

---

Electronic Theses and Dissertations, 2020-

---

2023

## Clustering of Diverse Multiplex Networks

Yaxuan Wang  
*University of Central Florida*



Part of the [Mathematics Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd2020>

University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Wang, Yaxuan, "Clustering of Diverse Multiplex Networks" (2023). *Electronic Theses and Dissertations, 2020-*. 1692.

<https://stars.library.ucf.edu/etd2020/1692>



CLUSTERING OF DIVERSE MULTIPLEX NETWORKS

by

YAXUAN WANG

M.S. University of Central Florida, 2020

M.S. Capital Normal University, 2018

B.S. Capital Normal University, 2015

A dissertation submitted in partial fulfilment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Mathematics  
in the College of Sciences  
at the University of Central Florida  
Orlando, Florida

Spring Term  
2023

Major Professor: Marianna Pensky

© 2023 Yaxuan Wang

## ABSTRACT

This dissertation introduces the DIverse MultiPLEx Generalized Dot Product Graph (DIMPLE-GDPG) network model where all layers of the network have the same collection of nodes and follow the Generalized Dot Product Graph (GDPG) model. In addition, all layers can be partitioned into groups such that the layers in the same group are embedded in the same ambient subspace but otherwise all matrices of connection probabilities can be different. In common particular cases, where layers of the network follow the Stochastic Block Model (SBM) and Degree Corrected Block Model (DCBM), this setting implies that the groups of layers have common community structures but all matrices of block connection probabilities can be different. For DCBM, each group can also equip with nodes' specific weights. We refer to this two versions as the DIMPLE model and the DIMPLE-DECOR model. While the DIMPLE-GDPG model generalizes the COmmon Subspace Independent Edge (COSIE) random graph model, the DIMPLE model generalizes a multitude of papers that study multilayer networks with the same community structures in all layers (which include the tensor block model, the checker-board model as well as the Mixture Multilayer Stochastic Block Model (MMLSBM) as particular cases). This dissertation introduces novel algorithms for the recovery of similar groups of layers, for the estimation of the ambient subspaces in the groups of layers in the DIMPLE-GDPG setting, and for the within-layer clustering in the case of the DIMPLE model. We also consider applications of the DIMPLE models to real-life data, and its comparison with the MMLSBM. And the DIMPLE model with its SBM-imposed structures provided better descriptions of the organization of layers than the ones obtained on the basis of the MMLSBM setting.

## **ACKNOWLEDGMENTS**

I dedicate this dissertation to Marianna Penksy, my respected advisor, who has always encouraged me and helped me a lot not only in academic field but also in life. Without her kind support, I can't go through my entire doctoral degree. To my parents, Wenyan Gang and Mingjiang Wang, whose love and teachings from childhood to adulthood have enabled me to achieve what I am today. I also dedicate this work to my husband, Linli Shi, who has supported me every step of the way in my study and life. And lastly, I would like to thank the funding support from the National Science Foundation(NSF) grant DMS-2014928.

# TABLE OF CONTENTS

LIST OF FIGURES . . . . .	viii
LIST OF TABLES . . . . .	xiv
CHAPTER 1: INTRODUCTION TO EXISTING MODELS . . . . .	1
1.1 Network Theory . . . . .	1
1.2 Stochastic Block Model . . . . .	2
1.3 Degree Corrected Stochastic Block Model . . . . .	5
1.4 Generalized Random Dot Product Graph model . . . . .	7
1.5 Multiplex Networks . . . . .	8
1.6 COMmon Subspace Independent Edge (COSIE) random graph model . . . . .	10
1.7 Organization of this Dissertation . . . . .	11
CHAPTER 2: INTRODUCTION TO OUR GENERALISED MODELS . . . . .	14
2.1 DIverse MultiPLEx (DIMPLE) network models frameworks . . . . .	14
2.2 Justification of the model and related work . . . . .	17
2.3 Notations . . . . .	20

CHAPTER 3: FITTING THE MODELS . . . . .	23
3.1 Between-layer clustering . . . . .	23
3.2 Fitting invariant subspaces in groups of layers in the DIMPLE-GDPG model . . . . .	28
3.3 Within-layer clustering in the DIMPLE and DIMPLE-DECOR multiplex network . . . . .	30
CHAPTER 4: THEORETICAL ANALYSIS . . . . .	32
4.1 Definitions . . . . .	32
4.2 Assumptions . . . . .	33
4.3 The between-layer clustering error . . . . .	37
4.4 The subspace fitting errors in groups of layers in the DIMPLE-GDPG model . . . . .	38
4.5 The within-layer clustering error . . . . .	40
CHAPTER 5: SIMULATION STUDY . . . . .	41
5.1 Simulations settings . . . . .	41
5.2 Simulations Results . . . . .	43
5.3 Comparison of $k$ -means and $k$ -medians algorithms for community detection for DIMPLE-DECOR . . . . .	51
CHAPTER 6: APPLICATION TO REAL WORLD DATA . . . . .	55
6.1 Worldwide Food Trading Network Data . . . . .	55

6.2 Global Flights Network Data . . . . .	58
CHAPTER 7: DISCUSSION . . . . .	63
APPENDIX A: PROOF FOR DIMPLE MODEL . . . . .	66
A.1 Proof of Theorem 1 . . . . .	67
A.2 Proof of Theorem 2 . . . . .	68
A.3 Proof of Corollary 1 . . . . .	72
A.4 Proof of supplementary lemmas . . . . .	73
APPENDIX B: SIMULATIONS FOR COMPARISON OF DIMPLE WITH MMLSBM . . . . .	76



## LIST OF FIGURES

1.1	Multiplex networks examples with connection between layers(left penal) and no connection between layers(right penal) . . . . .	9
5.1	The between-layer clustering error rates of Algorithm 1 (left) and the within-layer error rates of Algorithms 3 (right), averaged over 500 simulation runs, for the DIMPLE model with $c = 0, d = 0.8$ (top) and $c = 0, d = 0.5$ (bottom), $L = 50$ and $n = 20, 25, 30, 40, 50, 60, 75, 100$ . The entries of $\mathbf{B}^{(l)}, l \in [L]$ , are generated as uniform random numbers between $c$ and $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by $w$ . . . . .	44
5.2	The between-layer clustering error rates of Algorithm 1 (left) and the within-layer error rates of Algorithms 3 (right), averaged over 500 simulation runs, for the DIMPLE model with $c = 0, d = 0.8$ (top) and $c = 0, d = 0.5$ (bottom), $n = 100$ and $L = 5, 10, 15, 20, 25, 30, 40, 50, 60, 75, 100, 150, 200, 250, 300, 350, 400$ . The entries of $\mathbf{B}^{(l)}, l \in [L]$ , are generated as uniform random numbers between $c$ and $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by $w$ . . . . .	45
5.3	The between-layer clustering error rates of Algorithm 1 (left) and the $\sin \Theta$ distances $R_{S,ave}$ of Algorithms 2 (right), averaged over 100 simulation runs, for the DIMPLE-GDPG model with $\alpha = 0.1, c = 0, d = 0.8$ (top) and $c = 0, d = 0.5$ (bottom), $L = 50$ and $n = 20, 25, 30, 40, 50, 60, 75, 100, 120, 140, 160$ . The entries of $\mathbf{B}^{(l)}, l \in [L]$ , are generated as uniform random numbers between $c$ and $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by $w$ . . . . .	46

5.4	The between-layer clustering error rates of Algorithm 1 (left) and the $\sin \Theta$ distances $R_{S,ave}$ of Algorithms 2 (right), averaged over 100 simulation runs, for the DIMPLE-GDPG model with $\alpha = 0.1$ , $c = 0$ , $d = 0.8$ (top) and $c = 0$ , $d = 0.5$ (bottom), $n = 100$ and $L = 5, 10, 15, 20, 25, 30, 40, 50, 60, 75, 100, 150, 200, 250, 300, 350, 400$ . The entries of $\mathbf{B}^{(l)}$ , $l \in [L]$ , are generated as uniform random numbers between $c$ and $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by $w$ . . . . .	47
5.5	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0$ , $d = 0.8$ , $w = 0.5$ , $t_1 = 0.2, 0.4, 0.6, 0.8$ , $t_2 = 1.11$ , $L = 50$ and $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs. . . . .	48
5.6	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0$ , $d = 0.8$ , $w = 0.5$ , $t_1 = 0.2$ , $t_2 = 0.65, 0.8, 0.95, 1.11$ , $L = 50$ and $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs. . . . .	48
5.7	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0$ , $d = 0.8$ , $w = 0.5$ , $(t_1, t_2) = (0.2, 0.65); (0.4, 0.8); (0.6, 0.95); (0.8, 1.11)$ , $L = 50$ and $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs. . . . .	48
5.8	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0$ , $d = 0.8$ , $w = 0.5$ , $(t_1, t_2) = (0.2, 1.11); (0.4, 0.95); (0.6, 0.8)$ , $L = 50$ and $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs. . . . .	49

5.9	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0, d = 0.8, w = 0.5, 0.75, 1, 1.25, t_1 = 0.2, t_2 = 1, L = 50$ and $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs. . . . .	49
5.10	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0, d = 0.8, w = 0.5, t_1 = 0.2, 0.4, 0.6, 0.8, t_2 = 1.11, L = 50, 100, 150, 200, 250, 300, 350, 400$ and $n = 100$ . Results are averaged over 100 simulation runs. . . . .	49
5.11	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0, d = 0.8, w = 0.5, t_1 = 0.2, t_2 = 0.65, 0.8, 0.95, 1.11, L = 50, 100, 150, 200, 250, 300, 350, 400$ and $n = 100$ . Results are averaged over 100 simulation runs. . . . .	50
5.12	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0, d = 0.8, w = 0.5, (t_1, t_2) = (0.2, 0.65); (0.4, 0.8); (0.6, 0.95); (0.8, 1.11), L = 50, 100, 150, 200, 250, 300, 350, 400$ and $n = 100$ . Results are averaged over 100 simulation runs. . . . .	50
5.13	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0, d = 0.8, w = 0.5, (t_1, t_2) = (0.2, 1.11); (0.4, 0.95); (0.6, 0.8), L = 50, 100, 150, 200, 250, 300, 350, 400$ and $n = 100$ . Results are averaged over 100 simulation runs. . . . .	51

5.14	The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for $c = 0$ , $d = 0.8$ , $w = 0.5, 0.75, 1.0, 1.25$ , $t_1 = 0.2$ , $t_2 = 1$ , $L = 50, 100, 150, 200, 250, 300, 350, 400$ and $n = 100$ . Results are averaged over 1000 simulation runs. . . . .	51
5.15	The comparison of the $k$ -means and $k$ -medians for Algorithm 3 for $c = 0$ , $d = 0.8$ , $w = 0.8$ , $t_1 = 0.4, 0.6, 0.8$ , $t_2 = 1.11$ , and $L = 100$ , $n = 25, 50, 75, 100, 150, 200$ . Results are averaged over 100 simulation runs. . . . .	52
5.16	The comparison of the $k$ -means and $k$ -medians for Algorithm 3 for $c = 0$ , $d = 0.5$ , $w = 0.5$ , $(t_1, t_2) = (0.4, 1), (0.6, 1.2), (0.8, 1.4)$ , and $L = 100$ , $n = 25, 50, 75, 100, 150, 200$ . Results are averaged over 100 simulation runs. . . . .	52
5.17	The comparison of the $k$ -means and $k$ -medians for Algorithm 3 for $c = 0$ , $t_1 = 0.6$ , $(d, w, t_2) = (0.4, 0.4, 1.58), (0.6, 0.6, 1.29), (0.8, 0.8, 1.11)$ , and $L = 100$ , $n = 25, 50, 75, 100, 150, 200$ . Results are averaged over 100 simulation runs. . . . .	53
5.18	The comparison of the $k$ -means and $k$ -medians for Algorithm 3 for $c = 0$ , $d = 0.8$ , $w = 0.8$ , $t_1 = 0.4, 0.6, 0.8$ , $t_2 = 1.11$ , and $L = 50, 100, 150, 200, 300, 400$ , $n = 100$ . Results are averaged over 100 simulation runs. . . . .	53
5.19	The comparison of the $k$ -means and $k$ -medians for Algorithm 3 for $c = 0$ , $d = 0.5$ , $w = 0.5$ , $(t_1, t_2) = (0.4, 1), (0.6, 1.2), (0.8, 1.4)$ , and $L = 50, 100, 150, 200, 300, 400$ , $n = 100$ . Results are averaged over 100 simulation runs. . . . .	53
5.20	The comparison of the $k$ -means and $k$ -medians for Algorithm 3 for $c = 0$ , $t_1 = 0.6$ , $(d, w, t_2) = (0.4, 0.4, 1.58), (0.6, 0.6, 1.29), (0.8, 0.8, 1.11)$ , $L = 50, 100, 150, 200, 300, 400$ , $n = 100$ . Results are averaged over 100 simulation runs. . . . .	54

6.1	Results of clustering of food networks layers into $M = 2$ clusters by Algorithm 1 in the paper . . . . .	57
6.2	Trading communities for the meat/dairy (top) and the fruit/vegetable (bottom) groups. Left panels: community sizes; right panels: community memberships . . . . .	57
6.3	Results of clustering of food networks layers into $M = 2$ clusters by ALMA algorithm of Fan et al. (2021) . . . . .	58
6.4	Communities for the four airlines groups. Group 1: airlines originated in China. Group 2: airlines originated in Asia, Australia, New Zealand, and Gulf States. Group 3: airlines originated in Europe and North Africa. Group 4: airlines originated in North or South America. . . . .	60
B.1	The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using DIMPLE model with $L = 50$ , $c = 0$ , $d = 0.8$ (top) and $c = 0$ , $d = 0.5$ (bottom), and $w = 0.7$ (left panel) or $w = 1$ (right panel). . . . .	78
B.2	The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using DIMPLE model with $L = 100$ , $c = 0$ , $d = 0.8$ (top) and $c = 0$ , $d = 0.5$ (bottom), $n = 20, 40, 60, 80, 100, 120, 140, 160$ and $w = 0.7$ (left panel) or $w = 1$ (right panel). . .	79
B.3	The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using MMLSBM with $L = 50$ , $c = 0$ , $d = 0.8$ (top) and $c = 0$ , $d = 0.5$ (bottom), $n = 20, 40, 60, 80, 100, 120, 140, 160$ and $w = 0.7$ (left panel) or $w = 1$ (right panel). . . . .	80

B.4	The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using MMLSBM with $L = 100$ , $c = 0$ , $d = 0.8$ (top) and $c = 0$ , $d = 0.5$ (bottom), and $w = 0.7$ (left panel) or $w = 1$ (right panel). . . . .	81
-----	--	----

## LIST OF TABLES

6.1	Airlines Groups obtained using Algorithm 1 with $K = 3$ and $M = 4$ . . . . .	61
6.2	Airlines Groups obtained using ALMA algorithm of Fan et al. (2021) with $K = 3$ and $M = 4$ . . . . .	62

# CHAPTER 1: INTRODUCTION TO EXISTING MODELS

## 1.1 Network Theory

As a part of graph theory, network theory has become more and more popular and useful over the past several years. Network is the study of graphs where connections are drawn between discrete objects and has been applied in many disciplines, including but not limited to, statistical physics (Cimini et al. (2019)), biology (Mason and Verwoerd (2007)), economics (Bates et al. (2014)) and neuroscience (Bassett et al. (2018)). A network refers to a graph consisting of nodes and edges. Nodes are the objects in the network and edges are connections between each node. For example, all computers can be nodes in a network and the wires or Internet connection between two computers are the edges connecting them. Nodes in a network can be different and therefore have different weights, and the edges between different nodes can be directed or undirected. Research on relations between individuals in several communities is a popular network problem, and people can be unweighted nodes in this network. Connections between people can be directed if research cares about who initiates a connection, and undirected if not.

In this dissertation, we consider a multilayer network where individual layer networks are unweighted and undirected. For unweighted networks of  $n$  nodes without multiple connections, the network structure can be represented by a  $n \times n$  adjacency matrix  $A$ . The component  $A_{ij}$  indicates a connection from node  $j$  and to node  $i$  and  $A_{ij} = 1$ , if there is an edge from node  $j$  to node  $i$  and  $A_{ij} = 0$  otherwise.

Since we only consider undirected graph with no self-edges in this dissertation,  $A_{ii} = 0$  and  $A_{ij} = A_{ji}$ ,  $1 \leq i, j \leq n$ . A common way to model adjacency matrix  $A$  is to assume that  $A_{ij}$  are independent Bernoulli variables for  $i < j$ , i.e.,  $A_{ij} \sim \text{Bernoulli}(P_{ij})$ ,  $1 \leq i < j \leq n$ .



$P \in [0, 1]^{n \times n}$  is the matrix of connection probabilities between nodes in the network, i.e.  $P_{ij}$  is the connection probability between node  $i$  and node  $j$ .

## 1.2 Stochastic Block Model

In many studies of networks, one assumes that nodes are divided into communities that have similar properties. The stochastic block model (SBM) is a generatively used model in network theory because of its simple analytic form and expressive power and has wide application to social networks, machine learning and computer science. According to Olhede and Wolfe (2014), SBM provides a universal tool for the description of time-independent stochastic network data. In fact, in the last few years, such models have been widely employed in brain research (see, e.g., Crossley et al. (2013), Faskowitz et al. (2018), Nicolini et al. (2017), Sporns (2018)).

SBM is equipped with communities, which are disjoint subsets of nodes in the network, and probabilities of connections between nodes are determined entirely by community membership of those nodes belong to. Communities are assortative if probabilities of connections between nodes within communities are higher than between communities. And they are disassortative if within communities connection probabilities are lower than between communities.

All nodes can be divided into communities in SBM, and the probability of connection between a pair of nodes is determined by the communities, to which those nodes belong. Assume all  $n$  nodes in the network can be partitioned into  $K$  communities, with each community to be  $G_k$ ,  $k = 1, \dots, K$  and  $\cup_{k=1}^K G_k = \{1, \dots, n\}$ . The probability of connection  $\mathbf{P}_{j_1, j_2}$  between nodes  $j_1$  and  $j_2$  is associated with  $\mathbf{B}_{k_1, k_2}$ , where node  $j_1$  is in community  $G_{k_1}$  and node  $j_2$  is in community  $G_{k_2}$ .  $\mathbf{P}_{j_1, j_2} = \mathbf{B}_{k_1, k_2}$ , if  $j_1 \in G_{k_1}$ ,  $j_2 \in G_{k_2}$ ,  $j_1, j_2 = 1, \dots, n$ ;  $k_1, k_2 = 1, \dots, K$ .  $\mathbf{B} \in [0, 1]^{K \times K}$  is the matrix of block connection probabilities and  $\mathbf{B}_{i, j}$  is the connection probability between community

$G_i$  and community  $G_j$ . For undirected network we consider here,  $\mathbf{P} = \mathbf{P}^T$  and  $\mathbf{B} = \mathbf{B}^T$ .

We can estimate connection probabilities if we know clustering function  $z : \{1, \dots, n\} \rightarrow \{1, \dots, K\}$ . And the clustering matrix  $\mathbf{Z} \in \{0, 1\}^{n \times K}$  corresponding to clustering function  $z$ ,  $\mathbf{Z}_{jk} = 1$ , if and only if  $j \in G_k$  and  $\mathbf{Z}_{jk} = 0$  otherwise. In this case, the matrix of probabilities of connection can be expressed as

$$\mathbf{P} = \mathbf{Z}\mathbf{B}\mathbf{Z}^T.$$

One observes that the adjacency matrix  $\mathbf{A}_{ij}$  are independent Bernoulli variables for  $\mathbf{A}_{ij} \sim \text{Bernoulli}(\mathbf{P}_{ij})$ . The objective is to recover the community assignment matrix  $\mathbf{Z}$  on the basis of the observed adjacency matrix  $\mathbf{A}$ .

One of typical statistical tasks in the study of SBM is to partition nodes into communities. And this attracted a lot of attention of researchers in this field of SBM. Decelle et al. (2011) conjectured phase transition for the community detection (a.k.a. weak recovery) problem. They used the cavity method to obtain an analysis of the phase diagram and developed an algorithm for inferring the group memberships of the nodes. Mossel et al. (2015) proved impossibility part of their prediction for the algorithmic threshold of clustering for two symmetric communities situation. Massoulié (2014) solved the positive part of the conjecture and recovered a weak form of the Ramanujan property. Theoretical investigation in the symmetric case at 4 communities (Abbe and Sandon (2018), Abbe and Sandon (2015)) and crossing at 5 communities Banks et al. (2016) are also considered to obtain the scaling of threshold for a growing number of communities for weak recovery problems.

Besides of weak recovery problem, partial recovery and exact recovery problems are concerned a lot. Mossel et al. (2014) obtained a result on the partial recovery of the communities. They developed an algorithm, which reconstructed sparse symmetric block models with two blocks under a general restrict, to achieve the optimal fraction of nodes labeled correctly. Deshpande et al. (2015)

developed an information-theoretic view of SBM and established explicit expression of the mutual information related to estimation-theoretic quantities in the case of two communities. Mossel and Xu (2016) showed the average minimum fraction of misclassified vertices and attained it by a local belief propagation algorithm. Abbe et al. (2015) identified a sharp threshold phenomenon for exact recovery problems and proposed an algorithm based on a semidefinite programming (SDP), which is proved to succeed in recovering the communities close to the threshold and another successful algorithm using a partial recovery algorithm combined with a local improvement procedure. SDP algorithm for community detection was extended by Perry and Wein (2017) by allowing for many communities of different sizes and this algorithm achieves exact recovery of the latent communities. Spectral methods are also a general one to consider about for network problems and SBM specifically. (see, e.g. McSherry (2001), Chin et al. (2015), Yun and Proutiere (2016)) For more detailed development for recovery problems can refer to Abbe (2017).

Among all methods for community recovery, spectral clustering is one of the most widely applied one in SBM for its easy implement and less computational cost qualities, and draw much attention of researchers and studied recent years. Rohe et al. (2011) originally studied the eigenvectors of random matrices and first showed that the eigenvectors of the normalized graph Laplacian asymptotically converge to the eigenvectors of a “population” normalized graph Laplacian. Su et al. (2019) proved the strong consistency of several methods based on the spectral clustering techniques for SBM. Lei and Rinaldo (2015) applied spectral clustering to the adjacency matrix of the network and consistently recover hidden communities in SBM. Other recovery methods include modularity maximization, likelihood methods, belief propagation and convex optimization, etc. More research work and specific introduction for spectral clustering method can be found in Baek and Kim (2021). And more topics on SBM can be referred to a detailed review of SBM in Lee and Wilkinson (2019). Some variants and their inference methods are also introduced by Funke and Becker (2019).

### 1.3 Degree Corrected Stochastic Block Model

Though being a widely used and essential model, a serious shortcoming of the SBM is that it treats all nodes in the same community as stochastically equivalent, which means that for nodes  $i$  and  $j$  in the same community,  $i$  has the same probability of connecting with node  $r$ , as  $j$  does (Lee and Wilkinson (2019)). On the contrary, in the real life networks, communities can have "hubs", which are nodes with high degrees, and many nodes with lower degrees, i.e., probabilities of connections are much more heterogeneous than the SBM allows. In order to address this shortcoming of SBM, Karrer and Newman (2011) introduced the Degree Corrected Stochastic Block Model (DCBM) which incorporates heterogeneous nodes' degrees.

In DCBM, the probability of connection between a pair of nodes is not only associated with the block connection probability, which, similarly to the SBM, is determined by the pair of communities, to which those nodes belong but also the nodes' specific weights. That is to say, nodes equipped with the same weights in SBM are different and they can be equipped with different weights in DCBM. Specifically,  $\Theta$  is the diagonal matrix of the node weights with elements  $\theta_i$  representing the weights of node  $i$  in the network. Then the matrix of probabilities of connection in DCBM can be expressed as

$$\mathbf{P} = \Theta \mathbf{Z} \mathbf{B} \mathbf{Z}^T \Theta^T.$$

where  $\mathbf{Z}$  is the clustering matrix. The adjacency matrix  $\mathbf{A}_{ij}$  are still independent Bernoulli variables for  $\mathbf{A}_{ij} \sim \text{Bernoulli}(\mathbf{P}_{ij})$ . And in this dissertation we ignore the node degree parameters and care more about recovering  $\mathbf{Z}$  based on the observed adjacency matrix  $\mathbf{A}$  in DCBM model.

DCBM, equipped with different nodes weights, have more parameters and wider application than SBM, which attracted many researchers' attention and derived various problems. Among all of these problems, the community detection problems in DCBM are so important and more difficult

because of various parameters for the nodes weights. Gao et al. (2018) derived asymptotic minimax risks of the community detection problem for a misclassification proportion loss and proposed an algorithm to perform consistent and even asymptotically optimal community detection in DCBMs. Zhao et al. (2012) established general theory for checking consistency of community detection for DCBM and compared several community detection criteria under both the standard and the degree-corrected models. And they applied methods to a political blogs network, which is a very famous and successful example in the application of DCBM. The nodes in the network are blogs of US politics and edges are hyperlinks between these blogs. With different degrees for nodes setting according to background of the blogs network, DCBM finds very similar partitions with the true one while block model likelihood behaves like random guessing.

Various methods for recovering community structure for DCBM are studied by researchers and spectral clustering is very popular among these methods in community detection problems for DCBM. Gulikers et al. (2017) proposed a spectral clustering algorithm based on a suitably normalized adjacency matrix, which consistently recovers the block membership of all but a vanishing fraction of nodes. Lei and Rinaldo (2015) recovered the community structure using a spherical k-median spectral clustering method which is an extension of its spectral clustering method to SBM. Inspired variations on the spectral clustering algorithm were proposed by Amini et al. (2013) and Chaudhuri et al. (2012), which artificially inflate the node degrees and improve statistical performance. Following that, Qin and Rohe (2013) extends the algorithm method of Amini et al. (2013) and Chaudhuri et al. (2012) more canonically by removing minimum degree assumption and providing guidance on the choice of tuning parameter.

## 1.4 Generalized Random Dot Product Graph model

In addition, the recent years saw a substantial advancement in the latent position graphical models. Specifically, the Random Dot Product Graph (RDPG) model of Athreya et al. (2018) and the Generalized Random Dot Product Graph (GDPG) model of Rubin-Delanchy et al. (2022) turned out to be very flexible and useful in applications.

RDPG is an special latent position graph, in which each node  $i$  is associated with a latent position  $x_i$  in a latent space  $\mathcal{X}$ , and the probability of connection between two nodes is given by a *kernel* function  $k: \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ . In RDPG, the latent space is an appropriately constrained subspace of Euclidean space  $\mathbb{R}^d$ , and the kernel function is given by the inner product of their latent positions. The latent positions associated to  $n$  nodes can be represented by a matrix  $\mathbf{X} \in \mathbb{R}^{n \times d}$  whose rows are the latent positions, and the matrix of connection probabilities is given by  $\mathbf{P} = \mathbf{X}\mathbf{X}^T$ . The adjacency matrix  $\mathbf{A}_{ij}$  are still independent Bernoulli variables for  $\mathbf{A}_{ij} \sim \text{Bernoulli}(\mathbf{P}_{ij})$ .

However, RDPG produces a non-negative-definite connection probability matrix and cannot explain significant negative eigenvalues in the adjacency matrix  $\mathbf{A}$ . In this way, we generalize the GDPG model to the Generalized Random Dot Product Graph (GDPG) model, in which the connection probability between two nodes is given by the indefinite inner product of their latent positions. Specifically, the matrix of connection probabilities can be expressed as

$$\mathbf{P} = \mathbf{X}\mathbf{I}_{p,q}\mathbf{X}^T.$$

$\mathbf{X}$  is the latent position matrix whose rows are the latent positions, and  $\mathbf{I}_{p,q}$  is a diagonal matrix with  $p$  ones followed by  $q$  minus ones on its diagonal, and  $p \geq 1$  and  $q \geq 0$  are two integers satisfying  $p + q = d$ . Next, we replace  $\mathbf{X}$  with  $\mathbf{V}\mathbf{M}$ , with  $\mathbf{V} \in \mathbb{R}^{n \times d}$  to be the orthonormal matrix, which is the eigenvector matrix of  $\mathbf{X}$ . In this way, let  $\mathbf{Q} = \mathbf{M}\mathbf{I}_{p,q}\mathbf{M}^T$ , then we can describe GDPG

as

$$\mathbf{P} = \mathbf{V}\mathbf{Q}\mathbf{V}^T.$$

$\mathbf{V}$  is the latent structure of the graph with orthonormal columns and  $\mathbf{Q} \in \mathbb{R}^{d \times d}$  is symmetric and representing a (generalized) inner product.

Following the description of GDPG, one can easily see that the Stochastic Block Model (SBM) and the Degree Corrected Stochastic Block Model (DCBM) can be represented as GDPG in which all nodes in the same community have the same latent positions. And  $\mathbf{P}$  in GDPG, not like in SBM and DCBM, doesn't need to have any constant block structure.

