significance of the dFCN

Before applying the proposed clustering approach to the dFCN, we first apply thresholding to the connectivity matrices to obtain sparse adjacency matrices with nonzero edges corresponding to the

significant connections. We generated surrogate data [138] from the ROI time series and we kept all edges that were significantly different from the null model.

For each ROI, surrogate data was generated using iterative amplitude-adjusted Fourier transform (IAAFT) [139]. For each subject, we created 100 surrogate time series for each ROI. We then computed Pearson correlation coefficient between each pair of ROIs using a sliding rectangular window of length 60 TRs and an overlap of 1 TR. This process was repeated for all 100 surrogate time series. For each pair of ROIs,  $i$  and  $j$ , and for each window, the correlation values obtained from the null model were normally distributed with a sample mean of  $\mu^{(t)}(i, j)$  and a standard deviation of  $\sigma^{(t)}(i, j)$ . For each edge,  $A_k^{(t)}(i, j)$ , in the original dFCN of each subject, we calculated the absolute  $z$ -score as  $z_k^{(t)}(i, j) = |A_k^{(t)}(i, j) - \mu^{(t)}(i, j)| / \sigma^{(t)}(i, j)$  and removed all the edges with  $z_k^{(t)}(i, j) < 3$ , which corresponds to a statistical test with significance of 99.8%. The  $\mathbf{B}_k^{(t)}\}_{t_1, \dots, t_M}$ . The final  $z$ -score thresholded dFCN,  $\{\mathbf{B}^{(t)}\}_{t_1, \dots, t_M}$ , is constructed at each time point by averaging  $\{\mathbf{B}_k^{(t)}\}_{t_1, \dots, t_M}$  across subjects, where  $B^{(t)}(i, j) = \frac{1}{K} \sum_{k=1}^K z_k^{(t)}(i, j)$ .

### 4.2.3 Reducibility of Brain dFCNs

The first step in our proposed approach is the reduction of dFCN. The goal of the reduction step is to cluster the time points that possess similar structure into a common partition to identify the distinct network states. The  $z$ -score thresholded dFCN,  $\{\mathbf{B}^{(t)}\}_{t_1, \dots, t_M}$ , constructed from surrogate data is used to reduce the original dFCN. The reduction method relies on quantum theory principles to group similar time points by maximizing the distinguishability between the reduced temporal network and the corresponding fully aggregated graph defined as,  $\mathbf{B}_{agg} = \sum_{m=1}^M \mathbf{B}^{(t_m)}$ . A detailed description and derivation of the method can be found in [140].

The method can be summarized in three main steps: (i) computation of Jensen-Shanon divergence,  $D_{JS}$ , between all pairs of time points in the temporal network, (ii) performing classical

hierarchical clustering of the  $M$  time points and, (iii) calculating the relative entropy,  $q$ , which is used as the quality function of the resulting partition. The partition that maximizes  $q$  is chosen.

In the first step,  $D_{JS}$  is computed between all pairs of time points, which is defined in terms of Von Neumann entropy of each graph,  $\mathbf{B}^{(t)}$ ,

$$h(\mathbf{B}^{(t)}) = -tr(\mathbf{L}_{\mathbf{B}^{(t)}} \log_2 \mathbf{L}_{\mathbf{B}^{(t)}}) = -\sum_{i=1}^n \lambda_i^{(t)} \log_2 \lambda_i^{(t)}, \quad (4.1)$$

where  $\mathbf{L}_{\mathbf{B}^{(t)}} = \frac{(\mathbf{D}_{\mathbf{B}^{(t)}} - \mathbf{B}^{(t)})}{\sum_{i,j=1}^n B^{(t)}(i,j)}$  is the rescaled Laplacian of graph  $\mathbf{B}^{(t)}$  which has all the properties of a density matrix and  $\{\lambda_1^{(t)}, \lambda_2^{(t)}, \dots, \lambda_n^{(t)}\}$  are the eigenvalues of  $\mathbf{L}_{\mathbf{B}^{(t)}}$ ,  $tr$ , is the trace of a matrix and  $\mathbf{D}_{\mathbf{B}^{(t)}}$ , is the diagonal degree matrix corresponding to  $\mathbf{B}^{(t)}$ .

Measuring the dissimilarity between any two pairs of time points in the temporal network can be achieved by computing  $D_{JS}$  between them. Given the rescaled Laplacian of any pair of time points as  $\mathbf{L}^{t_i}$  and  $\mathbf{L}^{t_j}$ , we can compute a new density matrix as  $\mathbf{L} = \frac{1}{2}(\mathbf{L}^{t_i} + \mathbf{L}^{t_j})$ , then  $D_{JS}$  can be calculated as,

$$D_{JS}(\mathbf{L}^{t_i} || \mathbf{L}^{t_j}) = h(\mathbf{L}) - \frac{1}{2}(h(\mathbf{L}^{t_i}) + h(\mathbf{L}^{t_j})). \quad (4.2)$$

In the second step, classical hierarchical clustering of the  $M$  time points is performed using  $D_{JS}$  between all pairs of time points as the dissimilarity measure. At each step of the algorithm, the time points that are separated by the smallest  $D_{JS}$  are aggregated by summing up the corresponding adjacency matrices. The distances between the resulting new pairs of the reduced temporal network are then updated. This procedure is repeated  $M - 1$  times until we form the fully aggregated graph,  $\mathbf{B}_{agg}$ . As a result of the hierarchical clustering algorithm, we get a dendrogram that is associated with the aggregation procedure.

In the third step, the relative entropy,  $q$ , is used as the quality function to quantify the distinguishability between the fully aggregated graph,  $\mathbf{B}_{agg}$ , and the reduced temporal network,  $\mathbf{G}_{red}$ , where  $\mathbf{G}_{red} : \{\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \dots, \mathbf{G}^{(\bar{M})}\}$ , with  $\bar{M} \leq M$ . The matrix,  $\mathbf{G}^{(\alpha)}$ , is either one of the  $z$ -score thresholded matrices or the sum of two or more of them with  $\alpha = 1, \dots, \bar{M}$ . The quality function of the reduced network is defined as,

$$q(\mathbf{G}_{red}) = 1 - \frac{\bar{H}(\mathbf{G}_{red})}{h(\mathbf{B}_{agg})}, \quad (4.3)$$

where  $\bar{H}(\mathbf{G}_{red})$  is the average Von Neumann entropy of the reduced  $z$ -score thresholded dynamic network. Finally, the partition that maximizes the relative entropy is chosen, where the  $z$ -score thresholded matrices of the time points that have similar information are clustered together.

The output of the reduction step is a set of multidimensional arrays or tensors, where the slices of each tensor consists of adjacency matrices from the original dFCN. The adjacency matrices included within each tensor are marked and selected from the original dFCN through the reduction step as they carry similar topological information. The reduced dFCN can be defined as  $\mathcal{Y} : \{\mathcal{Y}^{(1)}, \dots, \mathcal{Y}^{(\bar{M})}\}$  with  $\bar{M} \leq M$ , where  $\mathcal{Y}^{(\tau)} \in \mathbb{R}^{n \times n \times L_T^{(\tau)}}$  with  $\tau = 1, \dots, \bar{M}$  and  $L_T^{(\tau)}$  is the number of time points included in the tensor  $\mathcal{Y}^{(\tau)}$ , which refers to the third dimension of the tensor and might vary between multiple tensors. For each tensor,  $\mathcal{Y}^{(\tau)}$ , we define  $L^{(\tau)}$  as a vector of indices of the time points from the original dFCN included in the tensor. The  $l^{th}$  frontal slice of the 3-mode tensor,  $\mathcal{Y}^{(\tau)}(:, :, l)$ , is the normalized adjacency matrix  $\mathbf{A}_N^{L^{(\tau)}(l)}$ , where  $l \in L^{(\tau)}$ .

## 4.2.4 Dynamic Network Clustering

### 4.2.4.1 HOOI Clustering Based Approach

The evolutionary community structure of the reduced dFCN,  $\mathcal{Y}$ , can be determined by identifying the community structure for each partition,  $\mathcal{Y}^{(\tau)}$ . Each  $\mathcal{Y}^{(\tau)}$  can be represented as a multidimensional array and the optimal subspace  $\mathbf{U}^{(\tau)}$  that describes the collection of adjacency matrices can be found through spectral clustering of the weighted average of the adjacency matrices. This results in an optimization over both the subspace vectors matrix,  $\mathbf{U}^{(\tau)}$ , and the weight vector,  $W^{(\tau)}$ , for each partition. The optimization problem in (3.3) can then be rewritten for each tensor as,

$$\max_{\mathbf{U}^{(\tau)}, W^{(\tau)}} \left\| \mathbf{U}^{(\tau)\dagger} \left( \sum_{l \in L^{(\tau)}} w_l^{(\tau)} \mathbf{A}_N^{L^{(\tau)}(l)} \right) \mathbf{U}^{(\tau)} \right\|_F^2, s.t. \mathbf{U}^{(\tau)\dagger} \mathbf{U}^{(\tau)} = \mathbf{I}, W^{(\tau)} \geq 0, \|W^{(\tau)}\|_2 = 1, \quad (4.4)$$

where  $w_l^{(\tau)}$  are the weights for the adjacency matrices within  $\mathcal{Y}^{(\tau)}$  and  $L^{(\tau)}$  refers to the vector of indices of the time points from the original dFCN included in the tensor. The first two modes of the tensor refer to the connectivity and the third mode refers to time. A variable size set of tensors are constructed to represent the reduced dFCN across time and (4.4) can be reformulated in terms of Tucker decomposition, for each tensor as follows,

$$\max_{\mathbf{U}^{(\tau)}, W^{(\tau)}} \left\| \mathcal{Y}^{(\tau)} \times_1 \mathbf{U}^{(\tau)\dagger} \times_2 \mathbf{U}^{(\tau)\dagger} \times_3 W^{(\tau)\dagger} \right\|_F^2, s.t. \mathbf{U}^{(\tau)\dagger} \mathbf{U}^{(\tau)} = \mathbf{I}, \|W^{(\tau)}\|_2 = 1, \quad (4.5)$$

where  $\mathbf{U}^{(\tau)}$  is the basis along the first and second modes and  $W^{(\tau)}$  corresponds to the weighting vector across time. Since the networks are undirected, the components along the first two modes are the same. The optimization in (4.5) can be solved by higher-order orthogonal iteration (HOOI), where in each iteration either the matrix  $\mathbf{U}^{(\tau)}$  or the vector  $W^{(\tau)}$  is fixed to optimize the other one. This approach is summarized in the pseudo code in Algorithm 4.6.

---

**Algorithm 4.6** HOOI clustering based approach.

---

**Input:**  $\mathcal{Y} : \{\mathcal{Y}^{(1)}, \dots, \mathcal{Y}^{(\bar{M})}\}, L = \{L^{(1)}, \dots, L^{(\bar{M})}\}.$

**Output:** Clustering Labels.

```
1: for  $\tau = 1 : \bar{M}$  do
2:   Unfold the tensor  $\mathcal{Y}^{(\tau)}$  along mode 3 to get  $\mathbf{Y}_{(3)}^{(\tau)}$ .
3:   Initialize  $\mathbf{U}_0^{(\tau)}$  by HOSVD of  $\mathcal{Y}^{(\tau)}$ .
4:    $i = 0$ 
5:   Calculate  $\mathbf{W}_{i+1}^{(\tau)}$  as the dominant left singular vector of  $\mathbf{Y}_{(3)}^{(\tau)} \left( \mathbf{U}_i^{(\tau)} \otimes \mathbf{U}_i^{(\tau)} \right).$ 
6:   Compute  $\bar{\mathbf{A}} = \sum_{l \in L^{(\tau)}} (\mathbf{W}_{i+1}^{(\tau)})^l \mathbf{A}_N^{L^{(\tau)}(l)}$ , where  $\mathbf{A}_N^{L^{(\tau)}(l)} = \mathcal{Y}^{(\tau)}(:, :, l).$ 
7:   Obtain  $\mathbf{U}_{i+1}^{(\tau)}$  by eigenvalue decomposition of  $\bar{\mathbf{A}}$ .
8:   if  $\| \mathbf{U}_{i+1}^{(\tau)} - \mathbf{U}_i^{(\tau)} \|_F^2 > \varepsilon$  then
9:      $i = i + 1.$ 
10:    goto line 6.
11:  end if
12:  Use modularity to determine the number of clusters.
13:  Normalize the columns of  $\mathbf{U}^{(\tau)}$ .
14:  Apply  $k$ -means on  $\mathbf{U}^{(\tau)}$  to obtain clustering labels.
15: end for
```

---

#### 4.2.4.2 Choosing the Number of Clusters

The number of the clusters for each state were determined by modularity [10]. In the proposed approach, we calculated the modularity of the detected community structure for each tensor using the adjacency matrices for the time points included in each tensor separately. Then, an average modularity across all of the time points within that FC state was calculated, and the number of clusters that maximized the average modularity was chosen.

## 4.3 Results

### 4.3.1 dFCN Reduction

The first step of the proposed approach is to combine the time points that carry similar topological information, so that the time points with similar FC patterns are grouped together. Consequently,

the dimension of the dFCN is reduced while the distinguishability between the reduced dFCN and the fully aggregated graph is maximized. We performed reducibility for the subject averaged dFCN following the procedure explained in 4.2.2.2 and 4.2.3. The Jensen-Shannon distance matrix and quality function for the temporal network obtained after each step of the hierarchical clustering algorithm are shown in Fig.4.1 and Fig.4.2, respectively.

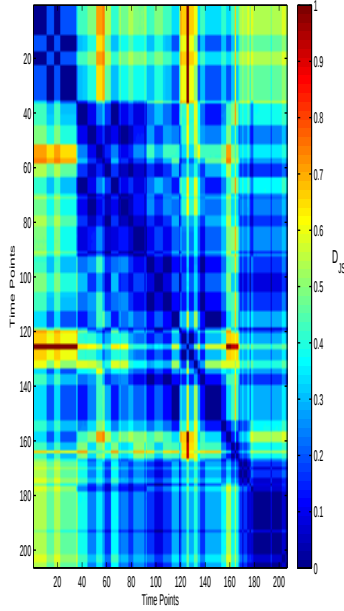


Figure 4.1: Jensen-Shannon divergence between all pairs of time points in the dFCN.

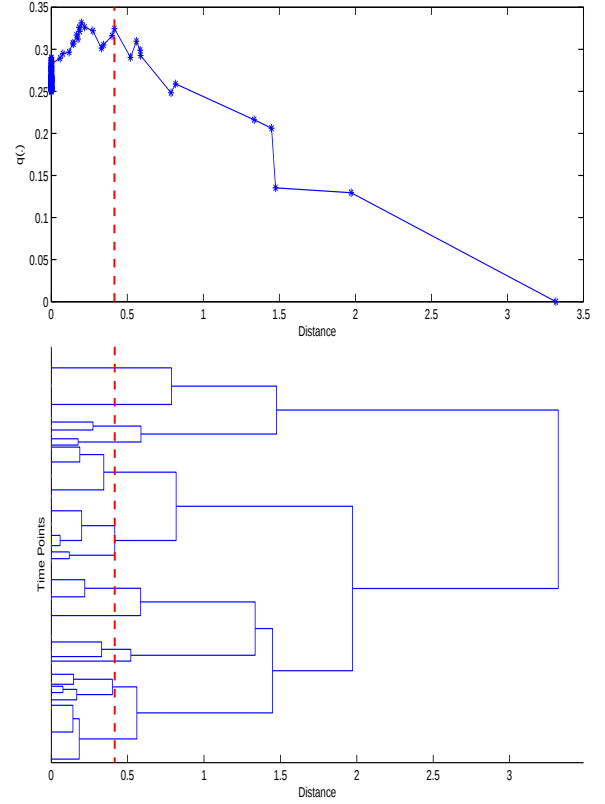


Figure 4.2: The dendrogram resulting from hierarchical clustering and the corresponding values of  $q$ . The dashed red line identifies the cut point of the dendrogram from the second peak of  $q$ .

