Mixture Degree-Corrected Stochastic Block Model for Multi-Group Community Detection in Multiplex Graphs

Anonymous authors
Paper under double-blind review

Abstract

Multiplex graphs have emerged as a powerful tool for modeling complex data structures due to their ability to handle multiple relational layers. Clustering within a multiplex graph can involve merging vertices into communities that are consistent across all layers, grouping similar layers into clusters, or creating overlapping clusters among vertices and layers. However, a multiplex graph may exhibit distinct vertex communities based on the specific layers to which a vertex is connected. This scenario, termed multi-group community detection, significantly enhances the accuracy of clustering processes and aids in the interpretation of partitions. To date, the current literature on state-of-the-art community detection has not extensively addressed this modeling approach. In this paper, we introduce a novel methodology referred to as the "Mixture Degree-Corrected Stochastic Block Model." This generative model, an extension of the widely utilized Degree-Corrected Stochastic Block Model (DCSBM), is designed to cluster similar layers by their community structures while simultaneously identifying communities within each layer's group. We provide a rigorous definition of the model and utilize an iterative technique to perform inference computations. Furthermore, we assess the identifiability of our proposed model and demonstrate the consistency of the maximum likelihood function through analytical analysis. The effectiveness of our method is evaluated using both real-word data sets and synthetic graphs.

1 Introduction

Recent technological advancements have precipitated an exponential increase in data accumulation, conducting to the era of big data. This new ecosystem presents novel challenges in exploring and analyzing extensive datasets, as highlighted in recent literature Elgendy & Elragal (2014). Often, data comes from multiple sources, presenting multiple perspectives from diverse sources and features. Such datasets require sophisticated models that capture intricate relationships and interdependencies across different data types and sources Devagiri et al. (2021); Niu et al. (2016). To address the increasing complexity, multiplex graphs have become a relevant asset Hammoud & Kramer (2020); Zweig (2016).

A multiplex graph consists of a set of vertices connected across multiple layers Han et al. (2023); Magnani et al. (2021). Each layer in a multiplex graph uniquely represents a set of edges that models specific similarities between vertices, thereby serving as an effective model for multi-relational data. Additionally, multiplex graphs are adept at modeling time-varying data, offering a robust framework for dynamic data analysis Xia et al. (2020).

Clustering has long been recognized as an effective means for understanding and exploring data across various domains by identifying groups of individuals with strong similarities Fortunato (2010a); Bedi & Sharma (2016). In multiplex graphs, the clustering process poses a significant challenge, yet offers promising solutions for analyzing complex data structures Wang et al. (2019). Community detection in multiplex graphs aims to identify groups characterized by high intra-connectivity and low inter-connectivity Fortunato (2010b). To this end, numerous algorithms employing diverse approaches such as optimization De Meo et al. (2011); Que et al. (2015), spectral computation Li et al. (2018), consensus clustering Mandaglio et al. (2018), and inference Shuo & Chai (2016) have been developed.

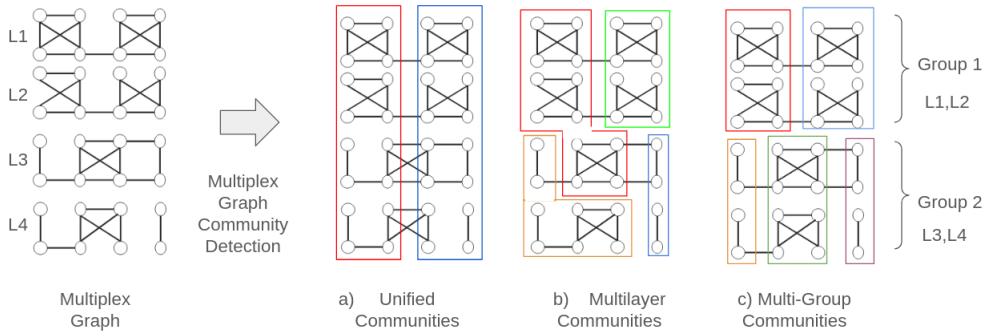


Figure 1: Presentation of a multiplex graph with 4 layers. The results of community detection algorithms on a multiplex graph are categorized into three distinct types of groupings. Each grouping is represented by color-coded boxes, with each color signifying a different community. In figure (a), unified communities are depicted, showing a single community structure shared across all layers. Figure (b) presents a visualization of overlapping communities, where communities extend over several but not all layers. Finally, figure (c) displays multi-group communities, in which similar layers are clustered together, and each cluster maintains a uniform community partition.

The results of these methods can generally be categorized into two groups. The first group optimizes communities to be identical and unified across all layers, which may overlook the inherent diversity within each layer, as presented in (a) from Figure 1. The second approach of multilayer communities allows for overlapping communities across layers and vertices, where vertices may belong to different communities in different layers, as presented in (b) from Figure 1. This model, while computationally intensive, also poses challenges in interpretation and constrains the re-computation of partitions when a new layer arrives, especially as the number of layers increases.

In this paper, we propose a novel approach for partitioning multiplex graphs, where layers are clustered into groups, and vertices within each group are organized into communities that remain consistent across all layers, as presented in the third clustering outcome in (c) from Figure 1. This multi-group perspective not only captures the intrinsic structure of each layer but also simplifies the interpretation of the results. Additionally, this method proves advantageous when integrating new layers post-partitioning, as it allows for the incorporation of these layers into existing partitions or the creation of new partitions if necessary.

Moreover, clustering multiplex layers into groups of similar networks has become an active research area, treating each layer as an individual within a population of networks Mantziou et al. (2023); van der Laan et al. (2022). However, previous approaches have often treated the clustering of layers and the partitioning of vertices independently, neglecting the structural differences in community configurations when layers are clustered. Our dual clustering approach thus aims to enhance both layer clustering and community detection within multiplex graphs, providing a comprehensive solution to the challenges posed by complex network structures.

Although multi-group community detection has not been widely addressed in the literature, its relevance is apparent in real-world datasets. For example, individuals sharing similar musical tastes may not necessarily align in their political or sports affiliations, that proving the structural diversity of layers. In the financial sector, multiplex graphs represent various transaction types within banking systems—such as online, wire, and ATM transactions. Grouping these transactions by similar dynamics and identifying communities within these groups are crucial for detecting anomalous behaviors and preventing fraud. Additionally, as we present in experiment section in the field of neuroscience, the study of brain connectivity through magnetic resonance imaging offers profound insights. Our experiments demonstrate that this approach can effectively group subjects with similar neurological diagnoses, illustrating the practical applications of multi-group community detection in medical research.