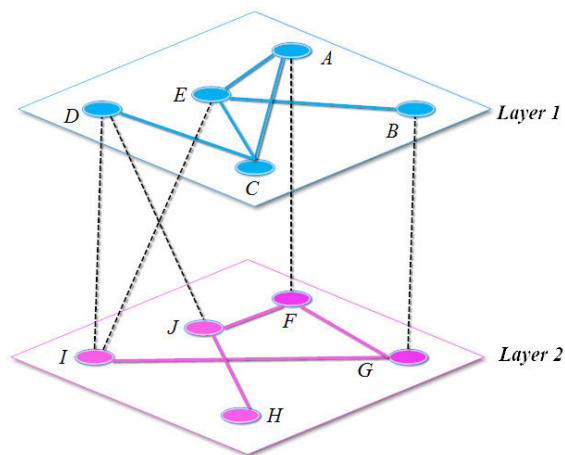
Given a graph distributed as an RDPG, the natural task is to recover the latent positions  $\mathbf{X}$  that give rise to the observed graph. Spectral embedding is a very popular method for this task. Athreya et al. (2016) used the spectral embedding of the adjacency matrix to construct consistent estimates for the latent positions, and showed that the scaled errors converge to a mixture of Gaussian random variables. And Rubin-Delanchy et al. (2022) used the spectral embedding method to produce uniformly consistent latent position estimates with asymptotically Gaussian error for the GDPG model. Other than the spectral embedding method, Xie and Xu (2021) proposed a one-step procedure (OSE) to estimate the latent positions in RDPG. Compared to the spectral methods, the asymptotic sum of squares error of OSE is not higher but the asymptotic covariance matrix of the corresponding row of OSE dominates those of the spectral embeddings in spectra.

## 1.5 Multiplex Networks

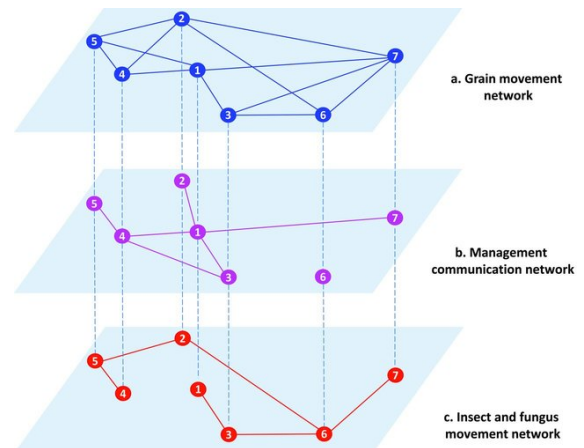
As the real world works with more complex system, not just single layer network in most cases, multilayer networks help us understand the structure and process of the world better. However, it is not natural to extend the theory of single layer network to the multilayer network, letting go

considering layers not just communities among nodes, research on multilayer networks has been flourished in recent several years. Nowadays, Internet affects the world dramatically and people relying on Internet to build and keep relationship becomes more and more popular. The application of multilayer networks in the social relationship draws a lot of researchers' interest. (Borondo et al. (2015); Oro et al. (2018)). Much more situation can be modeled by multilayer networks like transportation systems (Criado et al. (2007); Cardillo et al. (2013)), animal behaviour detection (Finn et al. (2019)) and global trading (A. Alves et al. (2018)), etc.

In a multiplayer network, each layer is equipped with nodes and edges. Nodes in each layer can be different and they can connect with the same layer nodes or different layers'. The following figures show multilayer networks with different nodes in two layers on the left and with same nodes in three layers on the right.



(a) Wang and Zou (2017) shows a schematic representation of multilayer networks with two layers, where each layer has different five nodes and interlayer links can connect different nodes on different layers.



(b) Hernandez Nopsa et al. (2015) introduces a stored-grain system multiplex network, in which nodes in each layer are all the same and there is no connection between nodes in different layers.

Figure 1.1: Multiplex networks examples with connection between layers(left panel) and no connection between layers(right panel)



Although there are many different ways of modeling a multilayer network (see, e.g., an excellent review article of Kivela et al. (2014)), in this dissertation, we consider the case where all layers have the same set of nodes, and all the edges between nodes are drawn within layers, i.e., there are no edges connecting the nodes in different layers. Many authors, who work in a variety of research fields, study this particular version of a multilayer network (Aleta and Moreno (2019), Durante et al. (2017), Han and Dunson (2018), Kao and Porter (2017), MacDonald et al. (2021)). MacDonald et al. (2021) called this type of multilayer network models the *Multiplex Network Model* and argued that it appears in a variety of real life situations.

For example, the multiplex network models include brain networks where nodes are associated with brain regions, and edges are drawn if signals in those regions exhibit some kind of similarity (Sporns (2018)). In this setting, the nodes are the same for each individual network, and there is no connection between brain regions of different individuals. Another type of multiplex networks are trade networks between a set of countries (De Domenico et al. (2015)), where nodes and layers represent, respectively, various countries and commodities in which they are trading. In this case, edges are drawn if countries trade specific products with each other.

## 1.6 COmmon Subspace Independent Edge (COSIE) random graph model

As a very popular case of multiplex network models, Arroyo et al. (2021) and Zheng and Tang (2022) introduced the COmmon Subspace Independent Edge (**COSIE**) random graph model which extends the RDPG and the GDPG to the multilayer setting. The COSIE model describes  $L$  random graphs in which each matrix of connection probability can be written as

$$\mathbf{P}^{(l)} = \mathbf{V}\mathbf{Q}^{(l)}\mathbf{V}^T, \quad l = 1, \dots, L.$$

$\mathbf{Q}^{(l)}$  is a low-dimensional matrix for different connectivity patterns for each graph and  $\mathbf{V}$  is a matrix with orthonormal columns describing a shared latent structure on the nodes. The  $\mathbf{Q}^{(l)}$  matrices need not be diagonal, and can vary with each graph. In particular, each graph can have a different distribution.

It is a natural idea to recover the common subspace  $\mathbf{V}$  after fitting this model. Arroyo et al. (2021) proposed a joint spectral embedding of adjacency matrices (MASE) to estimate the common subspace  $\mathbf{V}$  and the score matrices  $\mathbf{Q}^{(l)}$ . And they also infer that the COSIE model and the MASE embedding can be deployed for a number of subsequent network inference tasks. Jones and Rubin-Delanchy (2020) estimated  $\mathbf{V}$  by jointly embedding the associated adjacency matrices into a suitable latent space. Zheng and Tang (2022) studied parameters estimations using the procedure proposed in Arroyo et al. (2021), for both the undirected case (as considered in Arroyo et al. (2021)) and directed case (Jones and Rubin-Delanchy (2020)). Nielsen and Witten (2018) and Wang et al. (2019) proposed estimation procedures for the common subspace  $\mathbf{V}$  using alternating gradient descent. And for the case of the connectivity patterns  $\mathbf{Q}^{(l)}$  being scalars, they provide bounds for the accuracy of the resulting estimation.

## 1.7 Organization of this Dissertation

This dissertation focuses on the research of multiplex networks where layers can be partitioned into groups with the common subspace structures or community assignments, and all matrices of block connection probabilities can be different. The objective is to recover the clustering function between layers and find the subspace structures or community patterns within layer in the model.

In Chapter 2, this dissertation reviews related work and introduces DIMPLE, DIMPLE-GDPG and DIMPLE-DECOR models. We also explain why introduction of the DIMPLE and the DIMPLE-

GDPG models is imperative, and why analysis of those models requires development of new algorithms. Following it, Section 2.3 introduce notations, required for construction of the algorithms and their subsequent analysis.

Chapter 3 introduces more details about fitting the network models. We develop between layer clustering algorithm to cluster all  $L$  layers into  $M$  groups based on the construction of the adjacency matrix in each layer and then find the common subspace structures, or cluster  $n$  nodes into  $K_m$  communities within each group of layers. In particular, Section 3.1 proposes a between-layer clustering algorithm for both the DIMPLE, the DIMPLE-DECOR and the DIMPLE-GDPG models. Section 3.2 talks about estimation of invariant subspace matrices  $\mathbf{V}^{(m)}$  in the groups of layers in the DIMPLE-GDPG model in (2.1). Section 3.3 provides within-layer clustering procedures in the case of the DIMPLE and the DIMPLE-DECOR network.

Chapter 4 is dedicated to theoretical developments. Specifically, Section 4.2 introduces assumptions that guarantee the between-layer clustering error rates, the within-layer clustering error rates for the DIMPLE and the DIMPLE-DECOR model and the subspace fitting errors in groups of layers in the DIMPLE-GDPG model, that are derived in Sections 4.3, 4.5 and 4.4, respectively. And we show that they tend to zero at a high rate under very simple and intuitive assumptions.

The simulation study in Chapter 5 confirm that the algorithms deliver high precision in a finite parameter settings mainly with the number of nodes  $n$  varying or number of layers  $L$  varying while keeping other parameters still. We also compare the performance of between layer clustering algorithms for Algorithm 1 and a popular algorithm for MMLSBM in the setting of different matrices of block connection probabilities in different layers, and our algorithm performed much better in this setting.

In Chapter 6, the clustering algorithms are applied to the Worldwide Food Trading Network Data 6.1 and Airline Data 6.2. The former successfully separates meat and fruit/vegetable food accord-

ing to the trading community patterns among countries for foods and the latter forms naturally geographical areas distribution from where the flights are originated after clustered into groups.

Chapter 7 concludes the paper with the discussion of its results. Finally, Section 7 contains proofs of the statements in the paper and also provides additional simulations.

## CHAPTER 2: INTRODUCTION TO OUR GENERALISED MODELS

### 2.1 DIverse MultiPLEx (DIMPLE) network models frameworks

This dissertation considers an  $L$ -layer network on the same set of  $n$  vertices  $[n] = \{1, \dots, n\}$ , where the tensor of probabilities of connections  $\mathcal{P} \in [0, 1]^{n \times n \times L}$  is formed by layers  $\mathbf{P}^{(l)}$ ,  $l \in [L]$ , that can be partitioned into  $M$  groups with the common subspace structure or community assignment. We consider a multiplex network with  $L$  layers of  $M$  types, so that there exists a label function  $c : [L] \rightarrow [M]$ . We assume that the layers of the network follow the Generalized Dot Product Graph (**GDPG**) model of Rubin-Delanchy et al. (2022), where each group of layers is embedded in its own ambient subspace, but otherwise all matrices of connection probabilities can be different. Specifically,  $\mathbf{P}^{(l)}$ ,  $l \in [L]$ , are given by

$$\mathbf{P}^{(l)} = \mathbf{V}^{(m)} \mathbf{Q}^{(l)} (\mathbf{V}^{(m)})^T, \quad m = c(l), \quad l \in [L], \quad m \in [M], \quad (2.1)$$

where  $\mathbf{Q}^{(l)} = (\mathbf{Q}^{(l)})^T$  and  $\mathbf{V}^{(m)}$  are matrices with orthonormal columns, such that all entries of  $\mathbf{P}^{(l)}$  are in  $[0, 1]$ . We shall call this model the DIverse MultiPLEx Generalized Dot Product Graph (**DIMPLE-GDPG**).

In a common particular case, where layers of the network follow the Stochastic Block Models (**SBM**), (2.1) implies that the groups of layers have common community structures but matrices of block connection probabilities can be all different. Then, the matrix of probabilities of connection in layer  $l$  can be expressed as

$$\mathbf{P}^{(l)} = \mathbf{Z}^{(m)} \mathbf{B}^{(l)} (\mathbf{Z}^{(m)})^T, \quad m = c(l), \quad l \in [L], \quad m \in [M], \quad (2.2)$$

where  $\mathbf{Z}^{(m)}$  is the clustering matrix in the layer of type  $m = c(l)$  and  $\mathbf{B}^{(l)} = (\mathbf{B}^{(l)})^T$  is a matrix of block probabilities,  $l \in [L]$ . In order to distinguish this special case, we shall refer to (2.2) as simply the **DIMPLE** model.

Another particular case, where layers of the network follow the Degree Corrected Stochastic Block Models (**DCBM**), degree parameters supply more generalized application than **SBM**. the matrix of probabilities of connection in layer  $l$  can be expressed as

$$\mathbf{P}^{(l)} = \Theta^{(m)} \mathbf{Z}^{(m)} \mathbf{B}^{(l)} (\mathbf{Z}^{(m)})^T \Theta^{(m)}, \quad m = c(l), l = 1, \dots, L, m = 1, \dots, M \quad (2.3)$$

Matrix  $\Theta^{(m)}$  is the diagonal matrix of the node weights with elements  $\theta_i^{(m)}, i = 1, \dots, n$ . We shall refer to this model as the DEgree CORrected DIverse MultiPLEx model (**DIMPLE-DECOR**).

In all models, one observes the adjacency tensor  $\mathcal{A} \in \{0, 1\}^{n \times n \times L}$  with layers  $\mathbf{A}^{(l)}$  such that  $\mathbf{A}^{(l)}(i, j) = \mathbf{A}^{(l)}(j, i)$  and, for  $1 \leq i < j \leq n$  and  $l \in [L]$ , where  $\mathbf{A}^{(l)}(i, j)$  are the Bernoulli random variables with  $\mathbb{P}(\mathbf{A}^{(l)}(i, j) = 1) = \mathbf{P}^{(l)}(i, j)$ , and they are independent from each other. The objective is to recover the layer clustering matrix  $\mathbf{C}$ , as well as the community assignment matrices  $\mathbf{Z}^{(m)}$  in the case of model (2.2), or the subspaces  $\mathbf{V}^{(m)}$  in the case of model (2.1).

Note that, since the SBM and DCBM are particular cases of the GDPG, (2.2) and (2.3) are particular cases of (2.1) (see Section 3.1 for further explanations). Nevertheless, the problems associated with (2.1) and (2.2) and (2.3) are somewhat different. While recovering matrices  $\mathbf{V}^{(m)}$  is an estimation problem, finding communities in the groups of layers, corresponding to clustering matrices  $\mathbf{Z}^{(m)}$ , is a clustering problem. For this reason, we study all models, (2.1), (2.2) and (2.3), in this dissertation.

This dissertation makes several key contributions.

1. It is the first one that considers the SBM-equipped multiplex network, where both the probabilities of connections and the community structures can vary. And it considers the DCBM-equipped multiplex network with degree corrected parameters. In this sense, This dissertation generalizes both the models, where the community structure is identical in all layers, and the ones, where there are only  $M$  types of the matrices of the connection probabilities, so that the probability tensor has collections of identical layers. Those models correspond, respectively, to  $M = 1$ , and to  $\mathbf{B}^{(l)} = \mathbf{B}^{(m)}$  with  $m = c(l)$  in (2.2).
2. This dissertation generalizes the COmmon Subspace Independent Edge (**COSIE**) random graph model of Arroyo et al. (2021) and Zheng and Tang (2022), which corresponds to  $M = 1$  in (2.1).
3. This dissertation develops a novel between-layer clustering algorithm that works for all network models, DIMPLE, DIMPLE-GDPG and DIMPLE-DECOR, and derive expressions for the clustering errors under very simple and intuitive assumptions. Our simulations confirm that the between-layer and the within-layer clustering algorithms deliver high precision in a finite parameter settings. In addition, if  $M = 1$ , our subspace recovery error compares favorably to the ones in Arroyo et al. (2021) and Zheng and Tang (2022), due to employment of a different algorithm.
4. Since the DIMPLE model generalizes two types of popular SBM-equipped multiplex networks models, this dissertation opens a gateway for testing/model selection. In particular, one can test whether communities persist throughout the layers of the network, or whether layers can be partitioned into groups for which this is true, which is equivalent to testing the hypothesis that  $M = 1$  in (2.2). Alternatively, one can test the hypothesis that all matrices  $\mathbf{B}^{(l)}$  in a group of layers are the same that reduces to  $\mathbf{B}^{(l)} = \mathbf{B}^{(m)}$  with  $m = c(l)$  in (2.2). One can test similar hypotheses in the case of the DIMPLE-GDPG network model.

## 2.2 Justification of the model and related work

In the last few years, a number of authors studied multiplex network models. The vast majority of the paper assumed that all layers of the network follow the Stochastic Block Model (SBM). The latter is due to the fact that the SBM, according to Olhede and Wolfe (2014), provides a universal tool for description of time-independent stochastic network data. It is also very common in applications. For example, Sporns (2018) argues that stochastic block models provide a powerful tool for brain studies. In fact, in the last few years, such models have been widely employed in brain research (see, e.g., Crossley et al. (2013), Faskowitz et al. (2018), Nicolini et al. (2017), among others).

While the scientific community considered various types of multiplex networks in general, and the SBM-equipped multiplex networks in particular (see e.g., Brodka et al. (2018), Kao and Porter (2017), Mercado et al. (2018) among others), the theoretically inclined papers in the field of statistics mainly have been investigating the case where communities persist throughout all layers of the network. This includes studying the so called “checker board model” in Chi et al. (2020), where the matrices of block probabilities take only finite number of values, and communities are the same in all layers. The tensor block models of Wang and Zeng (2019) and Han et al. (2021) belong to the same category. In recent years, statistics publications extended this type of research to the case where community structure is preserved in all layers of the network, but the matrices of block connection probabilities can take arbitrary values (see, e.g., Bhattacharyya and Chatterjee (2020), Lei et al. (2019), Lei and Lin (2022), Paul and Chen (2016), Paul and Chen (2020) and references therein). The authors studied precision of community detection, and provided theoretical and numerical comparisons between various techniques that can be employed in this case.

The recent years saw the latent position graphical models are with much wider applications and they were extended to the COSIE random graph model for the multilayer setting. However, COSIE



postulates that the layer networks are embedded into the same invariant subspace, which is very similar to the assumption of persistent communities in all layers of a multiplex network. Nevertheless, there are many real life scenarios where the assumption, that all layers of the network have the same communities or are embedded into the same subspace is too restrictive. For example, it is known that some brain disorders are associated with changes in brain network organizations (see, e.g., Buckner and DiNicola (2019)), and that alterations in the community structure of the brain have been observed in several neuropsychiatric conditions, including Alzheimer disease (see, e.g., Chen et al. (2016)), schizophrenia (see, e.g., Stam (2014)) and epilepsy disease (see, e.g., Munsell et al. (2015)). In this case, one would like to examine brains networks of the individuals with and without brain disorder to derive the differences in community structures. Similar situations occur when one examines several groups of networks, often corresponding to subjects with different biological conditions (e.g., males/females, healthy/diseased, etc.)

One of the possible approaches here is to assume that both, the community structures and the probabilities of connections in the network layers, will be identical under the same biological condition and dissimilar for different conditions. This type of setting, called the **Mixture MultiLayer Stochastic Block Model (MMLSBM)** assumes that all layers can be partitioned into a few different types, such that each distinct type of layers is equipped with its own community structure and a unique matrix of block connection probabilities, and that both are identical within the same type of layers. In the context of a GDPG-based multiplex network, this extension leads directly to low-rank tensor estimation, the problem that received a great deal of attention in the last five years.

Specifically, if  $M = 1$ , then the DIMPLE model (2.2) reduces to the multiplex models in Bhattacharyya and Chatterjee (2020), Lei et al. (2019), Lei and Lin (2022), Paul and Chen (2016), Paul and Chen (2020) with the persistent communities, and it becomes the MMLSBM of Stanley et al. (2019), Jing et al. (2021) and Fan et al. (2021), if  $\mathbf{B}^{(l)}$  takes only  $M$  distinct values, i.e.,  $\mathbf{B}^{(l)} = \mathbf{B}^{(m)}$  for  $c(l) = m$ . Similarly, if  $M = 1$ , the DIMPLE-GDPG model in (2.1) reduces to

the COSIE model in Arroyo et al. (2021) and Zheng and Tang (2022), and it reduces to a low rank tensor estimation of Luo et al. (2021) and Zhang and Xia (2018a) if all matrices  $\mathbf{Q}^{(l)}$  are identical within a group of layers.

In essence, the conclusion of the discussion above is that so far authors considered two complementary types of settings for multiplex networks. In the first of them, all layers of the network are embedded into the same subspaces in the case of the GDPG, or have the same communities if the layers of the network are equipped with SBMs. In the second one, the layers may be embedded into different subspaces, but the tensor of connection probabilities has a low rank, which reduces to MMLSBM if layers follow the SBM.

Hence, the natural generalization of those two scenarios would be the setting, where the layers of the network can be partitioned into groups, each with the distinct subspace or community structure. Such multiplex network can be viewed as a concatenation of several multiplex networks that follow COSIE model or Stochastic Block Models with persistent community structure. On the other hand, such networks will reduce to a low rank tensor or the MMLSBM if networks in the group of layers have identical probabilities of connections.

We feel that the above extension is imperative for a variety of reasons. As one can easily see, the existing models are complementary in nature and are usually adopted without any consideration of the alternatives. The DIMPLE-GDPG and the DIMPLE models allow to forgo this choice. They also open the gate for testing this alternatives and adopting the one which better fits the data. Our real data examples show that in real life situations the DIMPLE or the DIMPLE-GDPG model provides a better summary of data than the MMLSBM.

The new DIMPLE-GDPG model requires development of new algorithms, since the probability tensor  $\mathcal{P}$  associated with the DIMPLE-GDPG model in (2.1) does not have a low rank, due to the fact that all matrices  $\mathbf{Q}^{(l)}$  are different. For this reason, techniques and theoretical assessments

developed for low rank tensors do not work in the case of the DIMPLE-GDPG model. Similarly, since the matrices of the block connection probabilities take different values in each of the layers, techniques employed in Jing et al. (2021) and Fan et al. (2021) cannot be applied in the new environment of DIMPLE.

Indeed, the TWIST algorithm of Jing et al. (2021) is based on the alternating regularized low rank approximations of the adjacency tensor, which relies on the fact that the tensor of connection probabilities is truly low rank in the case of MMLSBM. This, however, is not true for the DIMPLE model, where the matrices of block connection probabilities vary from layer to layer. On the other hand, the ALMA algorithm of Fan et al. (2021) exploits the fact that the matrices of connection probabilities are identical in the groups of layers with the same community structures. This is no longer the case in the environment of the DIMPLE model, where matrices of connection probabilities are all different for different layers. Specifically, Section B compares the MMLSBM and the DIMPLE model introduced in this paper and shows that, while algorithms designed for the DIMPLE model work well for the MMLSBM, the algorithms designed for the MMLSBM display poor performance if data are generated according to the DIMPLE model.

### 2.3 Notations

For any integer  $n$ , we denote  $[n] = \{1, \dots, n\}$ . We denote tensors by calligraphy letters and matrices by bold letters. Denote by  $\mathcal{M}_{N,K}$  the set of the *clustering* matrices for  $N$  objects partitioned into  $K$  groups

$$\mathcal{M}_{N,K} = \{ \mathbf{X} \in \{0,1\}^{N \times K}, \quad \mathbf{X}\mathbf{1} = \mathbf{1}, \quad \mathbf{X}^T\mathbf{1} \neq \mathbf{0} \},$$

where  $\mathbf{X} \in \mathcal{M}_{N,K}$  are such that  $\mathbf{X}_{i,j} = 1$  if node  $i$  is in cluster  $j$  and  $\mathbf{X}_{i,j} = 0$  otherwise. For any matrix  $\mathbf{X}$ , denote the Frobenius, the infinity and the operator norm by  $\|\mathbf{X}\|_F$ ,  $\|\mathbf{X}\|_\infty$  and  $\|\mathbf{X}\|$ ,

respectively, and its  $r$ -th largest singular value by  $\sigma_r(\mathbf{X})$ . Let  $\|\mathbf{X}\|_{2 \rightarrow \infty} = \sup_{\|z\|=1} \|\mathbf{X}z\|_\infty$ .

The column  $j$  and the row  $i$  of a matrix  $\mathbf{Q}$  are denoted by  $\mathbf{Q}(:, j)$  and  $\mathbf{Q}(i, :)$ , respectively. Denote the identity and the zero matrix of size  $K$  by, respectively,  $\mathbf{I}_K$  and  $\mathbf{0}_K$  (where  $K$  is omitted when this does not cause ambiguity). Denote

$$\mathcal{O}_{n,K} = \{\mathbf{X} \in \mathbb{R}^{n \times K} : \mathbf{X}^T \mathbf{X} = \mathbf{I}_K\}, \quad \mathcal{O}_n = \mathcal{O}_{n,n}. \quad (2.4)$$

Let  $\text{vec}(\mathbf{X})$  be the vector obtained from matrix  $\mathbf{X}$  by sequentially stacking its columns. Denote by  $\mathbf{X} \otimes \mathbf{Y}$  the Kronecker product of matrices  $\mathbf{X}$  and  $\mathbf{Y}$ . Denote  $n$ -dimensional vector with unit components by  $\mathbf{1}_n$ . Denote diagonal of a matrix  $\mathbf{A}$  by  $\text{diag}(\mathbf{A})$ . Also, denote the  $M$ -dimensional diagonal matrix with  $a_1, \dots, a_M$  on the diagonal by  $\text{diag}(a_1, \dots, a_M)$ .

For any matrix  $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ , denote its projection on the nearest rank  $K$  matrix by  $\Pi_K(\mathbf{X})$ , that is, if  $\sigma_k$  are the singular values, and  $u_k$  and  $v_k$  are the left and the right singular vectors of  $\mathbf{X}$ ,  $k = 1, \dots, r$ , then

$$\mathbf{X} = \sum_{k=1}^r \sigma_k u_k v_k^T \implies \Pi_K(\mathbf{X}) = \sum_{k=1}^{\min(r,K)} \sigma_k u_k v_k^T.$$

For any matrices  $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$  and  $\mathbf{U} \in \mathcal{O}_{n_1, K}$ ,  $K \leq n_1$ , projection of  $\mathbf{X}$  on the column space of  $\mathbf{U}$  and on its orthogonal space are defined, respectively, as

$$\Pi_{\mathbf{U}}(\mathbf{X}) = \mathbf{U}\mathbf{U}^T \mathbf{X}, \quad \Pi_{\mathbf{U}^\perp}(\mathbf{X}) = (\mathbf{I} - \Pi_{\mathbf{U}})\mathbf{X}.$$

Following Kolda and Bader (2009), we define the following tensor operations. For any tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n_3}$ , their product  $\mathcal{X} \times_3 \mathbf{A}$  along dimension 3 is a tensor in

$\mathbb{R}^{n_1 \times n_2 \times m}$  with elements

$$[\mathcal{X} \times_3 \mathbf{A}](i_1, i_2, j) = \sum_{i_3=1}^{n_3} \mathbf{A}(j, i_3) \mathcal{X}(i_1, i_2, i_3), \quad j = 1, \dots, m.$$

If  $\mathcal{Y} \in \mathbb{R}^{m \times n_2 \times n_3}$  is another tensor, the product between tensors  $\mathcal{X}$  and  $\mathcal{Y}$  along dimensions (2,3), denoted by  $\mathcal{X} \times_{2,3} \mathcal{Y}$ , is a matrix in  $\mathbb{R}^{n_1 \times m}$  with elements

$$[\mathcal{X} \times_{2,3} \mathcal{Y}](i_1, i_2) = \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3} \mathcal{X}(i_1, j_2, j_3) \mathcal{Y}(i_2, j_2, j_3), \quad i_1 = 1, \dots, n_1, \quad i_2 = 1, \dots, m.$$

The mode-3 matricization of tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is a matrix  $\mathcal{M}_3(\mathcal{X}) = \mathbf{X} \in \mathbb{R}^{n_3 \times (n_1 n_2)}$  with rows  $\mathbf{X}(i, :) = [\text{vec}(\mathcal{X}(:, :, i))]^T$ . Please, see Kolda and Bader (2009) for a more extensive discussion of tensor operations and their properties.

We use the  $\sin \Theta$  distances to measure the separation between two subspaces with orthonormal bases  $\mathbf{U} \in \mathcal{O}_{n,K}$  and  $\tilde{\mathbf{U}} \in \mathcal{O}_{n,K}$ , respectively. Suppose the singular values of  $\mathbf{U}^T \tilde{\mathbf{U}}$  are  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_K > 0$ . Then

$$\Theta(\mathbf{U}, \tilde{\mathbf{U}}) = \text{diag}(\cos^{-1}(\sigma_1), \dots, \cos^{-1}(\sigma_K))$$

are the principle angles. Quantitative measures of the distance between the column spaces of  $\mathbf{U}$  and  $\tilde{\mathbf{U}}$  are then

$$\|\sin \Theta(\mathbf{U}, \tilde{\mathbf{U}})\| = \sqrt{1 - \sigma_{\min}^2(\mathbf{U}^T \tilde{\mathbf{U}})} \quad \text{and} \quad \|\sin \Theta(\mathbf{U}, \tilde{\mathbf{U}})\|_F = \sqrt{K - \|\mathbf{U}^T \tilde{\mathbf{U}}\|_F^2} \quad (2.5)$$

Some convenient characterizations of those distances can be found in Section 8.1 of Cai and Zhang (2018a).

Finally, we shall use  $C$  for a generic positive constant that can take different values and is independent of  $L, n, M, K$  and graph densities.

## CHAPTER 3: FITTING THE MODELS

In this dissertation, we consider a multiplex network with  $L$  layers of  $M$  types, where  $L_m$  is the number of layers of type  $m$ ,  $m \in [M]$ . Let  $\mathbf{C} \in \mathcal{M}(L, M)$  be the layer clustering matrix. A layer of type  $m$  has an ambient dimension  $K_m$ . In the case of model (2.2), a layer of type  $m$  has  $K_m$  communities, and  $n_{k,m}$  is the number of nodes of type  $k$  in the layer of type  $m$ ,  $k \in [K_m]$ ,  $m \in [M]$ , so that

$$\mathbf{D}_z^{(m)} = (\mathbf{Z}^{(m)})^T \mathbf{Z}^{(m)} = \text{diag}(n_{1,m}, \dots, n_{K_m,m}). \quad (3.1)$$

For DIMPLE-DECOR, we denote

$$\Theta_z^{(m)} = (\mathbf{Z}^{(m)})^T (\Theta^{(m)})^2 (\mathbf{Z}^{(m)}), \quad \gamma_k^{(m)} = \sum_{z^{(m)}=k} (\theta_i^{(m)})^2, \quad (3.2)$$

and observe that

$$\Theta_z^{(m)} = (\Theta^{(m)} \mathbf{Z}^{(m)})^T (\Theta^{(m)} \mathbf{Z}^{(m)}) = \text{diag}(\gamma_1^{(m)}, \dots, \gamma_K^{(m)}). \quad (3.3)$$

Here  $\gamma_k^{(m)}$  is the squared norm of the node weights in class  $k$  in the layer of type  $m$ .