From the dendrogram shown in Fig.4.2, the reduced dFCN consists of 12 FC states represented by 12 tensors. The time points included in each tensor carry similar information. As can be seen from Fig.4.2, we cut the dendrogram using the quality function curve. The cut point is chosen as the second peak of the quality function, since the partition resulting from the first peak produced

18 tensors and some of these tensors corresponded to less than 5 time points.

### 4.3.2 FC States Summarization

The HOOI based clustering approach is applied to analyze the brain functional connectivity dynamics. A set of 12 tensors are constructed corresponding to the reduced dFCN. The third dimension of the tensors represents the number of time points included in each partition. The proposed approach is applied to each tensor to detect its community structure. The constructed tensors were approximated by Tucker decomposition as  $\mathcal{Y}^{(\tau)} \approx \mathcal{C}^{(\tau)} \times_1 \mathbf{U}^{(\tau)} \times_2 \mathbf{U}^{(\tau)} \times_3 W^{(\tau)}$ , where  $\mathcal{C}^{(\tau)} \in \mathbb{R}^{r \times r \times 1}$  is the core tensor,  $\mathbf{U}^{(\tau)} \in \mathbb{R}^{90 \times r}$  is the orthogonal low rank component matrix along the connectivity mode and  $W^{(\tau)} \in \mathbb{R}^{L_T^{(\tau)} \times 1}$  is the weighting vector across time mode.  $r$  is the rank which corresponds to the number of clusters. The number of clusters for each tensor is determined using modularity, as explained in 4.2.4.2, and varied between 3 and 4.

| State | Tensor Size              | Time Points (seconds)                        | Reoccurrence percentage |
|-------|--------------------------|----------------------------------------------|-------------------------|
| 1     | $90 \times 90 \times 16$ | (120 – 140)(154 – 162)                       | 7.8%                    |
| 2     | $90 \times 90 \times 19$ | (141 – 153)(163 – 188)                       | 9.2%                    |
| 3     | $90 \times 90 \times 25$ | (189 – 206)(246 – 256)(282 – 298)(301 – 302) | 12%                     |
| 4     | $90 \times 90 \times 32$ | (207 – 245)(257 – 281)(299 – 300)            | 15.5%                   |
| 5     | $90 \times 90 \times 19$ | (303 – 314)(330 – 342)(386 – 396)            | 9.2%                    |
| 6     | $90 \times 90 \times 20$ | (315 – 329)(356 – 358)(452 – 470)            | 9.7%                    |
| 7     | $90 \times 90 \times 19$ | (343 – 355)(397 – 422)                       | 9.2%                    |
| 8     | $90 \times 90 \times 13$ | (360 – 384)                                  | 6.3%                    |
| 9     | $90 \times 90 \times 14$ | (423 – 451)                                  | 6.7%                    |
| 10    | $90 \times 90 \times 29$ | (471 – 503)(505 – 530)                       | 14%                     |

Table 4.2: Key Attributes of the Tensors Constructed from the Reduced dFCN.

In this chapter, we are investigating the whole brain dynamics by detecting the community

structure of the brain over time. This is achieved by summarizing the dynamic behavior of the brain by extracting a set of “FC states”. As the detected community structure across some states is the same, the final resulting FC states are summarized into 10 states. Table 4.2, summarizes the key attributes of the tensors constructed to detect the community structure of the dFCN states.

Fig. 4.3 shows the detected community structures for all of the FC states. For each state, the optimally weighted average of the adjacency matrices within the state is given along with the cluster assignments (left). The clusters are labeled using the major ROIs detected within the cluster, i.e. the cluster might contain ROIs from multiple RSNs, but it is labeled by the the most dominant RSN within the cluster. Moreover, the circular graph (right) shows the detected community structures using circularGraph toolbox [1] and the connections between the ROIs of the different RSNs.

The extracted FC states are analyzed with respect to two aspects. First, the frequency of occurrence of each FC state is computed across time. Second, the variability in the community membership of the core ROIs corresponding to different RSNs is investigated across FC states and the reconfiguration of the community membership of the ROIs is analyzed.

The different FC states are first analyzed based on their frequency of occurrence. States 4 and 10 are the two states with the highest reoccurrence percentage with relative frequency of occurrence at 15.5% and 14%, respectively. In State 4, a cluster which includes most of the core ROIs known to belong to DMN, is detected. State 10, on the other hand, has a large SMN cluster that also includes ROIs of the SCN, CCN and BiN. The reoccurrence percentage of the remaining states varied between 6.3% and 9.7%, as shown in Table 4.2.

For the variability in the community membership of the ROIs, we observed that some ROIs have a relatively stable community membership across the multiple states, i.e. these ROIs were clustered together across the different states. For example, ROIs of the VN revealed a consistent

community membership across all states similar to the results in [141]. On the other hand, the VN expanded to include the superior parietal gyrus (SPG) during States 1, 2, 5, 6, 8, 9 and 10. Similar findings were reported in [128], where the innovation-driven co-activation patterns (iCAPs) from rs-fMRI were extracted and an iCAP that consists of ROIs of the VN and SPG has been identified. Another set of ROIs that clustered consistently together across all states are the core ROIs of the SMN, including paracentral lobule (PCL), precentral gyrus (PreCG), postcentral gyrus (PoCG) and supplementary motor area (SMA). Even though these ROIs were clustered together across all states, the SMN was extended to include ROIs from the SCN in States 1 – 4, 9 and 10 as previously noted in [55]. Also, the HIP is included within the SMN across States 1 – 4, 6 and 10. In contrast, States 5 and 8 exhibited a separation of the ROIs of the SMN and SCN. Moreover, we observed that the ROIs of the AN, superior temporal gyrus (STG) and heschl gyrus (HES), were co-activated with the SMN across all states and this agrees with the results found in [128]. Similar to the other RSNs, some of the ROIs of the SCN were relatively stable, such as putamen (PUT) and pallidum (PAL), whereas thalamus (THA) and caudate (CAU) have a variation in their community membership between DMN, SCN and SMN [128, 142]. Furthermore, the ROIs of the BiN exhibited a dynamic behavior across different states. For instance, in States 5 and 7 – 9, the hippocampus (HIP) and parahippocampal gyrus (PHG) are included in the DMN, as these ROIs are known for playing an important role in long-term memory [124, 125], previous studies comprised them as part of the DMN [126]. During States 1-3, the HIP and AMYG joined the SMN and PHG joined the DMN. On the other hand, the ROIs HIP, PHG and AMYG are clustered together during State 10.

As the DMN is a major network of interest during resting state [122, 123], we will present some interesting features of the detected DMN cluster across the different states. In State 4, which scored the highest reoccurrence percentage, a large DMN cluster is detected. This cluster consists of the

core ROIs of the DMN, including precuneus (PCUN), Posterior cingulate gyrus (PCG), anterior cingulate gyrus (ACG), angular gyrus (AG), superior frontal gyrus (SFG), middle frontal gyrus (MFG), inferior frontal gyrus (IFG) and middle temporal gyrus (MTG). Most of the ROIs of the DMN were stable across all states except for the PCUN. The PCUN exhibited different community membership across the FC states, and it is known to play a crucial role in the DMN [127, 143]. In particular, it is affiliated with DMN cluster only in State 4. On the other hand, a consistent co-activation between PCUN and the ROIs of the VN in States 1, 3, 5, 6 and 10 is revealed as can be seen in Fig. 4.3, and this agrees with previously reported results in [55, 127], where a FC state that included PCUN and the ROIs of the VN has been identified. Moreover, States 7-9 detected a cluster that revealed co-activation between PCUN and the ROIs of SCN and CCN, similar to the results in [55, 144].

By studying the community structures detected across the different FC states, we observe that some RSNs show a relatively stationary behavior, while other RSNs show a dynamic behavior. From the community structures shown in Fig. 4.3, it can be seen that VN and AN are more stationary across the states, where the ROIs of these networks are clustered together across multiple states and experienced less variability in their community membership. On the other hand, the DMN, SMN, SCN, BiN and CCN exhibited a dynamic behavior across multiple states, as most of the ROIs of these RSNs keep changing their community membership across states. Moreover, in some states the aforementioned RSNs joined each other to form bigger clusters. For instance, SMN, SCN, AN and BiN formed one cluster during States 1, 2, 4, 6, 9 and 10 and DMN and ROIs from CCN and SCN formed one cluster during States 2, 5 and 6.

## 4.4 Conclusions

In this chapter, we introduce a two step dynamic community detection algorithm to investigate the whole brain FC dynamics during resting state. The first step in the proposed approach relies on an information-theoretic function to reduce the dimensionality of the dFCN across time. This reduction step identifies the time points that possess similar topological information across time and combines them as a tensor. The second step introduces a tensor based clustering approach to detect the community structure of the reduced dFCN and summarizes it into a set of FC states that reoccur across time. The extracted FC states provide evidence for the temporal evolution of the brain dFCN, as the ROIs of the different RSNs reconfigure their community membership across time.

The extracted FC states are evaluated through the frequency of reoccurrence across time and the variability in the community membership of the ROIs across the different states. The results show that some RSNs are more dynamic across time than others. In particular, we observed that the VN and AN are the most stationary networks across states, whereas DMN, SCN, BiN, CCN and SMN reconfigure their community membership more often across the different states. Moreover, the results show that the FC state that is repeated most frequently across time is State 4, which has a cluster that includes most of the core ROIs of the DMN. This result provides further evidence for the important role played by the DMN during resting state.

The proposed approach makes some key contributions to the field of dFCN analysis. First, the proposed algorithm depends on quantum theory principles to reduce the dFCN across time into a set of FC states. Second, the proposed approach uses a tensor representation for the extracted partitions to detect the community structure across them, and this presentation preserves the topological structure of the dFCN. Third, the proposed tensor based clustering approach optimizes the

contribution of each time point within each state. Finally, the proposed approach detects and tracks the community structure and its evolution in dFCN without any prior assumptions.

Future work will focus on different extensions of the proposed approach. First, the reduction of the dFCN will be performed across time and subjects in order to construct a reduced multi-layer temporal FCN network. Second, FC states will be extracted across time and subjects and will be used to evaluate and compare the variability in the dFCN community structure between group and individuals.

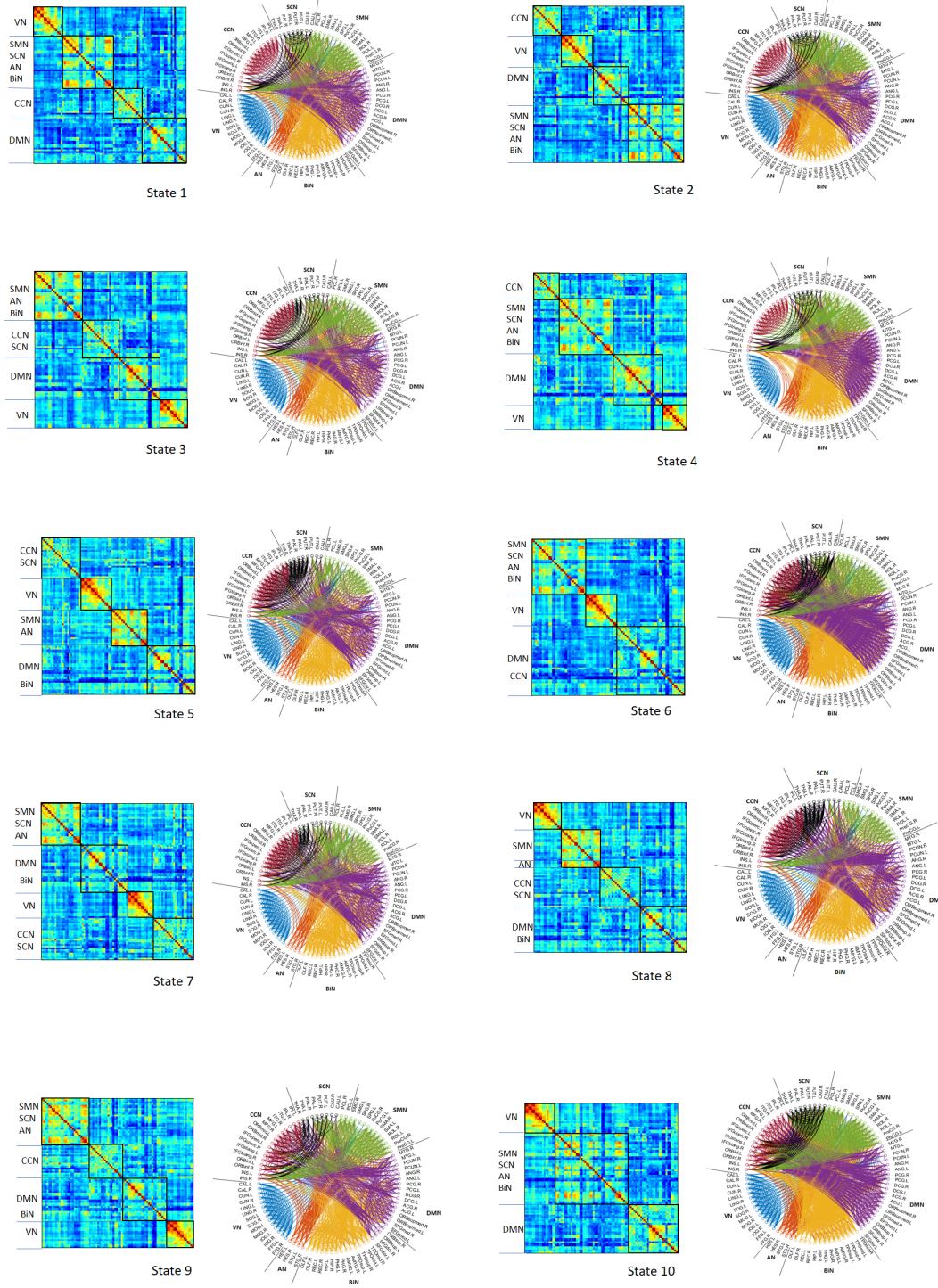


Figure 4.3: The detected community structure for the FC States (1 – 10). For each state, (left) the optimized weighted average of the adjacency matrices within the state. Each cluster is denoted by the main ROIs detected within the cluster. The clusters shown in the circular graphs (right) refer to the different RSNs defined in literature, and the connections between them are detected by the proposed algorithm. The number of clusters for the FC states varies between 3 and 4.

# **Chapter 5**

## **Temporal Block Spectral Clustering for Multi-layer Temporal Functional Connectivity Networks**

### **5.1 Introduction**

A network representation is useful for describing the structure of a large variety of complex systems in the biological, social and physical sciences [145]. Traditional studies of networks generally assume that nodes are connected to each other by a single type of static edge. However, this assumption is almost always an oversimplification, and ignores important network dynamics such as time dependence. Multi-layer networks explicitly incorporate multiple channels of connectivity in a system. A fundamental aspect of describing multi-layer networks is defining and quantifying the interconnectivity between different layers. One common approach to characterizing these connections is community detection.

While different community detection approaches have been developed for static networks [7, 9–11], there is a growing need to develop algorithms that detect the community structure

in multi-layer networks [6, 37]. Examples of multi-layer community detection methods include temporal clustering approaches [31–33], subspace learning based methods [146], stochastic block model (SBM) based multi-layer community detection methods [147, 148] and tensor based methods [149, 150]. More recently, a multi-objective optimization based genetic algorithm [41] was proposed for detecting and tracking community structure in dynamic multi-layer networks. However, none of the aforementioned methods take the inter-layer couplings into account and are thus mere extensions of multi-view data clustering approaches.