We introduce a new approach for Multi-group community detection based on the Stochastic Block Model (SBM), a generative model extensively developed for community detection, which recognizes the presence of communities within a graph by defining the probability of edges between vertices based on the communities to which they belong Lee & Wilkinson (2019). SBM has been applied to various graph types including single-layer Abbe (2017), multiplex Barbillon et al. (2017), and dynamic networks Corneli (2017). The Degree-Corrected Stochastic Block Model (DCSBM), an advancement of SBM, addresses the heterogeneity of vertex degrees within communities, relaxing the assumption of uniform degree distribution Karrer & Newman (2011). The previous work in Stanley et al. (2015) proposed a model for SBM with multiple strata, which bears some resemblance to multi-group community detection. However, their implementation faces significant limitations when applied to real-world scenarios. These limitations arise due to the strong assumption within the SBM model of a uniform distribution within communities, challenges in initialization, and difficulties in determining the optimal number of communities when used in practical applications.

In response, this paper introduces the Mixture DCSBM (MDCSBM), a novel model for multi-group partitioning of a multiplex graph. Our approach **groups similar layers and partitions the vertices of each layer within these groups into distinct blocks**, as depicted on the right of Figure 1. We hypothesize that layers within the same group share an identical DCSBM distribution. To facilitate this dual clustering, we employ the Expectation Maximization (EM) algorithm to determine the groupings of layers, and the Variational Expectation Maximization (VEM) technique to assign vertices to their respective blocks within each group. Throughout this document, we use the term "partitioning" to refer to the formation of vertex communities and "clustering" to describe the grouping of layers.

2 Related Works

Community detection in multiplex graphs can yield diverse types of partitions. For instance, some approaches employ spectral representation-based algorithms and tensor-based algorithms that fuse layers into a centroid graph. This graph is constrained to have K connected components, each representing a community, where K is the number of clusters aimed at capturing highly consistent communities Kang et al. (2019); Wang et al. (2019); Papalexakis et al. (2016). Other methods focus on consensus clustering, which involves computing communities for each layer individually and then identifying the most consistent community across all layers for multiplex community detection Berlingerio et al. (2013); Tagarelli et al. (2017); Tang et al. (2012). Additionally, algorithms originally designed for mono-layer community detection have been adapted for multiplex graphs, often allowing for overlapping between vertices and layers to form multilayer communities, such as Generalized Modularity and Multilayer Infomap Mucha et al. (2010); Afsarmanesh & Magnani (2016); De Domenico et al. (2015); Wilson et al. (2017).

Recent research has also focused on identifying clusters of similar layers that may exhibit the same structure van der Laan et al. (2022). In this approach, each graph is treated as an individual within a population of networks. Various algorithms aim to identify a set of similar layers using methods such as mixture parametric models Mantziou et al. (2023); Kemp et al. (2006), nonparametric methods derived from minimum code length descriptions Kirkley et al. (2023); Kirkley & Newman (2022), models based on graph distance measurements La Rosa et al. (2015), and latent space models Young et al. (2022).

However, these models, which aim to partition the graph into communities, often overlook the potential for multiple clusters within the multiplex graph and do not consider the similarity of communities between different layers. This dual consideration is crucial for enhancing the accuracy and comprehensibility of results, especially as the number of layers increases. In the filed of tabular data, the dual clustering of both features and samples—referred to as co-clustering—has been extensively explored and has proven beneficial for in-depth data analysis Ailem et al. (2015); Nadif & Govaert (2010). Thus, bridging this gap in graph-type data is crucially relevant.

The Stochastic Block Model (SBM) is a generative probabilistic model widely used for community detection in networks. Originally developed for single-layer networks, SBM has been extended to accommodate multi-layer clustering. A notable example is the introduction of the Multi-Layer Stochastic Block Model (MLSBM), which incorporates various types of layer aggregation Vallè s-Català et al. (2016). In contrast, a directed graph model with layers generated independently from the same distribution was explored in De Bacco et al.

(2017). Furthermore, Paul & Chen (2016) adapted SBM by assigning each layer its own affinity matrix, yet constrained these matrices to yield consistent communities across layers. Additional developments include the work of Amini et al. (2024); Roy et al. (2006), which focuses on a hierarchical generalization of SBM. This approach seeks to uncover underlying community structures by considering the hierarchical nature of real-world networks. Other adaptations of MLSBM utilize techniques such as Variational Expectation Maximization to infer multi-layer graph structures Barbillon et al. (2017); Corneli et al. (2016); Han et al. (2015). However, these models often grapple with the challenge of exponential parameter scaling, which complicates their application to real-world graphs. Moreover, they generally lack mechanisms for constraining the formation of multi-group structures within the multiplex graph, a critical aspect for capturing the complexity of such networks.

In this paper, we present a novel extension of the DCSBM model that jointly represents the affiliation of each layer to a group (set of layers), and vertices within each group are assigned to blocks (set of vertices). We employ the EM-VEM technique to infer the model's parameters and estimate the assignment variables effectively, offering an improved approach for community detection in multiplex graphs.

3 Mixture DCSBM

This work aims to achieve a dual clustering process by grouping similar layers into a group of layers, and vertices within each group are further clustered into a shared block of vertices. The estimation process of MDCSBM involves determining layer-to-group variables to identify the group of each layer. It is assumed that edges within each group follow a unique DCSBM distribution, treating layers within the same group as independently sampled from the same distribution. Therefore, vertex-to-block variables are estimated for each group to identify the block of each vertex within the same group.

3.1 Model Definition

Consider a multiplex graph denoted as $\mathcal{G} = \{G^1, ...G^L\}$ comprising L layer, $G^l = \{V, E^l\}$ represents a single layer, where l, s.t $l \in [1, L]$ indicates the layer index, V indicates the set of vertices with $|V| = N$, E^l the set of edges within layer l. Let $\mathcal{A} = \{A^1, ..., A^L\}$ be the corresponding adjacency matrix of multiplex graph $\mathcal{G}$, where A^l stands for the adjacency matrix of the graph G^l. The underlying graph model in this study is an unweighted and undirected multiplex graph, where the edge distributions follow a Bernoulli distribution. The generalization of this model to a directed graph is straightforward. Finally, an edge A^l_{ij} is defined by a dyad i, j, representing its extremity.

Let us consider partitioning multiplex's graph layers into K groups and assume that the vertices of group k are divided into Q^k blocks, where $k \in [1, K]$. The probability of having an edge $A^l_{i,j}$ in layer l, within group k, giving the block assigned to each vertex, under MDCSBM model is expressed as follows:

$$P(A^l_{i,j}|\mathbf{Z}^k, \mathbf{\Pi}^k, \mathbf{\Theta}^k) = \theta^k_i \theta^k_j \pi^k_{Z^k_i, Z^k_j} \tag{1}$$

where $\mathbf{Z}^k = \{Z^k_1, ...Z^k_N\}$ is the set of vertex-to-block assignments in the group k and $Z^k_i \in \{1, ..., Q^k\}$, $\mathbf{\Theta}^k = \{\theta^k_1, ..., \theta^k_N\}$ is the set of degree heterogeneity parameter for vertices in group k, s.t $\theta^k_i > 0$ and $\sum_{i, Z^k_i = q} \theta^k_i = 1 \forall i \in V, \forall q \in [1, Q^k]$. The matrix $\mathbf{\Pi}^k$ has $Q^k \times Q^k$ elements $\pi^k_{q,w}, \forall q, w \in \{1, ..., Q^k\}^2$. Each element represents the probability of an edge existing within the group k, depending on the block of its dyad $\{i, j\}$.