### 3.1 Between-layer clustering

First, we show that model (2.2) is a particular case of model (2.1). Indeed, denote  $\mathbf{U}_z^{(m)} = \mathbf{Z}^{(m)} \left( \mathbf{D}_z^{(m)} \right)^{-1/2}$ , where matrices  $\mathbf{D}_z^{(m)}$  are defined in (3.1). For DIMPLE-DECOR,  $\mathbf{U}_z^{(m)} = \Theta^{(m)} \mathbf{Z}^{(m)} (\Theta_z^{(m)})^{-1/2}$ . Since  $\mathbf{U}_z^{(m)} \in \mathcal{O}_{n, K_m}$  in both DIMPLE and DIMPLE-DECOR, matrices

$\mathbf{P}^{(l)}$  in (2.2) and (2.3) can be written as

$$\mathbf{P}^{(l)} = \mathbf{U}_z^{(m)} \mathbf{B}_D^{(l)} (\mathbf{U}_z^{(m)})^T, \quad (3.4)$$

Here,  $\mathbf{B}_D^{(l)} = \sqrt{\mathbf{D}_z^{(m)}} \mathbf{B}^{(l)} \sqrt{\mathbf{D}_z^{(m)}}$  for DIMPLE model and  $\mathbf{B}_D^{(l)} = \sqrt{\mathbf{\Theta}_z^{(m)}} \mathbf{B}^{(l)} \sqrt{\mathbf{\Theta}_z^{(m)}}$  for DIMPLE-DECOR model. Therefore, (2.2) and (2.3) are particular cases of (2.1) with  $\mathbf{V}^{(m)} = \mathbf{U}_z^{(m)}$  and  $\mathbf{Q}^{(l)} = \mathbf{B}_D^{(l)}$ . For this reason, we are going to cluster groups of layers in the more general setting (2.1) of DIMPLE-GDPG.

In order to find the clustering matrix  $\mathbf{C}$ , observe that matrices  $\mathbf{P}^{(l)}$  in (2.1) can be written as

$$\mathbf{P}^{(l)} = \mathbf{V}^{(m)} \mathbf{O}_Q^{(l)} \mathbf{S}_Q^{(l)} (\mathbf{O}_Q^{(l)})^T (\mathbf{V}^{(m)})^T, \quad l \in [L] \quad (3.5)$$

where

$$\mathbf{Q}^{(l)} = \mathbf{O}_Q^{(l)} \mathbf{S}_Q^{(l)} (\mathbf{O}_Q^{(l)})^T, \quad l \in [L], \quad (3.6)$$

is the singular value decomposition (SVD) of  $\mathbf{Q}^{(l)}$  with  $\mathbf{O}_Q^{(l)} \in \mathcal{O}_{n, K_m}$ ,  $m = c(l)$ , and diagonal matrix  $\mathbf{S}_Q^{(l)}$ . In order to extract common information from matrices  $\mathbf{P}^{(l)}$ , we consider the SVD of  $\mathbf{P}^{(l)}$

$$\mathbf{P}^{(l)} = \mathbf{U}_{P,l} \mathbf{\Lambda}_{P,l} (\mathbf{U}_{P,l})^T, \quad \mathbf{U}_{P,l} \in \mathcal{O}_{n, K_m}, \quad l \in [L], \quad m = c(l) \quad (3.7)$$

and relate it to expansion (3.5). If, as we assume later, matrices  $\mathbf{Q}^{(l)}$  are of full rank, then  $\mathbf{O}_Q^{(l)} \in \mathcal{O}_{K_m}$ , so that  $\mathbf{O}_Q^{(l)} (\mathbf{O}_Q^{(l)})^T = (\mathbf{O}_Q^{(l)})^T \mathbf{O}_Q^{(l)} = \mathbf{I}_{K_m}$ ,  $m = c(l)$ . Therefore,  $\mathbf{V}^{(m)} \mathbf{O}_Q^{(l)} \in \mathcal{O}_{n, K_m}$ , and expansion (3.5) is just another way of writing the SVD of  $\mathbf{P}^{(l)}$ . Hence, up to the  $K_m$ -dimensional rotation  $\mathbf{O}_Q^{(l)}$ , matrices  $\mathbf{V}^{(m)}$  and  $\mathbf{U}_{P,l}$  are equal to each other when  $c(l) = m$ .

Since matrices  $\mathbf{O}_Q^{(l)}$  are unknown, we introduce alternatives to  $\mathbf{U}_{P,l}$ :

$$\mathbf{U}_{P,l} (\mathbf{U}_{P,l})^T = \mathbf{V}^{(m)} \mathbf{O}_Q^{(l)} (\mathbf{V}^{(m)} \mathbf{O}_Q^{(l)})^T = \mathbf{V}^{(m)} (\mathbf{V}^{(m)})^T, \quad m = c(l), \quad (3.8)$$

---

**Algorithm 1:** The between-layer clustering
 

---

**Input:** Adjacency tensor  $\mathcal{A} \in \{0, 1\}^{n \times n \times L}$ ; number of groups of layers  $M$ ; ambient dimension  $K^{(l)}$  of each layer  $l \in [L]$ ; parameter  $\epsilon$

**Output:** Estimated clustering matrix  $\widehat{\mathbf{C}} \in \mathcal{M}_{L,M}$

**Steps:**

- 1: Find the SVDs  $\Pi_{K^{(l)}}(\mathbf{A}^{(l)}) = \widehat{\mathbf{U}}_{A,l} \widehat{\mathbf{\Lambda}}_{P,l} (\widehat{\mathbf{U}}_{A,l})^T, l \in [L]$
  - 2: Form matrix  $\widehat{\mathbf{\Theta}} \in \mathbb{R}^{n^2 \times L}$  with columns  $\widehat{\mathbf{\Theta}}(:, l) = \text{vec}(\widehat{\mathbf{U}}_{A,l} (\widehat{\mathbf{U}}_{A,l})^T)$
  - 3: Construct the SVD of  $\widehat{\mathbf{\Theta}}$  using (3.12) and obtain matrix  $\widehat{\mathbf{W}} = \widehat{\mathbf{W}}(:, 1 : M) \in \mathcal{O}_{L,M}$
  - 4: Cluster  $L$  rows of  $\widehat{\mathbf{W}}$  into  $M$  clusters using  $(1 + \epsilon)$ -approximate  $K$ -means clustering. Obtain estimated clustering matrix  $\widehat{\mathbf{C}}$
- 

which depend on  $l$  only via  $m = c(l)$  and are uniquely defined for  $l \in [L]$ . The latter implies that the between-layer clustering can be based on the matrices  $\mathbf{U}_{P,l} (\mathbf{U}_{P,l})^T, l \in [L]$ , or rather on their vectorized versions. Denote

$$\mathbf{D}_c = \mathbf{C}^T \mathbf{C} = \text{diag}(L_1, \dots, L_M), \quad \mathbf{W} = \mathbf{C} (\mathbf{D}_c)^{-1/2} \in \mathcal{O}_{L,M} \quad (3.9)$$

Consider matrices  $\mathbf{\Psi} \in \mathbb{R}^{n^2 \times M}$  and  $\mathbf{\Theta} \in \mathbb{R}^{n^2 \times L}$  with respective columns

$$\mathbf{\Psi}(:, m) = \text{vec}(\mathbf{V}^{(m)} (\mathbf{V}^{(m)})^T), \quad \mathbf{\Theta}(:, l) = \text{vec}(\mathbf{V}^{(c(l))} (\mathbf{V}^{(c(l))})^T) = \text{vec}(\mathbf{U}_{P,l} (\mathbf{U}_{P,l})^T),$$

where  $m \in [M], l \in [L]$ . It is easy to see that

$$\mathbf{\Theta} = \mathbf{\Psi} \mathbf{C}^T, \quad \mathbf{\Psi} = \mathbf{\Theta} \mathbf{C} \mathbf{D}_c^{-1}, \quad (3.10)$$

so that clustering assignment can be recovered by spectral clustering of columns of an estimated version of matrix  $\mathbf{\Theta}$ .

For this purpose, consider layers  $\mathbf{A}^{(l)} = \mathcal{A}(:, :, l)$  of the adjacency tensor  $\mathcal{A}$  and construct the SVDs



of their rank  $K_m$  projections  $\Pi_{K_m}(\mathbf{A}^{(l)})$ :

$$\Pi_{K_m}(\mathbf{A}^{(l)}) = \widehat{\mathbf{U}}_{A,l} \widehat{\mathbf{\Lambda}}_{P,l} (\widehat{\mathbf{U}}_{A,l})^T, \quad \widehat{\mathbf{U}}_{A,l} \in \mathcal{O}_{n,K_m}, \quad m = c(l), \quad l \in [L]. \quad (3.11)$$

Then, replace matrix  $\Theta$  by its proxy  $\widehat{\Theta}$  with columns  $\widehat{\Theta}(:, l) = \text{vec}(\widehat{\mathbf{U}}_{A,l} (\widehat{\mathbf{U}}_{A,l})^T)$ . The major difference between  $\Theta$  and  $\widehat{\Theta}$ , however, is that, under assumptions in Section 4.2,  $\text{rank}(\Theta) = M$  while, in general,  $\text{rank}(\widehat{\Theta}) = L \gg M$ . If the SVD of  $\widehat{\Theta}$  is

$$\widehat{\Theta} = \widetilde{\mathbf{V}} \widetilde{\mathbf{\Lambda}} \widetilde{\mathbf{W}}, \quad \widetilde{\mathbf{V}} \in \mathcal{O}_{n^2,L}, \quad \widetilde{\mathbf{W}} \in \mathcal{O}_L, \quad (3.12)$$

then, we can form reduced matrices

$$\widehat{\mathbf{V}} = \widetilde{\mathbf{V}}(:, 1 : M) \in \mathcal{O}_{n^2,M}, \quad \widehat{\mathbf{W}} = \widetilde{\mathbf{W}}(:, 1 : M) \in \mathcal{O}_{L,M}, \quad (3.13)$$

and apply clustering to the rows of  $\widehat{\mathbf{W}}$  rather than to the rows of  $\widetilde{\mathbf{W}}$ . The latter results in Algorithm 1. We use  $(1+\epsilon)$ -approximate  $K$ -means clustering to obtain the final clustering assignments. There exist efficient algorithms for solving the  $(1+\epsilon)$ -approximate  $K$ -means problem (see, e.g., Kumar et al. (2004)). We denote

$$\widehat{\mathbf{D}}_c = \widehat{\mathbf{C}}^T \widehat{\mathbf{C}}, \quad \widehat{\mathbf{W}} = \widehat{\mathbf{C}} \widehat{\mathbf{D}}_c^{-1/2} \in \mathcal{O}_{L,M} \quad (3.14)$$

Observe that clustering procedure above relies on the knowledge of the ambient dimension  $K_m$ , which is associated with the unknown group membership  $m = c(l)$ . Instead of assuming that  $K_m$  are known, as it is done in Jing et al. (2021) and Fan et al. (2021), we assume that one knows the ambient dimension  $K^{(l)}$  of the GDPG in every layer  $l \in [L]$  of the network. This is a very common assumption and is imposed in almost every paper that studies latent position or block model equipped networks (see, e.g., Athreya et al. (2018), Rubin-Delanchy et al. (2022), Gao et al.

(2018), Gao et al. (2017)). In this case, one can replace  $K_m$  in (3.11) by  $K^{(l)}$ . We further discuss this issue in Remark 2.

**Remark 1. Unknown number of layers.** While Algorithm 1 assumes  $M$  to be known, in many practical situations this is not true, and the value of  $M$  has to be discovered from data. Identifying the number of clusters is a common issue in data clustering, and it is a separate problem from the process of actually solving the clustering problem with a known number of clusters. A common method for finding the number of clusters is the so called “elbow” method that looks at the fraction of the variance explained as a function of the number of clusters. The method is based on the idea that one should choose the smallest number of clusters, such that adding another cluster does not significantly improve fitting of the data by a model. There are many ways to determine the “elbow”. For example, one can base its detection on evaluation of the clustering error in terms of an objective function, as in, e.g., Zhang et al. (2012). Another possibility is to monitor the eigenvalues of the non-backtracking matrix or the Bethe Hessian matrix, as it is done in Le and Levina (2015). One can also employ a simple technique of checking the eigen-gaps of the matrix  $\tilde{\Lambda}$  in (3.12), as it has been discussed in von Luxburg (2007), or use a scree plot as it is done in Zhu and Ghodsi (2006).

**Remark 2. Unknown ambient dimensions.** In this paper, for the purpose of methodological developments, we assume that the ambient dimension  $K^{(l)}$  of each layer of the network is known (which corresponds to the known number of communities in the case of the DIMPLE model). This is a common assumption, and everything in the Remark 1 can also be applied to this case. Here,  $K^{(l)} = K_m$  with  $m = c(l)$ . One can, of course, assume that the values of  $K_m$ ,  $m \in [M]$ , are known. However, since group labels are interchangeable, in the case of non-identical subspace dimensions (numbers of communities), it is hard to choose, which of the values corresponds to which of the groups. This is actually the reason why Jing et al. (2021) and Fan et al. (2021), who imposed this assumption, used it only in theory, while their simulations and real data examples are

all restricted to the case of equal number of communities in all layers  $K_m = K$ ,  $m \in [M]$ . On the contrary, knowledge of  $K^{(l)}$  allows one to deal with different ambient dimensions (number of communities) in the groups of layers in simulations and real data examples.

Of course, if  $K_m$  are all different, e.g.,  $M = 3$ ,  $K_1 = 2$ ,  $K_2 = 3$  and  $K_3 = 4$ , this seems to imply that one can use this information for clustering of layers. However, this is not true in general. Also, in practice, the values of  $K^{(l)}$  are estimated, so precision of the clustering procedure based entirely on the ambient dimensions of layers is questionable at best.

### 3.2 Fitting invariant subspaces in groups of layers in the DIMPLE-GDPG model

If we knew the true clustering matrix  $\mathbf{C}$  and the true probability tensor  $\mathcal{P} \in \mathbb{R}^{n \times n \times L}$  with layers  $\mathbf{P}^{(l)}$  given by (2.1), then we could average layers with identical subspace structures. Precision of estimating  $\mathbf{V}^{(m)}$ , however, depends on whether the eigenvalues of  $\mathbf{Q}^{(l)}$  with  $c(l) = m$  add up. Since the latter is not guaranteed, one can alternatively add the squares  $\mathbf{G}^{(l)} = (\mathbf{P}^{(l)})^2$ , obtaining

$$\sum_{c(l)=m} \mathbf{G}^{(l)} = \sum_{c(l)=m} (\mathbf{P}^{(l)})^2 = \sum_{c(l)=m} \mathbf{V}^{(m)} (\mathbf{Q}^{(l)})^2 (\mathbf{V}^{(m)})^T, \quad m \in [M]$$

In this case, the eigenvalues of  $(\mathbf{Q}^{(l)})^2$  are all positive which ensures successful recovery of matrices  $\mathbf{V}^{(m)}$ .

Note that, however,  $(\mathbf{A}^{(l)})^2$  is not an unbiased estimator of  $(\mathbf{P}^{(l)})^2$ . Indeed, while  $\mathbb{E}((\mathbf{A}^{(l)})^2)_{i,j} = ((\mathbf{P}^{(l)})^2)_{i,j}$  for  $i \neq j$ , for the diagonal elements, one has

$$\mathbb{E}((\mathbf{A}^{(l)})^2)_{i,i} = (\mathbf{P}^{(l)})^2_{i,i} + \sum_j [(\mathbf{P}^{(l)})_{i,j} - (\mathbf{P}^{(l)})^2_{i,j}].$$

Therefore, following Lei and Lin (2022), we evaluate the degree vector  $\hat{\mathbf{d}}^{(l)} = \mathbf{A}^{(l)} \mathbf{1}_n$  and form

---

**Algorithm 2:** Estimating invariant subspaces

---

**Input:** Adjacency tensor  $\mathcal{A} \in \{0, 1\}^{n \times n \times L}$ ; number of groups of layers  $M$ ; ambient dimensions  $K_m, m \in [M]$ , of each group of layers; estimated clustering matrix  $\widehat{\mathbf{C}} \in \mathcal{M}_{L, M}$

**Output:** Estimated invariant subspaces  $\widehat{\mathbf{V}}^{(m)}, m \in [M]$

**Steps:**

**1:** Construct tensor  $\widehat{\mathcal{G}}$  with layers  $\widehat{\mathbf{G}}^{(l)}$  given by (3.15),  $l \in [L]$

**2:** Construct tensor  $\widehat{\mathcal{H}}$  using formula (3.16)

**3:** Construct the SVDs of layers  $\widehat{\mathbf{H}}^{(m)} = \widetilde{\mathbf{U}}_{\widehat{\mathbf{H}}}^{(m)} \widehat{\mathbf{\Lambda}}_{\widehat{\mathbf{H}}}^{(m)} (\widetilde{\mathbf{U}}_{\widehat{\mathbf{H}}}^{(m)})^T, m \in [M]$

**4:** Find  $\widehat{\mathbf{V}}^{(m)} = \widetilde{\mathbf{U}}_{\widehat{\mathbf{H}}}^{(m)}(:, 1 : K_m) = \Pi_{K_m}(\widetilde{\mathbf{U}}_{\widehat{\mathbf{H}}}^{(m)}), m \in [M]$

---

diagonal matrices  $\text{diag}(\widehat{\mathbf{d}}^{(l)})$  with vectors  $\widehat{\mathbf{d}}^{(l)}$  on the diagonals. We construct a tensor  $\widehat{\mathcal{G}} \in \mathbb{R}^{n \times n \times L}$  with layers  $\widehat{\mathbf{G}}^{(l)} = \widehat{\mathcal{G}}(:, :, l)$  of the form

$$\widehat{\mathbf{G}}^{(l)} = (\mathbf{A}^{(l)})^2 - \text{diag}(\widehat{\mathbf{d}}^{(l)}), \quad l \in [L] \quad (3.15)$$

Subsequently, we combine layers of the same types, obtaining tensor  $\widehat{\mathcal{H}} \in \mathbb{R}^{n \times n \times M}$

$$\widehat{\mathcal{H}} = \widehat{\mathcal{G}} \times_3 \widehat{\mathbf{W}}^T, \quad (3.16)$$

where  $\widehat{\mathbf{W}}$  is defined in (3.14). After that,  $\mathbf{V}^{(m)}, m \in [M]$ , can be estimated using the SVD. The procedure is described in Algorithm 2.

**Remark 3. Estimating invariant subspaces by averaging adjacency matrices.** If one knew that all matrices  $\mathbf{Q}^{(l)}, l \in [L]$ , in (2.1) have only positive eigenvalues, then estimation of invariant subspaces  $\mathbf{V}^{(m)}$  could have been done by averaging adjacency matrices of the graphs, since

$$\sum_{c(l)=m} \mathbf{P}^{(l)} = \mathbf{V}^{(m)} \left( \sum_{c(l)=m} \mathbf{Q}^{(l)} \right) (\mathbf{V}^{(m)})^T, \quad m \in [M]$$

Indeed, the accuracy of spectral clustering relies on the relationship between the ratio of the largest

and the smallest nonzero eigenvalues. The largest eigenvalues of matrices  $\mathbf{P}^{(l)}$  are always positive due to the Perron-Frobenius theorem (see, e.g., Rao and Rao (1998)) and, hence, add up. However, the same may not be true for the smallest nonzero eigenvalues that can be positive or negative, so that their sum may not be large enough. In this situation, in the case of one-group ( $M = 1$ ) SBM-equipped multilayer network, simulation studies in Paul and Chen (2020) show that averaging of the adjacency matrices may not lead to improved precision of community detection in groups of layers. Furthermore, we studied averaging of the adjacency matrices in the DIMPLE model under the assumption that all eigenvalues of matrices  $\mathbf{P}^{(l)}$  are nonnegative. However, even in the presence of this assumption, averaging of adjacency matrices does not substantially improve the accuracy in comparison with the bias-adjusted spectral clustering in Algorithm 2, while performing significantly worse when this assumption does not hold. For this reason, we shall avoid presentation of this algorithm in our exposition.

### 3.3 Within-layer clustering in the DIMPLE and DIMPLE-DECOR multiplex network

---

**Algorithm 3:** The within-layer clustering

---

**Input:** Adjacency tensor  $\mathcal{A} \in \{0, 1\}^{n \times n \times L}$ ; number of groups of layers  $M$ ; number of communities  $K_m$ ,  $m \in [M]$ ; estimated clustering matrix  $\hat{\mathbf{C}} \in \mathcal{M}_{L, M}$ ; parameter  $\epsilon$

**Output:** Estimated community assignments  $\hat{\mathbf{Z}}^{(m)} \in \mathcal{M}_{n, K_m}$ ,  $m \in [M]$

**Steps:**

**1:** Construct tensor  $\hat{\mathcal{G}}$  with layers  $\hat{\mathbf{G}}^{(l)}$  given by (3.15),  $l \in [L]$

**2:** Construct tensor  $\hat{\mathcal{H}}$  using formula (3.16)

**3:** Construct the SVDs of layers  $\hat{\mathbf{H}}^{(m)} = \tilde{\mathbf{U}}_{\hat{\mathbf{H}}}^{(m)} \hat{\mathbf{\Lambda}}_{\hat{\mathbf{H}}}^{(m)} (\tilde{\mathbf{U}}_{\hat{\mathbf{H}}}^{(m)})^T$ ,  $m \in [M]$

**4:** Find  $\hat{\mathbf{V}}^{(m)} = \tilde{\mathbf{U}}_{\hat{\mathbf{H}}}^{(m)}(:, 1 : K_m) = \Pi_{K_m}(\tilde{\mathbf{U}}_{\hat{\mathbf{H}}}^{(m)})$ ,  $m \in [M]$

**5:** Cluster rows of  $\hat{\mathbf{V}}^{(m)}$  into  $K_m$  clusters using  $(1 + \epsilon)$ -approximate  $K$ -means clustering.

Obtain clustering matrices  $\hat{\mathbf{Z}}^{(m)}$ ,  $m \in [M]$

---

After the matrices  $\mathbf{V}^{(m)}$  have been estimated, one can find clustering matrices  $\mathbf{Z}^{(m)}$  in (2.2) and (2.3) by approximate  $K$ -means clustering. Indeed, up to a rotation,  $\mathbf{V}^{(m)}$  is equal to  $\mathbf{U}_z^{(m)} =$

$\mathbf{Z}^{(m)}(\mathbf{D}_z^{(m)})^{-1/2}$  for DIMPLE and  $\mathbf{U}_z^{(m)} = \mathbf{\Theta}^{(m)}\mathbf{Z}^{(m)}(\mathbf{\Theta}_z^{(m)})^{-1/2}$  for DIMPLE-DECOR, where  $\mathbf{Z}^{(m)}$  is the clustering matrix of the layer  $m$ . Hence, there are only  $K_m$  distinct rows in the matrix  $\mathbf{V}^{(m)}$ , and clustering assignment can be obtain using Algorithm 3.

**Remark 4.  $k$ -means versus  $k$ -medians clustering.** *In Algorithm 3, one can replace the  $(1 + \epsilon)$ -approximate  $k$ -means clustering by  $(1 + \epsilon)$ -approximate  $k$ -medians clustering. While the former procedure is simpler, the latter is more common for networks obeying the DCBM. In the simulation section we shall provide the comparison between the two methods for DIMPLE-DECOR.*

## CHAPTER 4: THEORETICAL ANALYSIS

### 4.1 Definitions

In this section, we study the between-layer clustering error rates of the Algorithm 1, the error of estimation of invariant subspaces for the DIMPLE-GDPG model of Algorithm 2, and the within-layer clustering error rates of Algorithm 3. Since the clustering is unique only up to a permutation of cluster labels, denote the set of  $K$ -dimensional permutation functions of  $[K]$  by  $\aleph(K)$  and the set of  $K \times K$  permutation matrices by  $\mathfrak{F}(K)$ . The misclassification error rate of the between-layer clustering is then given by

$$R_{BL} = (2L)^{-1} \min_{\mathcal{P} \in \mathfrak{F}(M)} \|\widehat{\mathbf{C}} - \mathbf{C} \mathcal{P}\|_F^2. \quad (4.1)$$

Similarly, the local community detection error in the layer of type  $m$  is

$$R_{WL}(m) = (2n)^{-1} \min_{\mathcal{P}_m \in \mathfrak{F}(K_m)} \|\widehat{\mathbf{Z}}^{(m)} - \mathbf{Z}^{(m)} \mathcal{P}_m\|_F^2, \quad m \in [M]. \quad (4.2)$$

Note that, since the numbering of layers is defined also up to a permutation, the errors  $R_{WL}(1), \dots, R_{WL}(M)$  should be minimized over the set of permutations  $\aleph(M)$ . The average error rate of the within-layer clustering is then given by

$$R_{WL} = \frac{1}{M} \min_{\aleph(M)} \sum_{m=1}^M R_{WL}(m) = \frac{1}{2Mn} \min_{\aleph(M)} \sum_{m=1}^M \left[ \min_{\mathcal{P}_m \in \mathfrak{F}(K_m)} \|\widehat{\mathbf{Z}}^{(m)} - \mathbf{Z}^{(m)} \mathcal{P}_m\|_F^2 \right] \quad (4.3)$$

We shall measure the differences between the true and the estimated subspace bases matrices  $\mathbf{V}^{(m)}$  and  $\widehat{\mathbf{V}}^{(m)}$  using the average  $\sin \Theta$  distances defined in (2.5). Here, again we need to seek the

minimum over permutations of labels. We measure the errors as  $R_{S,max}$  and  $R_{S,ave}$  where

$$R_{S,max} = \min_{\mathfrak{N}(M)} \max_{m \in [M]} \left\| \sin \Theta \left( \mathbf{V}^{(m)}, \widehat{\mathbf{V}}^{(\mathfrak{N}(m))} \right) \right\|_F \quad (4.4)$$

$$R_{S,ave} = \frac{1}{M} \min_{\mathfrak{N}(M)} \sum_{m=1}^M \left\| \sin \Theta \left( \mathbf{V}^{(m)}, \widehat{\mathbf{V}}^{(\mathfrak{N}(m))} \right) \right\|_F^2 \quad (4.5)$$

## 4.2 Assumptions

In order the layers are identifiable, we assume that matrices  $\mathbf{V}^{(m)}$  in (2.1) or  $\mathbf{Z}^{(m)}$  in (2.2) and (2.3) correspond to different linear subspaces for different values of  $m$ . Furthermore, the performances of Algorithms 2 and 3 depend on the success of the between-layer clustering in Algorithm 1, which, in turn, relies on the fact that matrices  $\mathbf{V}^{(m)}(\mathbf{V}^{(m)})^T$  in (2.1) or  $\mathbf{Z}^{(m)}(\mathbf{Z}^{(m)})^T$  in (2.2) and (2.3),  $m \in [M]$ , are not too similar to each other for different values of  $m$ .

For the between layer clustering errors and the accuracy of the subspaces recovery, we develop our theory for the general case of the DIMPLE-GDPG model (2.1). Subsequently, we derive the within-layer clustering errors for the DIMPLE model (2.2). Denote

$$\bar{K} = \frac{1}{M} \sum_{m=1}^M K_m, \quad K = \max_{m \in [M]} K_m \quad (4.6)$$

Consider matrix  $\bar{\mathbf{Z}} \in \mathbb{R}^{n \times M\bar{K}}$ , which is obtained as horizontal concatenation of matrices  $\mathbf{V}^{(m)} \in \mathbb{R}^{n \times K_m}$ ,  $m \in [M]$ . Let the SVD of  $\bar{\mathbf{Z}}$  be

$$\bar{\mathbf{Z}} = [\mathbf{V}^{(1)} | \dots | \mathbf{V}^{(M)}] = \bar{\mathbf{U}} \bar{\mathbf{D}} \bar{\mathbf{V}}^T, \quad \bar{\mathbf{U}} \in \mathcal{O}_{n,r}, \bar{\mathbf{V}} \in \mathcal{O}_{M\bar{K},r}, \quad r \geq M + 1 \quad (4.7)$$

Here,  $r$  is the rank of  $\bar{\mathbf{Z}}$ , and  $\bar{\mathbf{D}}$  is an  $r$ -dimensional diagonal matrix. In the case of the DIMPLE



model (2.2) and the DIMPLE-DECOR model (2.3), one has  $\bar{\mathbf{Z}} = [\mathbf{U}_z^{(1)} | \dots | \mathbf{U}_z^{(M)}]$ . Since matrices  $\mathbf{V}^{(m)}$  represent different subspaces, one has  $M + 1 \leq r < n$ .

We impose the following assumptions.

**A1.** Clusters of layers are balanced, so that there exist absolute positive constants  $C_K$ ,  $\underline{c}$  and  $\bar{c}$  such that

$$C_K K \leq K_m \leq K, \quad \underline{c}L/M \leq L_m \leq \bar{c}L/M, \quad m \in [M] \quad (4.8)$$

where  $L_m$  is the number of networks in the layer of type  $m$ . In the case of the DIMPLE model (2.2) and the DIMPLE-DECOR model (2.3), local communities are balanced, so that

$$\underline{c}n/K \leq n_{k,m} \leq \bar{c}n/K, \quad k \in [K_m], m \in [M]$$

where  $n_{k,m}$  is the number of nodes in the  $k$ -th community in the layer of type  $m$ .