Recently modularity optimization and spectral clustering based methods for multi-layer networks with inter-layer connections have been proposed to address this limitation [35, 151]. In [35], a modularity optimization framework to determine the community structure in multi-slice networks is introduced. Since this framework depends on modularity, it has multiple drawbacks as discussed in [20]. Moreover, the approach is limited to the case where each node is connected to only itself in the adjacent time windows, i.e., multiplex networks with the inter-layer coupling set to a constant parameter. In [151], a spectral clustering method to detect the community structure in multi-layer networks with inter-layer couplings is introduced. However, this approach is limited as it constructs a single block adjacency matrix to represent all the graphs in the network and cannot capture the temporal variation in the community structure.

Therefore, there is an increasing need to look at the dynamic community structure of multi-layer temporal networks with unconstrained inter-layer or inter-temporal connectivity [6, 152–154]. In this chapter, a framework to detect and track the community structure in multi-layer temporal networks (MLTNs) is introduced. In the proposed approach, the temporal relationships between all node pairs within and across time window are modeled using a sliding window correlation approach. In this manner, non-trivial intra- and inter-adjacency matrices are constructed resulting in temporal supra-adjacency matrices. The dynamic community structure is detected by applying

spectral clustering on these temporal block supra-adjacency matrices. This work makes significant contributions to the field of dynamic community detection and differs from existing multi-layer community detection methods in some key ways. First, unlike the work in [35], the inter-layer adjacency matrices across time are not limited to the interactions between each node and itself across time and consider the interactions across all nodes and all time points. Second, unlike the work in [151] that constructs a single block matrix for the whole network, the community structure at each time point is updated by taking the history of networks into account. A particular focus of this chapter is community detection in dynamic functional connectivity networks (dFCNs) of the brain. More specifically, the dFCNs are modeled for each subject as a MLTN where the different brain regions correspond to the different nodes and the different time windows correspond to the different layers.

## 5.2 Methods

In this chapter, a temporal block spectral clustering approach is introduced to detect and track the community structure in temporal multi-layer temporal networks. In the proposed approach, a set of supra-adjacency matrices that captures the intra- and inter-temporal relations is constructed as follows.

### 5.2.1 Construction of multi-layer temporal networks

#### 5.2.1.1 Construction of a temporal network from signals

Given a set of multivariate signals,  $\{x_1(t), x_2(t), \dots, x_n(t)\}$  with  $t \in \{t_1, t_2, \dots, t_M\}$ , a set of adjacency matrices is constructed at each time point describing the temporal network with the signals corresponding to the nodes in the networks. A sliding window of length  $W$  is used to construct

both the intra-adjacency matrix,  $\mathbf{A}^{(t_m)}$ , with  $t_m \in \{t_W, t_{W+1}, \dots, t_M\}$ , and the inter-adjacency matrix,  $\mathbf{A}^{(t_k, t_l)}$ , with  $t_k, t_l \in \{t_W, t_{W+1}, \dots, t_M\}$ .  $\mathbf{A}^{(t_m)}$  captures the correlations between the different signals within a time window  $W$ , whereas  $\mathbf{A}^{(t_k, t_l)}$  captures the correlations between signals across different time windows.  $\mathbf{A}^{(t_k, t_l)}$  is defined as:

$$\mathbf{A}^{(t_k, t_l)} = \begin{bmatrix} \rho_{x_1 x_1}^{W(t_k, t_l)} & \rho_{x_1 x_2}^{W(t_k, t_l)} & \cdots & \rho_{x_1 x_n}^{W(t_k, t_l)} \\ \rho_{x_2 x_1}^{W(t_k, t_l)} & \rho_{x_2 x_2}^{W(t_k, t_l)} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{x_n x_1}^{W(t_k, t_l)} & \rho_{x_n x_2}^{W(t_k, t_l)} & \cdots & \rho_{x_n x_n}^{W(t_k, t_l)} \end{bmatrix},$$

where  $\rho_{x_i x_j}^{W(t_k, t_l)} = \rho(x_i(t_{k-W+1} : t_k), x_j(t_{l-W+1} : t_l))$  is the absolute correlation between  $x_i$  in time window  $W^{(t_k)}$  and  $x_j$  in time window  $W^{(t_l)}$ . The constructed  $\mathbf{A}^{(t_k, t_l)}$  measures the similarities between each node in window  $W^{(t_k)}$  and all other nodes in window  $W^{(t_l)}$ , unlike previous work that focused mostly on node-to-node edges [35].  $\mathbf{A}^{(t_m)}$  is considered as a special case of  $\mathbf{A}^{(t_k, t_l)}$ , where  $\mathbf{A}^{(t_m)} = \mathbf{A}^{(t_m, t_m)}$ . Note that  $\mathbf{A}^{(t_l, t_k)} = \mathbf{A}^{(t_k, t_l)\dagger}$ .

### 5.2.1.2 Construction of multi-layer temporal network

In this chapter, the variations in the community structure of temporal networks is studied. In order to achieve this, a set of block supra-adjacency matrices is constructed to reflect the relationships between the different networks across time.

In the proposed approach, a sliding window technique is used over the time-varying networks  $\mathbf{A}^{(t_m)}$  to construct a set of intra- and inter- block adjacency matrices,  $\mathbf{A}_{intra}$  and  $\mathbf{A}_{inter}$ , respectively. Each window considers  $T$  consecutive networks to construct:

$$\mathbf{A}_{intra}^{(t_{m'})} = \begin{bmatrix} \mathbf{A}^{(t_{m'}-T+1)} & 0 \\ & \ddots \\ 0 & \mathbf{A}^{(t_{m'})} \end{bmatrix}, \text{ and}$$

$$\mathbf{A}_{inter}^{(t_{m'})} = \begin{bmatrix} 0 & \mathbf{A}^{(t_{m'}-T+1, t_{m'}-T+2)} & \mathbf{A}^{(t_{m'}-T+1, t_{m'})} \\ \mathbf{A}^{(t_{m'}-T+2, t_{m'}-T+1)} & 0 & \vdots & \vdots \\ \vdots & \vdots & 0 \end{bmatrix}, \quad (5.1)$$

where  $t_{m'} \in \{t_{W+1}, \dots, t_M\}$ ,  $T < M - W + 1$  and  $\mathbf{A}_{intra}^{(t_{m'})}, \mathbf{A}_{inter}^{(t_{m'})} \in \mathbb{R}^{nT \times nT}$ . Note that  $T$  is different from  $W$  used in Section 5.2.1.1, where  $T$  is the number of networks included in constructing  $\mathbf{A}_{intra}^{(t_{m'})}$  and  $\mathbf{A}_{inter}^{(t_{m'})}$ , while  $W$  is the signal length used to construct the adjacency matrices.

For each set of  $T$  consecutive networks, a supra-adjacency matrix with the intra-adjacency matrices on the diagonal and the inter-adjacency matrices on the off-diagonal is constructed. As this matrix is symmetric with positive entries, the corresponding Laplacian is positive semi-definite.

$\mathbf{A}_{Supra}^{(t_{m'})} \in \mathbb{R}^{nT \times nT}$  is

$$\mathbf{A}_{Supra}^{(t_{m'})} = \mathbf{A}_{intra}^{(t_{m'})} + \mathbf{A}_{inter}^{(t_{m'})}. \quad (5.2)$$

A supra-normalized Laplacian matrix can then be constructed following the definition in [151, 155],

$$\mathbf{L}_{Supra}^{(t_{m'})} = \mathbf{D}_{Supra}^{(t_{m'})}{}^{-\frac{1}{2}} (\mathbf{L}_{intra}^{(t_{m'})} + \mathbf{L}_{inter}^{(t_{m'})}) \mathbf{D}_{Supra}^{(t_{m'})}{}^{-\frac{1}{2}}, \quad (5.3)$$

where  $\mathbf{L}_{intra}^{(t_{m'})} = \mathbf{D}_{intra}^{(t_{m'})} - \mathbf{A}_{intra}^{(t_{m'})}$  and  $\mathbf{L}_{inter}^{(t_{m'})} = \mathbf{D}_{inter}^{(t_{m'})} - \mathbf{A}_{inter}^{(t_{m'})}$ , with  $\mathbf{D}_{intra}^{(t_{m'})}$ ,  $\mathbf{D}_{inter}^{(t_{m'})}$  and  $\mathbf{D}_{Supra}^{(t_{m'})}$  defined as the degree matrices of  $\mathbf{A}_{intra}^{(t_{m'})}$  and  $\mathbf{A}_{inter}^{(t_{m'})}$  and  $\mathbf{A}_{Supra}^{(t_{m'})}$ , respectively.

### 5.2.2 Temporal block spectral clustering approach (TBSCA)

Block spectral clustering [155], [151] simplifies the clustering problem of multi-layer networks to the standard spectral clustering of a single  $\mathbf{L}_{Supra}$  that corresponds to all graphs in the network. The advantage of this approach is that it can provide a globally optimal solution for the clustering problem using the eigenvectors of  $\mathbf{L}_{Supra}$ .

In order to obtain temporally smooth communities, it is important to determine a community structure that simultaneously fits the network structure at time  $t$  and does not deviate from the past community structures. To accomplish this objective, a temporal block spectral clustering approach (TBSCA) is introduced. In this approach, a set of supra-adjacency matrices is constructed as described in (5.1) and (5.2) across time to capture the dynamics of the network. From each  $\mathbf{A}_{Supra}^{(t_{m'})}$ , the corresponding supra-normalized Laplacian is constructed,  $\mathbf{L}_{Supra}^{(t_{m'})}$ , for  $T$  consecutive networks using (5.3). The clustering problem is then formulated as,

$$\min_{\hat{\mathbf{U}}^{(t_{m'})} \in \mathbb{R}^{nT \times r_{m'}}} \text{trace} \left( \hat{\mathbf{U}}^{(t_{m'})\dagger} \mathbf{L}_{Supra}^{(t_{m'})} \hat{\mathbf{U}}^{(t_{m'})} \right) \text{ s.t. } \hat{\mathbf{U}}^{(t_{m'})\dagger} \hat{\mathbf{U}}^{(t_{m'})} = \mathbf{I}, \quad (5.4)$$

$\hat{\mathbf{U}}^{(t_{m'})}$  can be obtained from the eigendecomposition of  $\mathbf{L}_{Supra}^{(t_{m'})}$ . The number of clusters is determined using the eigen-gap criterion [7]. By applying  $k$ -means on  $\hat{\mathbf{U}}^{(t_{m'})}$ , a cluster label vector  $C \in \mathbb{R}^{1 \times nT}$  is obtained for the  $nT$  vertices. Finally, the cluster labels for time point  $t_{m'}$  are obtained as  $C(1, nT - n + 1 : nT)$ .

## 5.3 Results

### 5.3.1 Simulated networks

A temporal network with 150 nodes is generated for 60 time points. This temporal network is created with 3 clusters within time intervals 1 – 20 and 40 – 60 and 2 clusters within time interval 21 – 40. At each time point,  $\mathbf{A}_{intra}$  and  $\mathbf{A}_{inter}$  are constructed. The intra- and inter-adjacency matrices are generated where the within- and between-cluster edges are selected randomly from a uniform distribution in the range  $[0, 1]$  and  $[0, 0.5]$ , respectively. 30% of the within-cluster and 5% of the between-cluster edges are randomly set to 1. The TBSCA is implemented to detect and track the community structure of the network over time with  $T = 4$  time points. This simulation is repeated 100 times. The number of clusters at each time point is detected using the eigengap criterion. The proposed algorithm is able to detect the correct community structure and the change points across time as shown in Fig. 5.1. The performance of the proposed approach (TBSCA), BNLSC [151] and the generalized Louvain for multi-layer modularity maximization (GenLov) [35] is compared using Variation of information (VI) metric [4]. A VI value of 0 means the partitions are exactly the same and as the VI value increases, the degree of agreement between partitions decreases. The results in Fig.5.1 indicate that the proposed algorithm is very comparable to the GenLov method, which is considered as the state-of-the-art algorithm for detecting the community structure in temporal networks, and is better than BNLSC.

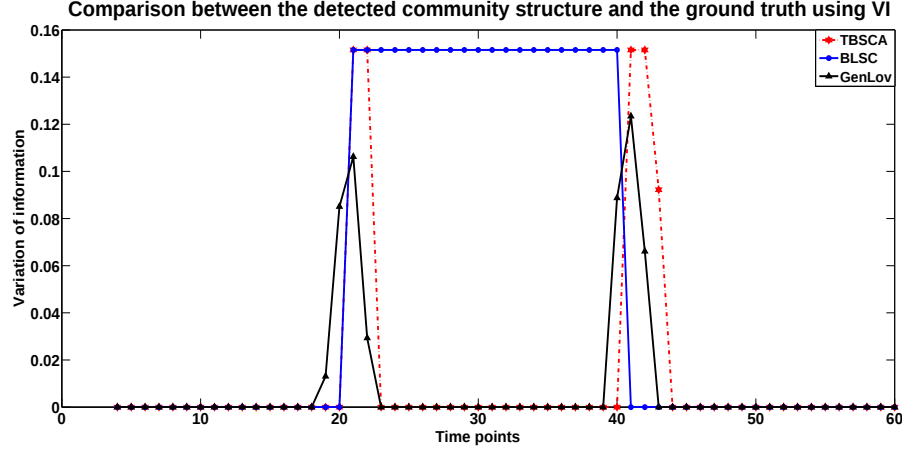


Figure 5.1: Comparison between the detected community structure by TBSCA, BNLSC and GenLov and the ground truth for the synthetic data set in terms of variation of information. The number of clusters changes from 3 to 2 at time point 21, and from 2 to 3 at time point 41.

### 5.3.2 Community Detection in dynamic functional connectivity networks (dFCNs)

Resting state fMRI (rs-fMRI) is commonly used to investigate spontaneous brain activity. Previous work [55, 59] indicates that there is a set of resting state networks (RSNs) that appears consistently in healthy subjects. These RSNs are default mode network (DMN), visual network (VN), somato-motor network (SMN), cognitive control network (CCN), bilateral network (BiN) and auditory network (AN). Even though early studies in rs-fMRI assume stationary functional connectivity, recent studies revealed evidence for temporal evolution of the brain functional connectivity during resting state [53, 60].

#### 5.3.2.1 Data and preprocessing

Bangor rs-fMRI dataset (<http://www.nitrc.org/projects/fcon1000>) which includes 20 healthy subjects (males, aged between 19 – 38) is used in this chapter. The rs-fMRI data was collected using a 3T Magnet scanner from all subjects while their eyes were open. TR equals to 2s yielding 34 slices

and 265 time points. The pre-processing was done using SPM12 (<http://www.fil.ion.ucl.ac.uk/spm/>) and CONN (<http://www.nitrc.org/projects/conn>) toolboxes. The structural images were registered to the mean functional image and spatially normalized to MNI space for each subject. Slice timing and motion correction were performed on every functional image. The functional images were regionally parcellated by automated anatomical labeling (AAL) atlas [118] to 90 regions of interest (ROIs) and smoothed by spatial convolution with 4-mm FWHM Gaussian kernel. Confounds such as motion parameters obtained from realignment and BOLD signals obtained from white matter and CSF masks were regressed out and band-pass (0.008-0.09 Hz) temporal filtering was applied to functional images of each subject.