Let consider a given layer G^l with known vertex-to-block variable assignments for each group $\mathbf{Z} = \{\mathbf{Z}^1, ..., \mathbf{Z}^K\}$. The probability of existing edge A^l_{ij} between dyad (i, j) from the MDCSBM model, conditioned on $\mathbf{Z}$, can be described as a mixture distribution of K independent DCSBMs, expressed as:

$$
\begin{aligned}
&P(A^l_{ij} = 1|\mathbf{Z}; \boldsymbol{\beta}, \mathbf{\Pi}, \mathbf{\Theta}) = \sum_{k=1}^{K} \beta^k \theta^k_i \theta^k_j \pi^k_{Z^k_i, Z^k_j} \\
&s.t \sum_k \beta^k = 1, \\
&\sum_{i, Z^k_i = q} \theta^k_i = 1, \forall q \in [1, Q^k], \forall k \in [1, K]
\end{aligned} \tag{2}
$$

where $\boldsymbol{\beta} = \{\beta^1, ..., \beta^K\}$ denotes the set of probabilities of layer l to be generated from group k, representing the mixture weights of MDCSBM, $\mathbf{\Pi} = \{\mathbf{\Pi}^1, \mathbf{\Pi}^2, ..., \mathbf{\Pi}^K\}$ is the set of parameters of each group, and $\mathbf{\Theta} = \{\mathbf{\Theta}^1, \mathbf{\Theta}^2, ..., \mathbf{\Theta}^K\}$ is the set of degree heterogeneity for each group. This model incorporates K distributions from which layers can be generated.

To address the challenge of maximizing the log-likelihood function due to the sum in the mixture model, we introduce a set of latent variables $\mathbf{Y}$ that represents the layer-to-group assignment. Specifically, y_{lk}, s.t $l \in [1, L]$ and $k \in [1, K]$, takes the value of one when layer l is generated from group k and zero otherwise. The updated formulation for the probability of an existing edge is as follows:

$$
\begin{aligned}
P(A^l_{ij} = 1|\mathbf{Y}, \mathbf{Z}; \mathbf{\Pi}, \mathbf{\Theta}) &= \prod_{k=1}^{K} (\theta^k_i \theta^k_j \pi^k_{Z^k_i, Z^k_j})^{y_{lk}} \\
P(y_{lk} = 1; \beta) &= \prod_{k=1} (\beta^k)^{y_{lk}}
\end{aligned} \tag{3}
$$

Additionally, for any group k, we identify the probability of a vertex i to be assigned to block q as follows:

$$
\begin{aligned}
P(Z^k_i = q; \boldsymbol{\alpha}^k) &= \alpha^k_q \\
s.t \sum_{q=1}^{Q^k} \alpha^k_q &= 1
\end{aligned} \tag{4}
$$

such that $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, ..., \boldsymbol{\alpha}^K\}$ and $\boldsymbol{\alpha}^k = \{\alpha^k_1, \alpha^k_2, ..., \alpha^k_{Q_k}\}$.

Let us consider $\mathbf{\Delta} = \{\mathbf{\Delta}^1, \mathbf{\Delta}^2, ..., \mathbf{\Delta}^K\}$, such that $\mathbf{\Delta}^k = \{\mathbf{\Pi}^k, \mathbf{\Theta}^k, \boldsymbol{\alpha}^k\}$ be the aggregation of group parameters, the log-likelihood of the proposed model is written as follows:

$$
\mathcal{L}(\mathcal{A}, \mathbf{Y}, \mathbf{Z}; \boldsymbol{\beta}, \mathbf{\Delta}) = \sum_{l=1}^{L} \sum_{k=1}^{K} y_{lk} \Big[ln\beta^k + \mathcal{L}(A^l, \mathbf{Z}^k; \mathbf{\Delta}^k) \Big] \tag{5}
$$

where $\mathcal{L}(A^l, \mathbf{Z}^k; \mathbf{\Delta}^k)$ is the complete log-likelihood of layer l in group k with parameters $\mathbf{\Delta}^k$, formulated as follow:

$$
\begin{aligned}
\mathcal{L}(A^l, \mathbf{Z}^k; \mathbf{\Delta}^k) &= ln(P(A^l|\mathbf{Z}^k; \mathbf{\Pi}^k, \mathbf{\Theta}^k)) + ln(P(\mathbf{Z}^k; \boldsymbol{\alpha}^k)) \\
&= \sum_{i,j,i \neq j} A^l_{ij} ln(\pi^k_{Z_i Z_j}) + (1 - A^l_{ij}) ln(1 - \pi^k_{Z_i Z_j}) + \sum_{i,j,i \neq j} ln(\theta^k_i \theta^k_j) + \sum_{i=1} ln(\alpha^k_{Z_i})
\end{aligned} \tag{6}
$$

The verification of the parameter's identifiability and the assessment of the maximum likelihood consistency have been performed in supplementary materials.

In the context of inferring information from a given multiplex graph, the primary objectives involve assigning each layer to a specific group k using variable y_{lk}, then assigning each vertex i within group k to a particular block q using variable Z^k_i, and optimizing the parameters $\boldsymbol{\beta}$ and $\mathbf{\Delta}$.

4 Optimization of Log Likelihood Function

As explained previously, the MDCSBM depends on layer-to-group and vertex-to-block assignment variables. We set an iterative approach to address this joint assignment clustering challenge. To elaborate, we estimate layer-to-group assignment variables by utilizing the Expectation Maximization (EM) technique. Then, to infer the DCSBM model within each group, we use the Variational EM (VEM) technique. This technique proves its performance in maximizing DCSBM distribution parameters while estimating the latent vertex-to-block variables.