**A2.** For some absolute constant  $\kappa_0$ , one has  $\sigma_1(\bar{\mathbf{D}}) \leq \kappa_0 \sigma_r(\bar{\mathbf{D}})$  in (4.7).

**A3.** The layers  $\mathbf{P}^{(l)}$  of the probability tensor  $\mathcal{P}$  in (2.1) are such that, for some absolute constant  $C_\rho$

$$\mathbf{P}^{(l)} = \rho_{n,l} \mathbf{P}_0^{(l)}, \quad \|\mathbf{P}_0^{(l)}\|_\infty = 1, \quad \min_{l \in [L]} \rho_{n,l} \geq C_\rho n^{-1} \log n, \quad l \in [L] \quad (4.9)$$

In the case of the DIMPLE model (2.2) and the DIMPLE-DECOR model (2.3), (4.9) reduces to  $\mathbf{B}^{(l)} = \rho_{n,l} \mathbf{B}_0^{(l)}, \|\mathbf{B}_0^{(l)}\|_\infty = 1$ .

**A4.** Matrices  $\mathbf{Q}^{(l)}$  in (2.1) are such that, for some absolute constant  $C_\lambda \in (0, 1)$ , one has

$$\min_{l=1, \dots, L} [\sigma_{K_m}(\mathbf{Q}^{(l)}) / \sigma_1(\mathbf{Q}^{(l)})] \geq C_\lambda, \quad m = c(l). \quad (4.10)$$

In the case of the DIMPLE model and the DIMPLE-DECOR model, (4.10) appears as

$\min_{l \in [L]} [\sigma_{K_m}(\mathbf{B}_0^{(l)}) / \sigma_1(\mathbf{B}_0^{(l)})] \geq C_\lambda$  for  $m = c(l)$ .

**A5.** There exist absolute constants  $\underline{c}_\rho$  and  $\bar{c}_\rho$  such that

$$\underline{c}_\rho \rho_n \leq \rho_{n,l} \leq \bar{c}_\rho \rho_n \quad \text{with} \quad \rho_n = L^{-1} \sum_{l=1}^L \rho_{n,l} \quad (4.11)$$

**A6.** For some absolute constant  $C_{0,P}$  one has

$$\|\mathbf{P}_0^{(l)}\|_F^2 \geq C_{0,P}^2 K^{-1} n^2 \quad (4.12)$$

Assumptions above are very common and are present in many other network papers. Specifically, Assumption **A1** is identical to Assumptions **A3** and **A4** in Jing et al. (2021), or Assumption **A3** in Fan et al. (2021). Assumption **A2** is identical to Assumption **A2** in Jing et al. (2021). Assumption **A3** is present in majority of papers that study community detection in individual networks (see, e.g. Lei and Rinaldo (2015)). It is required here since we rely on similarity of the sets of eigenvectors in the groups of similar layers, and, hence, need the sample eigenvectors to converge to the true ones. Assumption **A4** is equivalent to Assumption **A1** in Jing et al. (2021), Assumption **A4** in Fan et al. (2021) and an equivalent assumption in Zheng and Tang (2022). Finally, Assumption **A5** requires that the sparsity factors are of approximately the same order of magnitude. The latter guarantees that the discrepancies between the true and the sample-based eigenvectors are similar across all layers of the network. Hypothetically, Assumption **A5** can be removed, and one can trace the impact of different scales  $\rho_{n,l}$  on the clustering errors. This, however, will make clustering error bounds very complicated, so we leave this case for future investigation.

Assumption **A6** postulates that matrices  $\mathbf{P}_0^{(l)}$  have enough of non-negligible entries. Assumption **A6** naturally holds in the case of the balanced DIMPLE model (2.2). Indeed, in this case,

$\|\mathbf{P}_0^{(l)}\|_F^2 \geq \underline{c}^2 n^2 K^{-2} \|\mathbf{B}_0^{(l)}\|_F^2$ . Due to Assumption **A3**, one has  $1 = \|\mathbf{B}_0^{(l)}\|_\infty \leq \|\mathbf{B}_0^{(l)}\|$  and, therefore, by Assumptions **A1** and **A4**

$$\|\mathbf{B}_0^{(l)}\|_F^2 \geq K_m \sigma_{K_m}^2(\mathbf{B}_0^{(l)}) \geq C_\lambda^2 K_m \|\mathbf{B}_0^{(l)}\|^2 \geq C_\lambda^2 C_K K,$$

which implies  $\|\mathbf{P}_0^{(l)}\|_F^2 \geq Cn^2/K$ .

**Remark 5. Extra assumptions for DIMPLE-DECOR** For DIMPLE-DECOR, assumptions for the degree corrected parameters are required. Specifically, there exist absolute constants  $\underline{c}_\theta$  and  $\bar{c}_\theta$  such that for the elements  $\theta_i^{(m)}$  of  $\Theta^{(m)}$  one has

$$\underline{c}_\theta \leq \theta_i^{(m)} \leq \bar{c}_\theta, \quad i = 1, \dots, n, \quad m = 1, \dots, M. \quad (4.13)$$

In addition, it follows from **B1** that  $\underline{C}n/K \leq \gamma_k^{(m)} \leq \bar{C}n/K$  with  $\bar{C} = \bar{c}(\bar{c}_\theta)$  and  $\underline{C} = \underline{c}(\underline{c}_\theta)$ .

Note that Assumption **A3** implies that  $n \rightarrow \infty$ . In what follows, we assume that  $L$  can grow at most polynomially with respect to  $n$ , specifically, that for some constant  $\tau_0$

$$L \leq n^{\tau_0}, \quad 0 < \tau_0 < \infty \quad (4.14)$$

Condition (4.14) is hardly restrictive. Indeed, Jing et al. (2021) assume that  $L \leq n$ , so, in their paper, (4.14) holds with  $\tau_0 = 1$ . We allow any polynomial growth of  $L$  with respect to  $n$ .

### 4.3 The between-layer clustering error

Evaluation of the between-layer clustering error relies on the Tucker decomposition of the tensor with layers  $\mathbf{U}_{P,l}(\mathbf{U}_{P,l})^T$ ,  $l \in [L]$ . Consider tensor  $\mathfrak{S} \in \mathbb{R}^{n \times n \times L}$  with layers

$$\mathfrak{S}(:, :, l) = \mathbf{U}_{P,l}(\mathbf{U}_{P,l})^T = \mathbf{V}^{(m)}(\mathbf{V}^{(m)})^T, \quad m = c(l), \quad l \in [L] \quad (4.15)$$

and its clustered version  $\mathcal{U} \in \mathbb{R}^{n \times n \times M}$  of the form

$$\mathcal{U} = \mathfrak{S} \times_3 [\mathbf{C}(\mathbf{D}_c)^{-1}]^T, \quad (4.16)$$

where  $\mathbf{D}_c$  is defined in (3.9). Here, tensor  $\mathcal{U}$  has layers identical to the set of distinct layers of tensor  $\mathfrak{S}$ , so that  $\mathcal{U}(:, :, m) = \mathbf{V}^{(m)}(\mathbf{V}^{(m)})^T$ ,  $m \in [M]$ .

Recall that, according to (4.15) and (4.16), one has  $\mathfrak{S} = \mathcal{G} \times_3 \mathbf{C}$ . Then, using matrix  $\bar{\mathbf{Z}}$  in (4.7), one can rewrite  $\mathfrak{S}$  as  $\mathfrak{S} = \mathcal{B} \times_1 \bar{\mathbf{Z}} \times_2 \bar{\mathbf{Z}} \times_3 \mathbf{C}$ , where  $\mathcal{B} \in \mathbb{R}^{\bar{K}M \times \bar{K}M \times M}$  is the core tensor with layers

$$\mathcal{B}(:, :, m) = \text{diag}(\mathbf{0}_{K_1}, \dots, \mathbf{0}_{K_{m-1}}, \mathbf{I}_{K_m}, \mathbf{0}_{K_{m+1}}, \dots, \mathbf{0}_{K_M}) \in \{0, 1\}^{\bar{K}M \times \bar{K}M}$$

Using the SVD in (4.7) and the definition of  $\mathbf{W}$  in (3.9), we obtain

$$\mathfrak{S} = \mathcal{F} \times_1 \bar{\mathbf{U}} \times_2 \bar{\mathbf{U}} \times_3 \mathbf{W}, \quad \mathcal{F} = \bar{\mathcal{R}} \times_1 \bar{\mathbf{D}} \times_2 \bar{\mathbf{D}} \times_3 \mathbf{D}_c^{1/2}, \quad \bar{\mathcal{R}} = \mathcal{B} \times_1 \bar{\mathbf{V}}^T \times_2 \bar{\mathbf{V}}^T, \quad (4.17)$$

where  $\mathcal{F}, \bar{\mathcal{R}} \in \mathbb{R}^{r \times r \times M}$ . Now, in order to use representation (4.17) for analyzing matrix  $\Theta$  in (3.10), note that  $\Theta$  is the transpose of mode 3 matricization of  $\mathfrak{S}$ , i.e.,  $\Theta = \mathfrak{S}_{(3)}^T$ . Using Proposition 1 of Kolda and Bader (2009), obtain

$$\Theta = (\bar{\mathbf{U}} \otimes \bar{\mathbf{U}}) \mathbf{F} \mathbf{W}^T, \quad \mathbf{F} = \mathcal{F}_{(3)}^T \in \mathbb{R}^{r^2 \times M}. \quad (4.18)$$

Here, by (3.9) and (4.7),  $\mathbf{W} = \mathbf{C}\mathbf{D}\bar{c}^{-1/2} \in \mathcal{O}_{L,M}$  and  $\bar{\mathbf{U}} \in \mathcal{O}_{n,r}$ . The following statement explores the structure of matrix  $\mathbf{F}$  in (4.18).

**Lemma 1.** *Matrix  $\mathbf{F}$  can be presented as  $\mathbf{F} = (\bar{\mathbf{D}} \otimes \bar{\mathbf{D}})\bar{\mathbf{R}}\mathbf{D}_c^{1/2}$  where  $\bar{\mathbf{R}} = (\bar{\mathbf{V}} \otimes \bar{\mathbf{V}})^T \mathbf{R}$  and  $\mathbf{R} = \mathcal{B}_{(3)}^T$ . Here,  $\text{rank}(\mathbf{F}) = M$ , and, under Assumptions **A1–A6**, one has*

$$\sigma_{\min}^2(\mathbf{F}) = \sigma_M^2(\mathbf{F}) \geq \frac{\underline{c}}{\bar{c}\kappa_0^4 M} \|\mathbf{F}\|_F^2 \geq \frac{\underline{c}C_K \bar{K} L}{\bar{c}\kappa_0^4 M} \quad (4.19)$$

Let the SVD of  $\mathbf{F}$  be of the form  $\mathbf{F} = \mathbf{U}_F \mathbf{\Lambda}_F \mathbf{V}_F$ , where  $\mathbf{U}_F \in \mathcal{O}_{r^2,M}$  and  $\mathbf{V}_F \in \mathcal{O}_M$ . Then, the SVD of  $\Theta$  in (4.18) can be written as

$$\Theta = \mathbf{V}\mathbf{\Lambda}\mathbf{W}, \quad \mathbf{V} = (\bar{\mathbf{U}} \otimes \bar{\mathbf{U}})\mathbf{U}_F \in \mathcal{O}_{n^2,M}, \quad \mathbf{W} = \mathbf{W}\mathbf{V}_F \in \mathcal{O}_{L,M}, \quad \mathbf{\Lambda} = \mathbf{\Lambda}_F \quad (4.20)$$

Representation (4.20) allows one to bound above the between-layer clustering error.

**Theorem 1.** *Let Assumptions **A1–A6** and (4.14) hold. Then, for any  $\tau > \tau_0$ , there exists a constant  $C$  that depends only on  $\tau$ ,  $C_K$ ,  $\kappa_0$ ,  $\bar{c}$ ,  $\underline{c}$ ,  $\bar{c}_\rho$  and  $\underline{c}_\rho$  in Assumptions **A1–A6**, such that the between-layer clustering error, defined in (4.1), satisfies*

$$\mathbb{P} \left\{ R_{BL} \leq \frac{CK^2}{n\rho_n} \right\} \geq 1 - Ln^{-\tau} \geq 1 - n^{-(\tau-\tau_0)} \quad (4.21)$$

#### 4.4 The subspace fitting errors in groups of layers in the DIMPLE-GDPG model

In this section, we provide upper bounds for the divergence between matrices  $\mathbf{V}^{(m)}$  and their estimators  $\widehat{\mathbf{V}}^{(m)}$ ,  $m \in [M]$ . We measure their discrepancies by  $R_{S,max}$  and  $R_{S,ave}$  defined in, respectively, (4.4) and (4.5).

**Theorem 2.** *Let Assumptions **A1–A6** and (4.14) hold, and matrices  $\widehat{\mathbf{V}}^{(m)}$ ,  $m \in [M]$ , be obtained using Algorithm 2. Let*

$$\lim_{n \rightarrow \infty} \frac{MK^2}{n\rho_n} = 0. \quad (4.22)$$

*Then, for any  $\tau > 0$ , there exists a constant  $C$  that depends only on constants in Assumptions **A1–A6**, and a constant  $C_{\tau,\epsilon}$  which depends only on  $\tau$  and  $\epsilon$ , such that the subspace estimation errors  $R_{S,max}$  and  $R_{S,ave}$  defined in, respectively, (4.4) and (4.5), satisfy*

$$\mathbb{P} \left\{ R_{S,max} \leq C \frac{K^{5/2} M}{\sqrt{n} \rho_n} \left( 1 + \frac{\sqrt{\log n}}{\sqrt{L} M} + \frac{K \sqrt{\log n}}{\sqrt{n} \rho_n} \right) \right\} \geq 1 - C_{\tau,\epsilon} L n^{1-\tau} \quad (4.23)$$

$$\mathbb{P} \left\{ R_{S,ave} \leq C \frac{K^5 M}{n \rho_n} \left( 1 + \frac{\log n}{L} + \frac{K^2 \log n}{n \rho_n} \right) \right\} \geq 1 - C_{\tau,\epsilon} L n^{1-\tau} \quad (4.24)$$

Note that, due to condition (4.14), if  $\tau > \tau_0 + 1$ , then the upper bounds in (4.23) and (4.24) hold with probability at least  $1 - \tilde{C}_{\tau,\epsilon} n^{-(\tau-\tau_0-1)}$ .

**Remark 6. Subspace estimation error for a homogeneous multilayer GDPG.** Consider the case when  $M = 1$ , so that all layers of the network can be embedded into the same invariant subspace. Since the dominant terms in (4.23) and (4.24) are due to clustering of layers, it follows from the proof of Theorem 2 in Section A.2, where  $\|\widehat{\mathbf{H}}^{(m)} - \mathbf{H}^{(m)}\|$  is replaced with  $\Delta_1^{(m)}$ ,  $m = M = 1$ , that

$$\left\| \sin \Theta \left( \widehat{\mathbf{V}}, \mathbf{V} \right) \right\|_F \leq C \frac{K^{5/2} \left[ \rho_n^{3/2} n^{3/2} \sqrt{\log n} + \rho_n^2 n \sqrt{L} \right]}{n^2 \rho_n^2 \sqrt{L}} = C K^{5/2} \left[ \frac{\sqrt{\log n}}{\sqrt{n} \rho_n L} + \frac{1}{n} \right]$$

Consequently, one has much smaller subspaces estimation error

$$\mathbb{P} \left\{ R_{S,max} \leq C K^{5/2} \left[ \frac{\sqrt{\log n}}{\sqrt{n} \rho_n L} + \frac{1}{n} \right] \right\} \geq 1 - \tilde{C}_{\tau,\epsilon} L n^{1-\tau} \quad (4.25)$$

#### 4.5 The within-layer clustering error

Since the within-layer clustering for each group of layers is carried out by clustering rows of the matrices  $\widehat{\mathbf{V}}^{(m)}$ , the upper bound for  $R_{WL}$  defined in (4.3) can be easily obtained as a by-product of Theorem 2. Specifically, the following statement holds.

**Corollary 1.** *Let assumptions of Theorem 2 hold. Then, for any  $\tau > 0$ , there exists a constant  $C$  that depends only on constants in Assumptions **A1–A6**, and  $C_{\tau,\epsilon}$  which depends only on  $\tau$  and  $\epsilon$ , such that for the DIMPLE model*

$$\mathbb{P} \left\{ R_{WL} \leq C \frac{K^4 M}{n \rho_n} \left( 1 + \frac{\log n}{L} + \frac{K^2 \log n}{n \rho_n} \right) \right\} \geq 1 - C_{\tau,\epsilon} L n^{1-\tau} \quad (4.26)$$

*For the DIMPLE-DECOR model,*

$$\mathbb{P} \left\{ R_{WL} \leq C \frac{K^4 M}{n \rho_n} \left( \frac{\log n}{L} + K^2 \right) \right\} \geq 1 - C_{\tau,\epsilon} L n^{1-\tau} \quad (4.27)$$

Note that in the case of  $M = 1$  for DIMPLE, Corollary 1 yields, with high probability, that

$$R_{WL} \leq C K^4 \left[ \frac{\log n}{n \rho_n L} + \frac{1}{n^2} \right] \quad (4.28)$$

## CHAPTER 5: SIMULATION STUDY

### 5.1 Simulations settings

In order to study performances of our methodology for various combinations of parameters, we carry out a limited simulation study with models generated from DIMPLE, DIMPLE-DECOR and DIMPLE-GDPG. We use Algorithm 1 for finding the groups of layers and Algorithms 2 and 3, respectively, for recovering the ambient subspaces in the DIMPLE-GDPG setting, and for finding communities in groups of layers for the DIMPLE and DIMPLE-DECOR models.

To obtain a multilayer network that complies with our assumptions in Section 4.2, we fix  $n$ ,  $L$ ,  $M$ ,  $K$ , the sparsity parameters  $c$  and  $d$ , the assortativity parameter  $w$ , the Dirichlet parameter  $\alpha$  used for generating a DIMPLE-GDPG network, and the degree parameters  $t_1$  and  $t_2$  in each degree corrected matrix for generating a DIMPLE-DECOR network. We use the multinomial distribution with equal probabilities  $1/M$  to assign group memberships to individual networks.

In the case of the DIMPLE model, we generate  $K$  communities in each of the groups of layers using the multinomial distribution with equal probabilities  $1/K$ . In this manner, we obtain community assignment matrices  $\mathbf{Z}^{(m)}$ ,  $m \in [M]$ , in each layer  $l$  with  $c(l) = m$ , where  $c : [L] \rightarrow [M]$  is the layer assignment function. Next, we generate the entries of  $\mathbf{B}^{(l)}$ ,  $l \in [L]$ , as uniform random numbers between  $c$  and  $d$ , and then multiply all the non-diagonal entries of those matrices by  $w$ . In this manner, if  $w < 1$  is small, then the network is strongly assortative, i.e., there is a higher probability for nodes in the same community to connect. If  $w > 1$  is large, then the network is disassortative, i.e., the probability of connection for nodes in different communities is higher than for nodes in the same community. Finally, since entries of matrices  $\mathbf{B}^{(l)}$  are generated at random, when  $w$  is close to one, the networks in all layers are neither assortative or disassortative. After the



community assignment matrices  $\mathbf{Z}^{(m)}$  and the block probability matrices  $\mathbf{B}^{(l)}$  have been obtained, we construct the probability tensor  $\mathcal{P}$  with layers  $\mathcal{P}(:, :, l) = \mathbf{Z}^{(m)} \mathbf{B}^{(l)} (\mathbf{Z}^{(m)})^T$ , where  $m = c(l)$ ,  $l \in [L]$ .

As for DIMPLE-DECOR, we also generate the degree parameters  $\Theta^{(m)}$  in each group of layers as uniform random numbers between  $t_1$  and  $t_2$ . And then with the same generating ways with DIMPLE, after  $\mathbf{Z}^{(m)}$  and  $\mathbf{B}^{(l)}$  have been obtained, we construct the probability tensor  $\mathcal{P}$  with layers  $\mathcal{P}(:, :, l) = \Theta^{(m)} \mathbf{Z}^{(m)} \mathbf{B}^{(l)} (\mathbf{Z}^{(m)})^T \Theta^{(m)}$ , where  $m = c(l)$ ,  $l = 1, \dots, L$ .

In the case of the DIMPLE-GDPG setting, we obtain matrices  $\mathbf{X}^{(m)} \in [0, 1]^{n \times K}$ ,  $m \in [M]$ , with independent rows, generated using the Dirichlet distribution with parameter  $\alpha$ . We obtain matrices  $\mathbf{B}^{(l)}$ , in exactly the same manner as in the case of the DIMPLE model and construct  $\mathcal{P}$  with layers  $\mathcal{P}(:, :, l) = \mathbf{X}^{(m)} \mathbf{B}^{(l)} (\mathbf{X}^{(m)})^T$ , where  $m = c(l)$ ,  $l \in [L]$ . In this case, the matrices  $\mathbf{V}^{(m)}$  are obtained from the SVD  $\mathbf{X}^{(m)} = \mathbf{V}^{(m)} \mathbf{\Lambda}_X^{(m)} \mathbf{W}_X^{(m)}$  of  $\mathbf{X}^{(m)}$ . Matrices  $\mathbf{Q}^{(l)}$  are defined as  $\mathbf{Q}^{(l)} = \mathbf{\Lambda}_X^{(m)} \mathbf{W}_X^{(m)} \mathbf{B}^{(l)} (\mathbf{W}_X^{(m)})^T \mathbf{\Lambda}_X^{(m)}$  in (2.1),  $l \in [L]$ .

After the probability tensor  $\mathcal{P}$  is generated, the layers  $\mathbf{A}^{(l)}$  of the adjacency tensor  $\mathcal{A}$  are obtained as symmetric matrices with zero diagonals and independent Bernoulli entries  $\mathbf{A}^{(l)}(i, j)$  for  $1 \leq i < j \leq n$ . Subsequently, we use Algorithm 1 for finding the groups of layers for all models, followed by Algorithm 2 for estimating matrices  $\mathbf{V}^{(m)}$  in the case of the DIMPLE-GDPG network, or Algorithm 3 for clustering nodes in each group of layers of the network into communities for the DIMPLE and DIMPLE-DECOR models.

In cases of DIMPLE-GDPG and DIMPLE, we have two sets of simulations, one with fixed  $L$  and varying  $n$ , another with the fixed  $n$  and varying  $L$ . In all simulations, we set  $M = 3$  and  $K_m = 3$  for  $m = 1, 2, 3$ , and study two sparsity scenarios,  $c = 0, d = 0.8$  or  $c = 0, d = 0.5$ , with four values of assortativity parameter  $w = 0.6, 0.8, 1.0$  and  $1.2$ . In all simulations, we set  $\alpha = 0.1$ . In the case of DIMPLE-DECOR, we set  $c = 0, d = 0.8$  and consider either fixed assortativity

parameter  $w = 0.5$ , or varying one:  $w = 0.5, 0.75, 1, 1.25$ . In addition, we look at four settings for the pairs of parameters  $t_1$  and  $t_2$ :

1.  $t_1$  varying,  $t_2$  fixed:  $t_1 = 0.2, 0.4, 0.6, 0.8$  and  $t_2 = 1.11$  (Figure 5.5 and Figure 5.10)
2.  $t_1$  fixed,  $t_2$  varying:  $t_1 = 0.2$  and  $t_2 = 0.65, 0.8, 0.95, 1.11$  (Figure 5.6 and Figure 5.11)
3. both  $t_1$  and  $t_2$  are varying:  $(t_1, t_2) = (0.2, 0.65); (0.4, 0.8); (0.6, 0.95); (0.8, 1.11)$  (Figure 5.7 and Figure 5.12), or  $(t_1, t_2) = (0.2, 1.11); (0.4, 0.95); (0.6, 0.8)$  (Figure 5.8 and Figure 5.13)
4. both  $t_1$  and  $t_2$  are fixed:  $t_1 = 0.2, t_2 = 1$  (Figure 5.9 and Figure 5.14)

We report the average between-layer clustering errors  $R_{BL}$  defined in (4.1), and also the average within-layer clustering error  $R_{WL}$  defined in (4.3) in the case of the DIMPLE setting and the average  $\sin \Theta$  distance  $R_{S,ave}$  defined in (4.5) between the true and the estimated subspaces in the case of the DIMPLE-GDPG network. We first present simulations results for the DIMPLE model and the DIMPLE-GDPG model, followed by the study of DIMPLE-DECOR.

## 5.2 Simulations Results

Simulations results for the DIMPLE and DIMPLE-GDPG models are summarized in Figures 5.1–5.2 and Figures 5.3–5.4, respectively. Note that, while the between-layer clustering errors (left panels in Figures 5.1–5.4), as well as the within-layer clustering errors (right panels in Figures 5.1–5.2) are between 0 and 1, the average errors of estimation of subspaces  $R_{S,ave}$  defined in (4.5) (right panels in Figures 5.3–5.4) lie between 0 and  $K$ , so they are on a different scale.

As it is expected, both estimation and clustering are harder when a network is more sparse, therefore, all errors are smaller when  $d = 0.8$  (top panels) than when  $d = 0.5$  (bottom). Figures 5.1–5.4

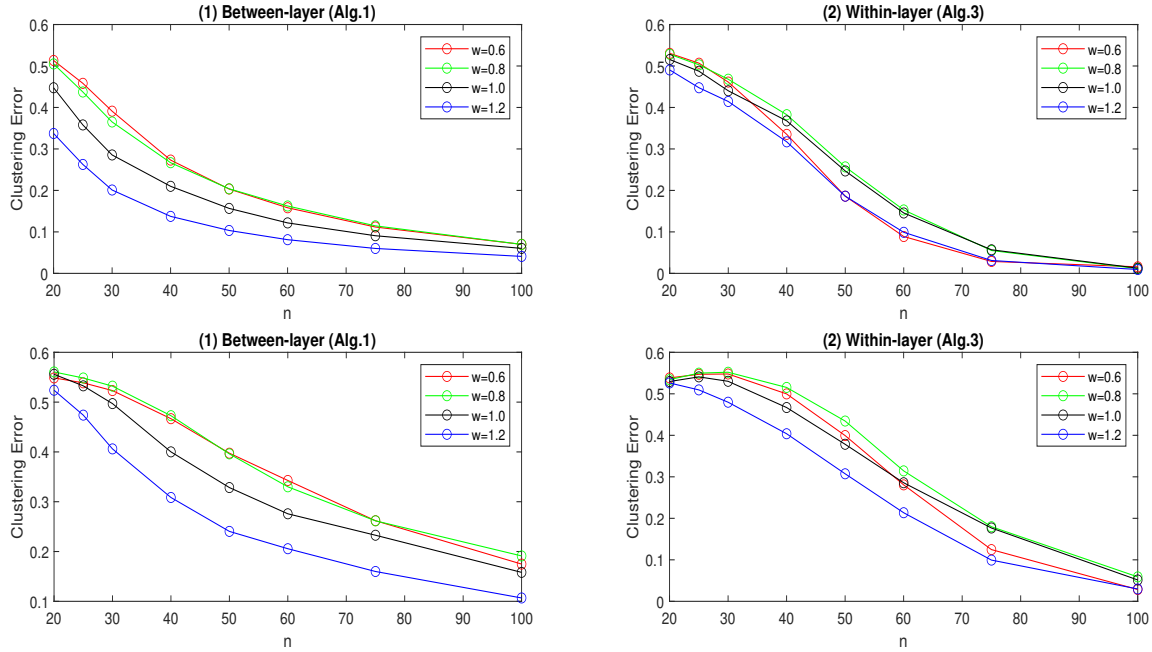


Figure 5.1: The between-layer clustering error rates of Algorithm 1 (left) and the within-layer error rates of Algorithms 3 (right), averaged over 500 simulation runs, for the DIMPLE model with  $c = 0, d = 0.8$  (top) and  $c = 0, d = 0.5$  (bottom),  $L = 50$  and  $n = 20, 25, 30, 40, 50, 60, 75, 100$ . The entries of  $\mathbf{B}^{(l)}$ ,  $l \in [L]$ , are generated as uniform random numbers between  $c$  and  $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by  $w$ .

show that the value of the assortativity parameter does not play a significant role in the between-layer clustering. Indeed, as the left panels in all figures show, the smallest between-layer clustering errors occur for  $w = 1.2$  followed by  $w = 1.0$ . The latter confirms that the difficulty of the between-layer clustering is predominantly controlled by the sparsity of the network. The results are somewhat different for the community detection errors and the subspace estimation errors in, respectively, the DIMPLE and the DIMPLE-GDPG models. Indeed, as the right panels in Figures 5.1–5.4 show, the smallest errors occur in the more assortative/disassortative models with  $w = 0.6$  and  $w = 1.2$ .