### 5.3.2.2 Dynamic Community structure

The proposed approach is applied to dFCNs constructed from rs-fMRI for each subject following the steps described in Sec. 5.2.1 with  $W = 60$  (2 minutes) similar to previous studies [56]. The constructed dFCN consists of 206 time points. The TBSCA with  $T = 2$  networks is applied to detect and track the community structure for each subject separately. The detected number of clusters varied between (2 – 7) for the different subjects across time.

The application of the TBSCA to multi-layer temporal networks for each subject resulted in a community structure for each subject at each time point. For a typical subject, Fig. 5.2 shows the similarity between the detected community structures across consecutive time points measured by VI. From Fig. 5.2, it can be seen that the detected community structure changes smoothly across time and there are certain time points where the structure shows significant changes indicating the dynamics of resting state connectivity. Fig. 5.3, displays the detected community structure at some of these change points. In an attempt to quantify the similarity of the community structure across time and subjects, a  $4100 \times 4100$  similarity matrix is constructed to reflect the degree of agreement

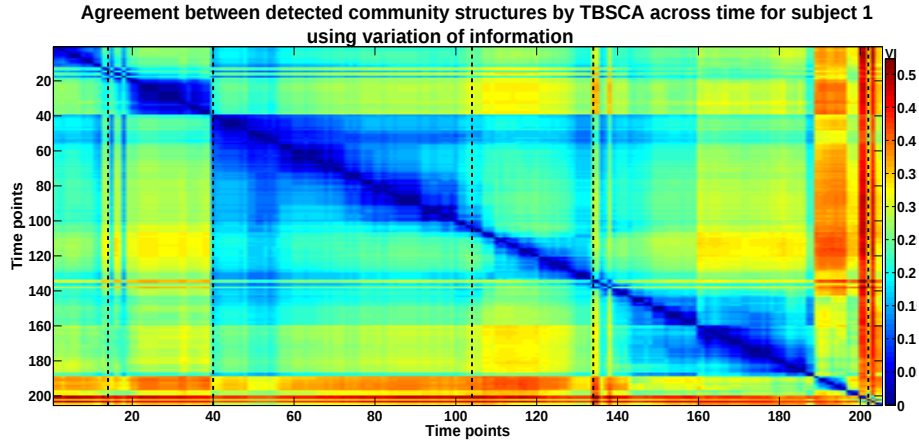


Figure 5.2: Agreement between the detected community structures across time for subject 1 measured by variation of information (VI). The dashed lines refer to the detected change points.

between the detected community structures across time and subjects in terms of VI, as shown in Fig. 5.4. From Fig. 5.4, it can be seen that the dFCNs exhibit a smooth variation in the community structure across time for a given subject indicating the dynamic behavior of the FCNs during rest. Furthermore, the community structure of the dFCNs for the different subjects follows different patterns across time as during resting state the subjects are not necessarily in the same network state at the same time.

### 5.3.2.3 Consistency of the RSNs across time

In order to quantify the dynamics of the RSNs across time and subjects, the consistency of each RSN is calculated. In order to calculate the consistency of a RSN, the recruitment coefficient defined in [129] is calculated for each ROI within the RSN. The ROI's recruitment coefficient is defined as the probability that the ROI is in the same cluster as other ROIs of its own subnetwork at each time point. The RSN's consistency is then calculated as the average of the recruitment coefficients of all ROIs that correspond to that RSN.

The consistency is calculated for the different RSNs across time and subjects after applying

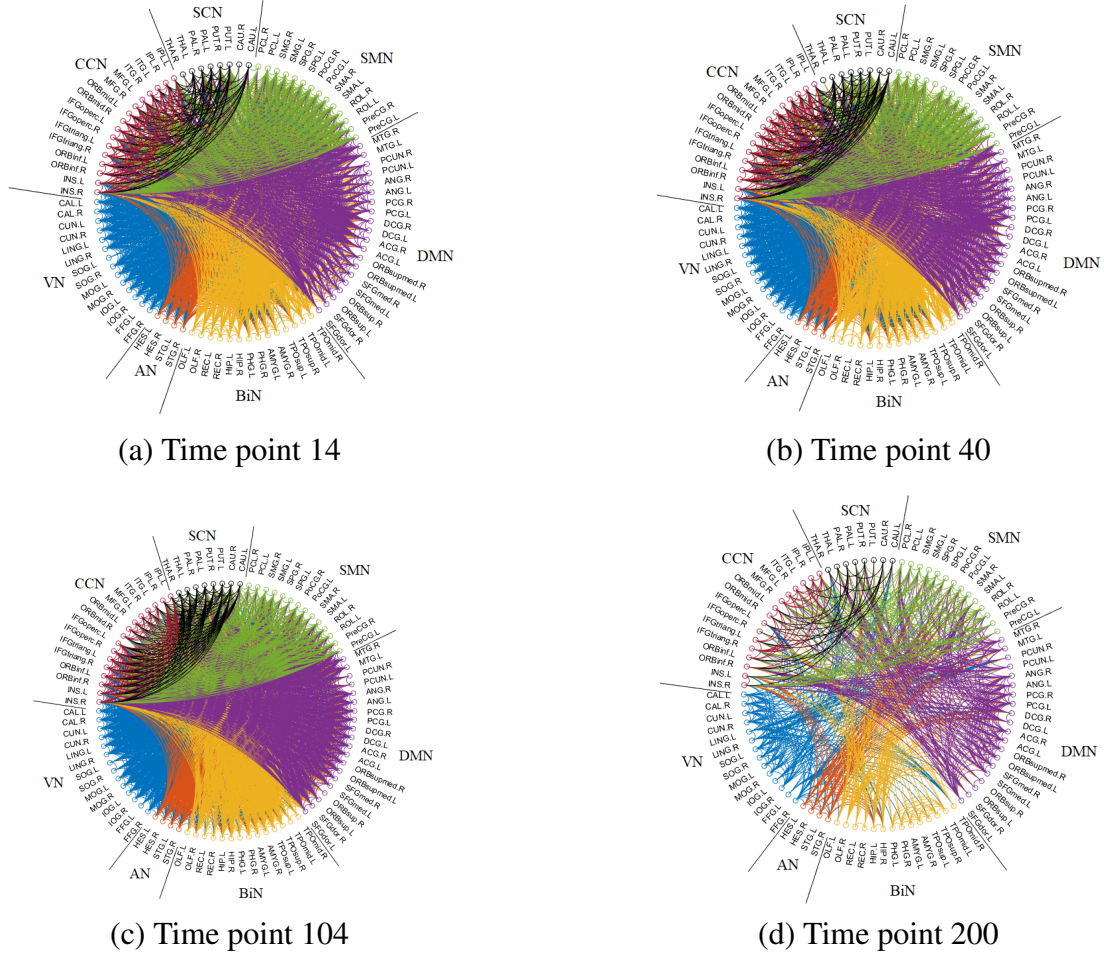


Figure 5.3: Detected community structure from selected change points. The ROIs in different networks reconfigured their community membership across time. Visualized by circularGraph toolbox [1].

TBSCA to the dFCN. The variation in consistency across subjects is shown in Fig. 5.5. From Fig. 5.5, the VN, AN and SMN have a relatively higher consistency than the other RSNs. This reflects a relatively stronger connection between the ROIs within these networks over time. On the other hand, the ROIs of the DMN, SCN, CCN and BiN exhibit more variation in their community membership over time which indicates the dynamic behavior of these RSNs during rest.

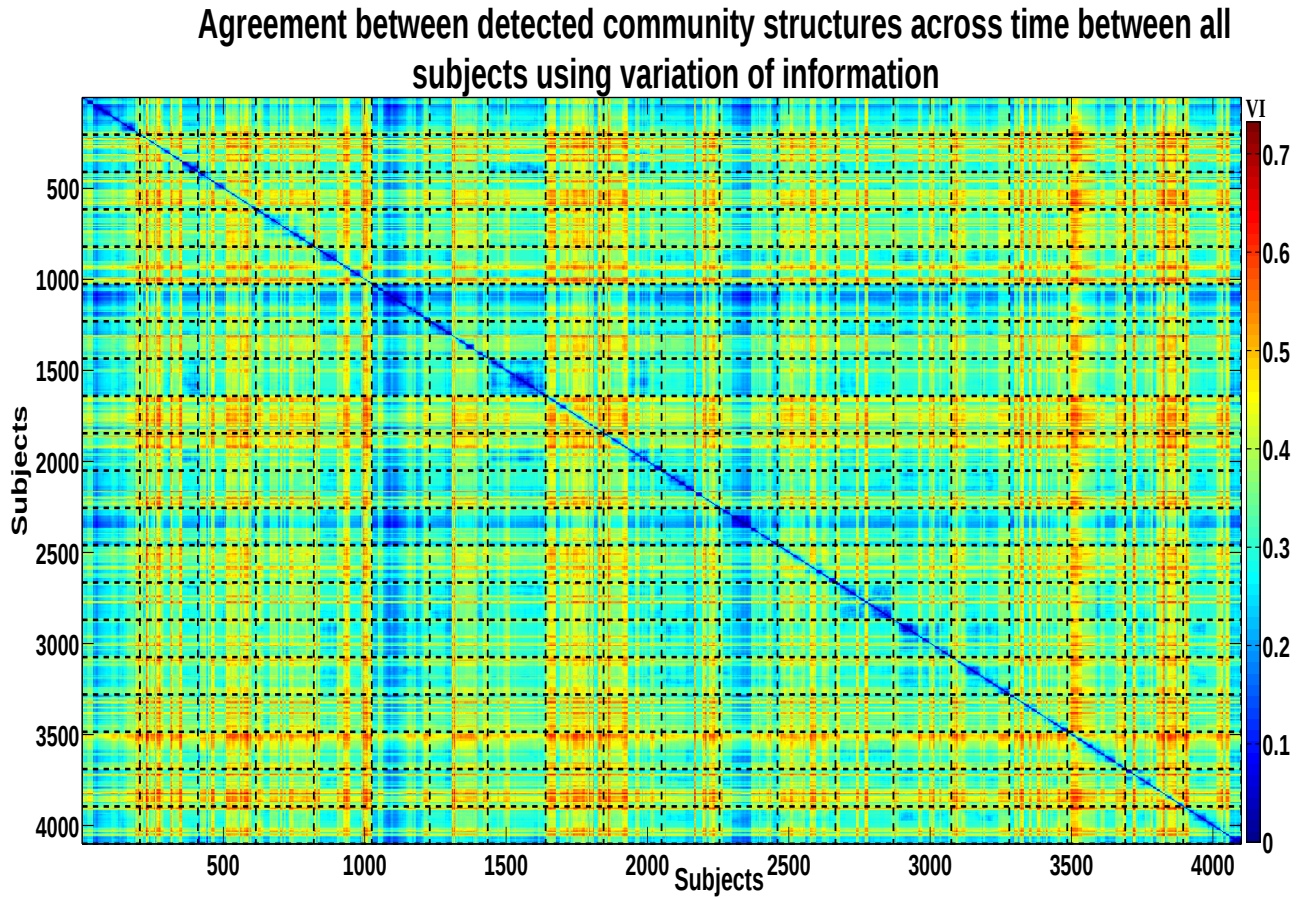


Figure 5.4: Agreement between the detected community structures across time for all subjects measured by variation of information (VI).

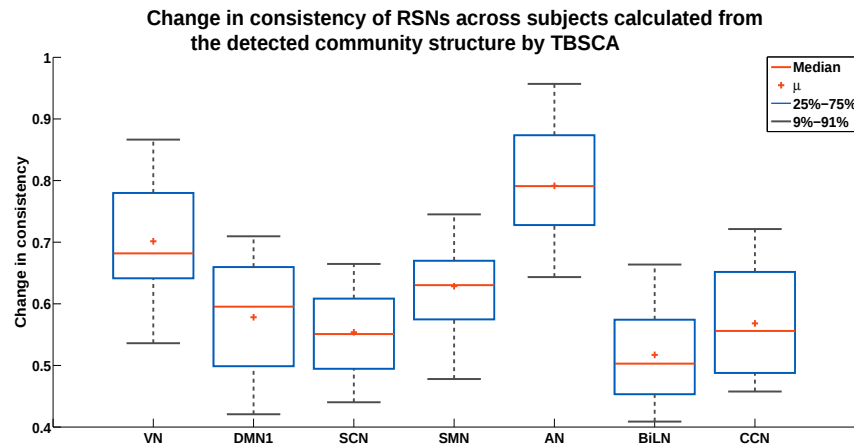


Figure 5.5: Consistency of different RSNs across subjects calculated from the detected community structure by TBSCA.

## 5.4 Conclusions

In this chapter, a temporal block spectral clustering framework is introduced to detect and track the community structure of multi-layer temporal networks. In the proposed approach, a sliding window correlation approach is used to model the pair-wise temporal relationships between all nodes within and across time windows. A set of intra- and inter-adjacency matrices is constructed and combined to create a set of temporal supra-adjacency matrices. The dynamic community structure is then detected by applying spectral clustering to these supra-adjacency matrices. The simulation results show that the proposed algorithm is able to detect and track the correct community structure over time. The proposed method is also evaluated on dFCNs constructed from rs-fMRI across time and subjects revealing dynamic connectivity patterns between the RSNs. Moreover, the detected community structures for the different subjects are found to follow different patterns across time as this is rs-fMRI. Finally, a consistency measure is used to quantify the dynamics of the RSNs across time and subjects. From this measure, certain RSNs such as the DMN, SCN, CCN and BiN have been found to exhibit more dynamic behavior during rest.

# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusions

In this thesis, we proposed various methods to detect and track the community structure in dynamic and multi-layer networks. In order to better understand the community structure in these complex networks, we started our work by analyzing dynamic single-layer networks and then generalized it to analyze dynamic multi-layer networks. A particular focus of this thesis is detecting the community structure in social networks and functional connectivity networks of the brain.

In Chapter 2, we approached the problem of community detection in dynamic networks by introducing a low-rank + sparse estimation based evolutionary spectral clustering approach. The proposed method decomposes the network into low-rank and sparse parts and obtains smooth cluster assignments by minimizing the subspace distance between consecutive time points. The method tries to obtain smooth cluster assignments by minimizing the subspace chordal distance between consecutive time points. Moreover, the proposed method considers explicitly the changes in the number of clusters between the consecutive time points and reformulates the objective function depending on these changes. The proposed framework is robust to noise and outliers and can detect the community structure in both binary and weighted temporal networks efficiently. Furthermore, a cost function to track changes in the community structure across time is introduced through the

subspace distance measure. Therefore, unlike existing evolutionary clustering methods, the proposed algorithm does not require any prior knowledge or make assumptions about the community structure and does not assume any particular model for the network. The proposed approach is evaluated on both simulated temporal networks and social temporal networks. The results show that the proposed algorithm is able to detect and track the correct community structure over time even in noisy networks.

As the method proposed in Chapter 2 is limited to temporal single-layer networks and can take limited history into account, a computationally efficient tensor based method is proposed in Chapter 3 to detect the community structure in dynamic single-layer networks dynamic multi-layer networks and track change across longer history. The proposed method relies on a tensor based representation of the dynamic network. A 3-mode tensor is used to represent the dynamic network, while a 4-mode tensor is used to represent the dynamic multi-layer network. In the case of dynamic single-layer networks, the framework is implemented through two approaches: 3D-windowed tensor approach (3D-WTA) and running time tensor approach (RTTA). The proposed framework uses a sliding window approach in order to take the past history of the networks into account and optimizes the contribution of each time point within the window. In 3D-WTA, a fixed length sliding window is used to construct the tensors, while a variable length sliding window is used in RTTA. For each window, a common subspace across all time points within the window is identified using Tucker decomposition. Furthermore, a normalized cost function is introduced to track the changes in the community structure over time. The performance of the proposed framework is evaluated using multiple synthetic and real networks including social networks and FCNs of the brain. The change points detected by the introduced normalized cost function reflect the true changes in these networks. Moreover, the application of proposed framework to the dFCNs of the brain constructed from EEG data during ERN and from rs-fMRI revealed changes in the

community structure across time that is well-aligned with prior work.