4.1 Estimation of Layer-to-Group Variables

The computation of layer-to-group latent variables estimation is derived from equation 5, which defines the complete log-likelihood. The estimation process involves calculating the expectation of the log-likelihood based on the posterior distribution of layer-to-group latent variables, and it can be expressed as follows:

$$E_{\mathbf{Y}}[\mathcal{L}(\mathcal{A}, \mathbf{Y}, \mathbf{Z}; \boldsymbol{\beta}, \boldsymbol{\Delta})] = \sum_{l=1}^{L} \sum_{k=1}^{K} E(y_{lk}) \Big[ln\beta^k + \mathcal{L}(A^l, \mathbf{Z^k}; \boldsymbol{\Delta}^k) \Big] \tag{7}$$

where $E(y_{lk})$ is the posterior expectation probability of layer l to be generated from group k, defined as $p(y_{lk}|A^l, \mathbf{Z}^k)$. Using Bayes theorem, the estimation of layer-to-group is computed as follows:

$$E(y_{lk}) = \frac{\beta^k P(A^l, \mathbf{Z}^k|\boldsymbol{\Delta}^k)}{\sum_j \beta^j P(A^l, \mathbf{Z}^j|\boldsymbol{\Delta}^j)} \tag{8}$$

where $P(A^l, \mathbf{Z}^k; \boldsymbol{\Delta}^k)$ is written as follows:

$$P(A^l, \mathbf{Z}^k; \boldsymbol{\Delta}^k) = P(A^l|\mathbf{Z}^k; \boldsymbol{\Pi}^k, \boldsymbol{\Theta}^k) P(\mathbf{Z^k}; \boldsymbol{\alpha}^{\mathbf{k}}) \tag{9}$$

The estimation of layer-to-group prioritizes the layer that maximizes the likelihood distribution for a specific group. In order to assign each layer to a single group, the selection of the group is based on the following:

$$y_{lk} = \underset{j}{\operatorname{argmax}}\, y_{lj} \tag{10}$$

4.2 Maximization of Likelihood Parameters and Vertex-to-Block Variables

Once the layer-to-group assignment is identified, the MDCSBM parameters can be maximized, and the vertex-to-block variable can be estimated too.

4.2.1 Maximization of β

Considering equation 5, the optimization of $\boldsymbol{\beta}$ involves expressing the complete log-likelihood as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{A}, \mathbf{Y}; \boldsymbol{\beta}, \boldsymbol{\Delta}) &= \sum_{l=1}^{L^k} ln\beta^k + C(\beta^k) \\ s.t \sum_{k=1}^{K} \beta^k &= 1 \end{aligned} \tag{11}$$

where $L^k = \{l \in [1, L], s.t \;\; y_{lk} = 1\}$, set of layer of group k, and $C(\beta^k)$ defined as a constant regarding on β^k. By employing the Lagrange multiplier approach, the solution that satisfies the Karush-Kuhn-Tucker (KKT) conditions can be expressed as follows:

$$\beta^k = \frac{N^k}{N} \tag{12}$$

where $N_k = |L^k|$ is the number of layers in the groups k.

4.2.2 Estimation of vertex-to-block and Maximization of Parameter Δ_k

Our mathematical models assume that the layers within the same group are generated independently from the same DCSBM distribution specific to that group. Let $\mathcal{A}^k$ the multiplex graph that contains the layers of group k, based on equation 6, the log-likelihood of group k is expressed as follows:

$$\begin{aligned} L(\mathcal{A}^k, \mathbf{Z}^k; \mathbf{\Delta}^k) = \sum_{l \in L^k} \sum_{i,j,i \neq j} ln(\theta_i^k \theta_j^k) + \sum_{i=1} ln(\alpha_{Z_i}^k) + \sum_{l \in L^k} \sum_{i,j,i \neq j} A_{ij}^l ln(\pi_{Z_i Z_j}^k) + (1 - A_{ij}^l) ln(1 - \pi_{Z_i Z_j}^k) \\ s.t \sum_{q}^{Q^k} \alpha_q^k = 1, \sum_{i, Z_i^k = q} \theta_i^k = 1 \end{aligned} \tag{13}$$

In order to optimize the parameters that maximize the previous equation, it is essential to first estimate the latent assignment variables. This task is addressed using the Expectation Maximization (EM) algorithm, which requires computing the posterior probability of the latent variable $\mathbf{Z}^k$ with respect to the observed layers, denoted as $P(\mathbf{Z}^k | \mathcal{A}^k)$. However, for single-layer graphs, it has been demonstrated that calculating this conditional probability is computationally intractable Celisse et al. (2012). Various approaches have been proposed in the literature to tackle this challenge Li et al. (2015); Lee & Wilkinson (2019), but they tend to suffer from the curse of dimensionality, mainly when dealing with large-scale datasets.

The Variational EM technique has been adopted to address this issue as an alternative technique for handling DCSBM estimation challenges. Previous studies have established the VEM technique's convergence for single-layer SBM and DCSBM models and multiplex SBM graphs Celisse et al. (2012); Barbillon et al. (2017). The VEM approach involves approximating the posterior distribution $P(\mathbf{Z}^k | \mathcal{A}^k)$, by another distribution $R_{\mathcal{A}^k}$ over Z^k. By leveraging this approximation, the marginal log-likelihood over $\mathbf{Z}^k$ can be expressed as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{A}^k; \mathbf{\Delta}^k) = \sum_{\mathbf{Z}^k} R_{\mathcal{A}^k}(\mathbf{Z^k}) \mathcal{L}(\mathcal{A}^k, \mathbf{Z}^k; \mathbf{\Delta}^k) - \sum_{\mathbf{Z}^k} R_{\mathcal{A}^k}(\mathbf{Z^k}) ln\Big(R_{\mathcal{A}^k}(\mathbf{Z^k})\Big) + \\ \mathbf{KL}\big[R_{\mathcal{A}^k}(\mathbf{Z^k}), P(\mathbf{Z}^k | \mathcal{A}^k; \mathbf{\Delta}^k)\big] \end{aligned} \tag{14}$$

where $\mathbf{KL}$ is the Kullback-Leibler divergence. Therefore, instead of maximizing $\mathcal{L}(\mathcal{A}^k; \theta^k)$ for the observed data, the VEM technique optimizes a lower bound of $\mathcal{L}(\mathcal{A}^k; \theta^k)$, denoted as $\mathcal{I}_\theta(R_{\mathcal{A}^k})$. This lower bound is known as the evidence lower bound, and it can be defined as follows:

$$\begin{aligned} \mathcal{I}_\theta(R_{\mathcal{A}^k}) &= \mathcal{L}(\mathcal{A}^k; \boldsymbol{\theta}^k) - \mathbf{KL}\big[R_{\mathcal{A}^k}(\mathbf{Z^k}), P(\mathbf{Z}^k | \mathcal{A}^k; \mathbf{\Delta}^k)\big] \\ &= \sum_{\mathbf{Z}^k} R_{\mathcal{A}^k}(\mathbf{Z^k}) \mathcal{L}(\mathcal{A}^k, \mathbf{Z}^k; \mathbf{\Delta}^k) - \sum_{\mathbf{Z}^k} R_{\mathcal{A}^k}(\mathbf{Z^k}) log R_{\mathcal{A}^k}(\mathbf{Z^k}) \\ &\leq \mathcal{L}(\mathcal{A}^k, \mathbf{Z}^k; \mathbf{\Delta}^k) \end{aligned} \tag{15}$$