One can see from Figures 5.1 and 5.3 that, when  $n$  grows, all errors decrease. The influence of  $L$  on the error rates is more complex. As Theorem 1 implies, the between-layer clustering errors are

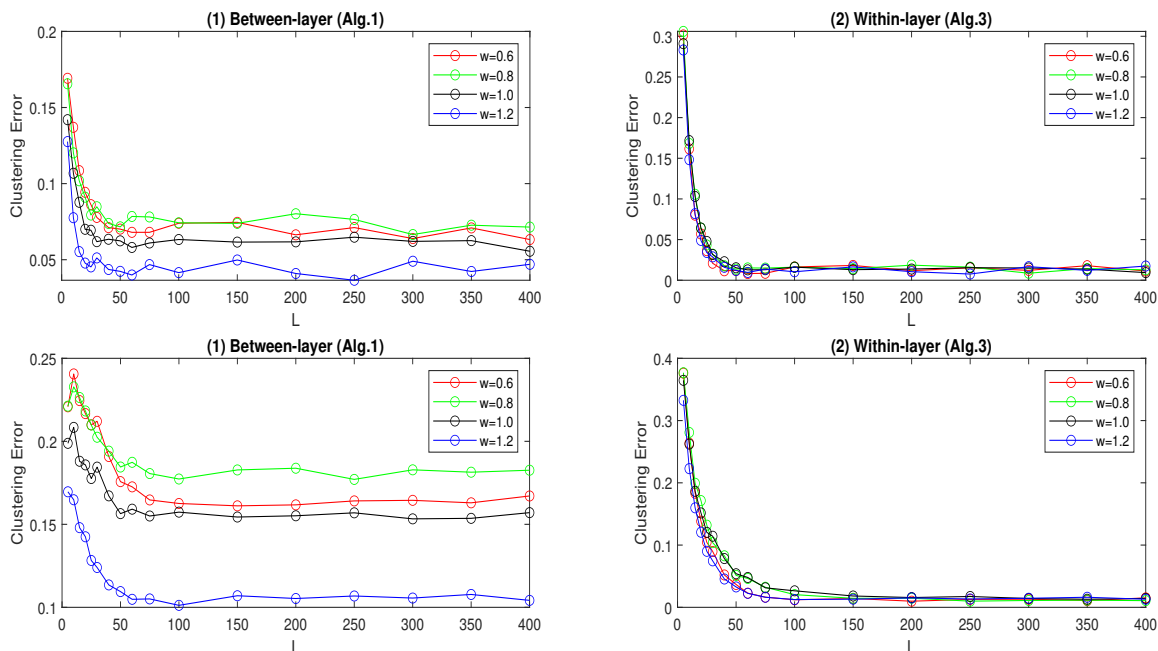


Figure 5.2: The between-layer clustering error rates of Algorithm 1 (left) and the within-layer error rates of Algorithms 3 (right), averaged over 500 simulation runs, for the DIMPLE model with  $c = 0, d = 0.8$  (top) and  $c = 0, d = 0.5$  (bottom),  $n = 100$  and  $L = 5, 10, 15, 20, 25, 30, 40, 50, 60, 75, 100, 150, 200, 250, 300, 350, 400$ . The entries of  $\mathbf{B}^{(l)}$ ,  $l \in [L]$ , are generated as uniform random numbers between  $c$  and  $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by  $w$ .

of the order  $(n\rho_n)^{-1}$  for fixed values of  $M$  and  $K$ . This agrees with the left panels in Figures 5.2 and 5.4 where curves exhibit constant behavior for when  $L$  grows (small fluctuations are just due to random errors). For the right panels in Figures 5.2 and 5.4 this, however, happens only when  $L$  is relatively large.

The explanation for such behavior lies in the fact that the between-layer clustering error (corresponding to the left panels in Figures 5.2 and 5.4) is of the order  $K^2 (n\rho_n)^{-1}$  and is independent of  $L$ . On the other hand, for fixed  $K$  and  $M$ , the errors  $R_{WL}$  and  $R_{S,ave}$  (corresponding to the right panels in, respectively, Figures 5.2 and 5.4) are of the order  $(n\rho_n)^{-1} + \log n (n\rho_n L)^{-1}$ . While  $L$  is small the second term is dominant but, as  $L$  grows, the first term becomes dominant and the

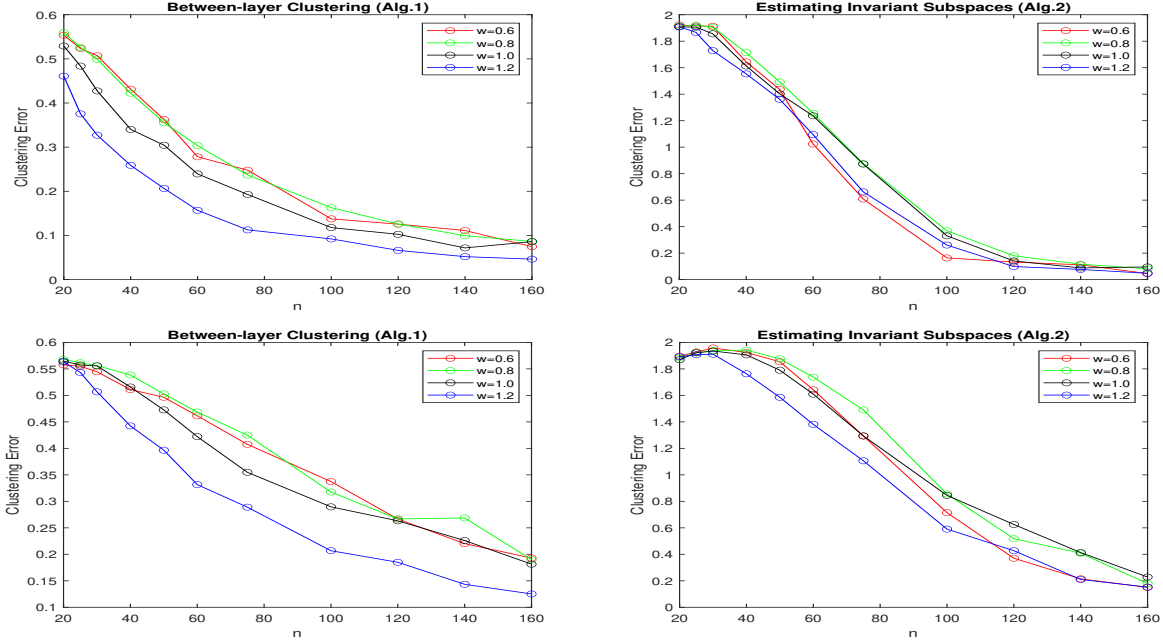


Figure 5.3: The between-layer clustering error rates of Algorithm 1 (left) and the  $\sin \Theta$  distances  $R_{S,ave}$  of Algorithms 2 (right), averaged over 100 simulation runs, for the DIMPLE-GDPG model with  $\alpha = 0.1$ ,  $c = 0$ ,  $d = 0.8$  (top) and  $c = 0$ ,  $d = 0.5$  (bottom),  $L = 50$  and  $n = 20, 25, 30, 40, 50, 60, 75, 100, 120, 140, 160$ . The entries of  $\mathbf{B}^{(l)}$ ,  $l \in [L]$ , are generated as uniform random numbers between  $c$  and  $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by  $w$ .

errors stop declining as  $L$  grows.

Simulation results for DIMPLE-DECOR are summarized in Figures 5.5–5.14. One can see from these figures that, when  $n$  and  $L$  grow while the other stays the same, clustering errors follow similar trends with DIMPLE and DIMPLE-GDPG.

The node weight parameters  $t_1$  and  $t_2$  have multiple effects on the error rates of both the between-layer and the within-layer clustering. In particular, the value of  $\rho_n$  is proportional to  $(t_1 + t_2)/2$ , the average of  $t_1$  and  $t_2$ . Therefore, when  $t_1$  becomes larger in Figures 5.5 and 5.10, or  $t_2$  becomes larger in Figures 5.6 and 5.11, both error rates decrease. Figures 5.7 and 5.12 exhibit the similar effect.

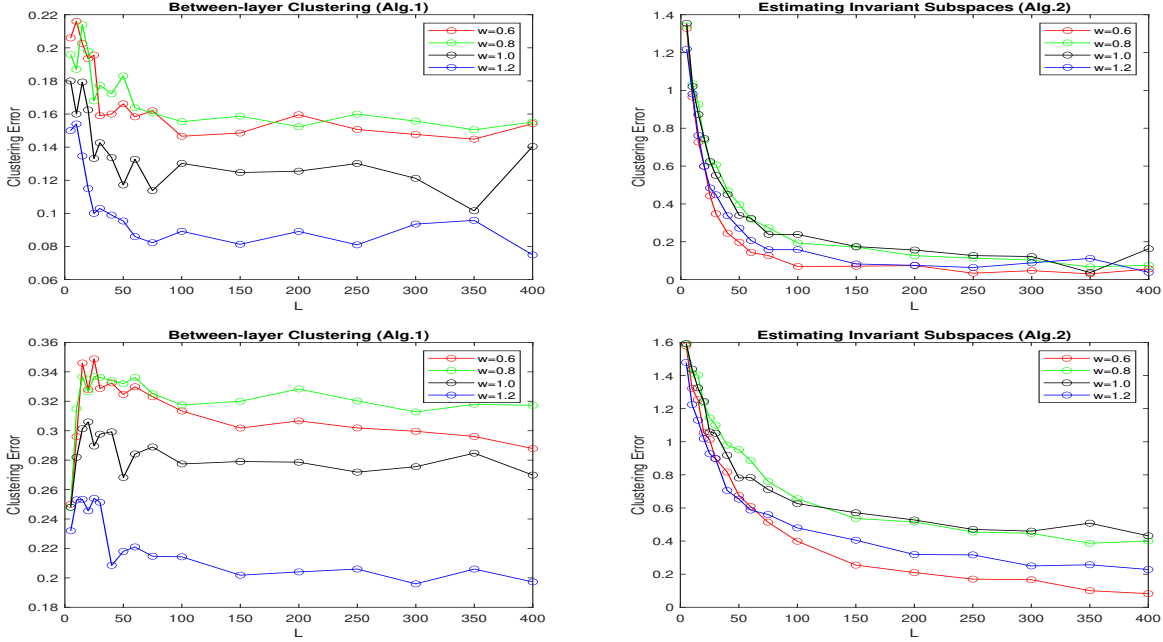


Figure 5.4: The between-layer clustering error rates of Algorithm 1 (left) and the  $\sin \Theta$  distances  $R_{S,ave}$  of Algorithms 2 (right), averaged over 100 simulation runs, for the DIMPLE-GDPG model with  $\alpha = 0.1$ ,  $c = 0, d = 0.8$  (top) and  $c = 0, d = 0.5$  (bottom),  $n = 100$  and  $L = 5, 10, 15, 20, 25, 30, 40, 50, 60, 75, 100, 150, 200, 250, 300, 350, 400$ . The entries of  $\mathbf{B}^{(l)}$ ,  $l \in [L]$ , are generated as uniform random numbers between  $c$  and  $d$ . All the non-diagonal entries of those matrices are subsequently multiplied by  $w$ .

The difference  $t_2 - t_1$  also influences the error rates. In the one-layer network, increase in the heterogeneity between nodes' degrees usually leads to the decline in clustering precision. However, in the case of the DIMPLE-DECOR, the dynamics is different. When  $t_2 - t_1$  becomes larger, the layers become more diverse, which is beneficial for the accuracy of Algorithm 1. Specifically, as Figures 5.8 and 5.13 demonstrate, when  $t_2 - t_1$  grow with the same average, the error rates of the between-layer clustering become smaller which, in turn, leads to the decrease of the within-layer clustering errors. For this reason, the best precision of the between and within-layer clustering occurs when the difference between  $t_2$  and  $t_1$  is the largest.

Figures 5.9 and 5.14 study the impact of assortativity parameter  $w$  and confirm that it significantly

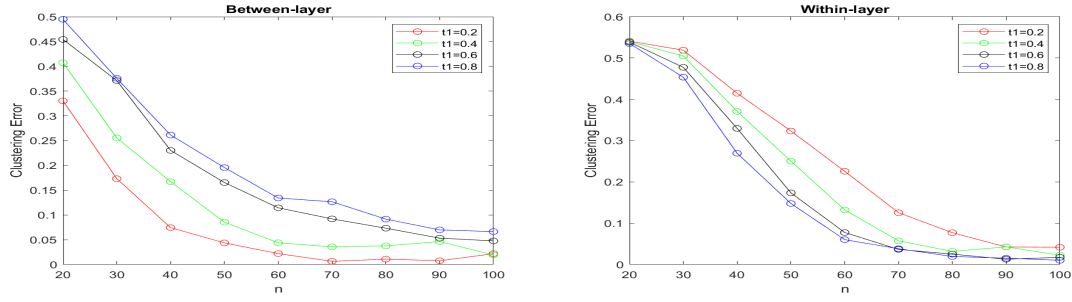


Figure 5.5: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $t_1 = 0.2, 0.4, 0.6, 0.8$ ,  $t_2 = 1.11$ ,  $L = 50$  and  $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs.

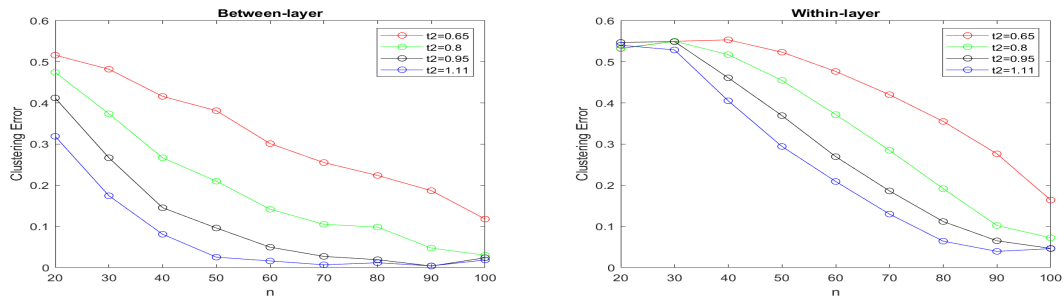


Figure 5.6: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $t_1 = 0.2$ ,  $t_2 = 0.65, 0.8, 0.95, 1.11$ ,  $L = 50$  and  $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs.

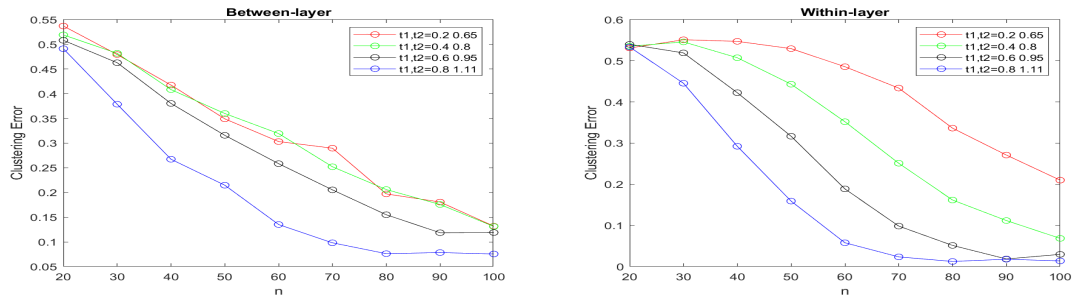


Figure 5.7: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $(t_1, t_2) = (0.2, 0.65); (0.4, 0.8); (0.6, 0.95); (0.8, 1.11)$ ,  $L = 50$  and  $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs.

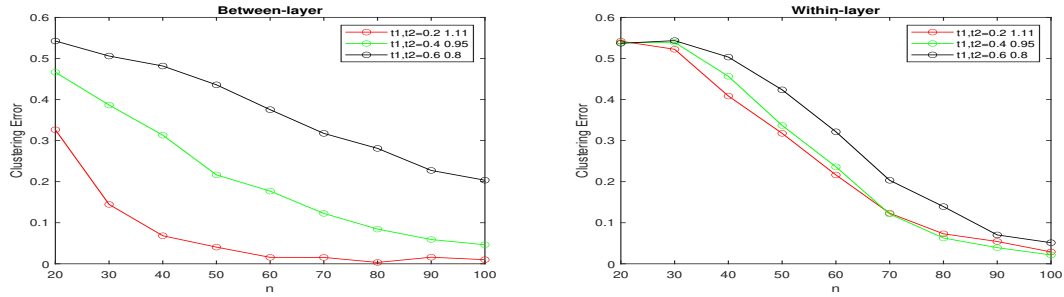


Figure 5.8: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $(t_1, t_2) = (0.2, 1.11); (0.4, 0.95); (0.6, 0.8)$ ,  $L = 50$  and  $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs.

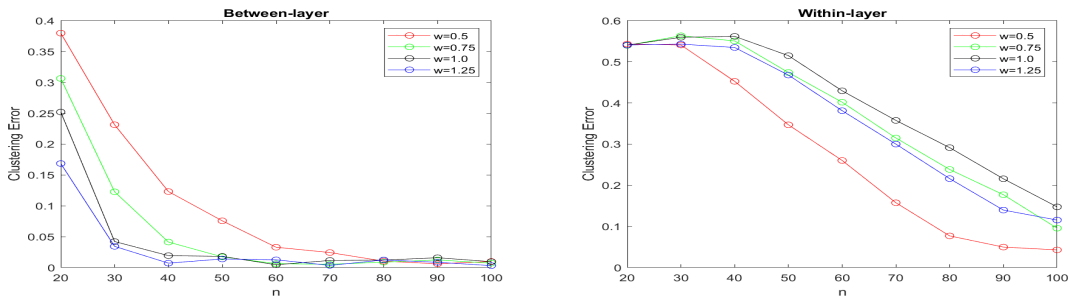


Figure 5.9: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5, 0.75, 1, 1.25$ ,  $t_1 = 0.2, t_2 = 1$ ,  $L = 50$  and  $n = 20, 30, 40, 50, 60, 70, 80, 90, 100$ . Results are averaged over 100 simulation runs.

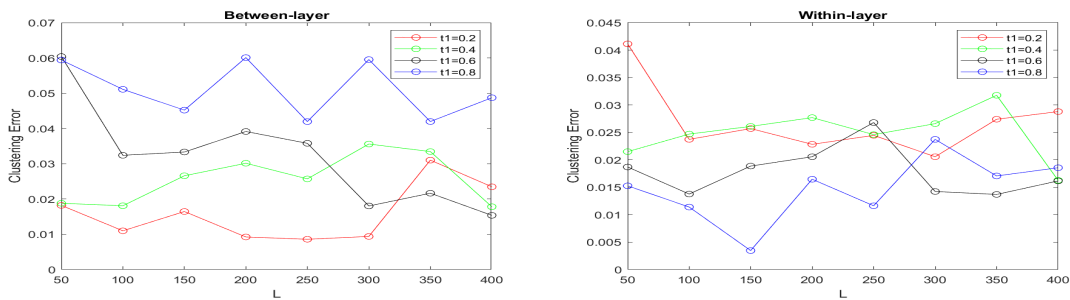


Figure 5.10: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $t_1 = 0.2, 0.4, 0.6, 0.8$ ,  $t_2 = 1.11$ ,  $L = 50, 100, 150, 200, 250, 300, 350, 400$  and  $n = 100$ . Results are averaged over 100 simulation runs.



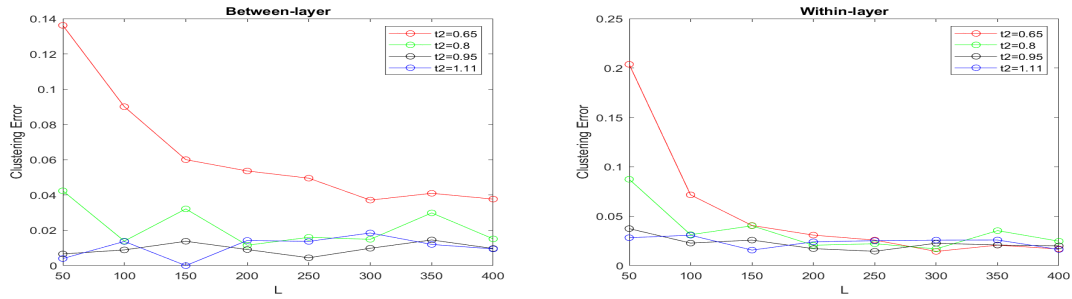


Figure 5.11: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $t_1 = 0.2$ ,  $t_2 = 0.65, 0.8, 0.95, 1.11$ ,  $L = 50, 100, 150, 200, 250, 300, 350, 400$  and  $n = 100$ . Results are averaged over 100 simulation runs.

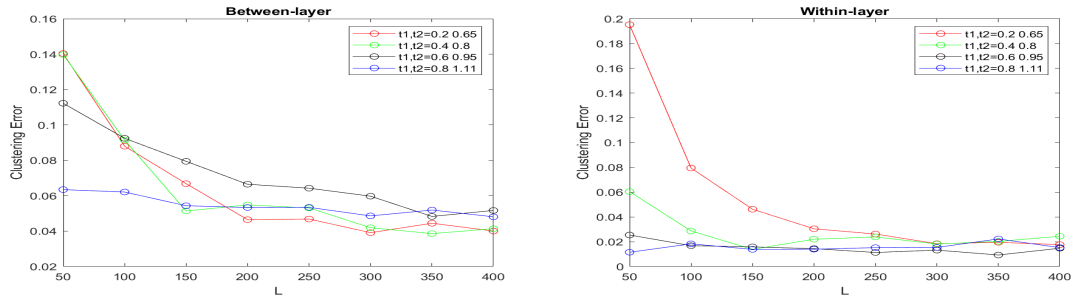


Figure 5.12: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $(t_1, t_2) = (0.2, 0.65); (0.4, 0.8); (0.6, 0.95); (0.8, 1.11)$ ,  $L = 50, 100, 150, 200, 250, 300, 350, 400$  and  $n = 100$ . Results are averaged over 100 simulation runs.

affects the within-layer clustering error rates. Indeed, if one had just one layer, then parameter  $w$  would be directly proportional to the signal-to-noise ratio in the DCBM (Gao et al. (2018)). Hence, the errors are larger when the value of  $w$  is closer to one. On the other hand, the between-layer clustering errors are less affected by the value of  $w$  since the algorithm relies on the differences between clustering assignments.

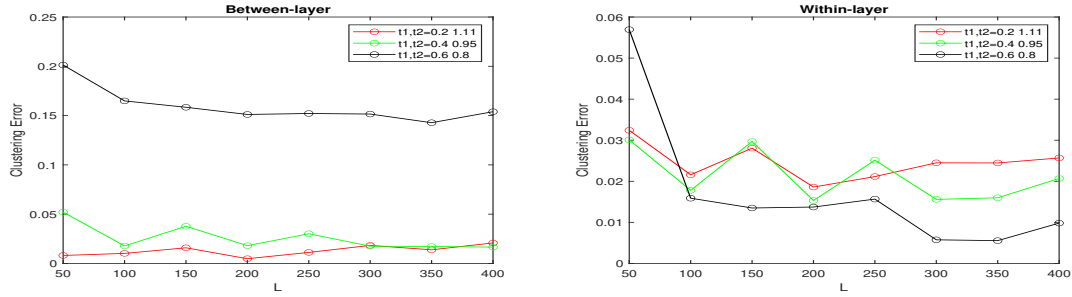


Figure 5.13: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5$ ,  $(t_1, t_2) = (0.2, 1.11); (0.4, 0.95); (0.6, 0.8)$ ,  $L = 50, 100, 150, 200, 250, 300, 350, 400$  and  $n = 100$ . Results are averaged over 100 simulation runs.

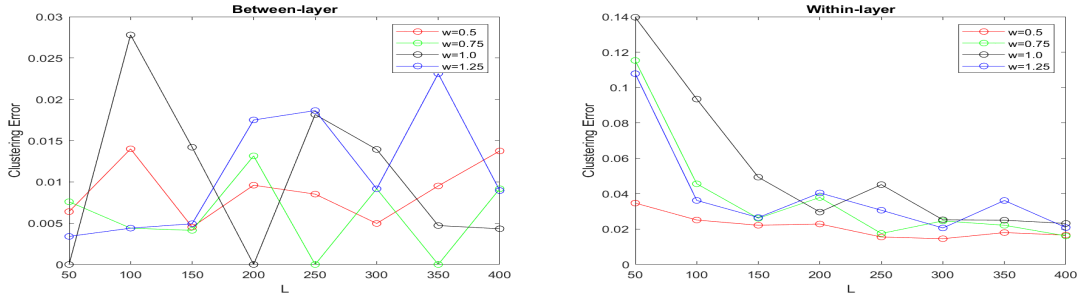


Figure 5.14: The between-layer clustering error rates of Algorithm 1 and the within-layer error rates of Algorithms 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.5, 0.75, 1.0, 1.25$ ,  $t_1 = 0.2, t_2 = 1$ ,  $L = 50, 100, 150, 200, 250, 300, 350, 400$  and  $n = 100$ . Results are averaged over 1000 simulation runs.

### 5.3 Comparison of $k$ -means and $k$ -medians algorithms for community detection for

#### DIMPLE-DECOR

Traditionally, community detection in the one-layer DCBM is carried out using the  $k$ -median clustering algorithm (Gao et al. (2018)), however, we use  $k$ -means clustering in Algorithm 3. This section serves as justification for this choice. In particular, we compare performances of the within-layer clustering in Algorithm 3, where in Step 5,  $k$ -means and  $k$ -medians algorithms are used for clustering. We report results in Figures 5.15–5.20. Specifically, Figures 5.15–5.17 consider fixed

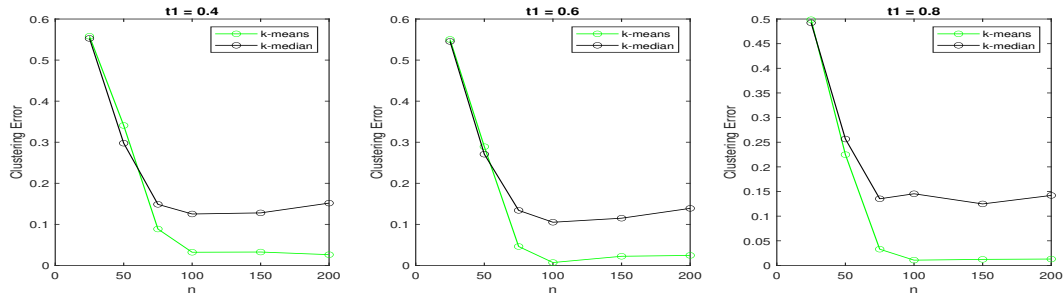


Figure 5.15: The comparison of the  $k$ -means and  $k$ -medians for Algorithm 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.8$ ,  $t_1 = 0.4, 0.6, 0.8$ ,  $t_2 = 1.11$ , and  $L = 100$ ,  $n = 25, 50, 75, 100, 150, 200$ . Results are averaged over 100 simulation runs.

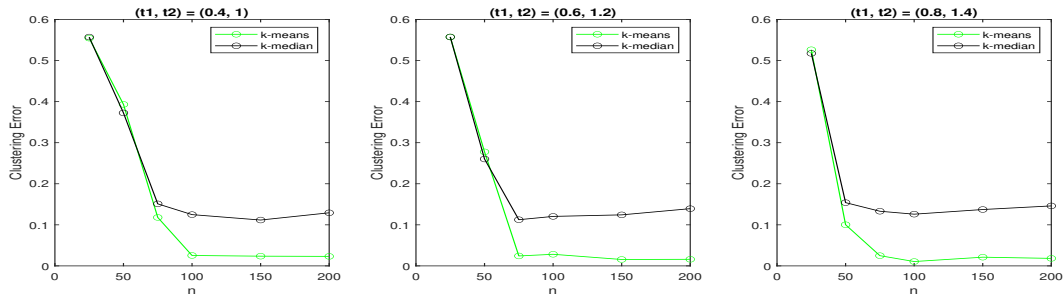


Figure 5.16: The comparison of the  $k$ -means and  $k$ -medians for Algorithm 3 for  $c = 0$ ,  $d = 0.5$ ,  $w = 0.5$ ,  $(t_1, t_2) = (0.4, 1), (0.6, 1.2), (0.8, 1.4)$ , and  $L = 100$ ,  $n = 25, 50, 75, 100, 150, 200$ . Results are averaged over 100 simulation runs.

$L = 100$  and varying values of  $n$ , while Figures 5.18–5.20 display errors for  $n = 100$  and varying values of  $L$ .

Our simulations show that, for small values of  $n$  ( $n \leq 50$ ),  $k$ -means and  $k$ -medians have very similar precision. However, as  $n$  grows,  $k$ -means algorithm leads to significantly smaller errors.

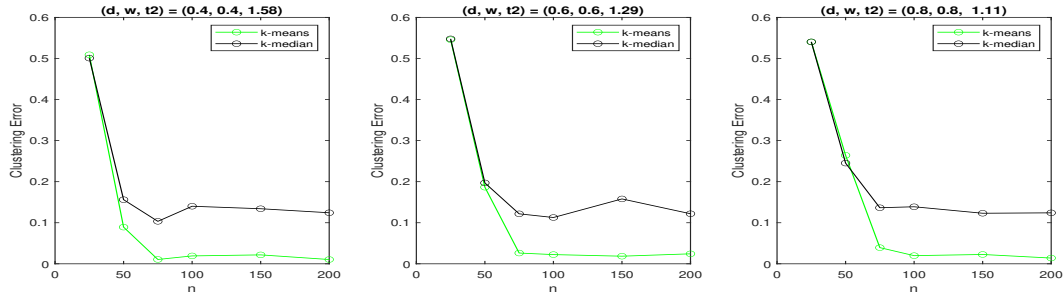


Figure 5.17: The comparison of the  $k$ -means and  $k$ -medians for Algorithm 3 for  $c = 0$ ,  $t_1 = 0.6$ ,  $(d, w, t_2) = (0.4, 0.4, 1.58), (0.6, 0.6, 1.29), (0.8, 0.8, 1.11)$ , and  $L = 100$ ,  $n = 25, 50, 75, 100, 150, 200$ . Results are averaged over 100 simulation runs.