As one of the major applications of our work is exploring the dFCNs from rs-fMRI, we applied 4D-WTA to identify and track the brain network community structure across time and subjects. In particular, this approach is adopted to study the temporal evolution of communities in the resting state networks. In 4D-WTA, the contribution of each subject and time point within the framework is optimized. Moreover, the dFCNs are summarized into a set of FC states defined through their community structures that repeat over time and subjects. The detected community structures are evaluated through a consistency measure and the dynamic behavior of the FCNs is compared to the results from prior work.

In Chapter 4, we introduce an information-theoretic approach to reduce the dynamic network and identify the time points that carry similar topological structure to aggregate them into a tensor. A tensor based spectral clustering approach, similar to the one in Chapter 3, is then applied to identify the community structure of the constructed tensors. A particular focus of the proposed approach is detecting the dynamics of the community structure in the FCNs constructed from rs-fMRI data. The objective of the proposed algorithm is to reduce the dynamic FCNs by grouping the time points that are topologically similar into a set of functional connectivity states. A tensor based approach is then performed to detect the community structure of these states. Finally, the detected community structure is summarized as a set of FC states that reoccur across time. The extracted FC states revealed the dynamic behavior of the resting state networks. The detected community structure of the FC states is found to be in line with results reported in previous studies of rs-fMRI.

In Chapter 5, a temporal block spectral clustering framework is proposed to detect and track the community structure of multi-layer temporal networks. The main contribution of the proposed approach is the construction of data-driven intra- and inter-layer adjacency matrices. In particular,

the proposed approach considers all possible edges between nodes within and across layers to construct intra- and inter- layers adjacency matrices unlike existing methods that ignore inter-layer edges or considers inter-layer edges between the node and itself only, i.e. multiplex networks. This is performed by using a sliding window correlation approach to model the pair-wise temporal relationships between all nodes within and across time windows. A set of intra- and inter-adjacency matrices are constructed and combined to create a set of temporal supra-adjacency matrices. The community structure is then determined by applying spectral clustering to these supra-adjacency matrices. The proposed algorithm is evaluated on simulated networks and dFCNs from rs-fMRI across time and subjects and compared to results reported from previous studies.

## 6.2 Future Work

In Chapter 2, the proposed approach considers the minimization of the subspace distance between consecutive time points to obtain smooth cluster assignments. Future work can extend the current work by considering longer history and minimize the distance between the current subspace and the subspaces of multiple previous time points. One main challenge of the proposed approach is the computational complexity where the dominant cost for each time step corresponds to the computation of the nuclear proximal operator and eigenvalue decomposition (EVD). Future work will consider reducing this cost by using randomized algorithms for SVD [156] or partial SVD and partial EVD as suggested in [157] [158].

In Chapter 3, the proposed approaches rely on a sliding window technique that is applied at each time point to identify the current community structure. Future work will consider optimal selection of the window length to determine how many time points from past history should be included. Moreover, we will reduce the computational cost by reducing the constructed tensors’

sizes. This can be accomplished by adding extra constraints to the objective function. The added constraints can guarantee the selection of time points or layers that are sparser than others to be included in the framework, thus excluding noisy adjacency matrices.

In Chapter 4, future work will focus on different extensions of the proposed approach for temporal multi-layer networks. In particular, the aggregation of the multi-layer network can be performed by considering both intra- and inter-layer edges since the current work focuses on intra-layer edges only. The aggregation process can be achieved by a two-step approach: First, a sliding window approach can be used to construct supra-adjacency matrices that consider both intra- and inter-layer connections between nodes within the time window. Second, the time windows that possess similar topological structure can be aggregated together. This may result in a better representation of the temporal multi-layer network and reflect a more accurate community structure. After the temporal multi-layer network is reduced, a tensor based community detection approach can be applied to detect the underlying community structure.

In Chapter 5, the proposed approach detects the community structure in temporal multi-layer networks through a temporal block spectral clustering framework. Future research will investigate the extension of different definitions of graph cut to multi-layer networks and derive the corresponding spectral clustering algorithms. Moreover, a multi-layer network may also have multiple aspects. For example, each layer may correspond to a different frequency band and vary across time for neuronal networks. Extensions of spectral clustering for these types of multiple aspect multi-layer networks will also be considered.

## **APPENDIX**

## Update $\mathbf{L}^{(t)k+1}$

Update  $\mathbf{L}^{(t)}$  by keeping only the terms with  $\mathbf{L}^{(t)}$ :

$$\begin{aligned} \mathbf{L}^{(t)k+1} = & \underset{\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \|\mathbf{L}^{(t)}\|_* + \langle \mathbf{X}_1^k, \mathbf{A}^{(t)} - \mathbf{L}^{(t)} - \mathbf{S}^{(t)k} \rangle \\ & + \frac{\gamma_1}{2} \|\mathbf{A}^{(t)} - \mathbf{L}^{(t)} - \mathbf{S}^{(t)k}\|_F^2 + \langle \mathbf{X}_2^k, \mathbf{W}^{(t)k} - \mathbf{L}^{(t)} \rangle + \frac{\gamma_2}{2} \|\mathbf{W}^{(t)k} - \mathbf{L}^{(t)}\|_F^2, \end{aligned} \quad (1)$$

which can be simplified to:

$$\begin{aligned} & = \underset{\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \|\mathbf{L}^{(t)}\|_* + \frac{\gamma_1}{2} \|\mathbf{L}^{(t)} - (\mathbf{A}^{(t)} - \mathbf{S}^{(t)k} + \frac{\mathbf{X}_1^k}{\gamma_1})\|_F^2 \\ & \quad + \frac{\gamma_2}{2} \|\mathbf{L}^{(t)} - (\mathbf{W}^{(t)k} + \frac{\mathbf{X}_2^k}{\gamma_2})\|_F^2 \\ & = \underset{\mathbf{L}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \|\mathbf{L}^{(t)}\|_* + \frac{\gamma_1 + \gamma_2}{2} \|\mathbf{L}^{(t)} - \frac{\gamma_1 \mathbf{Y}_1^k + \gamma_2 \mathbf{Y}_2^k}{\gamma_1 + \gamma_2}\|_F^2 \\ & = \operatorname{prox}_{\frac{1}{\gamma_1 + \gamma_2} \|\mathbf{L}^{(t)}\|_*} \left( \frac{\gamma_1 \mathbf{Y}_1^k + \gamma_2 \mathbf{Y}_2^k}{\gamma_1 + \gamma_2} \right), \end{aligned} \quad (2)$$

where  $\mathbf{Y}_1^k = \mathbf{A}^{(t)} - \mathbf{S}^{(t)k} + \frac{\mathbf{X}_1^k}{\gamma_1}$ ,  $\mathbf{Y}_2^k = \mathbf{W}^{(t)k} + \frac{\mathbf{X}_2^k}{\gamma_2}$  and  $\operatorname{prox}_f(\mathbf{Y}) = \operatorname{argmin}_{\mathbf{L} \in \mathbb{R}^{n \times n}} f(\mathbf{L}) + \frac{1}{2} \|\mathbf{L} - \mathbf{Y}\|_F^2$  is the proximity operator of the convex function  $f$  [88]. Let  $\mathbf{Y} = \frac{\gamma_1 \mathbf{Y}_1^k + \gamma_2 \mathbf{Y}_2^k}{\gamma_1 + \gamma_2}$ ,  $\gamma = \frac{\gamma_1 + \gamma_2}{2}$  and  $\mathbf{Y} = \mathbf{Q}_Y \mathbf{\Sigma}_Y \mathbf{V}_Y^\dagger$  be the SVD of the matrix  $\mathbf{Y}$ , singular value soft thresholding is then used to update  $\mathbf{L}^{(t)k+1}$  as:

$$\mathbf{L}^{(t)k+1} = \mathbf{Q}_Y \Omega_{\frac{1}{\gamma}}(\mathbf{\Sigma}_Y) \mathbf{V}_Y^\dagger, \quad (3)$$

where  $\Omega_\tau$  is the element-wise thresholding operator defined as  $\Omega_\tau(a) = \operatorname{sgn}(a) \max(|a| - \tau, 0)$ .

## Update $\mathbf{S}^{(t)k+1}$

Update  $\mathbf{S}^{(t)}$  by keeping only the terms with  $\mathbf{S}^{(t)}$ :

$$\begin{aligned}
\mathbf{S}^{(t)k+1} &= \\
&\underset{\mathbf{S}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \|\mathbf{S}^{(t)}\|_1 + \langle \mathbf{X}_1^k, \mathbf{A}^{(t)} - \mathbf{S}^{(t)} - \mathbf{L}^{(t)k+1} \rangle + \frac{\gamma_1}{2} \|\mathbf{A}^{(t)} - \mathbf{S}^{(t)} - \mathbf{L}^{(t)k+1}\|_F^2 \\
&= \underset{\mathbf{S}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \|\mathbf{S}^{(t)}\|_1 + \frac{\gamma_1}{2} \|\mathbf{S}^{(t)} - (\mathbf{A}^{(t)} - \mathbf{L}^{(t)k+1} + \frac{\mathbf{X}_1^k}{\gamma_1})\|_F^2
\end{aligned} \tag{4}$$

Similar to updating  $\mathbf{L}^{(t)k+1}$ ,  $\mathbf{S}^{(t)k+1}$  can be computed as:

$$\begin{aligned}
\mathbf{S}^{(t)k+1} &= \operatorname{prox}_{\frac{\mu}{\gamma_1} \|\cdot\|_1}(\mathbf{A}^{(t)} - \mathbf{L}^{(t)k+1} + \frac{\mathbf{X}_1^k}{\gamma_1}) \\
&= \Omega_{\frac{\mu}{\gamma_1}}(\mathbf{A}^{(t)} - \mathbf{L}^{(t)k+1} + \frac{\mathbf{X}_1^k}{\gamma_1})
\end{aligned} \tag{5}$$

## Update $\mathbf{W}^{(t)k+1}$

Update  $\mathbf{W}^{(t)}$  by keeping only the terms with  $\mathbf{W}^{(t)}$ :

$$\begin{aligned}
\mathbf{W}^{(t)k+1} &= \underset{\mathbf{W}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} f(\mathbf{W}^{(t)}) \\
&= \underset{\mathbf{W}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \lambda_1 \operatorname{Tr}(\mathbf{U}^{(t)k\ddagger} (\mathbf{I} - (\mathbf{D}^{(t)k})^{-0.5} \mathbf{W}^{(t)} (\mathbf{D}^{(t)k})^{-0.5}) \mathbf{U}^{(t)k}) \\
&\quad + \langle \mathbf{X}_2^k, \mathbf{W}^{(t)} - \mathbf{L}^{(t)k+1} \rangle + \frac{\gamma_2}{2} \|\mathbf{W}^{(t)} - \mathbf{L}^{(t)k+1}\|_F^2, \\
&\text{s.t. } \mathbf{W}^{(t)} = \mathbf{W}^{(t)\ddagger}, W^{(t)}_{ij} \geq 0, W^{(t)}_{ii} = 0,
\end{aligned} \tag{6}$$

where  $\mathbf{D}^{(t)k}$  is the degree matrix calculated from the previous iteration. Eq. (6) can be then simplified to:

$$\begin{aligned} & \underset{\mathbf{W}^{(t)} \in \mathbb{R}^{n \times n}}{\operatorname{argmin}} \quad \lambda_1 \operatorname{Tr}(\mathbf{U}^{(t)k\ddagger} (\mathbf{D}^{(t)k})^{-0.5} \mathbf{W}^{(t)} (\mathbf{D}^{(t)k})^{-0.5} \mathbf{U}^{(t)k}) \\ & + \frac{\gamma_2}{2} \|\mathbf{W}^{(t)} - (\mathbf{L}^{(t)k+1} - \frac{\mathbf{X}_2^k}{\gamma_2})\|_F^2 \\ & s.t \quad \mathbf{W}^{(t)} = \mathbf{W}^{(t)\ddagger}, W^{(t)}_{ij} \geq 0, W^{(t)}_{ii} = 0. \end{aligned} \quad (7)$$

The gradient of the function  $f(\mathbf{W}^{(t)})$  can be written as:

$$\nabla_{\mathbf{W}^{(t)}} f(\mathbf{W}^{(t)}) = \gamma_2 (\mathbf{W}^{(t)} - \mathbf{L}^{(t)k+1} + \frac{\mathbf{X}_2^k}{\gamma_2}) - \lambda_1 (\mathbf{D}^{(t)k})^{\frac{-1}{2}} \mathbf{U}^{(t)k} \mathbf{U}^{(t)k\ddagger} (\mathbf{D}^{(t)k})^{-0.5} \quad (8)$$

A closed form solution for  $\mathbf{W}^{(t)}$  can be calculated as:

$$\mathbf{W}^{(t)k+1} = \mathbf{L}^{(t)k+1} - \frac{\mathbf{X}_2^k}{\gamma_2} + \frac{\lambda_1}{\gamma_2} (\mathbf{D}^{(t)k})^{-0.5} \mathbf{U}^{(t)k} \mathbf{U}^{(t)k\ddagger} (\mathbf{D}^{(t)k})^{-0.5}. \quad (9)$$

In order to guarantee that  $\mathbf{W}^{(t)}$  satisfies all the constraints, spectral projected gradient (SPG) [89] method is used to update  $\mathbf{W}^{(t)}$  by following the steps in Algorithm 1 with the projection operator,  $\mathcal{P}(\mathbf{W}^{(t)})$ , defined as:

$$\mathcal{P}(\mathbf{W}^{(t)}) = \begin{cases} W_{ij}^{(t)} & \text{if } i \neq j \text{ and } W_{ij}^{(t)} \geq 0. \\ 0 & \text{if } i = j \text{ or } W_{ij}^{(t)} < 0. \end{cases}$$

## Update $\mathbf{U}^{(t)k+1}$

Update  $\mathbf{U}^{(t)}$  by keeping all the terms with  $\mathbf{U}^{(t)}$  only:

$$\begin{aligned} \mathbf{U}^{(t)k+1} &= \underset{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}}{\operatorname{argmin}} f(\mathbf{U}^{(t)}) \\ &= \underset{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}}{\operatorname{argmin}} \lambda_1 \operatorname{Tr}(\mathbf{U}^{(t)\ddagger} \mathbf{\Phi}^{(t)k+1} \mathbf{U}^{(t)}) + \lambda_2 d_{ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)}), \quad s.t \quad \mathbf{U}^{(t)\ddagger} \mathbf{U}^{(t)} = \mathbf{I}. \end{aligned} \quad (10)$$

---

**Algorithm 1** Spectral projected gradient method (SPG)

---

**Input:**  $\mathbf{W}^{(t)}, \mathbf{L}^{(t)}, \mathbf{X}_2^{(t)}, \mathbf{U}^{(t)}, \lambda_1, \lambda_2, \varepsilon$ .