The equality between the evidence lower bound and the log-likelihood holds when $R_{\mathcal{A}^k}(\mathbf{Z^k})$ is equal to the true posterior distribution $P(\mathbf{Z}^k | \mathcal{A}^k; \mathbf{\Delta}^k)$. Maximizing the lower bound $\mathcal{I}_{\boldsymbol{\theta}}(R_{\mathcal{A}^k})$ is equivalent to minimizing the Kullback-Leibler divergence $\mathbf{KL}\big[R_{\mathcal{A}^k}(\mathbf{Z^k}), P(\mathbf{Z}^k | \mathcal{A}^k; \mathbf{\Delta}^k)\big]$. Regarding to integer nature of vertex-to-block variables, to approximate the posterior distribution, we select $R_{\mathcal{A}^k}(\mathbf{Z^k})$ as follows:

$$R_{\mathcal{A}^k}(\mathbf{Z^k}) = \prod_{i=1}^{N} h(\mathbf{Z}_i^k; \tau_i^k) \tag{16}$$

where $h(:; \tau_i^k)$ is a multinomial distribution with parameters $\boldsymbol{\tau} = \{\tau_1^k, ...\tau_{Q^k}^k\}$. The entity τ_{iq}^k approximates the probability that vertex i belongs to the community q in group k. The $\mathcal{I}_\theta(R_{\mathcal{G}^k})$ can be written as follows:

Algorithm 1 Inference of Likelihood of MDCSBM

Input: $\mathcal{G}, K, \mathbf{Q} = [Q^1, ..., Q^K]$
Output: $\mathbf{Y}, \mathbf{Z}, \mathbf{\Pi}, \mathbf{\Theta}, \boldsymbol{\beta}, \boldsymbol{\alpha}$

```
Initialize Y, Z with 2
while Iteration < Iteration max ∧ Not Converge do
    Estimate y_lk with 8
    Compute y_lk with 10
    Compute α_q^k with 19
    Compute π_qw^k with 20
    Compute θ_i^k with 18
    Compute τ_qw^k with 21
end while
```

$$\mathcal{I}_\theta(R_{\mathcal{A}^k}) = \sum_{l \in L^k} \sum_{i \neq j} \sum_{qw} \tau_{iq}^k \tau_{jw}^k \Big[A_{ij}^l ln(\pi_{qw}^k) + (1 - A_{ij}^l) ln(1 - \pi_{qw}^k) \Big] + \sum_i \sum_q \tau_{iq}^k ln(\alpha_q^k) + \sum_i \sum_q \tau_{iq}^k ln(\theta_i^k) - \sum_i \sum_q \tau_{iq}^k ln(\tau_{iq}^k) \tag{17}$$

The parameters that maximize $\mathcal{I}_\theta(R_{\mathcal{A}^k})$ are derived directly from the previously presented formula. To ensure that the vector $\boldsymbol{\alpha}^k$ and matrix $\mathbf{\Pi}^k$ satisfy the constraints $\sum_q \alpha_q^k = 1$ and $0 \leq \pi_{qw} \leq 1, \forall q, w \in \{1, ..., Q^k\}^2$, Lagrange multipliers are employed as follows:

$$\hat{\theta}_i^k = \frac{\sum_{l \in L^k} \sum_j A_{ij}^l}{\sum_{l \in L^k} \sum_{i,j \in q} A_{ij}^l} \tag{18}$$

$$\hat{\alpha}_q^k = \sum_i \frac{\tau_{iq}^k}{N} \tag{19}$$

$$\hat{\pi}_{qw}^k = \frac{\sum_{l \in L^k} \sum_{i \neq j} \tau_{iq}^k \tau_{jw}^k A_{ij}^l}{\sum_{l \in L^k} \sum_{i \neq j} \tau_{iq}^k \tau_{jw}^k} \tag{20}$$

$$\hat{\tau}_{iq}^k \propto \hat{\alpha}_q^k \prod_{l \in L^k} \prod_{i \neq j} \prod_w \Big[\hat{\pi}_{qw}^{k\ A_{ij}^l} + (1 - \hat{\pi}_{qw}^k)^{(1 - A_{ij}^l)} \Big]^{\hat{\tau}_{jw}^k} \tag{21}$$

where $\hat{\boldsymbol{\alpha}}^k, \hat{\mathbf{\Theta}}^k, \hat{\mathbf{\Pi}}^k, \hat{\boldsymbol{\tau}}^k$ are the best current parameters for the group k. Due to the interdependence between $\hat{\Pi}^k$ and $\hat{\tau}^k$, an effective way to determine the best estimation is to alternate between updating $\hat{\Pi}^k$ and $\hat{\tau}^k$ iteratively until convergence. The optimized parameters define the distribution of DCSBM and vertex-to-block assignments for a group k. The same computation is executed for each group independently. The overall method is summarized in the algorithm 1.

4.3 Initialization model

The initialization process of MDCSBM involves setting up values to layer-to-group variables $\mathbf{Y}$ and vertex-to-block variables $\mathbf{Z}$. Effective initialization of these assignment variables contributes to faster convergence and a higher chance of recovering accurate ground truth values. In the context of mixture models, the K-means algorithm is commonly employed for initializing assignment variables due to its simplicity and quick response. In this paper, we introduce a novel spectral technique that computes layer-to-group and vertex-to-block variables, such that clustering results are used as an initialization for inferring the MDCSBM model.

Consider $\mathbf{U} = \{\mathcal{U}^1, \mathcal{U}^2, ..., \mathcal{U}^K\}$ set of centroid graphs, with each graph $\mathcal{U}^k$ being the centroid that represents the group k . We aim to find layer-to-group variables by optimizing centroids that best represent each group.

Then, each centroid $\mathcal{U}^k$ gets clustered into Q_k community to initialize the vertex-to-block assignment variable. One way to find the communities of centroid k is to ensure that it is composed of Q^k disconnected components, where each component corresponds to a community in the graph. In network theory, a graph with a Q^k component exhibits a multiplicity of Q^k null eigenvalues in its corresponding Laplacian matrix. These null eigenvalues are the smallest eigenvalues of the Laplacian matrix. Thus, minimizing the Q^k smallest eigenvalue of the Laplacian matrix associated with $\mathcal{U}^k$ facilitates the formation of Q^k disconnected components within the centroid. Therefore, the model aiming to optimize these representations can be formulated as follows:

$$\begin{aligned}
&\min_{\mathcal{U}^1,\ldots,\mathcal{U}^K,\mathbf{F},\mathbf{Y}} \sum_{l=1}^{L}\sum_{k=1}^{K} y_{lk}||\mathcal{U}^k - \mathcal{A}^l||_F^2 + 2\lambda \sum_{k=1}^{K} Tr(\mathbf{F^k}^T \mathbf{L}_{U^k}\mathbf{F^k}) \\
&s.t \;\; \forall i, u_{ij}^k \geq 0, \mathbf{1}^T\mathbf{u}_i^k = 1, \forall k, \\
&(\mathbf{F^k})^{\mathbf{T}}\mathbf{F^k} = \mathbf{I}, y_{lk} \in \{0,1\}, \sum_{k=1}^{K} y_{lk} = 1
\end{aligned} \tag{22}$$

where $||.||_F^2$ denote the Frobenius norm, and u_{ij}^k represents element of the centroid $\mathcal{U}^k$, where $\forall i, j \in V$. The Laplacian representation of centroid $\mathcal{U}^k$ is denoted by $L_{\mathcal{U}^k}$, and $\mathbf{F}^k$ represents an embedding vector. The Laplacian matrix is computed in its unnormalized version as follows:

$$L_{\mathcal{U}^k} = D_{\mathcal{U}^k} - \mathcal{U}^k \tag{23}$$

where D_{U^k} denotes the degree matrix, a diagonal matrix. $\mathbf{F}^k \in \mathbf{R}^{N\times Q^k}$ helps to get the number of connected components in the graph. The embedding vector was included in the cost function to relax the non-linear constraint of constructing a centroid with Q^k disconnected components that effectively represent the communities, leveraging the number of null eigenvalues corresponding to the number of components in the graph. The optimization process for this model is described in the next subsection. Experimentally, this initialization helps the MDCSBM to converge faster than random initialization, up to more than 30 times faster, which generally depends on the data structure.