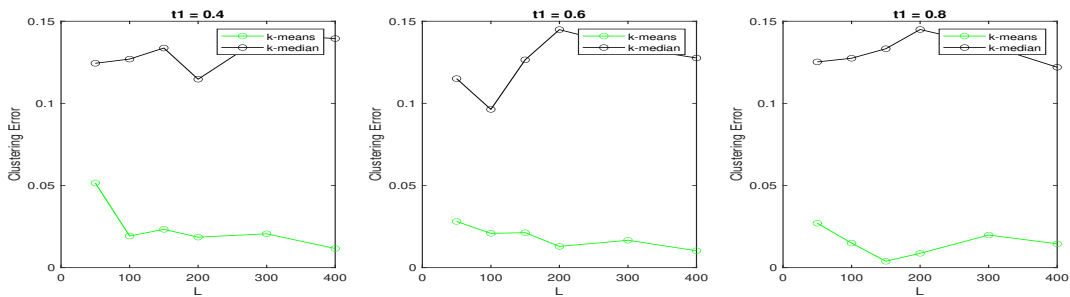


Figure 5.18: The comparison of the  $k$ -means and  $k$ -medians for Algorithm 3 for  $c = 0$ ,  $d = 0.8$ ,  $w = 0.8$ ,  $t_1 = 0.4, 0.6, 0.8$ ,  $t_2 = 1.11$ , and  $L = 50, 100, 150, 200, 300, 400$ ,  $n = 100$ . Results are averaged over 100 simulation runs.

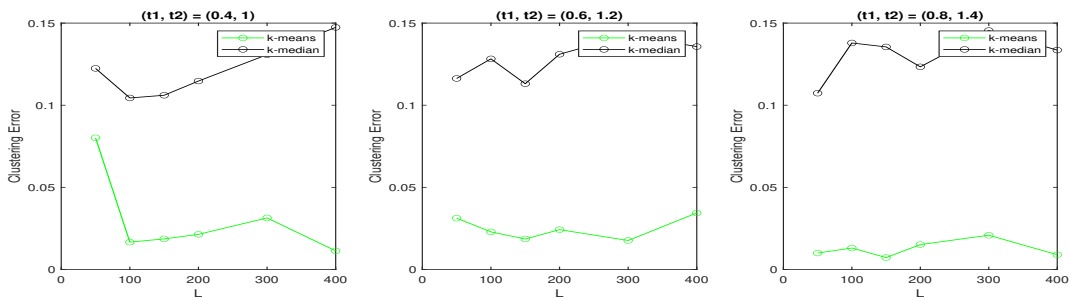


Figure 5.19: The comparison of the  $k$ -means and  $k$ -medians for Algorithm 3 for  $c = 0$ ,  $d = 0.5$ ,  $w = 0.5$ ,  $(t_1, t_2) = (0.4, 1), (0.6, 1.2), (0.8, 1.4)$ , and  $L = 50, 100, 150, 200, 300, 400$ ,  $n = 100$ . Results are averaged over 100 simulation runs.

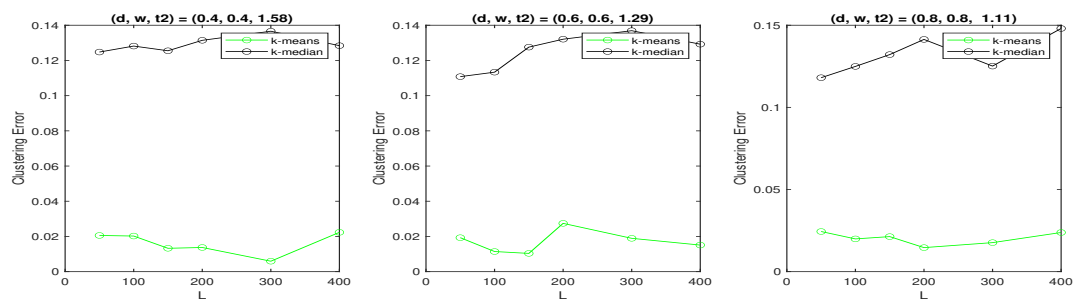


Figure 5.20: The comparison of the  $k$ -means and  $k$ -medians for Algorithm 3 for  $c = 0$ ,  $t_1 = 0.6$ ,  $(d, w, t_2) = (0.4, 0.4, 1.58), (0.6, 0.6, 1.29), (0.8, 0.8, 1.11)$ ,  $L = 50, 100, 150, 200, 300, 400$ ,  $n = 100$ . Results are averaged over 100 simulation runs.

## CHAPTER 6: APPLICATION TO REAL WORLD DATA

In this section, we consider applications of the DIMPLE and the DIMPLE-GDPG models to real-life data, and its comparison with the MMLSBM. Note that the between-layer clustering is carried out by Algorithm 1 for both the DIMPLE and the DIMPLE-GDPG models, so one can decide which of the models to use later in the analysis.

In our examples, the DIMPLE model with its SBM-imposed structures provided better descriptions of the organization of layers in each group than its GDPG-based DIMPLE-GDPG counterpart. Furthermore, we compared our between layer clustering partitions with the ones obtained on the basis of the MMLSBM setting.

### 6.1 Worldwide Food Trading Network Data

In this subsection, we consider applying our clustering algorithms to the Worldwide Food Trading Networks data collected by the Food and Agriculture Organization of the United Nations. The data have been described in De Domenico et al. (2015), and it is available at <https://www.fao.org/faostat/en/#data/TM>. The data includes export/import trading volumes among 245 countries for more than 300 food items. These data can be modeled as a multiplex network, in which layers represent different products, nodes are countries, and edges at each layer represent trading relationships of a specific food product among countries. A part of the data set was analyzed in Jing et al. (2021) and Fan et al. (2021).

Similarly to Jing et al. (2021) and Fan et al. (2021), we used data for the year 2010. We start with pre-processing the data by adding the export and import volumes for each pair of countries in each layer of the network, to produce undirected networks that fit in our model. To avoid sparsity, we

select 104 countries, whose total trading volumes are higher than the median among all countries. We choose 58 meat/dairy and fruit/vegetable items and constructed a network with 104 nodes and 58 layers.

While pre-processing the data, we observe that global trading patterns are different for the meat/dairy and the fruit/vegetable groups. Specifically, the trading volumes in meat/dairy group are much smaller than the trading volumes in the fruit/vegetable group. For this reason, we choose the thresholds that keep similar sparsity levels for the adjacency matrices. In particular, we set threshold to be equal to 1 unit for the meat/dairy group and 300 units for the fruit/vegetable group, and draw an edge between two nodes (countries) if the total trading volume between them is at or above the threshold.

We scramble the 58 layers and apply Algorithm 1 for the between-layer clustering. Since the food items consist of a meat/dairy and a fruit/vegetable group, we set  $M = 2$ . Due to the fact that there are five food regions (continents) in the world, Asia, America, Europe, Africa and Australia, we start with the number of communities in each layer to be  $K = 5$ . However, the latter leads to an unbalanced community structure, specifically, two communities that consists of only one country. For this reason, after experimenting, we set  $K = 3$ . Results of the between-layer clustering are presented in Figure 6.1. As it is evident from Figure 6.1, Algorithm 1 separates the food items into the meat/dairy and the fruit/vegetable groups.

Furthermore, we investigate the communities of countries that form trade clusters in each of the two layers. We use Algorithm 3 in the paper, and exhibit results of the within-layer clustering in Figure 6.2. The left panels in Figure 6.2 show the number of nodes (countries) in communities 1,2 and 3 in the meat/dairy and the fruit/vegetable group, respectively. The right panels in Figure 6.2 project those countries onto the world map. Here, the red color is used for community 1, the yellow color for community 2 and the green color for for community 3. Since we only select 104 countries

Meat Group Cluster 1	Bacon and ham	Meat, pig	
	Butter, cow milk	Meat, pig sausages	
	Fat, pigs	Meat, pork	
	Eggs, liquid	Offals, pigs, edible	
	Meat, chicken	Meat, beef, preparations	
	Meat, cattle	Meat, turkey	
	Meat, cattle, boneless (beef & veal)	Milk, whole dried	
	Meat, pig, preparations	Meat, sheep	
	Offals, edible, cattle	Meat, nes	
	Meat, chicken, canned	Meat, duck	
	Tallow	Offals, sheep, edible	
	Fruit/Vegetables Group Cluster 2	Apples	Grapefruit (inc. pomelos)
		Avocados	Peaches and nectarines
Cabbages and other brassicas		Plums and sloes	
Carrots and turnips		Tangerines, mandarins, clementines	
Cauliflowers and broccoli		Watermelons	
Cherries		Strawberries	
Chillies and peppers, green		Tomatoes	
Cucumbers and gherkins		Beans, green	
Figs dried		Fruit, fresh nes	
Kiwi fruit		Fruit, tropical fresh nes	
Oranges		Asparagus	
Papayas		Cassava dried	
Maize, green		Pineapples	
Persimmons		Leeks, other alliaceus vegetables	
Vegetables, fresh nes		Juice, pineapple, concentrated	
Spinach		Peas, green	
Sweet potatoes		Onions, shallots, green	
Roots and tubers, nes		Mangoes, mangosteens, guavas	

Figure 6.1: Results of clustering of food networks layers into  $M = 2$  clusters by Algorithm 1 in the paper

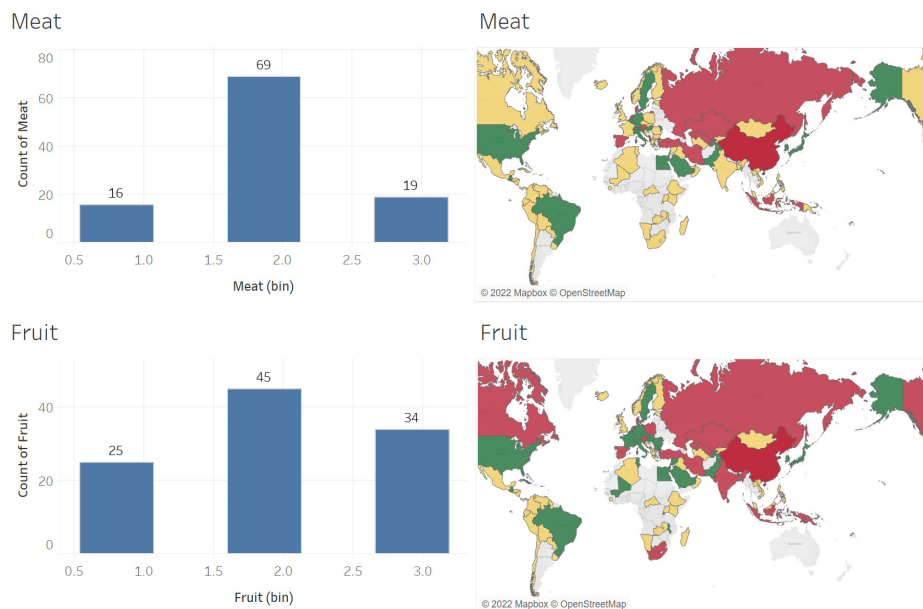


Figure 6.2: Trading communities for the meat/dairy (top) and the fruit/vegetable (bottom) groups. Left panels: community sizes; right panels: community memberships



Cluster1	Butter, cow milk	Meat, pig
	Eggs, liquid	Meat, pig sausages
	Meat, chicken	Meat, pork
	Meat, cattle	Meat, beef, preparations
	Meat, cattle, boneless (beef & veal)	Meat, turkey
	Meat, pig, preparations	Milk, whole dried
	Offals, edible, cattle	Meat, sheep
	Meat, chicken, canned	Meat, nes
		Tallow
Cluster 2	Bacon and ham	Tomatoes
	Fat, pigs	Beans, green
	Offals, pigs, edible	Fruit, fresh nes
	Meat, duck	Fruit, tropical fresh nes
	Offals, sheep,edible	Asparagus
	Apples	Cassava dried
	Avocados	Pineapples
	Cabbages and other brassicas	Leeks, other alliaceus vegetables
	Carrots and turnips	Juice, pineapple, concentrated
	Cauliflowers and broccoli	Peas, green
	Cherries	Onions, shallots, green
	Chillies and peppers, green	Mangoes, mangosteens, guavas
	Cucumbers and gherkins	Papayas
	Figs dried	Maize, green
	Kiwi fruit	Persimmons
	Oranges	Vegetables, fresh nes
	Grapefruit (inc. pomelos)	Spinach
	Peaches and nectarines	Sweet potatoes
	Plums and sloes	Roots and tubers, nes
	Tangerines, mandarins, clementines	Strawberries
Watermelons		

Figure 6.3: Results of clustering of food networks layers into  $M = 2$  clusters by ALMA algorithm of Fan et al. (2021)

to be a part of the network, some regions in the map are colored grey.

In order to justify application of the DIMPLE model, we also carry out data analysis assuming that data were generated using the MMLSBM. Specifically, we applied ALMA algorithm of Fan et al. (2021) for the layer clustering with the same parameters  $M = 2$  and  $K = 3$ . Results are presented in Figure 6.3. It is easy to notice that ALMA algorithm places some of the meat/dairy items into the fruit/vegetable group. We believe that this is due to the fact that MMLSBM is sensitive to the probabilities of connections rather than connection patterns.

## 6.2 Global Flights Network Data

In this section, we applied our clustering algorithms to the Global Flights Network data collected by the OpenFlights. As of June 2014, the OpenFlights Database contains 67663 routes between

3321 airports on 548 airlines spanning the globe. It is available at <https://openflights.org/data.html#airport>.

These data can be modeled as a multiplex network, in which layers represent different airlines, nodes are airports where airlines depart and land, and edges at each layer represent existing routes of a specific airline company between two airports. To avoid sparsity, we selected 224 airports, where over 150 airline companies have rights to depart and land in. Furthermore, we chose 81 airlines that have at least 240 routes between those airports, constructing a network with 224 nodes and 81 layers.

We scrambled the 81 layers and applied Algorithm 1 for the between-layer clustering. After experimenting with various values of  $M$  and  $K$ , we partitioned the airlines into  $M = 4$  groups, and used the ambient dimension  $K = 3$  for each of the groups. Results of the between-layer clustering are presented in Table 6.1.

We also partitioned airports in each of the groups of airlines into communities. Results are presented in Figure 6.4.

It is easy to see that in Table 6.1, the airlines are naturally grouped by geographical areas from where the flights are originated. Group 1 is constituted by Chinese airline and one Japanese airline which has flights predominantly in Far East. Group 2 consists of airlines that belong to countries in Asia, such as India, Japan, South Korea and Vietnam, Australia and New Zealand, and few big airlines in Gulf States (Saudi Arabia, United Arab Emirates, Qatar) that have a large number of flights to both Asia and Australia. Group 3 is formed by airlines originated from Europe and North Africa while Group 4 is comprised of airlines that fly in or from North or South America. Not surprisingly, this group includes two big European airlines, KLM and Air France, since those airlines are members of the SkyTeam alliance and share many flights originated in USA with Delta airlines.

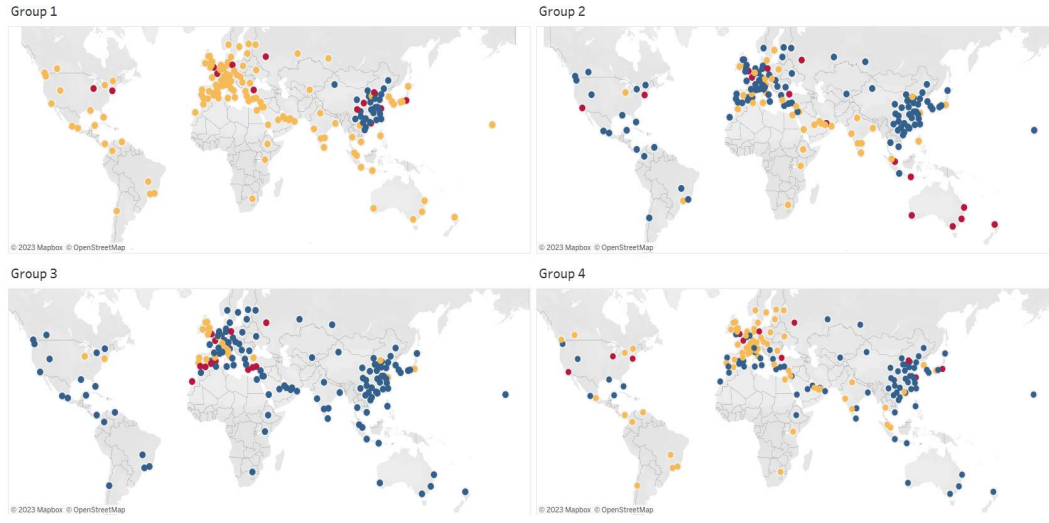


Figure 6.4: Communities for the four airlines groups. Group 1: airlines originated in China. Group 2: airlines originated in Asia, Australia, New Zealand, and Gulf States. Group 3: airlines originated in Europe and North Africa. Group 4: airlines originated in North or South America.

We also analyzed the airline data under the assumption that they follow the MMLSBM. To this end, we applied ALMA algorithm of Fan et al. (2021) for the layer clustering, with the same parameters  $M = 4$  and  $K = 3$ . Results are presented in Table 6.2. It is easy to see that while the DIMPLE model ensures a logical geography-based partition of the airlines, the MMLSBM does not. Indeed, the MMLSBM lumps almost all airlines into Group 1, placing few Chinese airlines into Group 2, few United States owned airlines together with Air France, Alitalia and KLM into Group 3, and Ryanair (Ireland), Transavia and Air Bourbon (France), easyJet and Jet2.com (United Kingdom) into Group 4. On the contrary, Algorithm 1 associated with the DIMPLE model delivers four balanced (similar in size) groups. This is due to the fact that MMLSBM groups airlines by the volume of operation rather than the structure of roots.

Table 6.1: Airlines Groups obtained using Algorithm 1 with  $K = 3$  and  $M = 4$

Airlines Groups under the DIMPLE-GDPG Model			
Group 1		Group 2	
China	Hainan Airlines	New Zealand	Air New Zealand
China	Air China	Republic of Korea	Korean Air
China	Sichuan Airlines	Singapore	Singapore Airlines
China	Shenzhen Airlines	Australia	Qantas
China	China Southern Airlines	Vietnam	Vietnam Airlines
China	Shandong Airlines	India	Air India Limited
China	China Eastern Airlines	India	IndiGo Airlines
China	Xiamen Airlines	Australia	Virgin Australia
Japan	Japan Air System	South Africa	South African Airways
Group 3		Indonesia	Garuda Indonesia
Germany	Lufthansa	Republic of Korea	Asiana Airlines
Russia	Ural Airlines	Malaysia	Malaysia Airlines
Switzerland	Swiss International Air Lines	India	Jet Airways
Morocco	Royal Air Maroc	Japan	Japan Airlines
Norway	Norwegian Air Shuttle	Japan	All Nippon Airways
Ireland	Ryanair	Qatar	Qatar Airways
Turkey	Turkish Airlines	Saudi Arabia	Saudi Arabian Airlines
Greece	Aegean Airlines	United Arab Emirates	Emirates
Algeria	Air Algerie	United Arab Emirates	Etiihad Airways
Ethiopia	Ethiopian Airlines	Group 4	
United Kingdom	Jet2.com	United States	JetBlue Airways
United Kingdom	Flybe	United States	US Airways
Russia	Transaero Airlines	United States	Alaska Airlines
Germany	Condor Flugdienst	United States	Southwest Airlines
Germany	TUIfly	United States	Delta Air Lines
Sweden	Scandinavian Airlines	United States	AirTran Airways
Portugal	TAP Portugal	United States	Spirit Airlines
France	Transavia France	United States	United Airlines
United Kingdom	British Airways	United States	American Airlines
Russia	S7 Airlines	United States	Frontier Airlines
Ireland	Aer Lingus	Canada	Air Canada
Germany	Germanwings	Canada	WestJet
Egypt	Egyptair	Mexico	AeroMexico
Austria	Austrian Airlines	Chile	LAN Airlines
Spain	Iberia Airlines	Brazil	TAM Brazilian Airlines
Germany	Air Berlin	South America	Avianca
Italy	Alitalia	Netherlands	KLM Royal Dutch Airlines
Hungary	Wizz Air	France	Air France
Finland	Finnair		
Russia	Aeroflot		
France	Air Bourbon		
Netherlands	Transavia Holland		
United Kingdom	easyJet		

Table 6.2: Airlines Groups obtained using ALMA algorithm of Fan et al. (2021) with  $K = 3$  and  $M = 4$

Airlines Groups under the MMLSBM			
Group 1		Group 2	
Japan	Japan Air System	China	Hainan Airlines
China	Sichuan Airlines	China	Air China
China	Shandong Airlines	China	Shenzhen Airlines
China	Xiamen Airlines	China	China Southern Airlines
Republic of Korea	Korean Air	China	China Eastern Airlines
Singapore	Singapore Airlines	Group 3	
Vietnam	Vietnam Airlines	Italy	Alitalia
India	Air India Limited	France	Air France
United States	US Airways	United States	Delta Air Lines
Australia	Qantas	United States	AirTran Airways
Mexico	AeroMexico	United States	Southwest Airlines
India	IndiGo Airlines	United States	American Airlines
South Africa	South African Airways	Netherlands	KLM Royal Dutch Airlines
Indonesia	Garuda Indonesia	Group 4	
Republic of Korea	Asiana Airlines	Ireland	Ryanair
Saudi Arabia	Saudi Arabian Airlines	United Kingdom	easyJet
Hong Kong	Cathay Pacific	United Kingdom	Jet2.com
South America	Avianca	France	Air Bourbon
Japan	Japan Airlines	France	Transavia France
Qatar	Qatar Airways	Group 1: Continuation	
Australia	Virgin Australia	United States	United Airlines
Japan	All Nippon Airways	New Zealand	Air New Zealand
Malaysia	Malaysia Airlines	United States	Frontier Airlines
India	Jet Airways	Canada	WestJet
United Arab Emirates	Etihad Airways	United Arab Emirates	Emirates
Germany	Lufthansa	Russia	Ural Airlines
Turkey	Pegasus Airlines	Morocco	Royal Air Maroc
Switzerland	Swiss International Airlines	Turkey	Turkish Airlines
Norway	Norwegian Air Shuttle	Ethiopia	Ethiopian Airlines
Greece	Aegean Airlines	Algeria	Air Algerie
United Kingdom	Flybe	Germany	Condor Flugdienst
Germany	TUIfly	Sweden	Scandinavian Airlines
Portugal	TAP Portugal	United Kingdom	British Airways
Russia	S7 Airlines	Austria	Austrian Airlines
Ireland	Aer Lingus	Spain	Iberia Airlines
Germany	Germanwings	Russia	Aeroflot
Egypt	Egyptair	Germany	Air Berlin
Hungary	Wizz Air	Russia	Transaero Airlines
Finland	Finnair	United States	Alaska Airlines
Netherlands	Transavia Holland	Brazil	TAM Brazilian Airlines
United States	JetBlue Airways	United States	Spirit Airlines
Chile	LAN Airlines	Canada	Air Canada

## CHAPTER 7: DISCUSSION

In this dissertation, we introduce the GDPG-equipped DIMPLE-GDPG multiplex network model where layers can be partitioned into groups with similar ambient subspace structures while the matrices of connections probabilities can be all different. In the common case when each layer follows the SBM and DCBM, the latter reduces to the DIMPLE and DIMPLE-DECOR models, where community affiliations are common for each group of layers while the matrices of block connection probabilities vary from one layer to another. The DIMPLE-GDPG model generalizes the COmmon Subspace Independent Edge (COSIE) random graph model of Arroyo et al. (2021) and Zheng and Tang (2022), while the DIMPLE model generalizes a multitude of the SBM-equipped multiplex network settings. Specifically, it includes, as its particular cases, the Mixture MultiLayer Stochastic Block Model (MMLSBM) of Stanley et al. (2016), Jing et al. (2021) and Fan et al. (2021), and the multitude of papers that assume that communities persist in all layers of the network (see, e.g., Bhattacharyya and Chatterjee (2020), Lei and Lin (2022), Lei et al. (2019), Paul and Chen (2016), Paul and Chen (2020)).

Our real data examples in Chapter 6 show that our models deliver more understandable description of data than the MMLSBM, due to the flexibility of the DIMPLE model.

If  $M = 1$ , the DIMPLE-GDPG reduces to COSIE model, and we believe that our paper provides some improvements due to employment of a different algorithm for the matrix  $\mathbf{V}$  estimation. Indeed, Arroyo et al. (2021) showed that

$$\mathbb{E} \left\| \sin \Theta(\widehat{\mathbf{V}}, \mathbf{V}) \right\| \leq C \left[ \frac{K^{3/2}}{\sqrt{n \rho_n L}} + \frac{K^{5/2}}{n \rho_n} \right], \quad (7.1)$$

while Zheng and Tang (2022), who use a different technique for recovery of  $\mathbf{V}$ , state that, with high

probability,  $\|\sin \Theta(\widehat{\mathbf{V}}, \mathbf{V})\|_{2 \rightarrow \infty} \leq C K n^{-1} \sqrt{\log n} / \sqrt{\rho_n}$ . The latter leads to  $\|\sin \Theta(\widehat{\mathbf{V}}, \mathbf{V})\|_F \leq C K \sqrt{(n \rho_n)^{-1} \log n}$ . Thus, the upper bound (7.1) is similar to our upper bound (4.25), which is derived for the (larger) Frobenius norm and holds not only in expectation but with the high probability. The upper bound of Zheng and Tang (2022) is larger (if one uses the Frobenius norm) and, in addition, does not decline when  $L$  grows.

As our theory (Theorems 1 and 2, and also Corollary 1) the simulation results imply, when  $K$  and  $M$  are fixed constants, the clustering precision in both algorithms cease to decrease for a given number of nodes  $n$  when  $L$  grows:

$$R_{BL} \lesssim C \rho_n^{-1} n^{-1}, \quad R_{S,max} \asymp R_{S,ave} \asymp R_{WL} \lesssim C (\rho_n^{-1} n^{-1} + n^{-1} L^{-1} \rho_n^{-1} \log n)$$

We believe that this is not caused by the deficiency of our methodology but is rather due to the fact, that the number of parameters in the model grows linearly in  $L$  for a fixed  $n$ . Indeed, even in the case of the simplest, SBM-based DIMPLE model, the total number of independent parameters in the model is  $O(K^2 L + M n \log K + L \log M)$ , since we have  $L$  matrices  $\mathbf{B}^{(l)}$ ,  $M$  clustering matrices for the SBMs in the groups of layers, and a clustering matrix of the layers, while the total number of observations is  $O(n^2 L)$ . The latter implies that, while for small values of  $L$ , the term  $(M n \log K) / (n^2 L)$  may dominate the error, eventually, as  $L$  grows, the term  $L(K^2 + \log M) / (n^2 L)$  becomes larger for a fixed  $n$ .

Incidentally, we observe that a similar phenomenon holds in the MMLSBM, where the block probability matrices are the same in all layers of each of the groups. While Stanley et al. (2016) does not produce relevant theoretical results, Jing et al. (2021) simply assume that  $L \leq n$ , which makes the issue of error rates for a growing value of  $L$  inconsequential. Similarly, the ALMA

clustering error rates in Fan et al. (2021)

$$R_{BL}^{ALMA} \lesssim C (\rho_n^{-1} n^{-2} + \rho_n^{-2} n^{-2} [\min(n, L)]^{-1}),$$

$$R_{WL}^{ALMA} \lesssim C (n^{-1} L^{-1} \rho_n^{-1} + \rho_n^{-1} n^{-2} + \rho_n^{-2} n^{-2} [\min(n, L)]^{-1}),$$

imply that, for given  $n$  and  $\rho_n$ , as  $L$  grows, the clustering errors flatten.

Our simulation study also exhibits similar dynamics. In particular, the between-layer clustering errors flatten when  $n$  is fixed and  $L$  grows, while the errors of subspace estimation and of the within-layer clustering, for a fixed  $n$ , decrease initially and then stop decreasing as  $L$  become larger and larger.

We remark that, unlike the ALMA methodology in Fan et al. (2021) or the TWIST algorithm in Jing et al. (2021), all three algorithms in this paper are not iterative. It is known, that if one needs to recover a low rank tensor, then the power iterations can improve precision guarantees. This has been shown in the context of estimation of a low rank tensor in, e.g., Zhang and Xia (2018b), and in the context of the clustering in the tensor block model in Han et al. (2021). While both ALMA and TWIST are designed for the MMLSBM, which results in a low rank probability tensor, the DIMPLE model does not lead to a low rank probability tensor. Therefore, it is not clear whether iterative techniques are advantageous in the DIMPLE setting. Our very limited experimentation with iterative algorithms did not lead to significant improvement of clustering precision. Investigation of this issue is a matter of future research.