**Output:**  $\mathbf{W}^{(t)}$ .

```

1:  $l \leftarrow 0$ 
2: Initialize  $\mathbf{W}^{(t)0}$  using Eq. (9).
3:  $\sigma \leftarrow 1$ .
4: while  $\| \mathbf{W}^{(t)l+1} - \mathbf{W}^{(t)l} \|_F^2 < \varepsilon$  do
5:    $\mathbf{R} \leftarrow \mathcal{P}(\mathbf{W}^{(t)l} - \sigma \nabla f(\mathbf{W}^{(t)l})) - \mathbf{W}^{(t)l}$ .
6:   Compute the step length  $\rho$  using line search [89].
7:    $\mathbf{W}^{(t)l+1} \leftarrow \mathbf{W}^{(t)l} + \rho \mathbf{R}$ .
8:    $\mathbf{s} \leftarrow \text{vec}(\mathbf{W}^{(t)l+1} - \mathbf{W}^{(t)l})$ .
9:    $\mathbf{y} \leftarrow \text{vec}(\nabla f(\mathbf{W}^{(t)l+1}) - \nabla f(\mathbf{W}^{(t)l}))$ .
10:   $\sigma \leftarrow \mathbf{y}^\dagger \mathbf{y} / \mathbf{s}^\dagger \mathbf{y}$ .
11:   $l \leftarrow l + 1$ 
12: end while

```

---

Considering the three different cases for  $d_{Ch}^2(\mathbf{U}^{(t)}, \mathbf{U}^{(t-1)})$  as discussed in Section 2.3.2, the function,  $f(\mathbf{U}^{(t)})$ , in Eq. (10) can be rewritten for the first case as:

$$\begin{aligned}
& \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) + \frac{\lambda_2}{\lambda_1} \{r - \text{Trace}(\mathbf{U}^{(t)} \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger})\}, \\
& s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \\
& \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) + \left( -\frac{\lambda_2}{\lambda_1} \text{Trace}(\mathbf{U}^{(t)} \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}) \right) + \frac{\lambda_2}{\lambda_1} r, \\
& s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I},
\end{aligned} \tag{11}$$

using the cyclic property of the trace operator, Eq. (11) becomes,

$$\begin{aligned}
& \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) + \left( -\frac{\lambda_2}{\lambda_1} \text{Trace}(\mathbf{U}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger} \mathbf{U}^{(t)}) \right) + \frac{\lambda_2}{\lambda_1} r, \\
& s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}.
\end{aligned} \tag{12}$$

By ignoring the constant term and rearranging the problem using properties of the trace function we get,

$$\lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \left( \boldsymbol{\Phi}^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger} \right) \mathbf{U}^{(t)} \right), \quad s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \tag{13}$$

which can be rewritten as,

$$\min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \lambda_1 \text{Trace}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}_{mod}^{(t)k+1} \mathbf{U}^{(t)}), \text{ s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \quad (14)$$

where  $\boldsymbol{\Phi}_{mod}^{(t)k+1}$  is defined as the modified Laplacian matrix at time point  $t$ .  $\boldsymbol{\Phi}_{mod}^{(t)k+1}$  is calculated as  $\boldsymbol{\Phi}_{mod}^{(t)k+1} = \boldsymbol{\Phi}^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}$ .

For the second case, the function,  $f(\mathbf{U}^{(t)})$ , in Eq. (10) can be rewritten as:

$$\begin{aligned} & \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \mathbf{U}^{(t)}) + \frac{\lambda_2}{\lambda_1} \{r_1 - \text{Trace}(\mathbf{U}^{(t)} \mathbf{U}^{(t)\dagger} \hat{\mathbf{U}}^{(t-1)} \hat{\mathbf{U}}^{(t-1)\dagger})\}, \\ & \text{s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \\ & \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \mathbf{U}^{(t)}) + (-\frac{\lambda_2}{\lambda_1} \text{Trace}(\mathbf{U}^{(t)} \mathbf{U}^{(t)\dagger} \hat{\mathbf{U}}^{(t-1)} \hat{\mathbf{U}}^{(t-1)\dagger})) + \frac{\lambda_2}{\lambda_1} r_1, \\ & \text{s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \end{aligned} \quad (15)$$

using the cyclic property of the trace operator and ignoring the constant term, Eq. 15 becomes,

$$\begin{aligned} & \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}^{(t)k+1} \mathbf{U}^{(t)}) + (-\frac{\lambda_2}{\lambda_1} \text{Trace}(\mathbf{U}^{(t)\dagger} \hat{\mathbf{U}}^{(t-1)} \hat{\mathbf{U}}^{(t-1)\dagger} \mathbf{U}^{(t)})), \\ & \text{s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \end{aligned} \quad (16)$$

which can be rewritten as,

$$\min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \lambda_1 \text{Trace}(\mathbf{U}^{(t)\dagger} \boldsymbol{\Phi}_{mod}^{(t)k+1} \mathbf{U}^{(t)}), \text{ s.t. } \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \quad (17)$$

where  $\boldsymbol{\Phi}_{mod}^{(t)k+1} = \boldsymbol{\Phi}^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \hat{\mathbf{U}}^{(t-1)} \hat{\mathbf{U}}^{(t-1)\dagger}$ .

The solutions to the problems in Eq. (14) and Eq. (17) are found by choosing  $\mathbf{U}^{(t)}$  as the matrix containing the  $r$  eigenvectors that correspond to the smallest  $r$  eigenvalues of  $\boldsymbol{\Phi}_{mod}^{(t)k+1}$  where  $\boldsymbol{\Phi}_{mod}^{(t)k+1} \in \mathbb{S}_n^+$  and  $\mathbb{S}_n^+$  is the set of positive semidefinite matrices.

For the third case, the problem in Eq. (10) can be rewritten as:

$$\begin{aligned}
& \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) + \frac{\lambda_2}{\lambda_1} \{r - \text{Trace}(\hat{\mathbf{U}}^{(t)} \hat{\mathbf{U}}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger})\}, \\
& s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}, \\
& \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) + \left( -\frac{\lambda_2}{\lambda_1} \text{Trace}(\hat{\mathbf{U}}^{(t)} \hat{\mathbf{U}}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}) \right) + \frac{\lambda_2}{\lambda_1} r, \\
& s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I},
\end{aligned} \tag{18}$$

by ignoring the constant term, Eq. 18 becomes,

$$\begin{aligned}
& \lambda_1 \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \text{Trace} \left( \mathbf{U}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) + \left( -\frac{\lambda_2}{\lambda_1} \text{Trace}(\hat{\mathbf{U}}^{(t)} \hat{\mathbf{U}}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}) \right), \\
& s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}.
\end{aligned} \tag{19}$$

Considering the first term in Eq. (19), and defining  $\mathbf{U}^{(t)}$  as,  $\mathbf{U}^{(t)} = \begin{bmatrix} \hat{\mathbf{U}}^{(t)} & \bar{\mathbf{U}}^{(t)} \end{bmatrix}$  with  $\hat{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times r_1}$  and  $\bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}$ , then

$$\begin{aligned}
\text{Trace} \left( \mathbf{U}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \mathbf{U}^{(t)} \right) &= \begin{bmatrix} \hat{\mathbf{U}}^{(t)\dagger} \{ \mathbf{\Phi}^{(t)k+1} \hat{\mathbf{U}}^{(t)} & \hat{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \\ \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \hat{\mathbf{U}}^{(t)} & \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \end{bmatrix} \\
&= \text{Trace} \left( \hat{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \hat{\mathbf{U}}^{(t)} \right) + \text{Trace} \left( \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \right).
\end{aligned} \tag{20}$$

The optimization problem in Eq. (19) can be rewritten as:

$$\begin{aligned}
& \min_{\hat{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times r_1}, \bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}} \text{Trace} \left( \hat{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \hat{\mathbf{U}}^{(t)} \right) + \text{Trace} \left( \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \right) \\
& + \left( -\frac{\lambda_2}{\lambda_1} \text{Trace}(\hat{\mathbf{U}}^{(t)} \hat{\mathbf{U}}^{(t)\dagger} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}) \right), \quad s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}.
\end{aligned} \tag{21}$$

$$\begin{aligned}
& \min_{\hat{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times r}, \bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}} \text{Trace} \left( \hat{\mathbf{U}}^{(t)\dagger} \left( \mathbf{\Phi}^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger} \right) \hat{\mathbf{U}}^{(t)} \right) \\
& + \text{Trace} \left( \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \right), \quad s.t \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}.
\end{aligned} \tag{22}$$

A modified Laplacian using the current Laplacian and the eigenvectors found for the previous time point can be defined as  $\Phi_{mod}^{(t)k+1} = \Phi^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}$ .

The modified version of the optimization problem in Eq. 22 can be written as two separate optimization problems for the third case, which will be solved by splitting the single constraint in Eq. 21 into the multiple orthogonality constraints shown below:

$$\begin{aligned} \min_{\mathbf{U}^{(t)} \in \mathbb{R}^{n \times r}} \quad & \lambda_1 \text{Trace}(\hat{\mathbf{U}}^{(t)\dagger} \Phi_{mod}^{(t)k+1} \hat{\mathbf{U}}^{(t)}) \\ & + \lambda_1 \text{Trace}(\bar{\mathbf{U}}^{(t)\dagger} \Phi^{(t)k+1} \bar{\mathbf{U}}^{(t)}), \quad s.t. \quad \mathbf{U}^{(t)\dagger} \mathbf{U}^{(t)} = \mathbf{I}. \end{aligned} \quad (23)$$

The optimization problem in Eq. (23) can be written as two separate optimization problems which will be solved by splitting the single constraint in Eq. (23) into multiple orthogonality constraints as follows:

$$\begin{aligned} \min_{\hat{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times r_1}} \quad & \lambda_1 \text{Tr}(\hat{\mathbf{U}}^{(t)\dagger} \Phi_{mod}^{(t)k+1} \hat{\mathbf{U}}^{(t)}) \quad s.t. \quad \hat{\mathbf{U}}^{(t)\dagger} \hat{\mathbf{U}}^{(t)} = \mathbf{I}, \\ \min_{\bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}} \quad & \lambda_1 \text{Tr}(\bar{\mathbf{U}}^{(t)\dagger} \Phi^{(t)k+1} \bar{\mathbf{U}}^{(t)}), \quad s.t. \quad \bar{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} = \mathbf{I}, \quad \hat{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} = \mathbf{0}, \end{aligned} \quad (24)$$

where  $\Phi_{mod}^{(t)k+1} = \Phi^{(t)k+1} - \frac{\lambda_2}{\lambda_1} \mathbf{U}^{(t-1)} \mathbf{U}^{(t-1)\dagger}$  is defined as the modified Laplacian matrix at time point  $t$  and  $\Phi_{mod}^{(t)k+1} \in \mathbb{S}_n^+$  where  $\mathbb{S}_n^+$  is the set of positive semidefinite matrices. Similar to the previous cases, the solution of the first part of this problem can be found as the first  $r_1$  eigenvectors corresponding to the smallest  $r_1$  eigenvalues of the modified Laplacian. After  $\hat{\mathbf{U}}^{(t)}$  is found, it is used to solve for the second part subject to the two constraints in Eq. (24). In particular, a complementary set of eigenvectors that corresponds to the graph at time point  $t$  is calculated. These vectors should be orthogonal within themselves and simultaneously orthogonal to the eigenvectors from the first part, i.e., the inner product between  $\hat{\mathbf{U}}^{(t)}$  and  $\bar{\mathbf{U}}^{(t)}$  is zero. The solution can be found using the trace penalty minimization model introduced in [159].

The trace penalty minimization model [159] that represents the second part in Eq. 24 is,

$$\begin{aligned} \min_{\bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}} f(\bar{\mathbf{U}}^{(t)}) &= \frac{1}{2} \text{Trace} \left( \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \right) + \frac{\zeta}{4} \|\bar{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} - \mathbf{I}\|_F^2, \\ s.t. \hat{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} &= \mathbf{0}, \end{aligned} \quad (25)$$

where  $\hat{\mathbf{U}}^{(t)\dagger} \hat{\mathbf{U}}^{(t)} = \mathbf{I}$  and  $\zeta$  is the penalty parameter. The problem in Eq. 25 can be solved using modified projected gradient method proposed in [159], initializing  $\bar{\mathbf{U}}_0^{(t)}$  such that  $\hat{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}_0^{(t)} = \mathbf{0}$  yields,

$$\bar{\mathbf{U}}_{j+1}^{(t)} = \bar{\mathbf{U}}_j^{(t)} - \delta_j \left( \mathbf{I} - \hat{\mathbf{U}}^{(t)} \hat{\mathbf{U}}^{(t)\dagger} \right) \nabla f(\bar{\mathbf{U}}_j^{(t)}), \quad (26)$$

where  $j$  refers to the  $j^{th}$  iteration and  $\delta_j$  is the step size. The initialization of  $\bar{\mathbf{U}}_0^{(t)}$  is done by solving the following optimization problem:

$$\min_{\bar{\mathbf{U}}^{(t)} \in \mathbb{R}^{n \times (r-r_1)}} \text{Trace} \left( \bar{\mathbf{U}}^{(t)\dagger} \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}^{(t)} \right), \quad s.t. \quad \bar{\mathbf{U}}^{(t)\dagger} \bar{\mathbf{U}}^{(t)} = \mathbf{I}. \quad (27)$$

A closed form solution can be used to calculate the gradient of  $f(\bar{\mathbf{U}}_j^{(t)})$  at each iteration, as follows:

$$\nabla f(\bar{\mathbf{U}}_j^{(t)}) = \mathbf{\Phi}^{(t)k+1} \bar{\mathbf{U}}_j^{(t)} + \zeta \bar{\mathbf{U}}_j^{(t)} \left( \bar{\mathbf{U}}_j^{(t)\dagger} \bar{\mathbf{U}}_j^{(t)} - \mathbf{I} \right). \quad (28)$$

The step size is computed as,

$$\delta_j = \frac{\text{Trace} \left( \mathbf{V}_j^\dagger \mathbf{H}_j \right)}{\|\mathbf{H}_j\|_F^2}, \quad (29)$$

where  $\mathbf{V}_j = \bar{\mathbf{U}}_{j+1}^{(t)} - \bar{\mathbf{U}}_j^{(t)}$  and  $\mathbf{H}_j = \nabla f(\bar{\mathbf{U}}_{j+1}^{(t)}) - \nabla f(\bar{\mathbf{U}}_j^{(t)})$ . The penalty parameter  $\zeta$  is chosen such that  $\zeta \in [\max(0, \lambda_{r-r_1}), \lambda_n]$  where  $\lambda_{r-r_1}$  is the  $(r-r_1)^{th}$  smallest generalized eigenvalue of the graph Laplacian  $\mathbf{\Phi}^{(t)k+1}$ . The iterations termination criterion is  $\|f(\bar{\mathbf{U}}_j^{(t)})\|_F \leq \varepsilon_1$ . At the end of the iterations  $\hat{\mathbf{U}}^{(t)}$  and  $\bar{\mathbf{U}}^{(t)}$  are combined to form  $\mathbf{U}^{(t)}$ .