4.4 Optimization of initialization model

The initialization model described in Section 4.3 in Equation 22 involves multiple variables, making it challenging to optimize them simultaneously. Therefore, we adopt an iterative technique where each variable will be optimized while the others are held fixed.

4.4.1 Optimizing Y, while U and F are fixed

The model can be represented as follows:

$$\begin{aligned}
&\min_{\mathbf{Y}} \sum_{l=1}^{L}\sum_{k=1}^{K} y_{lk}||\mathcal{U}^k - \mathcal{A}^l||_F^2 \\
&s.t \;\; y_{lk} \in \{0,1\}, \sum_{k=1}^{K} y_{lk} = 1
\end{aligned} \tag{24}$$

By relaxing the constraint $y_{lk} \in \{0,1\}$ to $y_{lk} \in [0,1]$, the model becomes linear, facilitating the application of analytical solutions that satisfy the Karush-Kuhn-Tucker (KKT) conditions using the Lagrange technique. The analytical solution is expressed as follows:

$$y_{lk} = \frac{||\mathcal{U}^k - \mathcal{A}^l||_F^2}{\sum_{k'=1}^{K} ||\mathcal{U}^{k'} - \mathcal{A}^l||_F^2} \tag{25}$$

The determination of the group to which the layer l will be assigned is carried out as follows:

$$y_{lk} = \underset{k}{\operatorname{argmax}}\, y_{lk} \tag{26}$$

4.4.2 Optimizing $\mathcal{U}$, while Y and F are fixed

Firstly, the optimization of centroids $\mathcal{U}^k$ is performed independently, and according to Wang et al. (2020), the objective function $Tr(\mathbf{F^k}^T \mathbf{L}_{U^k} \mathbf{F^k})$ can be expressed as $\sum i, j||\mathbf{f}_i - \mathbf{f}j||_2^2 ui, j$. Therefore, the optimization for each centroid can be formulated as follows:

$$\begin{aligned} \min_{\mathcal{U}^k} \sum_{l \in L^k} \sum_{i,j} ||u_{ij} - A_{ij}||_F^2 - \lambda \sum_{i,j} ||\mathbf{f}_i - \mathbf{f}_j||_2^2 u_{i,j} \\ s.t \;\; u_{ij} \geq 1, \mathbf{1}^T.\mathbf{u}_i = 1 \end{aligned} \tag{27}$$

L_k represent the set of layers for which y_{lk} equals one. We denote $||\mathbf{f}_i - \mathbf{f}j||_2^2$ as dij. Due to the independence of optimization for each vertex vector $\mathbf{u}_i$ from the others, the optimization of $\mathcal{U}^k$ can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{u}_i^k} \sum_{l \in L^k} ||\mathbf{u}_i - \frac{\lambda}{2|L^k|} \mathbf{d}_i||_F^2 \\ s.t \;\; \forall i, j \;\; u_{ij} \geq 1, \mathbf{1}^T.\mathbf{u}_i = 1 \end{aligned} \tag{28}$$

The model mentioned above is quadratic with linear constraints, indicating that it is convex. It can resolved using the augmented Lagrange multipliers. Otherwise, any quadratic solver can efficiently resolve this problem.

4.4.3 Optimizing $\mathbf{F}^k$ while U and Y are fixed

The optimization of each $\mathbf{F}^k$ for every group is performed independently of the remaining groups. For a given group, the model can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{F}^k} Tr({\mathbf{F}^k}^T L_{\mathcal{U}^k} \mathbf{F}^k) \\ s.t \;\; {\mathbf{F}^k}^T.\mathbf{F}^k = \mathbf{I} \end{aligned} \tag{29}$$

The optimal $\mathbf{F}^k$ can be obtained by extracting Q^k eigenvectors of the Laplacian matrix $L_{\mathcal{U}^k}$, which are associated with the smallest Q^k eigenvalues. It is important to note that Q^k denotes the number of communities within group k.

Algorithm 2 Multi Centroids algorithm initialization

Input $\mathcal{G}, K, \mathbf{Q} = [Q^1, ..., Q^K]$ **Output**: $\mathbf{U}, \mathbf{Y}$
  Initialize $\mathbf{U} = \{\mathcal{U}^1, \mathcal{U}^2, ..., \mathcal{U}^K\}$
  while Iteration < Iteration max || Not Converge **do**
    optimize y_{lk} with 26
    optimize $\mathbf{u}_i$ with 28
    compute $\mathbf{F}^k$ from the eigen vector associated to Q^k smallest eigen value of $L_{\mathcal{U}^k}$
  end while

5 Model Selection

To determine the optimal number of groups K and the number of blocks Q^k in each group k, we propose using the Bayesian Information Criterion (BIC). The BIC is formulated as follows:

$$\begin{aligned} BIC =& 2\sum_{k=1}^{K} (Q^k)^2 \log(|L^k|N(N-1)) + (N - Q^k) log(|L^k|N) + (Q^k - 1)\log(|L^k|N) \\ &+ 2(K-1)\log(L) - \mathcal{L}(\mathcal{G}, \mathbf{Y}, \mathbf{Z}; \boldsymbol{\beta}, \boldsymbol{\Delta}) \end{aligned} \tag{30}$$

In this expression, $|L^k|$ represents the number of layers associated with group k. The BIC helps determining the number of group and blocks by balancing the fit of the model to data penalized by its complexity. The selected model is the one with the lowest BIC.

6 Experiments

To assess the properties of the Mixture Degree-Corrected Stochastic Block Model (MDCSBM), we evaluated its performance across various datasets. This analysis includes real-world data on brain connectivity from cerebral imaging and reality mining data concerning proximity interactions among students within a university. Additionally, we utilized multiple synthetic data partitions to explore the limits of MDCSBM and assess its scalability in handling large datasets.