## **APPENDIX A: PROOF FOR DIMPLE MODEL**

## A.1 Proof of Theorem 1

Use notations of the paper, note that

$$\left\| \widehat{\mathbf{U}}_{A,l}(\widehat{\mathbf{U}}_{A,l})^T - \mathbf{U}_{P,l}(\mathbf{U}_{P,l})^T \right\|_F^2 = 2 \left\| \sin \Theta(\mathbf{U}_{P,l}, \widehat{\mathbf{U}}_{A,l}) \right\|_F^2$$

where  $\widehat{\mathbf{U}}_{A,l}$  and  $\mathbf{U}_{P,l}$  are defined in (3.11) and (3.7), respectively. By Davis-Kahan Theorem,

$$\left\| \widehat{\mathbf{U}}_{A,l}(\widehat{\mathbf{U}}_{A,l})^T - \mathbf{U}_{P,l}(\mathbf{U}_{P,l})^T \right\|_F \leq \frac{2\sqrt{K_m} \|\mathbf{A}^{(l)} - \mathbf{P}^{(l)}\|}{\sigma_{K_m}(\mathbf{P}^{(l)})}, \quad m = c(l)$$

By Theorem 5.2 of Lei and Rinaldo (2015), if  $n\rho_n \geq C_\rho \log n$ , then, for any  $\tau > 0$ , there exists a constant  $C_\tau$ , such that

$$\mathbb{P} \left\{ \|\mathbf{A}^{(l)} - \mathbf{P}^{(l)}\| \leq C_\tau \sqrt{n\rho_n} \right\} \geq 1 - n^{-\tau}$$

Then

$$\mathbb{P} \left\{ \max_{l \in [L]} \|\mathbf{A}^{(l)} - \mathbf{P}^{(l)}\| \leq C_\tau \sqrt{n\rho_n} \right\} \geq 1 - Ln^{-\tau}$$

In order to construct a lower bound for  $\sigma_{K_m}(\mathbf{P}^{(l)})$ , note that under Assumptions **A1–A6**, one has

$$\sigma_{K_m}(\mathbf{P}^{(l)}) = \sigma_{K_m}(\mathbf{Q}^{(l)}) \geq C_\lambda K_m^{-1/2} \|\mathbf{Q}^{(l)}\| \geq \underline{c}_\rho C_\lambda K^{-1/2} \rho_n \|\mathbf{P}_0^{(l)}\| \geq \underline{c}_\rho C_\lambda C_{0,P} n \rho_n K^{-1} \tag{A.1}$$

Combining the formulas and taking into account that

$$\|\widehat{\Theta} - \Theta\|_F^2 \leq L \max_{l \in [L]} \left\| \widehat{\mathbf{U}}_{A,l}(\widehat{\mathbf{U}}_{A,l})^T - \mathbf{U}_{P,l}(\mathbf{U}_{P,l})^T \right\|_F^2,$$

obtain

$$\mathbb{P} \left\{ \left\| \widehat{\Theta} - \Theta \right\|_F^2 \leq C \frac{LK^3}{n\rho_n} \right\} \geq 1 - Ln^{-\tau}$$

Also, by Davis-Kahan Theorem,

$$\|\sin \Theta(\widehat{\mathcal{W}}, \mathcal{W})\|_F \leq \frac{\|\widehat{\Theta} - \Theta\|_F}{\sigma_M(\Theta)}$$

By formula (4.18) and (4.19),

$$\sigma_M(\Theta) \geq \sqrt{\frac{\bar{c}}{c}} \frac{1}{\kappa_0^2} \|\Theta\| \geq \sqrt{\frac{\bar{c}}{c}} \frac{1}{\kappa_0^2} \frac{\|\Theta\|_F}{\sqrt{M}} \geq \sqrt{\frac{\bar{c}}{c}} \frac{\sqrt{KL}}{\kappa_0^2 \sqrt{M}}$$

Hence,

$$\mathbb{P} \left\{ \left\| \sin \Theta(\widehat{\mathcal{W}}, \mathcal{W}) \right\|_F^2 \leq \frac{CK^2M}{n\rho_n} \right\} \geq 1 - Ln^{-\tau}$$

Use Lemma C.1 of Lei and Lin (2022):

**Lemma 2. ( Lemma C.1 of Lei and Lin (2022)).** *Let  $\mathbf{X}$  be an  $m \times d$  matrix with  $K$  distinct rows and minimum pairwise Euclidean norm separation  $\gamma$ . Let  $\widehat{\mathbf{X}}$  be another  $(m \times d)$  matrix and  $(\widehat{\Theta}, \widehat{\mathbf{Q}})$  be an  $(1 + \epsilon)$ -approximate solution to  $K$ -means problem with input  $X$ . Then, the number of errors in  $\widehat{\Theta}$  as an estimate number of errors in  $\widehat{\Theta}$  as an estimate of row clusters of  $X$  is no larger than  $C_\epsilon \left\| \sin \Theta(\widehat{\mathbf{X}}, \mathbf{X}) \right\|_F^2 \gamma^{-2}$ , where  $C_\epsilon$  depends only on  $\epsilon$ .*

Since the row separation of  $\mathcal{W}$  is at least  $1/\sqrt{L_m} \geq \sqrt{M}/(\bar{c}\sqrt{L})$ , the number of errors is bounded above by  $CK^2L(n\rho_n)^{-1}$ , with probability at least  $1 - Ln^{-\tau}$ . The latter, in combination with (4.14), implies (4.21).

## A.2 Proof of Theorem 2

The proof requires the following lemma.

**Lemma 3.** *Let  $\mathcal{W}$  and  $\widehat{\mathcal{W}}$  be defined as in (3.9) and (3.14), respectively. Let assumptions of*

Theorem 2 hold. Then, on the set  $\Omega$ , with  $\mathbb{P}(\Omega) \geq 1 - L n^{-\tau}$ , on which (4.21) holds, one has

$$0.5 L_m^{-1} \leq \widehat{L}_m \leq 2 L_m^{-1}, \quad m \in [M] \quad (\text{A.2})$$

$$|\widehat{L}_m^{-1/2} - L_m^{-1/2}| \leq C(n\rho_n \sqrt{L_m})^{-1} M K^2, \quad m \in [M] \quad (\text{A.3})$$

$$\min_{\mathcal{P} \in \widehat{\mathfrak{F}}(M)} \|\widehat{\mathbf{W}} - \mathbf{W} \mathcal{P}\|_F^2 \leq C(n\rho_n)^{-1} M K^2 \quad (\text{A.4})$$

Consider tensors  $\mathcal{G} \in \mathbb{R}^{n \times n \times L}$  and  $\mathcal{H} = \mathcal{G} \times_3 \mathbf{W}^T \in \mathbb{R}^{n \times n \times M}$  with layers, respectively,  $\mathbf{G}^{(l)} = \mathcal{G}(:, :, l)$  and  $\mathbf{H}^{(m)} = \mathcal{H}(:, :, m)$  of the forms

$$\mathbf{G}^{(l)} = (\mathbf{P}^{(l)})^2, \quad \mathbf{H}^{(m)} = L_m^{-1/2} \sum_{c(l)=m} \mathbf{G}^{(l)}, \quad l \in [L], m \in [M] \quad (\text{A.5})$$

In order to assess  $R_{S,max}$  and  $R_{S,ave}$ , one needs to examine the spectral structure of matrices  $\mathbf{H}^{(m)}$  and their deviation from the sample-based versions  $\widehat{\mathbf{H}}^{(m)} = \widehat{\mathcal{H}}(:, :, m)$ . We start with the first task.

It follows from (3.5) and (3.6) that

$$\mathbf{H}^{(m)} = \mathbf{V}^{(m)} \overline{\mathbf{Q}}^{(m)} (\mathbf{V}^{(m)})^T \quad \text{with} \quad \overline{\mathbf{Q}}^{(m)} = L_m^{-1/2} \sum_{c(l)=m} (\mathbf{Q}^{(l)})^2 \quad (\text{A.6})$$

Here, by (3.5), one has  $(\mathbf{Q}^{(l)})^2 = \mathbf{O}_Q^{(l)} (\mathbf{S}_Q^{(l)})^2 (\mathbf{O}_Q^{(l)})^T$ , so that all eigenvalues of  $(\mathbf{Q}^{(l)})^2$  are positive. Applying the Theorem in Complement 10.1.2 on page 327 of Rao and Rao (1998) and Assumptions **A1–A6**, obtain

$$\begin{aligned} \sigma_{K_m}(\mathbf{H}^{(m)}) &= \sigma_{K_m}(\overline{\mathbf{Q}}^{(m)}) \geq L_m^{-1/2} \sum_{c(l)=m} \sigma_{K_m}((\mathbf{Q}^{(l)})^2) \\ &\geq C_\lambda^2 L_m^{-1/2} K_m^{-1} \sum_{c(l)=m} \|\mathbf{Q}^{(l)}\|_F^2 \geq C_\lambda^2 L_m^{-1/2} K^{-1} \sum_{c(l)=m} \rho_{n,l}^2 \|\mathbf{P}_0^{(l)}\|_F^2 \\ &\geq C_\lambda^2 c_\rho^2 C_{0,P}^2 L_m^{-1/2} K^{-2} n^2 \rho_n^2 L_m \end{aligned}$$

so that

$$\sigma_{K_m}(\mathbf{H}^{(m)}) \geq C (K^2 \sqrt{M})^{-1} n^2 \rho_n^2 \sqrt{L} \quad (\text{A.7})$$

Using Davis-Kahan theorem, Lemma 1 of Cai and Zhang (2018b) and formula (A.7), obtain

$$\left\| \sin \Theta \left( \widehat{\mathbf{V}}^{(m)}, \mathbf{V}^{(m)} \right) \right\|_F \leq \frac{2 \sqrt{2K_m} \|\widehat{\mathbf{H}}^{(m)} - \mathbf{H}^{(m)}\|}{\sigma_{K_m}(\mathbf{H}^{(m)})} \leq \frac{C K^{5/2} M^{1/2} \|\widehat{\mathbf{H}}^{(m)} - \mathbf{H}^{(m)}\|}{n^2 \rho_n^2 \sqrt{L}} \quad (\text{A.8})$$

Recall that  $\mathbf{H}^{(m)} = [\mathcal{G} \times_3 \mathbf{W}^T](:, :, m)$  and  $\widehat{\mathbf{H}}^{(m)} = [\widehat{\mathcal{G}} \times_3 \widehat{\mathbf{W}}^T](:, :, m)$ . Denote

$$\overline{\mathbf{G}}^{(m)} = \sum_{c(l)=m} \mathbf{G}^{(l)} = \sqrt{L_m} \mathbf{H}^{(m)}, \quad \widehat{\mathbf{G}}^{(m)} = \sqrt{L_m} [\widehat{\mathcal{G}} \times_3 \mathbf{W}^T](:, :, m) = \sum_{c(l)=m} \widehat{\mathbf{G}}^{(l)} \quad (\text{A.9})$$

Observe that

$$\|\widehat{\mathbf{H}}^{(m)} - \mathbf{H}^{(m)}\| \leq \Delta_1^{(m)} + \Delta_2^{(m)}, \quad \sum_{m=1}^M \|\widehat{\mathbf{H}}^{(m)} - \mathbf{H}^{(m)}\|^2 \leq 2(\Delta_1 + \Delta_2) \quad (\text{A.10})$$

where

$$\Delta_1^{(m)} = L_m^{-1/2} \left\| \widehat{\mathbf{G}}^{(m)} - \overline{\mathbf{G}}^{(m)} \right\|, \quad \Delta_2^{(m)} = \left\| [\widehat{\mathcal{G}} \times_3 (\widehat{\mathbf{W}} - \mathbf{W})^T](:, :, m) \right\|,$$

$$\Delta_i = \sum (\Delta_i^{(m)})^2, \quad i = 1, 2$$

To upper-bound  $\Delta_1^{(m)}$  and  $\Delta_2^{(m)}$ , we use the following lemma that modifies upper bounds in Theorem 3 of Lei and Lin (2022) in the absence of the sparsity assumption  $\rho_n n \leq C$ :

**Lemma 4.** *Let Assumptions A1–A6 hold,  $\mathbf{G}^{(l)} = (\mathbf{P}^{(l)})^2$  and  $\widehat{\mathbf{G}}^{(l)} = (\mathbf{A}^{(l)})^2 - \text{diag}(\mathbf{A}^{(l)} \mathbf{1})$ , where  $c(l) = m, l \in [\widetilde{L}]$ . Let*

$$\mathbf{G} = \sum_{l=1}^{\widetilde{L}} \mathbf{G}^{(l)}, \quad \widehat{\mathbf{G}} = \sum_{l=1}^{\widetilde{L}} \widehat{\mathbf{G}}^{(l)}$$

Then, for any  $\tau > 0$ , there exists a constant  $C$  that depends only on constants in Assumptions **A1–A6**, and a constant  $\tilde{C}_{\tau,\epsilon}$  which depends only on  $\tau$  and  $\epsilon$ , such that one has

$$\mathbb{P} \left\{ \|\widehat{\mathbf{G}} - \mathbf{G}\| \leq C \left[ \rho_n^{3/2} n^{3/2} \sqrt{\tilde{L} \log(\tilde{L} + n)} + \rho_n^2 n \tilde{L} \right] \right\} \geq 1 - \tilde{C}_{\tau,\epsilon} n^{1-\tau} \quad (\text{A.11})$$

Applying Lemma 4 with  $\tilde{L} = L_m$  and taking into account that, by assumption (4.14), one has  $\log n \leq \log(L + n) \leq (1 + \tau_0) \log n$ , obtain that, with probability at least  $1 - \tilde{C}_{\tau,\epsilon} n^{1-\tau}$ , one has  $\Delta_1^{(m)} \leq \tilde{C} [\rho_n^{3/2} n^{3/2} \sqrt{\log n} + \rho_n^2 n \sqrt{L_m}]$ . Therefore,

$$\mathbb{P} \left\{ \max_{m \in [M]} \Delta_1^{(m)} \leq C \left[ \rho_n^{3/2} n^{3/2} \sqrt{\log n} + \rho_n^2 n \sqrt{L/M} \right] \right\} \geq 1 - \tilde{C}_{\tau,\epsilon} M n^{1-\tau} \quad (\text{A.12})$$

and  $\Delta_1 \leq C [\rho_n^{3/2} n^{3/2} M \sqrt{\log n} + \rho_n^2 n \sqrt{LM}]$  with the same probability.

In the case of  $\Delta_2^{(m)}$ , we start with an upper bound for  $\Delta_2$ . Note that, by Cauchy inequality and Lemma 3, with probability at least  $1 - L n^{-\tau}$ , one has

$$\begin{aligned} \Delta_2 &= \sum_{m=1}^M \left\| \sum_{l=1}^L \widehat{\mathbf{G}}^{(l)} (\widehat{\mathbf{W}}_{l,m} - \mathbf{W}_{l,m}) \right\|^2 \leq \sum_{m=1}^M \left[ \sum_{l=1}^L \left\| \widehat{\mathbf{G}}^{(l)} \right\| \left| \widehat{\mathbf{W}}_{l,m} - \mathbf{W}_{l,m} \right| \right]^2 \\ &\leq \|\widehat{\mathbf{W}} - \mathbf{W}\|_F^2 \sum_{l=1}^L \|\widehat{\mathbf{G}}^{(l)}\|^2 \leq C(n\rho_n)^{-1} M K^2 \sum_{l=1}^L \|\widehat{\mathbf{G}}^{(l)}\|^2 \end{aligned} \quad (\text{A.13})$$

In order to obtain an upper bound for the sum of  $\|\widehat{\mathbf{G}}^{(l)}\|^2$ , use Lemma 4 with  $\tilde{L} = 1$ . Derive

$$\mathbb{P} \left\{ \max_{l \in [L]} \|\widehat{\mathbf{G}}^{(l)} - \mathbf{G}^{(l)}\|^2 \leq C \left[ \rho_n^3 n^3 \log n + \rho_n^4 n^2 \right] \right\} \geq 1 - \tilde{C}_{\tau,\epsilon} L n^{1-\tau}$$

On the other hand,

$$\|\mathbf{G}^{(l)}\| \leq \|\mathbf{P}^{(l)}\|^2 \leq C_\lambda^{-2} [\sigma_{K_m}(\mathbf{P}^{(l)})]^2 \leq C_\lambda^{-2} K_m^{-1} \|\mathbf{P}^{(l)}\|_F^2 \leq C_\lambda^{-2} C_K^{-1} K^{-1} (n\rho_n)^2$$

Since  $\|\widehat{\mathbf{G}}^{(l)}\| \leq \|\mathbf{G}^{(l)}\| + \|\widehat{\mathbf{G}}^{(l)} - \mathbf{G}^{(l)}\|$ , with probability at least  $1 - \widetilde{C}_{\tau,\epsilon} L n^{1-\tau}$ , obtain

$$\max_{l \in [L]} \|\widehat{\mathbf{G}}^{(l)}\|^2 \leq C (K^{-2} n^4 \rho_n^4 + \rho_n^3 n^3 \log n + \rho_n^4 n^2) \leq C K^{-2} n^4 \rho_n^4 (1 + (n\rho_n)^{-1} K^2 \log n)$$

Plugging the latter upper bound into (A.13), obtain

$$\mathbb{P} \left\{ \Delta_2 \leq C n^3 \rho_n^3 L M (1 + (n\rho_n)^{-1} K^2 \log n) \right\} \geq 1 - \widetilde{C}_{\tau,\epsilon} L n^{1-\tau} \quad (\text{A.14})$$

To complete the proof, combine formulas (A.8), (A.10), (A.12) and (A.14) take into account that  $\Delta_2^{(m)} \leq \sqrt{\Delta_2}$  for any  $m \in [M]$ .

### A.3 Proof of Corollary 1

To find the clustering errors for each group of clusters, we again use Lemma 2 which yields that the number of clustering errors in the layer  $m \in [M]$  is bounded above by

$C_\epsilon \left\| \sin \Theta(\widehat{\mathbf{V}}^{(m)}, \mathbf{V}^{(m)}) \right\|_F^2 \gamma_m^{-2}$ , where  $\gamma_m$  is the minimum pairwise Euclidean norm separation between rows of matrix  $\mathbf{V}^{(m)}$ . It is easy to see that under Assumptions **A1–A6**, one has

$$\gamma_m^2 \geq 2 \min(n_{k,m}^{-1}) \geq 2 C_K K / (\bar{c} n), \quad (\text{A.15})$$

so that the total number of errors is bounded above by  $C M K^{-1} n R_{S,ave}$  where  $R_{S,ave}$  is given by (4.24). Then, the average within layer clustering error is bounded above by  $K^{-1} R_{S,ave}$ , which completes the proof for the situation of the DIMPLE model. For the DIMPLE-DECOR model, as the bounds of the degree parameters are associated with some constants, we may need some changes for the constants in the process of proof and then we get (4.27).

#### A.4 Proof of supplementary lemmas

**Proof of Lemma 1** Note that, due to the structure of the tensor  $\mathcal{B}$ , for some  $s > 0$ , one has  $\sigma_{\min}(\overline{\mathbf{R}}) = \sigma_{\max}(\overline{\mathbf{R}}) = s$ , so that

$$\sigma_1(\mathbf{F}) \leq \sigma_1^2(\overline{\mathbf{D}}) s \sqrt{\max_{m \in [M]} L_m}, \quad \sigma_M(\mathbf{F}) \geq \sigma_M^2(\overline{\mathbf{D}}) s \sqrt{\min_{m \in [M]} L_m}.$$

Then, by Assumptions **A1** and **A4**,  $\sigma_1^2(\mathbf{F}) \leq \kappa_0^4 \sigma_M^2(\mathbf{F}) \bar{c}/\underline{c}$ . Therefore, the first inequality in (4.19) holds. To prove the second inequality, observe that

$$\|\Theta\|_F^2 = \text{Tr}(\mathbf{F}\mathbf{F}^T(\overline{\mathbf{U}}^T\overline{\mathbf{U}} \otimes \overline{\mathbf{U}}^T\overline{\mathbf{U}})) = \|\mathbf{F}\|_F^2$$

and, on the other hand,

$$\|\Theta\|_F^2 = \sum_{l=1}^L \|\mathbf{U}_{P,l}(\mathbf{U}_{P,l})^T\|_F^2 = \sum_{m=1}^M L_m \|\mathbf{V}^{(m)}(\mathbf{V}^{(m)})^T\|_F^2 = \sum_{m=1}^M L_m K_m \geq C_K K L, \quad (\text{A.16})$$

which together complete the proof.

**Proof of Lemma 3** Note that, for  $m \in [M]$ ,  $|\widehat{L}_m - L_m| \leq LR_{BL} \leq C(n\rho_n)^{-1} L K^2$ . Then,

$$\left| \frac{1}{\widehat{L}_m} - \frac{1}{L_m} \right| = \frac{|\widehat{L}_m - L_m|}{\widehat{L}_m L_m} \leq \frac{CMK^2}{n\rho_n} \frac{1}{\widehat{L}_m}$$

Then, due to assumption (4.22), the coefficient in front of  $\widehat{L}_m^{-1}$  is bounded by 1/2 and, hence, (A.2) holds. Inequality (A.3) follows directly from the upper bound on  $|\widehat{L}_m - L_m|$  and (A.2).



To prove (A.4), recall that formulas (3.9) and (3.14) imply that

$$\begin{aligned} \|\widehat{\mathbf{W}} - \mathbf{W}\|_F^2 &\leq \left\| \widehat{\mathbf{C}}(\widehat{\mathbf{D}}_{\hat{c}})^{-1/2} - \mathbf{C}(\mathbf{D}_c)^{-1/2} \right\|_F^2 \\ &\leq 2 \left\| \widehat{\mathbf{C}}(\widehat{\mathbf{D}}_{\hat{c}})^{-1/2} \right\|_F^2 \left\| \mathbf{I}_M - (\widehat{\mathbf{D}}_{\hat{c}})^{1/2}(\mathbf{D}_c)^{-1/2} \right\|_F^2 + 2 \left\| \widehat{\mathbf{C}} - \mathbf{C} \right\|_F^2 \left\| (\mathbf{D}_c)^{-1/2} \right\|_F^2 \end{aligned} \quad (\text{A.17})$$

where  $\mathbf{D}_c = \text{diag}(L_1, \dots, L_M)$  and  $\widehat{\mathbf{D}}_{\hat{c}} = \text{diag}(\widehat{L}_1, \dots, \widehat{L}_M)$ . It is easy to see that  $\|\widehat{\mathbf{C}}(\widehat{\mathbf{D}}_{\hat{c}})^{-1/2}\|_F = 1$  in (A.17), and that, by Assumption **A1**,  $\|(\mathbf{D}_c)^{-1/2}\|_F^2 \leq (\min L_m)^{-1} \leq M/(\underline{c}L)$ . Also,  $\|\widehat{\mathbf{C}} - \mathbf{C}\|_F^2 \leq 2L R_{BL}$ , and

$$\begin{aligned} \|\mathbf{I}_M - (\widehat{\mathbf{D}}_{\hat{c}})^{1/2}(\mathbf{D}_c)^{-1/2}\|_F^2 &= \text{Tr}(\mathbf{I}_M + \widehat{\mathbf{D}}_{\hat{c}}\mathbf{D}_c^{-1} - 2(\widehat{\mathbf{D}}_{\hat{c}})^{1/2}(\mathbf{D}_c)^{-1/2}) \\ &= \sum_{m=1}^M \frac{(\widehat{L}_m^{1/2} - L_m^{1/2})^2}{L_m} \leq \sum_{m=1}^M \frac{|\widehat{L}_m - L_m|}{L_m} \leq \frac{M}{\underline{c}L} \sum_{m=1}^M |\widehat{L}_m - L_m|, \end{aligned}$$

due to Assumption **A1**, and  $(\sqrt{a} - \sqrt{b})^2 \leq |a - b|$  for any  $a, b > 0$ . Since  $\sum |\widehat{L}_m - L_m|$  is dominated by the number of clustering errors  $L R_{BL}$ , plugging all components into (A.17), obtain (A.4).

**Proof of Lemma 4** Let  $\mathbf{X}^{(l)} = \mathbf{A}^{(l)} - \mathbf{P}^{(l)}$ ,  $l = 1, \dots, \tilde{L}$ . With some abuse of notations, for any square matrix  $\mathbf{Q}$ , let  $\text{diag}(\mathbf{Q})$  be the diagonal matrix which diagonal entries are equal to the diagonal entries of  $\mathbf{Q}$ , while for any vector  $\mathbf{q}$ , let  $\text{diag}(\mathbf{q})$  be the diagonal matrix with the vector  $\mathbf{q}$  on the diagonal. Then,  $\widehat{\mathbf{G}} - \mathbf{G} = \mathbf{S}_1 + \mathbf{S}_2 + \mathbf{S}_3$  where

$$\begin{aligned} \mathbf{S}_1 &= \sum_{l=1}^{\tilde{L}} (\mathbf{P}^{(l)}\mathbf{X}^{(l)} + \mathbf{X}^{(l)}\mathbf{P}^{(l)}), \quad \mathbf{S}_2 = \sum_{l=1}^{\tilde{L}} [(\mathbf{X}^{(l)})^2 - \text{diag}((\mathbf{X}^{(l)})^2)], \\ \mathbf{S}_3 &= \sum_{l=1}^{\tilde{L}} [\text{diag}((\mathbf{X}^{(l)})^2) - \text{diag}(\mathbf{A}^{(l)}\mathbf{1})] \end{aligned}$$

Therefore,  $\|\widehat{\mathbf{G}} - \mathbf{G}\|^2 \leq 3(\|\mathbf{S}_1\|^2 + \|\mathbf{S}_2\|^2 + \|\mathbf{S}_3\|^2)$ .

To bound above  $\|\mathbf{S}_1\|^2$ ,  $\|\mathbf{S}_2\|^2$  and  $\|\mathbf{S}_3\|^2$ , apply Theorems 2 and 3 of Lei and Lin (2022) with  $v_1 = v_2 = 2\bar{c}_\rho\rho_n$ ,  $R_1 = R_2 = R'_2 = 1$  and  $v'_2 = 2\bar{c}_\rho^2\rho_n^2$ . Using Theorems 2 with  $m = r = n$  and  $t^2 = \tau\bar{c}_\rho^2C_\rho\rho_n^3n^3\tilde{L}\log n$ , obtain

$$\mathbb{P}\left\{\|\mathbf{S}_1\|^2 \leq \tilde{C}\rho_n^3n^3\tilde{L}\log n\right\} \geq 1 - 4n^{1-\tau}$$

The first part of Theorem 3 yields that, due to Assumption **A3**,

$$\mathbb{P}\left\{\|\mathbf{S}_2\|^2 \leq \tilde{C}\rho_n^2n^2\tilde{L}\log^2(n + \tilde{L})\right\} \geq 1 - C(n + \tilde{L})^{1-\tau}$$

Now,  $\|\mathbf{S}_3\| \leq \|\mathbf{S}_3 - \mathbb{E}(\mathbf{S}_3)\| + \max_i |(\mathbb{E}\mathbf{S}_3)(i, i)|$ , since  $\mathbf{S}_3$  is a diagonal matrix. Applying second part of Theorem 3 with  $\sigma_2 = 1$  and  $\sigma'_2 = \sqrt{\tilde{L}n}$ , obtain

$$\mathbb{P}\left\{\|\mathbf{S}_3 - \mathbb{E}(\mathbf{S}_3)\|^2 \leq \tilde{C}\rho_n n \tilde{L} \log^2(n + \tilde{L})\right\} \geq 1 - C(n + \tilde{L})^{1-\tau}$$

Finally,

$$|(\mathbb{E}\mathbf{S}_3)(i, i)| = \left| \sum_{l=1}^{\tilde{L}} \left[ \mathbb{E} \sum_{j=1}^n [\mathbf{X}^{(l)}(i, j)]^2 - \sum_{j=1}^n \mathbf{P}^{(l)}(i, j) \right] \right| = \sum_{l=1}^{\tilde{L}} \sum_{j=1}^n [\mathbf{P}^{(l)}(i, j)]^2 \leq \rho_n^2 n \tilde{L},$$

which completes the proof.

**APPENDIX B: SIMULATIONS FOR COMPARISON OF DIMPLE WITH  
MMLSBM**

As we have previously mentioned, in this paper we consider the DIMPLE model, which is a more general model than the MMLSBM. Specifically, the MMLSBM has only  $M$  types of layers in the tensor and, therefore, results in a low rank tensor. On the other hand, all tensor layers in the DIMPLE model can be different and, therefore, the tensor is not of low rank. In this section, we carry out a limited simulation study, the purpose of which is to convince a reader that, while our algorithms work in the case of the MMLSBM, the algorithms designed for the MMLSBM produce poor results when data are generated according to the DIMPLE models.

In particular, in both scenarios, we first fix  $n, L, M, K$  and generate  $M$  groups of layers using the multinomial distribution with equal probabilities  $1/M$ . Similarly, we generate  $K$  communities in each of the groups of layers using the multinomial distribution with equal probabilities  $1/K$ . In this manner, we obtain community assignment matrices  $\mathbf{Z}^{(m)}$ ,  $m = 1, \dots, M$ , in each layer  $l$  with  $c(l) = m$ , where  $c : [L] \rightarrow [M]$  is the layer assignment function. Next, we choose sparsity parameters  $c$  and  $d$  and assortativity parameter  $w$ .

In order to generate data according to the DIMPLE model, we obtain the entries of  $\mathbf{B}^{(l)}$ ,  $l = 1, \dots, L$ , as uniform random numbers between  $c$  and  $d$ , and then multiply all the non-diagonal entries of those matrices by  $w$ . Therefore, if  $w < 1$  is small, then the network is strongly assortative, i.e., there is higher probability for nodes in the same community to connect.

The next four figures present simulation results for  $K = 5, M = 3$  and various values of  $L, n, c, d$  and  $w$ . We present only the between layer clustering errors since, in the presence of the assortativity assumption, the within-layer clustering in the MMLSBM and the DIMPLE model can be carried out in a similar way. We compare the performances of Algorithm 1 in this paper with the Alternative Minimization Algorithm (ALMA) of Fan et al. (2021).