## ROIs in the AAL Template

| Index   | ROI                                                 |
|---------|-----------------------------------------------------|
| (1,2)   | Precentral gyrus (PreCG)                            |
| (3,4)   | Superior frontal gyrus, dorsolateral (SFGdor)       |
| (5,6)   | Superior frontal gyrus, orbital part (ORBsup)       |
| (7,8)   | Middle frontal gyrus (MFG)                          |
| (9,10)  | Middle frontal gyrus, orbital part (ORBmid)         |
| (11,12) | Inferior frontal gyrus, opercular part IFGoperc     |
| (13,14) | Inferior frontal gyrus, triangular part (IFGtriang) |
| (15,16) | Inferior frontal gyrus, orbital part (ORBinf)       |
| (17,18) | Rolandic operculum (ROL)                            |
| (19,20) | Supplementary motor area (SMA)                      |
| (21,22) | Olfactory cortex (OLF)                              |
| (23,24) | Superior frontal gyrus, medial (SFGmed)             |
| (25,26) | Superior frontal gyrus, medial orbital (ORBsupmed)  |

|          |                                                 |
|----------|-------------------------------------------------|
| (27, 28) | Gyrus rectus (REC)                              |
| (29, 30) | Insula (INS)                                    |
| (31, 32) | Anterior cingulate and paracingulate gyri (ACG) |
| (33, 34) | Median cingulate and paracingulate gyri (DCG)   |
| (35, 36) | Posterior cingulate gyrus (PCG)                 |
| (37, 38) | Hippocampus (HIP)                               |
| (39, 40) | Parahippocampal gyrus (PHG)                     |
| (41, 42) | Amygdala (AMYG)                                 |
| (43, 44) | Calcarine fissure and surrounding cortex (CAL)  |
| (45, 46) | Cuneus (CUN)                                    |
| (47, 48) | Lingual gyrus (LING)                            |
| (49, 50) | Superior occipital gyrus (SOG)                  |
| (51, 52) | Middle occipital gyrus (MOG)                    |
| (53, 54) | Inferior occipital gyrus (IOG)                  |
| (55, 56) | Fusiform gyrus (FFG)                            |

|          |                                                             |
|----------|-------------------------------------------------------------|
| (57, 58) | Postcentral gyrus (PoCG)                                    |
| (59, 60) | Superior parietal gyrus (SPG)                               |
| (61, 62) | Inferior parietal, but supramarginal and angular gyri (IPL) |
| (63, 64) | Supramarginal gyrus (SMG)                                   |
| (65, 66) | Angular gyrus (ANG)                                         |
| (67, 68) | Precuneus (PCUN)                                            |
| (69, 70) | Paracentral lobule (PCL)                                    |
| (71, 72) | Caudate (CAU)                                               |
| (73, 74) | Lenticular nucleus, putamen (PUT)                           |
| (75, 76) | Lenticular nucleus, pallidum (PAL)                          |
| (77, 78) | Thalamus (THA)                                              |
| (79, 80) | Heschl gyrus (HES)                                          |
| (81, 82) | Superior temporal gyrus (STG)                               |
| (83, 84) | Temporal pole: superior temporal gyrus (TPOsup)             |
| (85, 86) | Middle temporal gyrus (MTG)                                 |

|          |                                               |
|----------|-----------------------------------------------|
| (87, 88) | Temporal pole: middle temporal gyrus (TPOmid) |
| (89, 90) | Inferior temporal gyrus (ITG)                 |

Table 1: List of the ROIs in the AAL 90 template. The odd and even indices refer to the left and right brain hemispheres ROIs, respectively.

## **BIBLIOGRAPHY**

## BIBLIOGRAPHY

- [1] “Circulargraph toolbox,” <https://www.mathworks.com/matlabcentral/fileexchange/48576-circulargraph>.
- [2] S. Borgatti, A. Mehra, D. Brass, and G. Labianca, “Network analysis in the social sciences,” *science*, vol. 323, no. 5916, pp. 892–895, 2009.
- [3] M. Rubinov and O. Sporns, “Complex network measures of brain connectivity: uses and interpretations,” *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.
- [4] B. Malin, “Data and collocation surveillance through location access patterns,” in *Proc. NAACSOS Conf*, 2004.
- [5] P. Holme and J. Saramäki, “Temporal networks,” *Physics reports*, vol. 519, no. 3, pp. 97–125, 2012.
- [6] M. Kivelä, A. Arenas, M. Barthelemy, J. Gleeson, Y. Moreno, and M. Porter, “Multilayer networks,” *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.
- [7] U. Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [8] N. Andrew, J. Michael, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.
- [9] A. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [10] M. Newman, “Modularity and community structure in networks,” *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [11] M. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
- [12] A. Jain, M. Murty, and P. Flynn, “Data clustering: a review,” *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

- [13] P. Tan *et al.*, *Introduction to data mining*. Pearson Education India, 2006.
- [14] M. Stoer and F. Wagner, “A simple min-cut algorithm,” *Journal of the ACM (JACM)*, vol. 44, no. 4, pp. 585–591, 1997.
- [15] L. Hagen and A. Kahng, “New spectral methods for ratio cut partitioning and clustering,” *IEEE transactions on computer-aided design of integrated circuits and systems*, vol. 11, no. 9, pp. 1074–1085, 1992.
- [16] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [17] D. Wagner and F. Wagner, “Between min cut and graph bisection,” in *International Symposium on Mathematical Foundations of Computer Science*. Springer, 1993, pp. 744–750.
- [18] S. White and P. Smyth, “A spectral clustering approach to finding communities in graphs,” in *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 2005, pp. 274–285.
- [19] J. Forman, P. Clemons, S. Schreiber, and S. Haggarty, “Spectralnet—an application for spectral graph analysis and visualization,” *BMC bioinformatics*, vol. 6, no. 1, p. 260, 2005.
- [20] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” *Physics Reports*, vol. 659, pp. 1–44, 2016.
- [21] K. Steinhaeuser and N. Chawla, “Identifying and evaluating community structure in complex networks,” *Pattern Recognition Letters*, vol. 31, no. 5, pp. 413–421, 2010.
- [22] E. Acar and B. Yener, “Unsupervised multiway data analysis: A literature survey,” *IEEE transactions on knowledge and data engineering*, vol. 21, no. 1, pp. 6–20, 2009.
- [23] E. Acar, S. Camtepe, M. Krishnamoorthy, and B. Yener, “Modeling and multiway analysis of chatroom tensors,” *ISI*, vol. 2005, pp. 256–268, 2005.
- [24] F. Estienne, N. Matthijs, D. Massart, P. Ricoux, and D. Leibovici, “Multi-way modelling of high-dimensionality electroencephalographic data,” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 1, pp. 59–72, 2001.
- [25] T. Kolda and B. Bader, “Tensor decompositions and applications,” *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

- [26] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM journal on Matrix Analysis and Applications*, vol. 21, p. 12531278, January 2000.
- [27] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, “On the best rank-1 and rank-( $r_1, r_2, \dots, r_n$ ) approximation of higher-order tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, pp. 1324–1342, January 2000.
- [28] T. Kolda, *Multilinear operators for higher-order decompositions*. United States. Department of Energy, 2006.
- [29] G. Palla, A. Barabási, and T. Vicsek, “Quantifying social group evolution,” *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.
- [30] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, “Detecting communities and their evolutions in dynamic social networks a bayesian approach,” *Machine learning*, vol. 82, no. 2, pp. 157–189, 2011.
- [31] D. Chackrabarti, R. Kumar, and A. Tomkins, “Evolutionary clustering,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, vol. II. ACM, 2006, pp. 554–560.
- [32] Y. Chi, X. Song, D. Zhou, K. Hino, and B. Tseng, “Evolutionary spectral clustering by incorporating temporal smoothness,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, vol. II. ACM, 2007, pp. 153–162.
- [33] K. Xu, M. Kliger, and A. Hero, “Adaptive evolutionary clustering,” *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 304–336, 2014.
- [34] L. Gauvin, A. Panisson, and C. Cattuto, “Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach,” *PloS one*, vol. 9, no. 1, 2014.
- [35] P. Mucha, T. Richardson, K. Macon, M. Porter, and J. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks,” *science*, vol. 328, no. 5980, pp. 876–878, 2010.
- [36] S. Chevallier, Q. Barthélemy, and J. Atif, “Metrics for multivariate dictionaries,” *arXiv preprint arXiv:1302.4242*, 2013.
- [37] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

- [38] D. Krackhardt, “Cognitive social structures,” *Social networks*, vol. 9, no. 2, pp. 109–134, 1987.
- [39] J. Padgett and C. Ansell, “Robust action and the rise of the medici, 1400-1434,” *American journal of sociology*, vol. 98, no. 6, pp. 1259–1319, 1993.
- [40] A. Cardillo, J. Gómez-Gardenes, M. Zanin, M. Romance, D. Papo, F. D. Pozo, and S. Boccaletti, “Emergence of network features from multiplexity,” *Scientific reports*, vol. 3, 2013.
- [41] A. Amelio and C. Pizzuti, “Evolutionary clustering for mining and tracking dynamic multi-layer networks,” *Computational Intelligence*, vol. 33, no. 2, pp. 181–209, 2017.
- [42] F. Babiloni, F. Cincotti, C. Babiloni, F. Carducci, D. Mattia, L. Astolfi, A. Basilisco, P. Rossini, L. Ding, Y. Ni *et al.*, “Estimation of the cortical functional connectivity with the multimodal integration of high-resolution eeg and fmri data by directed transfer function,” *Neuroimage*, vol. 24, no. 1, pp. 118–131, 2005.
- [43] A. D. Martino, A. Scheres, D. Margulies, A. Kelly, L. Uddin, Z. Shehzad, B. Biswal, J. Walters, F. Castellanos, and M. Milham, “Functional connectivity of human striatum: a resting state fmri study,” *Cerebral cortex*, vol. 18, no. 12, pp. 2735–2747, 2008.
- [44] M. V. D. Heuvel and H. Pol, “Exploring the brain network: a review on resting-state fmri functional connectivity,” *European neuropsychopharmacology*, vol. 20, no. 8, pp. 519–534, 2010.
- [45] M. Steriade, P. Gloor, R. Llinas, F. D. Silva, and M. Mesulam, “Basic mechanisms of cerebral rhythmic activities,” *Electroencephalography and clinical neurophysiology*, vol. 76, no. 6, pp. 481–508, 1990.
- [46] S. Bressler, “Large-scale cortical networks and cognition,” *Brain Research Reviews*, vol. 20, no. 3, pp. 288–304, 1995.
- [47] F. Varela, J. Lachaux, E. Rodriguez, and J. Martinerie, “The brainweb: phase synchronization and large-scale integration,” *Nature reviews. Neuroscience*, vol. 2, no. 4, p. 229, 2001.
- [48] E. Bullmore and O. Sporns, “Complex brain networks: graph theoretical analysis of structural and functional systems,” *Nature reviews. Neuroscience*, vol. 10, no. 3, p. 186, 2009.
- [49] M. van den Heuvel, C. Stam, M. Boersma, and H. Pol, “Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain,” *Neuroimage*, vol. 43, no. 3, pp. 528–539, 2008.

- [50] C. Stam and J. Reijneveld, “Graph theoretical analysis of complex networks in the brain,” *Nonlinear biomedical physics*, vol. 1, no. 1, p. 3, 2007.
- [51] R. Prabhakaran, S. Blumstein, E. Myers, E. Hutchison, and B. Britton, “An event-related fmri investigation of phonological–lexical competition,” *Neuropsychologia*, vol. 44, no. 12, pp. 2209–2221, 2006.
- [52] P. Boveroux, A. Vanhaudenhuyse, M. Bruno, Q. Noirhomme, S. Lauwick, A. Luxen, C. Degueldre, A. Plenevaux, C. Schnakers, C. Phillips *et al.*, “Breakdown of within-and between-network resting state functional magnetic resonance imaging connectivity during propofol-induced loss of consciousness,” *Anesthesiology: The Journal of the American Society of Anesthesiologists*, vol. 113, no. 5, pp. 1038–1053, 2010.
- [53] C. Chang and G. Glover, “Time–frequency dynamics of resting-state brain connectivity measured with fmri,” *Neuroimage*, vol. 50, no. 1, pp. 81–98, 2010.
- [54] B. Rogers, V. Morgan, A. Newton, and J. Gore, “Assessing functional connectivity in the human brain by fmri,” *Magnetic resonance imaging*, vol. 25, no. 10, pp. 1347–1357, 2007.
- [55] E. Allen, E. Damaraju, S. Plis, E. Erhardt, T. Eichele, and V. Calhoun, “Tracking whole-brain connectivity dynamics in the resting state,” *Cerebral cortex*, p. bhs352, 2012.
- [56] N. Leonardi, J. Richiardi, M. Gschwind, S. Simioni, J. Annoni, M. Schluep, P. Vuilleumier, and D. V. D. Ville, “Principal components of functional connectivity: a new approach to study dynamic brain connectivity during rest,” *NeuroImage*, vol. 83, pp. 937–950, 2013.
- [57] B. Biswal, F. Yetkin, V. Haughton, and J. Hyde, “Functional connectivity in the motor cortex of resting human brain using echo-planar mri,” *Magnetic resonance in medicine*, vol. 34, no. 4, pp. 537–541, 1995.
- [58] C. Rosazza and L. Minati, “Resting-state brain networks: literature review and clinical applications,” *Neurological Sciences*, vol. 32, no. 5, pp. 773–785, 2011.
- [59] J. Damoiseaux, S. Rombouts, F. Barkhof, P. P. Scheltens, C. Stam, S. Smith, and C. Beckmann, “Consistent resting-state networks across healthy subjects,” *Proceedings of the national academy of sciences*, vol. 103, no. 37, pp. 13 848–13 853, 2006.
- [60] R. Hutchison, T. Womelsdorf, E. Allen, P. Bandettini, V. Calhoun, M. Corbetta, S. D. Penna, J. Duyn, G. Glover, J. Gonzalez-Castillo *et al.*, “Dynamic functional connectivity: promise, issues, and interpretations,” *Neuroimage*, vol. 80, pp. 360–378, 2013.

- [61] R. Goebel, F. Esposito, and E. Formisano, "Analysis of functional image analysis contest (fiac) data with brainvoyager qx: From single-subject to cortically aligned group general linear model analysis and self-organizing group independent component analysis," *Human brain mapping*, vol. 27, no. 5, pp. 392–401, 2006.
- [62] J. Schrouff, V. Perlberg, M. Boly, G. Marrelec, P. Boveroux, A. Vanhaudenhuyse, M. Bruno, S. Laureys, C. Phillips, M. Pélégriani-Issac *et al.*, "Brain functional integration decreases during propofol-induced loss of consciousness," *Neuroimage*, vol. 57, no. 1, pp. 198–205, 2011.
- [63] W. Jamal, S. Das, and K. Maharatna, "Existence of millisecond-order stable states in time-varying phase synchronization measure in eeg signals," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, 2013, pp. 2539–2542.
- [64] W. Jamal, S. Das, K. Maharatna, D. Kuyucu, F. Sicca, L. Billeci, F. Apicella, and F. Muratori, "Using brain connectivity measure of eeg synchrostates for discriminating typical and autism spectrum disorder," in *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*. IEEE, 2013, pp. 1402–1405.
- [65] W. Jamal, S. Das, I. Oprescu, K. Maharatna, F. Apicella, and F. Sicca, "Classification of autism spectrum disorder using supervised learning of brain connectivity measures extracted from synchrostates," *Journal of neural engineering*, vol. 11, no. 4, pp. 046–019, 2014.
- [66] W. Jamal, S. Das, K. Maharatna, F. Apicella, G. Chronaki, F. Sicca, D. Cohen, and F. Muratori, "On the existence of synchrostates in multichannel eeg signals during face-perception tasks," *Biomedical Physics & Engineering Express*, vol. 1, no. 1, pp. 015–002, 2015.
- [67] S. Dimitriadis, N. Laskaris, and A. Tzelepi, "On the quantization of time-varying phase synchrony patterns into distinct functional connectivity microstates (fc $\mu$ states) in a multi-trial visual erp paradigm," *Brain topography*, vol. 26, no. 3, pp. 397–409, 2013.
- [68] J. Ou, L. Xie, C. Jin, X. Li, D. Zhu, R. Jiang, Y. Chen, J. Zhang, L. Li, and T. Liu, "Characterizing and differentiating brain state dynamics via hidden markov models," *Brain topography*, vol. 28, no. 5, pp. 666–679, 2015.
- [69] S. Ma, V. Calhoun, R. Phlypo, and T. Adalı, "Dynamic changes of spatial functional network connectivity in healthy individuals and schizophrenia patients using independent vector analysis," *Neuroimage*, vol. 90, pp. 196–206, 2014.
- [70] S. Mehrkanoon, M. Breakspear, and T. Boonstra, "Low-dimensional dynamics of resting-state cortical activity," *Brain topography*, vol. 27, no. 3, pp. 338–352, 2014.