We conducted comparative analyses of MDCSBM against several established algorithms, including the multiplex DCSBM, Generalized Louvain Mucha et al. (2010), and Graph Fusion Spectral Clustering (GFSC) Kang et al. (2019). These algorithms are designed to optimize partitions within multilayer graphs. They are constrained to return a single partition for all multilayer graph, which will be used as the partition for each layer. To quantitatively evaluate the performance of each algorithm, we employed metrics such as Average Normalized Mutual Information (NMI) and Average Adjusted Mutual Information (AMI). These metrics were calculated by computing the NMI and AMI scores for each layer independently, then averaging these scores across all layers. Since the algorithms used for comparison return unified community structures across all layers, the same set of communities is applied in the assessment for each layer. The results presented here reflect only the mean values of these metrics.

6.1 Real Data Set

In our real-world application, we examined two datasets: diffusion Magnetic Resonance Imaging (dMRI) and the Reality Mining dataset. To the best of our knowledge, neither dataset includes expert annotations or ground truth labels, meaning there is no reference standard for comparison. Consequently, this lack of ground truth prevents a fair comparison with other algorithms. Instead, we conducted an in-depth analysis of these datasets using our MDCSBM algorithm, offering interpretations and insights based on the results obtained.

6.1.1 Brain Connectivity from Multiplex Graph Representation

To assess the relevance of the MDCSBM in real applications, we conducted a study focusing on brain connectivity using diffusion Magnetic Resonance Imaging (dMRI)[1]. We utilized the HNU1 dataset, which includes 300 undirected graphs derived from 10 brain-scanning sessions across 30 individuals Zuo et al. (2015). Each graph contains 200 vertices representing different regions of the brain, with edges indicating the observed neural connections between these regions. The dataset treats each graph as an independent observation, aligning with methodologies from prior studies Mantziou et al. (2023); Arroyo et al. (2020).

Our primary goal was to detect groups of subjects exhibiting similar brain connectivity patterns and to identify communities within these regions. Considering the brain's division into two hemispheres, each graph inherently comprises two blocks. However, inter-subject scan variations reflect the unique neural states of the individuals.

The MDCSBM was tasked with recognizing subjects with closely related brain scans, accounting for each hemisphere. The model was initialized to distinguish 30 groups, corresponding to the number of subjects, with two blocks per group reflecting the hemispheres. Using the centroid method for initialization, the model consistently identified two blocks for each snapshot across 100 runs, which supports neuroscientific evidence of significant hemispherical independence. However, it also tended to consolidate subjects into 27 distinct groups from the 300 graphs.

Remarkably, the model effectively grouped each subject's graphs within the identified clusters. Specifically, it clustered subjects 8 and 23, 11 and 14, and 10 and 28 together, as detailed in Table 1. The grouping of subjects 11 and 14 aligns with findings from a semi-supervised study Arroyo et al. (2020). Notably, while

[1] https://neurodata.io/mri/

that study employed a semi-supervised method to achieve these results, our MDCSBM model operates in a completely unsupervised manner. Despite the numerous hyper-parameters that need to be carefully adjusted, our approach still provides more comprehensive results. Additionally, in the study by Mantziou et al. (2023), the experiment limited clustering to only two groups across all layers, which is not optimal for this dataset containing 30 different subjects. Even with this simplified task, they were unable to consistently group all layers from the same subject within the same cluster. In contrast, our results demonstrate the robustness of the MDCSBM algorithm, as it successfully grouped all layers of each subject into same cluster in every instance.

Subject	Image	Cluster	Subject	Image	Cluster	Subject	Image	Cluster	Subject	Image	Cluster	Subject	Image	Cluster	Subject	Image	Cluster
	1	C1		1	C2		1	C3		1	C4		1	C5		1	C6
	2	C1		2	C2		2	C3		2	C4		2	C5		2	C6
	3	C1		3	C2		3	C3		3	C4		3	C5		3	C6
	4	C1		4	C2		4	C3		4	C4		4	C5		4	C6
S1	5	C1	S2	5	C2	S3	5	C3	S4	5	C4	S5	5	C5	S6	5	C6
	6	C1		6	C2		6	C3		6	C4		6	C5		6	C6
	7	C1		7	C2		7	C3		7	C4		7	C5		7	C6
	8	C1		8	C2		8	C3		8	C4		1	C5		8	C6
	9	C1		9	C2		9	C3		9	C4		9	C5		9	C6
	10	C1		10	C2		10	C3		10	C4		10	C5		10	C6
	1	C7		1	C8		1	C9		1	C10		1	C11		1	C12
	2	C7		2	C8		2	C9		2	C10		2	C11		2	C12
	3	C7		3	C8		3	C9		3	C10		3	C11		3	C12
	4	C7		4	C8		4	C9		4	C10		4	C11		4	C12
S7	5	C7	S8	5	C8	S9	5	C9	S10	5	C10	S11	5	C11	S12	5	C12
	6	C7		6	C8		6	C9		6	C10		6	C11		6	C12
	7	C7		7	C8		7	C9		7	C10		7	C11		7	C12
	8	C7		8	C8		8	C9		8	C10		8	C11		1	C12
	9	C7		9	C8		9	C9		9	C10		9	C11		9	C12
	10	C7		10	C8		10	C9		10	C10		10	C11		10	C12
	1	C13		1	C11		1	C14		1	C15		1	C16		1	C17
	2	C13		2	C11		2	C14		2	C15		2	C16		2	C17
	3	C13		3	C11		3	C14		3	C15		3	C16		3	C17
	4	C13		4	C11		4	C14		4	C15		4	C16		4	C17
S13	5	C13	S14	5	C11	S15	5	C14	S16	5	C15	S17	5	C16	S18	5	C17
	6	C13		6	C11		6	C14		6	C15		6	C16		6	C17
	7	C13		7	C11		7	C14		7	C15		7	C16		7	C17
	8	C13		8	C11		8	C14		8	C15		8	C16		8	C17
	9	C13		9	C11		9	C14		9	C15		9	C16		9	C17
	10	C13		10	C11		10	C14		10	C15		10	C16		10	C17
	1	C18		1	C19		1	C20		1	C21		1	C8		1	C22
	2	C18		2	C19		2	C20		2	C21		2	C8		2	C22
	3	C18		3	C19		3	C20		3	C21		3	C8		3	C22
	4	C18		4	C19		4	C20		4	C21		4	C8		4	C22
S19	5	C18	S20	5	C19	S21	5	C20	S22	5	C21	S23	5	C8	S24	5	C22
	6	C18		6	C19		6	C20		6	C21		6	C8		6	C22
	7	C18		7	C19		7	C20		7	C21		7	C8		7	C22
	1	C18		8	C19		8	C20		8	C21		8	C8		8	C22
	9	C18		9	C19		9	C20		9	C21		9	C8		9	C22
	10	C18		10	C19		10	C20		10	C21		10	C8		10	C22
	1	C23		1	C24		1	C25		1	C10		1	C26		1	C27
	2	C23		2	C24		2	C25		2	C10		2	C26		2	C27
	3	C23		3	C24		3	C25		3	C10		3	C26		3	C27
	4	C23		4	C24		4	C25		4	C10		4	C26		4	C27
S25	5	C23	S26	5	C24	S27	5	C25	S28	5	C10	S29	5	C26	S30	5	C27
	6	C23		6	C24		6	C25		6	C10		6	C26		6	C27
	7	C23		7	C24		7	C25		7	C10		7	C26		7	C27
	8	C23		1	C24		8	C25		8	C10		8	C26		8	C27
	9	C23		9	C24		9	C25		9	C10		9	C26		9	C27
	10	C23		10	C24		10	C25		10	C10		10	C26		10	C27

Table 1: The table shows the cluster obtained for each image of each subject by the MDCSBM algorithm. Each of the 30 existing subjects has ten images, which are presented in 10 layers.