As our simulations show, when data are generated according to the DIMPLE model, Algorithm 1 in our paper allows to reliably separate layers of the network into  $M$  types, while ALMA fails to

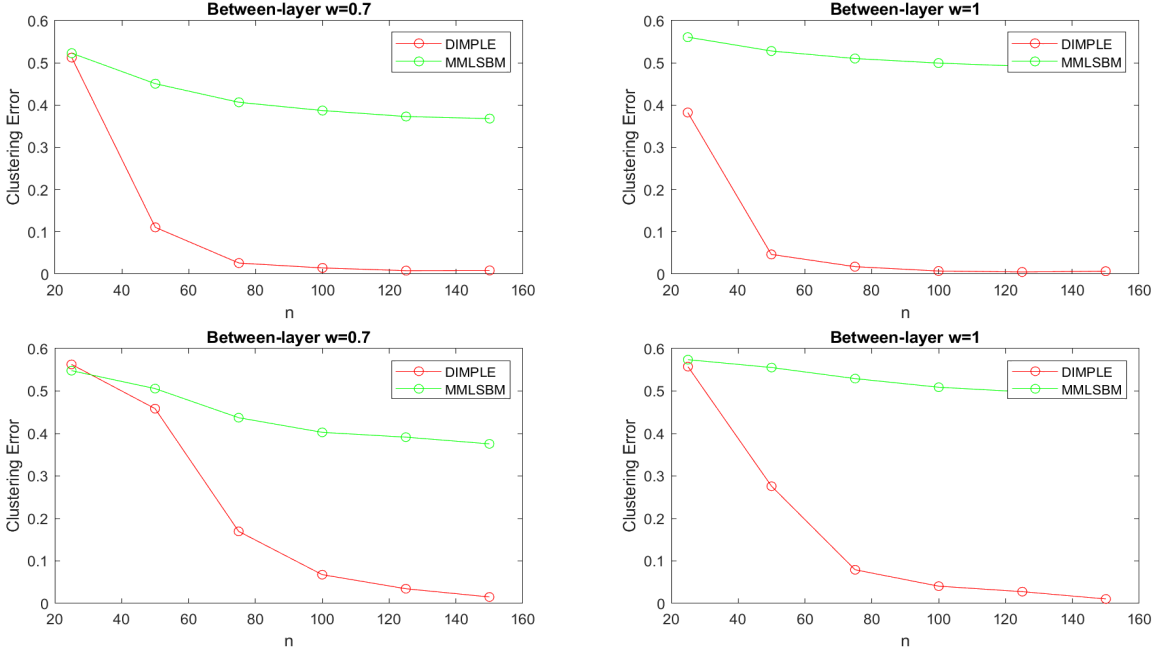


Figure B.1: The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using DIMPLE model with  $L = 50$ ,  $c = 0$ ,  $d = 0.8$  (top) and  $c = 0$ ,  $d = 0.5$  (bottom), and  $w = 0.7$  (left panel) or  $w = 1$  (right panel).

do so. The reason for this is that ALMA expects the matrices of probabilities to be identical in those layers, although, in reality, they are not. As a result, when  $n$  grows, the clustering errors do not tend to zero but just flatten.

Next, we generate data according to the MMLSBM. Note that the main difference between the MMLSBM and the DIMPLE model is that in MMLSBM one has only  $M$  distinct matrices  $\mathbf{B}^{(l)}$ , since  $\mathbf{B}^{(l)} = \mathbf{B}^{(c^{(l)})}$ ,  $l = 1, \dots, L$ . So, in order to generate MMLSBM, we generate  $M$  matrices  $\mathbf{B}^{(m)}$ ,  $m = 1, \dots, M$ , and then set  $\mathbf{B}^{(l)} = \mathbf{B}^{(c^{(l)})}$ ,  $l = 1, \dots, L$ . Figures B.1–B.4 exhibit results of application of Algorithm 1 and ALMA of Fan et al. (2021) to the generated data sets. As it is expected, for small values of  $n$ , ALMA of Fan et al. (2021) leads to a better clustering precision. The latter is due to the fact that Algorithm 1 relies on the SVDs of the layers of the adjacency tensor  $\mathcal{A}$ , that are not reliable for small values of  $n$ . In addition, Algorithm 1 cannot take into

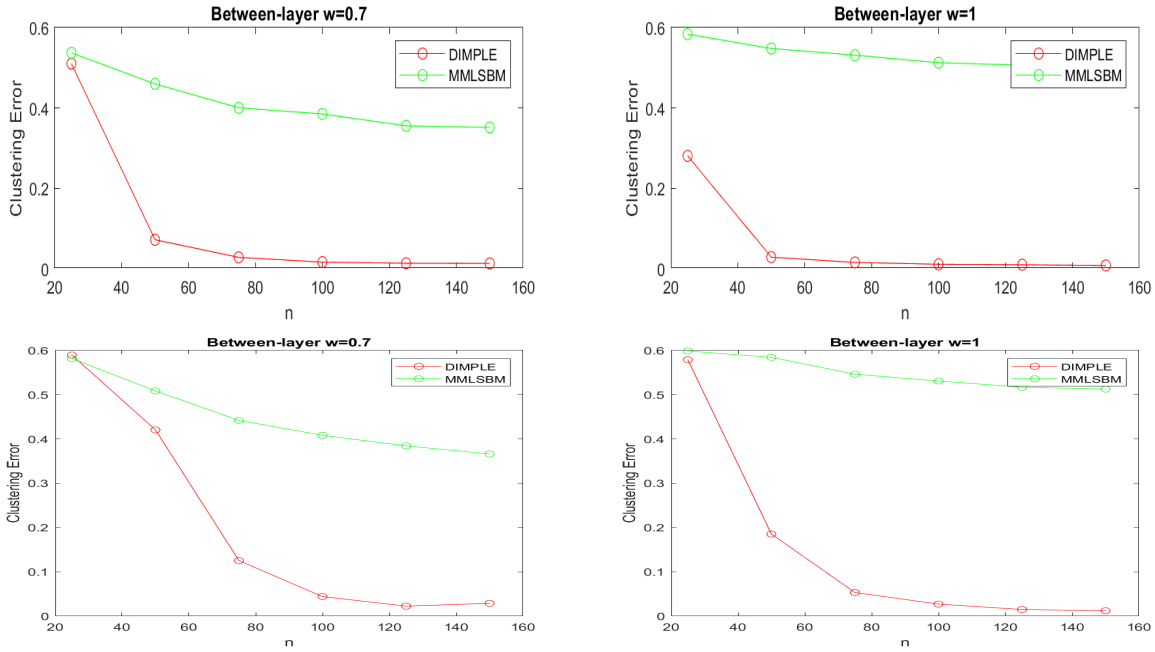


Figure B.2: The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using DIMPLE model with  $L = 100$ ,  $c = 0$ ,  $d = 0.8$  (top) and  $c = 0$ ,  $d = 0.5$  (bottom),  $n = 20, 40, 60, 80, 100, 120, 140, 160$  and  $w = 0.7$  (left panel) or  $w = 1$  (right panel).

account that the probability tensor is of a low rank since this is not true for the DIMPLE model. However, these advantages become less and less significant as  $n$  grows. As Figures B.1–B.4 show, both algorithms have similar clustering precision for larger values of  $n$ , specifically, for  $n \geq n_0$ , where  $n_0$  is between 60 and 100, depending on a particular simulations setting.

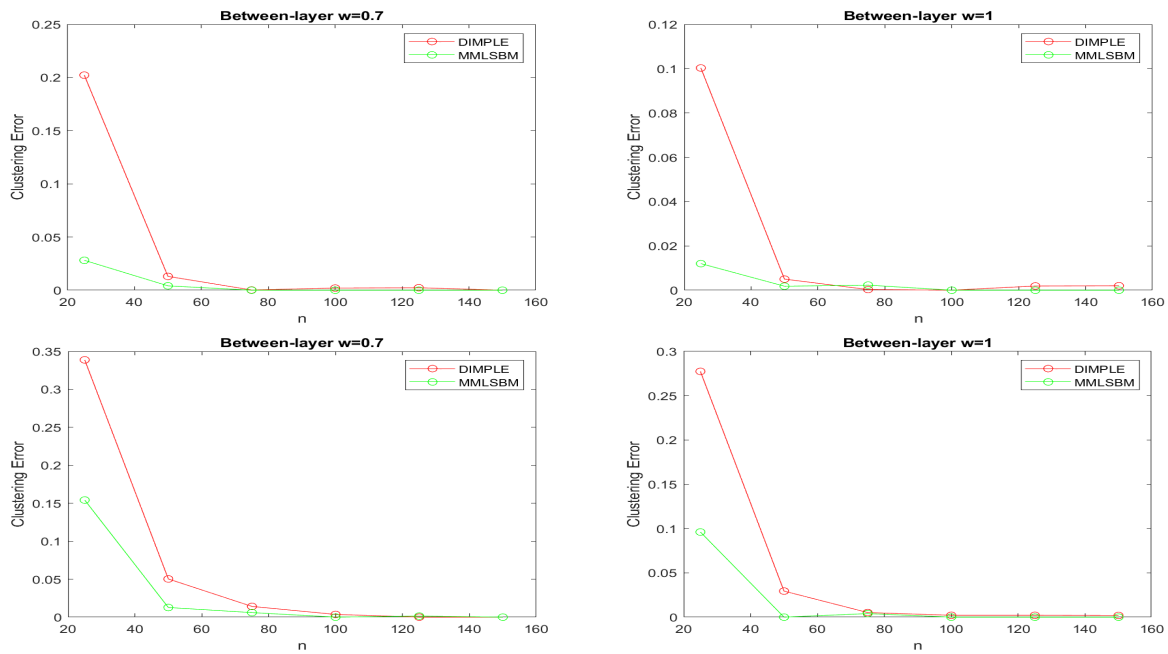


Figure B.3: The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using MMLSMBM with  $L = 50$ ,  $c = 0$ ,  $d = 0.8$  (top) and  $c = 0$ ,  $d = 0.5$  (bottom),  $n = 20, 40, 60, 80, 100, 120, 140, 160$  and  $w = 0.7$  (left panel) or  $w = 1$  (right panel).

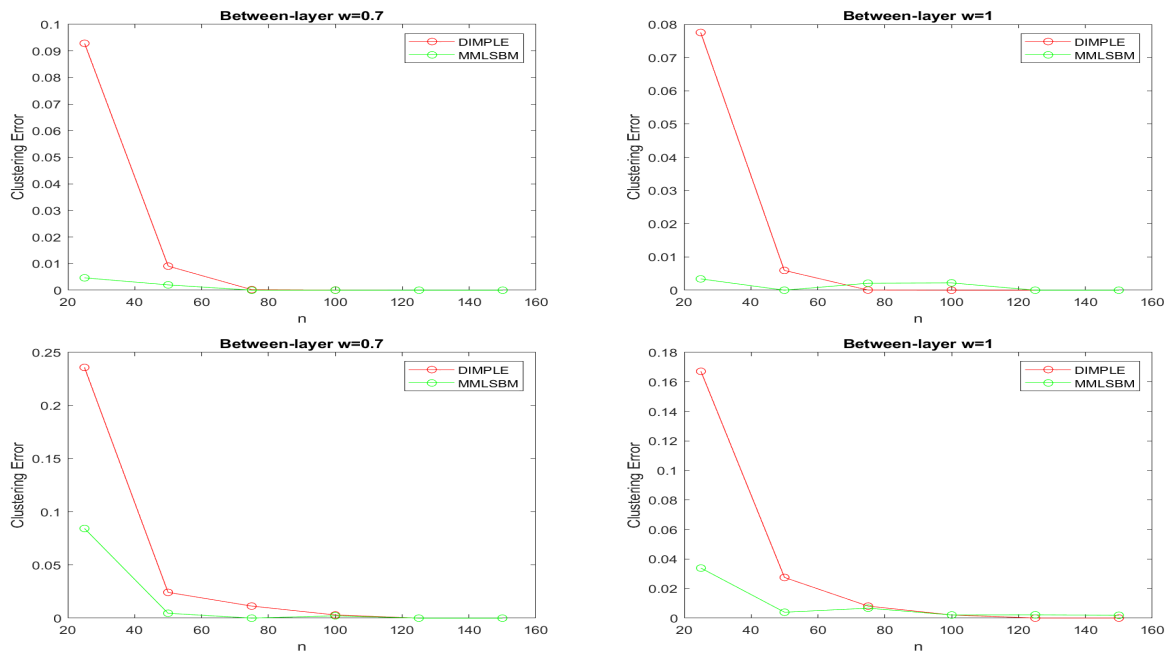


Figure B.4: The between-layer clustering error rates of Algorithm 1 and Alternative Minimization Algorithm of Fan et al. (2021). Data are generated using MMLSBM with  $L = 100$ ,  $c = 0$ ,  $d = 0.8$  (top) and  $c = 0$ ,  $d = 0.5$  (bottom), and  $w = 0.7$  (left panel) or  $w = 1$  (right panel).



## LIST OF REFERENCES

- A. Alves, L. G., Mangioni, G., Rodrigues, F. A., Panzarasa, P., and Moreno, Y. (2018), “Unfolding the Complexity of the Global Value Chain: Strength and Entropy in the Single-Layer, Multiplex, and Multi-Layer International Trade Networks,” *Entropy*, 20.
- Abbe, E. (2017), “Community detection and stochastic block models: recent developments,” *The Journal of Machine Learning Research*, 18, 6446–6531.
- Abbe, E., Bandeira, A. S., and Hall, G. (2015), “Exact recovery in the stochastic block model,” *IEEE Transactions on information theory*, 62, 471–487.
- Abbe, E. and Sandon, C. (2015), “Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap,” *ArXiv*, abs/1512.09080.
- (2018), “Proof of the Achievability Conjectures for the General Stochastic Block Model,” *Communications on Pure and Applied Mathematics*, 71.
- Aleta, A. and Moreno, Y. (2019), “Multilayer Networks in a Nutshell,” *Annual Review of Condensed Matter Physics*, 10, 45–62.
- Amini, A. A., Chen, A., Bickel, P. J., and Levina, E. (2013), “Pseudo-likelihood methods for community detection in large sparse networks,” *The Annals of Statistics*, 41, 2097–2122.
- Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C. E., and Vogelstein, J. T. (2021), “Inference for Multiple Heterogeneous Networks with a Common Invariant Subspace,” *Journal of Machine Learning Research*, 22, 1–49.
- Athreya, A., Fishkind, D. E., Tang, M., Priebe, C. E., Park, Y., Vogelstein, J. T., Levin, K., Lyzinski,

- V., Qin, Y., and Sussman, D. L. (2018), “Statistical Inference on Random Dot Product Graphs: a Survey,” *Journal of Machine Learning Research*, 18, 1–92.
- Athreya, A., Priebe, C. E., Tang, M., Lyzinski, V., Marchette, D. J., and Sussman, D. L. (2016), “A limit theorem for scaled eigenvectors of random dot product graphs,” *Sankhya A*, 78, 1–18.
- Baek, M. and Kim, C. (2021), “A review on spectral clustering and stochastic block models,” *Journal of the Korean Statistical Society*, 50, 818–831.
- Banks, J., Moore, C., Neeman, J., and Netrapalli, P. (2016), “Information-theoretic thresholds for community detection in sparse networks,” in *Conference on Learning Theory*, PMLR, pp. 383–416.
- Bassett, D. S., Zurn, P., and Gold, J. I. (2018), “On the nature and use of models in network neuroscience,” *Nature Reviews Neuroscience*, 19, 566–578.
- Bates, S., Angeon, V., and Ainouche, A. (2014), “The Pentagon of Vulnerability and Resilience: A Methodological Proposal in Development Economics by Using Graph Theory,” *Economic Modelling*, 42, 445–453.
- Bhattacharyya, S. and Chatterjee, S. (2020), “General Community Detection with Optimal Recovery Conditions for Multi-relational Sparse Networks with Dependent Layers,” *ArXiv:2004.03480*.
- Borondo, J., Morales, A. J., Benito, R. M., and Losada, J. C. (2015), “Multiple Leaders on A Multilayer Social Media,” *Chaos Solitons & Fractals*, 72, 90–98.
- Brodka, P., Chmiel, A., Magnani, M., and Ragozini, G. (2018), “Quantifying layer similarity in multiplex networks: a systematic study,” *Royal Society Open Science*, 5, 171747.
- Buckner, R. L. and DiNicola, L. M. (2019), “The brains default network: updated anatomy, physiology and evolving insights,” *Nature Reviews Neuroscience*, 1–16.

- Cai, T. T. and Zhang, A. (2018a), “Rate-optimal Perturbation Bounds for Singular Subspaces with Applications to High-dimensional Statistics,” *The Annals of Statistics*, 46, 60–89.
- (2018b), “Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics,” *Ann. Statist.*, 46, 60–89.
- Cardillo, A., Zanin, M., Gómez-Gardeñes, J., Romance, M., del Amo, A. J. G., and Boccaletti, S. (2013), “Modeling the Multi-layer Nature of the European Air Transport Network: Resilience and Passengers Re-scheduling under Random Failures,” *The European Physical Journal Special Topics*, 215, 23–33.
- Chaudhuri, K., Chung, F., and Tsias, A. (2012), “Spectral clustering of graphs with general degrees in the extended planted partition model,” in *Conference on Learning Theory, JMLR Workshop and Conference Proceedings*, pp. 35–1.
- Chen, X., Zhang, H., Gao, Y., Wee, C.-Y., Li, G., Shen, D., and the Alzheimer’s Disease Neuroimaging Initiative (2016), “High-order resting-state functional connectivity network for MCI classification,” *Human Brain Mapping*, 37, 3282–3296.
- Chi, E. C., Gaines, B. J., Sun, W. W., Zhou, H., and Yang, J. (2020), “Provable Convex Co-clustering of Tensors,” *Journal of Machine Learning Research*, 21, 1–58.
- Chin, P., Rao, A., and Vu, V. (2015), “Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery,” in *Conference on Learning Theory*, PMLR, pp. 391–423.
- Cimini, G., Squartini, T., Saracco, F., Garlaschelli, D., Gabrielli, A., and Caldarelli, G. (2019), “The Statistical Physics of Real-World Networks,” *Nature Reviews Physics*, 1, 58–71.
- Criado, R., Hernández-Bermejo, B., and Romance, M. (2007), “Efficiency, Vulnerability and Cost:

- an Overview with Applications to Subway Networks Worldwide,” *International Journal of Bifurcation and Chaos*, 17, 2289–2301.
- Crossley, N. A., Mechelli, A., Vértes, P. E., Winton-Brown, T. T., Patel, A. X., Ginestet, C. E., McGuire, P., and Bullmore, E. T. (2013), “Cognitive relevance of the community structure of the human brain functional coactivation network,” *Proceedings of the National Academy of Sciences*, 110, 11583–11588.
- De Domenico, M., Nicosia, V., Arenas, A., and Latora, V. (2015), “Structural reducibility of multilayer networks,” *Nature Communications*, 6.
- Decelle, A., Krzakala, F., Moore, C., and Zdeborová, L. (2011), “Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications,” *Physical Review E*, 84, 066106.
- Deshpande, Y., Abbe, E., and Montanari, A. (2015), “Asymptotic Mutual Information for the Two-Groups Stochastic Block Model,” *ArXiv*, abs/1507.08685.
- Durante, D., Mukherjee, N., and Steorts, R. C. (2017), “Bayesian Learning of Dynamic Multilayer Networks,” *Journal of Machine Learning Research*, 18, 1–29.
- Fan, X., Pensky, M., Yu, F., and Zhang, T. (2021), “ALMA: Alternating Minimization Algorithm for Clustering Mixture Multilayer Network,” *arXiv:2102.10226*.
- Faskowitz, J., Yan, X., Zuo, X.-N., and Sporns, O. (2018), “Weighted Stochastic Block Models of the Human Connectome across the Life Span,” *Scientific Reports*, 8, 12997.
- Finn, K. R., Silk, M. J., Porter, M. A., and Pinter-Wollman, N. (2019), “The Use of Multilayer Network Analysis in Animal Behaviour,” *Animal Behaviour*, 149, 7–22.
- Funke, T. and Becker, T. (2019), “Stochastic block models: A comparison of variants and inference methods,” *PloS one*, 14, e0215296.

- Gao, C., Ma, Z., Zhang, A. Y., and Zhou, H. H. (2017), “Achieving Optimal Misclassification Proportion in Stochastic Block Models,” *J. Mach. Learn. Res.*, 18, 1980–2024.
- (2018), “Community Detection in Degree-corrected Block Models,” *The Annals of Statistics*, 46, 2153–2185.
- Gulikers, L., Lelarge, M., and Massoulié, L. (2017), “A spectral method for community detection in moderately sparse degree-corrected stochastic block models,” *Advances in Applied Probability*, 49, 686–721.
- Han, R., Luo, Y., Wang, M., and Zhang, A. R. (2021), “Exact Clustering in Tensor Block Model: Statistical Optimality and Computational Limit,” *ArXiv:2012.09996*.
- Han, S. and Dunson, D. B. (2018), “Multiresolution Tensor Decomposition for Multiple Spatial Passing Networks,” *ArXiv:1803.01203*.
- Hernandez Nopsa, J. F., Daglish, G. J., Hagstrum, D. W., Leslie, J. F., Phillips, T. W., Scoglio, C., Thomas-Sharma, S., Walter, G. H., and Garrett, K. A. (2015), “Ecological Networks in Stored Grain: Key Postharvest Nodes for Emerging Pests, Pathogens, and Mycotoxins,” *BioScience*, 65, 985–1002.
- Jing, B., Li, T., Lyu, Z., and Xia, D. (2021), “Community detection on mixture multilayer networks via regularized tensor decomposition,” *The Annals of Statistics*, 49, 3181 – 3205.
- Jones, A. and Rubin-Delanchy, P. (2020), “The multilayer random dot product graph,” .
- Kao, T.-C. and Porter, M. A. (2017), “Layer Communities in Multiplex Networks,” *Journal of Statistical Physics*, 173, 1286–1302.
- Karrer, B. and Newman, M. E. J. (2011), “Stochastic Blockmodels and Community Structure in Networks,” *Physical Review E*, 83, 016107.

- Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014), “Multilayer Networks,” *Journal of Complex Networks*, 2, 203–271.
- Kolda, T. G. and Bader, B. W. (2009), “Tensor Decompositions and Applications,” *SIAM REVIEW*, 51, 455–500.
- Kumar, A., Sabharwal, Y., and Sen, S. (2004), “A Simple Linear Time  $(1 + \epsilon)$ -approximation Algorithm for K-means Clustering in any Dimensions,” in *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 454–462.
- Le, C. M. and Levina, E. (2015), “Estimating the number of communities in networks by spectral methods,” *ArXiv:1507.00827*.
- Lee, C. and Wilkinson, D. J. (2019), “A review of stochastic block models and extensions for graph clustering,” *Applied Network Science*, 4, 1–50.
- Lei, J., Chen, K., and Lynch, B. (2019), “Consistent community detection in multi-layer network data,” *Biometrika*, 107, 61–73.
- Lei, J. and Lin, K. Z. (2022), “Bias-Adjusted Spectral Clustering in Multi-Layer Stochastic Block Models,” *Journal of the American Statistical Association*, 0, 1–13.
- Lei, J. and Rinaldo, A. (2015), “Consistency of spectral clustering in stochastic block models,” *Ann. Statist.*, 43, 215–237.
- Luo, Y., Raskutti, G., Yuan, M., and Zhang, A. R. (2021), “A Sharp Blockwise Tensor Perturbation Bound for Orthogonal Iteration,” *Journal of Machine Learning Research*, 22, 1–48.
- MacDonald, P. W., Levina, E., and Zhu, J. (2021), “Latent space models for multiplex networks with shared structure,” *ArXiv:2012.14409*.

- Mason, O. and Verwoerd, M. (2007), “Graph Theory and Networks in Biology,” *IET systems biology*, 1, 89–119.
- Massoulié, L. (2014), “Community detection thresholds and the weak Ramanujan property,” in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 694–703.
- McSherry, F. (2001), “Spectral partitioning of random graphs,” in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 529–537.
- Mercado, P., Gautier, A., Tudisco, F., and Hein, M. (2018), “The Power Mean Laplacian for Multilayer Graph Clustering,” *ArXiv:1803.00491*.
- Mossel, E., Neeman, J., and Sly, A. (2014), “Belief propagation, robust reconstruction and optimal recovery of block models,” in *Conference on Learning Theory*, PMLR, pp. 356–370.
- (2015), “Reconstruction and estimation in the planted partition model,” *Probability Theory and Related Fields*, 162, 431–461.
- Mossel, E. and Xu, J. (2016), “Density evolution in the degree-correlated stochastic block model,” in *Conference on Learning Theory*, PMLR, pp. 1319–1356.
- Munsell, B., Wee, C.-Y., Keller, S., Weber, B., Elger, C., da Silva, L., Nesland, T., Styner, M., Shen, D., and Bonilha, L. (2015), “Evaluation of machine learning algorithms for treatment outcome prediction in patients with epilepsy based on structural connectome data,” *NeuroImage*, 118, 219–230.
- Nicolini, C., Bordier, C., and Bifone, A. (2017), “Community detection in weighted brain connectivity networks beyond the resolution limit,” *Neuroimage*, 146, 28–39.
- Nielsen, A. M. and Witten, D. (2018), “The Multiple Random Dot Product Graph Model,” .

- Olhede, S. C. and Wolfe, P. J. (2014), “Network histograms and universality of blockmodel approximation,” *Proceedings of the National Academy of Sciences*, 111, 14722–14727.
- Oro, E., Pizzuti, C., Procopio, N., and Ruffolo, M. (2018), “Detecting Topic Authoritative Social Media Users: A Multilayer Network Approach,” *IEEE Transactions on Multimedia*, 20, 1195–1208.
- Paul, S. and Chen, Y. (2016), “Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel,” *Electron. J. Statist.*, 10, 3807–3870.
- (2020), “Spectral and matrix factorization methods for consistent community detection in multi-layer networks,” *Ann. Statist.*, 48, 230–250.
- Perry, A. and Wein, A. S. (2017), “A semidefinite program for unbalanced multisection in the stochastic block model,” in *2017 International Conference on Sampling Theory and Applications (SampTA)*, IEEE, pp. 64–67.
- Qin, T. and Rohe, K. (2013), “Regularized spectral clustering under the degree-corrected stochastic blockmodel,” *Advances in neural information processing systems*, 26.
- Rao, C. and Rao, M. (1998), *Matrix Algebra and its Applications to Statistics and Econometrics*, World Scientific Publishing Co., 1st ed.
- Rohe, K., Chatterjee, S., and Yu, B. (2011), “Spectral clustering and the high-dimensional stochastic blockmodel,” *Ann. Statist.*, 39, 1878–1915.
- Rubin-Delanchy, P., Cape, J., Tang, M., and Priebe, C. E. (2022), “A statistical interpretation of spectral embedding: The generalised random dot product graph,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84, 1446–1473.
- Sporns, O. (2018), “Graph theory methods: applications in brain networks,” *Dialogues in Clinical Neuroscience*, 20, 111–121.



- Stam, C. J. (2014), “Modern network science of neurological disorders,” *Nature Reviews Neuroscience*, 15, 683–695.
- Stanley, N., Bonacci, T., Kwitt, R., Niethammer, M., and Mucha, P. J. (2019), “Stochastic block models with multiple continuous attributes,” *Applied Network Science*, 4, 54.
- Stanley, N., Shai, S., Taylor, D., and Mucha, P. J. (2016), “Clustering Network Layers with the Strata Multilayer Stochastic Block Model,” *IEEE Transactions on Network Science and Engineering*, 3, 95–105.
- Su, L., Wang, W., and Zhang, Y. (2019), “Strong consistency of spectral clustering for stochastic block models,” *IEEE Transactions on Information Theory*, 66, 324–338.
- von Luxburg, U. (2007), “A tutorial on spectral clustering,” *Statistics and Computing*, 17, 395–416.
- Wang, D. and Zou, X. (2017), “Control Energy And Controllability Of Multilayer Networks,” *Advances in Complex Systems (ACS)*, 20, 1–25.
- Wang, M. and Zeng, Y. (2019), “Multiway Clustering via Tensor Block Models,” in *Advances in Neural Information Processing Systems*, eds. Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F., Fox, E., and Garnett, R., Curran Associates, Inc., vol. 32, p. 715–725.
- Wang, S., Arroyo, J., Vogelstein, J. T., and Priebe, C. E. (2019), “Joint embedding of graphs,” *IEEE transactions on pattern analysis and machine intelligence*, 43, 1324–1336.
- Xie, F. and Xu, Y. (2021), “Efficient Estimation for Random Dot Product Graphs via a One-Step Procedure,” *Journal of the American Statistical Association*, 0, 1–14.
- Yun, S.-Y. and Proutiere, A. (2016), “Optimal cluster recovery in the labeled stochastic block model,” *Advances in Neural Information Processing Systems*, 29.

- Zhang, A. and Xia, D. (2018a), “Tensor SVD: Statistical and Computational Limits,” *IEEE Transactions on Information Theory*, 64, 7311–7338.
- (2018b), “Tensor SVD: Statistical and Computational Limits,” *IEEE Transactions on Information Theory*, 64, 7311–7338.
- Zhang, T., Szlam, A., Wang, Y., and Lerman, G. (2012), “Hybrid Linear Modeling via Local Best-Fit Flats,” *International Journal of Computer Vision*, 100, 217–240.
- Zhao, Y., Levina, E., and Zhu, J. (2012), “Consistency of Community Detection in Networks under Degree-corrected Stochastic Block Models,” *The Annals of Statistics*, 40, 2266–2292.
- Zheng, R. and Tang, M. (2022), “Limit results for distributed estimation of invariant subspaces in multiple networks inference and PCA,” *ArXiv: 2206.04306*.
- Zhu, M. and Ghodsi, A. (2006), “Automatic dimensionality selection from the scree plot via the use of profile likelihood,” *Computational Statistics & Data Analysis*, 51, 918–930.