- [71] N. Leonardi, W. Shirer, M. Greicius, and D. V. D. Ville, “Disentangling dynamic networks: Separated and joint expressions of functional connectivity patterns in time,” *Human brain mapping*, vol. 35, no. 12, pp. 5984–5995, 2014.
- [72] A. Mutlu, E. Bernat, and S. Aviyente, “A signal-processing-based approach to time-varying graph analysis for dynamic brain network identification,” *Computational and mathematical methods in medicine*, vol. 2012, 2012.
- [73] A. Mahyari, D. Zoltowski, E. Bernat, and S. Aviyente, “A tensor decomposition-based approach for detecting dynamic network states from eeg,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 1, pp. 225–237, 2017.
- [74] I. Cribben, R. Haraldsdottir, L. Atlas, T. Wager, and M. Lindquist, “Dynamic connectivity regression: determining state-related changes in brain connectivity,” *Neuroimage*, vol. 61, no. 4, pp. 907–920, 2012.
- [75] J. Zhang, X. Li, C. Li, Z. Lian, X. Huang, G. Zhong, D. Zhu, K. Li, C. Jin, X. Hu *et al.*, “Inferring functional interaction and transition patterns via dynamic bayesian variable partition models,” *Human brain mapping*, vol. 35, no. 7, pp. 3314–3331, 2014.
- [76] A. Ozdemir, E. Bernat, and S. Aviyente, “Recursive tensor subspace tracking for dynamic brain network analysis,” *IEEE Transactions on Signal and Information Processing over Networks*, 2017.
- [77] “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75 – 174, 2010.
- [78] K. Tu, B. Ribeiro, A. Swami, and D. Towsley, “Temporal clustering in dynamic networks with tensor decomposition,” *arXiv preprint arXiv:1605.08074*, 2016.
- [79] F. Folino and C. Pizzuti, “Multiobjective evolutionary community detection for dynamic networks,” in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*. ACM, 2010, pp. 535–536.
- [80] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
- [81] H. Hotelling, “Analysis of a complex of statistical variables into principal components.” *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.
- [82] N. Kishore Kumar and J. Schneider, “Literature survey on low rank approximation of matrices,” *Linear and Multilinear Algebra*, vol. 65, no. 11, pp. 2212–2244, 2017.

- [83] E. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
- [84] K. Ye and L. Lim, “Distance between subspaces of different dimensions,” *ArXiv e-prints*, 2014.
- [85] M. Meilă, “Comparing clusteringsan information based distance,” *Journal of multivariate analysis*, vol. 98, no. 5, pp. 873–895, 2007.
- [86] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [87] N. Parikh, S. Boyd *et al.*, “Proximal algorithms,” *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [88] P. Combettes and J. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.
- [89] E. Birgin, J. Martínez, M. Raydan *et al.*, “Spectral projected gradient methods: review and perspectives,” *J. Stat. Softw*, vol. 60, no. 3, pp. 1–21, 2014.
- [90] V. Traag, R. Aldecoa, and J. Delvenne, “Detecting communities using asymptotical surprise,” *Physical Review E*, vol. 92, no. 2, p. 022816, 2015.
- [91] M. Girvan and M. Newman, “Community structure in social and biological networks,” *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [92] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. Tseng, “Facetnet: a framework for analyzing communities and their evolutions in dynamic networks,” in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 685–694.
- [93] N. Eagle and A. Pentland, “Reality mining: sensing complex social systems,” *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.
- [94] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J. Pinton, M. Quaggiotto, W. V. den Broeck, C. Régis, B. B. Lina *et al.*, “High-resolution measurements of face-to-face contact patterns in a primary school,” *PloS one*, vol. 6, no. 8, p. e23176, 2011.

- [95] V. Gemmetto, A. Barrat, and C. Cattuto, “Mitigation of infectious disease at school: targeted class closure vs school closure,” *BMC infectious diseases*, vol. 14, no. 1, p. 695, 2014.
- [96] S. Fortunato, “Community detection in graphs,” *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.
- [97] M. Yaesoubi *et al.*, “Mutually temporally independent connectivity patterns: A new framework to study the dynamics of brain connectivity at rest with application to explain group difference based on gender,” *NeuroImage*, vol. 107, pp. 85–94, 2015.
- [98] “Metrics for community analysis: A survey.”
- [99] G. Rossetti and R. Cazabet, “Community discovery in dynamic networks: A survey,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 35, 2018.
- [100] “Global spectral clustering in dynamic networks.”
- [101] S. Fortunato and M. Barthelemy, “Resolution limit in community detection,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.
- [102] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” *Phys. Rev. E*, vol. 74, p. 016110.
- [103] M. Newman, “Community detection in networks: Modularity optimization and maximum likelihood are equivalent,” *arXiv preprint arXiv:1606.02319*, 2016.
- [104] S. Kullback and R. Leibler, “On information and sufficiencyannals of mathematical statistics, 22, 79–86,” *MathSciNet MATH*, 1951.
- [105] C. Nicolini *et al.*, “Community detection in weighted brain connectivity networks beyond the resolution limit,” *Neuroimage*, vol. 146, pp. 28–39, 2017.
- [106] S. Papadimitriou, S. Jimeng, and F. Christos, “Streaming pattern discovery in multiple time-series,” in *Proceedings of the 31st international conference on Very large data bases. VLDB Endowment*, 2005, pp. 697–708.
- [107] J. Sun, D. Tao, S. Papadimitriou, P. Yu, S. Philip, and C. Faloutsos, “Incremental tensor analysis: Theory and applications,” *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 3, p. 11, 2008.

- [108] M. Lavielle, “Using penalized contrasts for the change-point problem,” *Signal processing*, vol. 85, no. 8, pp. 1501–1510, 2005.
- [109] “findchangepts,” [www.mathworks.com/help/signal/ref/findchangepts.html](http://www.mathworks.com/help/signal/ref/findchangepts.html).
- [110] “A generalized louvain method for community detection implemented in matlab.”
- [111] C. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 25–34, 1987.
- [112] J. Cavanagh, M. Cohen, and J. Allen, “Prelude to and resolution of an error: Eeg phase synchrony reveals cognitive control dynamics during action monitoring,” *The Journal of Neuroscience*, vol. 29, pp. 98–105, 2009.
- [113] J. Kayser and C. Tenke, “Principal components analysis of laplacian waveforms as a generic method for identifying erp generator patterns: I. evaluation with auditory oddball tasks,” *Clinical neurophysiology*, vol. 117, no. 2, pp. 348–368, 2006.
- [114] S. Aviyente, E. Bernat, W. Evans, and S. Sponheim, “A phase synchrony measure for quantifying dynamic functional integration in the brain,” *Human brain mapping*, vol. 32, no. 1, pp. 80–93, 2011.
- [115] “1000 functional connectomes project,” [http://www.nitrc.org/projects/fcon\\_1000](http://www.nitrc.org/projects/fcon_1000)
- [116] “Statistical parametric mapping,” <http://www.fil.ion.ucl.ac.uk/spm/>
- [117] “Functional connectivity toolbox,” <http://www.nitrc.org/projects/conn>
- [118] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot, “Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain,” *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002.
- [119] N. Leonardi and D. V. D. Ville, “Identifying network correlates of brain states using tensor decompositions of whole-brain dynamic functional connectivity,” in *Pattern Recognition in Neuroimaging (PRNI), 2013 International Workshop on*. IEEE, 2013, pp. 74–77.
- [120] X. Li *et al.*, “Dynamic functional connectomics signatures for characterization and differentiation of ptsd patients,” *Human brain mapping*, vol. 35, no. 4, pp. 1761–1778, 2014.

- [121] H. Eavani, T. Satterthwaite, R. Gur, R. Gur, and C. Davatzikos, “Unsupervised learning of functional network dynamics in resting state fmri,” in *International Conference on Information Processing in Medical Imaging*. Springer, 2013, pp. 426–437.
- [122] M. Raichle, A. MacLeod, A. Snyder, W. Powers, D. Gusnard, and G. Shulman, “A default mode of brain function,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 676–682, 2001.
- [123] M. Greicius, K. Supekar, V. Menon, and R. Dougherty, “Resting-state functional connectivity reflects structural connectivity in the default mode network,” *Cerebral cortex*, vol. 19, no. 1, pp. 72–78, 2009.
- [124] Y. Zhou, N. Shu, Y. Liu, M. Song, Y. Hao, H. Liu, C. Yu, Z. Liu, and T. Jiang, “Altered resting-state functional connectivity and anatomical connectivity of hippocampus in schizophrenia,” *Schizophrenia research*, vol. 100, no. 1, pp. 120–132, 2008.
- [125] A. David and L. Pierre, “Hippocampal neuroanatomy,” in *The hippocampus book*. Oxford University Press, 2009.
- [126] J. Vincent, A. Snyder, M. Fox, B. Shannon, J. Andrews, M. Raichle, and R. Buckner, “Coherent spontaneous activity identifies a hippocampal-parietal memory network,” *Journal of neurophysiology*, vol. 96, no. 6, pp. 3517–3531, 2006.
- [127] Z. Yang, R. Craddock, D. Margulies, C. Yan, and M. Milham, “Common intrinsic connectivity states among posteromedial cortex subdivisions: Insights from analysis of temporal dynamics,” *Neuroimage*, vol. 93, pp. 124–137, 2014.
- [128] F. Karahanoğlu and D. V. D. Ville, “Transient brain activity disentangles fmri resting-state dynamics in terms of spatially and temporally overlapping networks,” *Nature communications*, vol. 6, 2015.
- [129] “Learning-induced autonomy of sensorimotor systems.”
- [130] M. Lowe, B. Mock, and J. Sorenson, “Functional connectivity in single and multislice echo-planar imaging using resting-state fluctuations,” *Neuroimage*, vol. 7, no. 2, pp. 119–132, 1998.
- [131] M. Greicius, B. Krasnow, A. Reiss, and V. Menon, “Functional connectivity in the resting brain: a network analysis of the default mode hypothesis,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 1, pp. 253–258, 2003.

- [132] D. Meunier, R. Lambiotte, and E. Bullmore, “Modular and hierarchically modular organization of brain networks,” *Frontiers in neuroscience*, vol. 4, p. 200, 2010.
- [133] E. Bullmore and D. Bassett, “Brain graphs: graphical models of the human brain connectome,” *Annual review of clinical psychology*, vol. 7, pp. 113–140, 2011.
- [134] E. Al-sharoha, M. Al-khassaweneh, and S. Aviyente, “A tensor based framework for community detection in dynamic networks,” in *Proceedings of ICASSP 2017*, to be published.
- [135] D. Meunier, S. Achard, A. Morcom, and E. Bullmore, “Age-related changes in modular organization of human brain functional networks,” *Neuroimage*, vol. 44, no. 3, pp. 715–723, 2009.
- [136] M. Rubinov and O. Sporns, “Weight-conserving characterization of complex functional brain networks,” *Neuroimage*, vol. 56, no. 4, pp. 2068–2079, 2011.
- [137] J. Power, A. Cohen, S. Nelson, G. Wig, K. Barnes, J. Church, A. Vogel, T. Laumann, F. Miezin, B. Schlaggar *et al.*, “Functional network organization of the human brain,” *Neuron*, vol. 72, no. 4, pp. 665–678, 2011.
- [138] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. Farmer, “Testing for nonlinearity in time series: the method of surrogate data,” *Physica D: Nonlinear Phenomena*, vol. 58, no. 1-4, pp. 77–94, 1992.
- [139] T. Schreiber and A. Schmitz, “Surrogate time series,” *Physica D: Nonlinear Phenomena*, vol. 142, no. 3, pp. 346–382, 2000.
- [140] M. D. Domenico, V. Nicosia, A. Arenas, and V. Latora, “Structural reducibility of multilayer networks,” *Nature communications*, vol. 6, p. 6864, 2015.
- [141] X. Di and B. Biswal, “Dynamic brain functional connectivity modulated by resting-state networks,” *Brain structure & function*, vol. 220, no. 1, p. 37, 2015.
- [142] A. Schaefer, D. Margulies, G. Lohmann, K. Gorgolewski, J. Smallwood, S. Kiebel, and A. Villringer, “Dynamic network participation of functional connectivity hubs assessed by resting-state fmri,” *Frontiers in human neuroscience*, vol. 8, 2014.
- [143] P. Fransson and G. Marrelec, “The precuneus/posterior cingulate cortex plays a pivotal role in the default mode network: Evidence from a partial correlation network analysis,” *Neuroimage*, vol. 42, no. 3, pp. 1178–1184, 2008.

- [144] L. Uddin, K. Clare, B. Biswal, C. Xavier, and M. Milham, “Functional connectivity of default mode network components: correlation, anticorrelation, and causality,” *Human brain mapping*, vol. 30, no. 2, pp. 625–637, 2009.
- [145] M. Newman, *Networks: an introduction*. Oxford university press, 2010.
- [146] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, “Clustering with multi-layer graphs: A spectral perspective,” *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820–5831, 2012.
- [147] N. Stanley, S. Shai, D. Taylor, and P. Mucha, “Clustering network layers with the strata multilayer stochastic block model,” *IEEE transactions on network science and engineering*, vol. 3, no. 2, pp. 95–105, 2016.
- [148] T. Valles-Catala, F. Massucci, R. Guimera, and M. Sales-Pardo, “Multilayer stochastic block models reveal the multilayer structure of complex networks,” *Physical Review X*, vol. 6, no. 1, p. 011036, 2016.
- [149] E. Al-Sharoa, M. Al-khassaweneh, and S. Aviyente, “A tensor based framework for community detection in dynamic networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2312–2316.
- [150] E. Al-sharoa, M. Al-khassaweneh, and S. Aviyente, “Detecting brain dynamics during resting state: a tensor based evolutionary clustering approach,” in *Wavelets and Sparsity XVII*, vol. 10394. International Society for Optics and Photonics, 2017, p. 103940G.
- [151] C. Chen, M. Ng, and S. Zhang, “Block spectral clustering for multiple graphs with interrelation,” *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 6, no. 1, p. 8, 2017.
- [152] R. Kanawati, “Multiplex network mining: a brief survey,” *IEEE Intelligent Informatics Bulletin*, vol. 16, pp. 24–28, 2015.
- [153] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi, “Evolving networks: Eras and turning points,” *Intelligent Data Analysis*, vol. 17, no. 1, pp. 27–48, 2013.
- [154] M. D. Domenico, A. Solé, S. Gómez, and A. Arenas, “Random walks on multiplex networks,” *arXiv preprint arXiv:1306.0519*, 2013.
- [155] C. Chen, M. Ng, and S. Zhang, “Block spectral clustering methods for multiple graphs,” *Numerical Linear Algebra with Applications*, vol. 24, no. 1, 2017.

- [156] R. Witten and E. Candes, “Randomized algorithms for low-rank matrix factorizations: sharp performance bounds,” *Algorithmica*, vol. 72, no. 1, pp. 264–281, 2015.
- [157] D. Goldfarb and S. Ma, “Convergence of fixed-point continuation algorithms for matrix rank minimization,” *Foundations of Computational Mathematics*, vol. 11, no. 2, pp. 183–210, 2011.
- [158] Y. Nakatsukasa and N. Higham, “Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the svd,” *SIAM Journal on Scientific Computing*, vol. 35, no. 3, pp. A1325–A1349, 2013.
- [159] Z. Wen, C. Yang, X. Liu, and Y. Zhang, “Trace-penalty minimization for large-scale eigenspace computation,” *Journal of Scientific Computing*, vol. 66, no. 3, pp. 1175–1203, 2016.