6.1.2 Reality Mining Study

In this second study, we analyze a physical proximity graph derived from the reality mining study conducted by Eagle and Pentland Eagle & (Sandy), featuring a group of college students and faculty. This dataset comprises undirected graphs representing interactions among 96 students from the Massachusetts Institute of Technology, collected over a nine-month period. Participants were equipped with mobile phones installed with special software that recorded close proximity encounters using Bluetooth technology. Therefore, each layer of this multiplex graph represents a day, with edges indicating at least one observed proximity encounter between individuals on that day. The dataset thus includes $L = 234$ layers, each with $|V| = 96$ vertices.

Figure 2 illustrates the results from the MDCSBM model applied to this dataset. Sub-figure (d) displays the BIC values relative to the number of groups. We identified five distinct groups, with two communities in groups 0 and 2, and one community in groups 1, 3, and 4. The single community structure in groups $1, 3$, and 4 is attributable to the high sparsity of layers within these groups. In contrast, groups like 0 exhibit dual communities as depicted in sub-figure (c), which shows a layer from group 0.

Sub-figures (a) and (b) reveal discernible patterns within these groups. Notably, group 3 (colored red) encompasses layers from weekends and holidays, particularly the Christmas period in December. The other groups capture monthly variations; for instance, group 0 (colored blue) corresponds to the start of the semester, characterized by denser layers. Group 1 (colored orange) represents mid-year, aligning with academic examinations and displaying sparser community interactions. Groups 2 and 4 (colored green and purple, respectively) reflect the end-of-year activities, showing varied layer densities. These interpretations

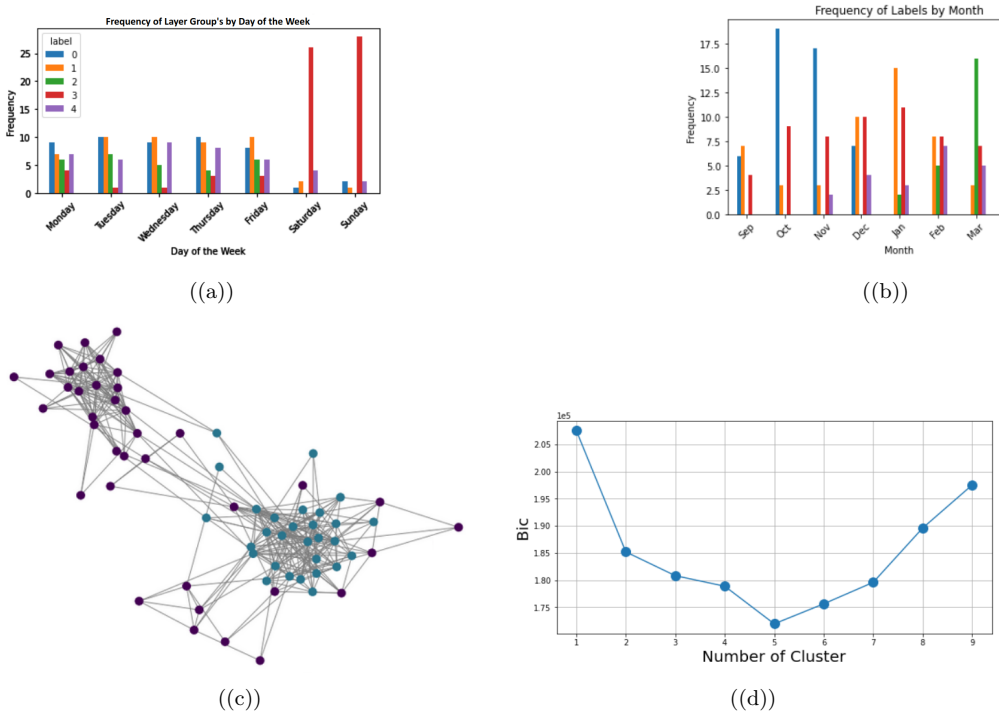


Figure 2: The figure illustrates the outcomes of the Multi-Group clustering applied to the reality mining distance proximity dataset. Panel (a) depicts the total number of days represented in each group. Panel (b) details the monthly distribution of days for each group. Panel (c) presents the community structures within a layer from the group 0. Finally, panel (d) displays the BIC values relative to the number of groups.

are consistent with typical university schedules, where students primarily interact with classmates during academic sessions but engage with a broader range of acquaintances outside of class times.

Sub-figures (a) and (b) reveal discernible patterns within these groups, particularly for groups 0 and 2, which each contain two communities. From both sub-figures, it is evident that group 0, represented in blue, consists of days of the week when courses are likely to occur in the first part of the year. This group is characterized by denser layers, indicating that students are taking two different courses. Conversely, group 2, represented in green, functions similarly to group 0 but pertains to the second part of the year, again involving students taking two different courses. This explains the presence of two communities within these groups.

Group 3, colored red, encompasses layers from weekends and holidays, with a notable focus on the Christmas period in December. Other groups capture monthly variations; for instance, group 1, colored orange, represents the mid-year period, which aligns with academic examinations and displays sparser community interactions. Group 4, colored purple, reflects end-of-year activities, showing varied layer densities. These interpretations align with typical university schedules, where students primarily interact with classmates during academic sessions but engage with a broader range of acquaintances outside of class times.

This experiment demonstrates the effectiveness of the DCMSBM in real-world applications, adeptly handling complex data structures and offering insightful interpretations of dynamic social interactions.

6.2 Synthetic Data

6.2.1 Variability in Block Size

In this experiment, we aim to assess each group's sensitivity to block size. Therefore, the dataset is composed of 3 groups. Each group of layers consists of 10 graphs, each with 100 vertices organized into four blocks. Vertices inside a block are randomly linked to each other with a probability $\pi_{\text{intra}} = 0.5$, and vertices from

different blocks are randomly connected with a probability $\pi_{\text{inter}} = 0.3$. What distinguishes the groups of layers is the number of vertex in each block, $G^1 = \{25, 25, 25, 25\}$, $G^2 = \{20, 25, 25, 30\}$ and $G^3 = \{30, 30, 20, 20\}$, where G^i is the ith group, and for each group, the first number indicates the number of vertex of the first block, the second number the one of the second block, and so forth, as shown in figure 3.

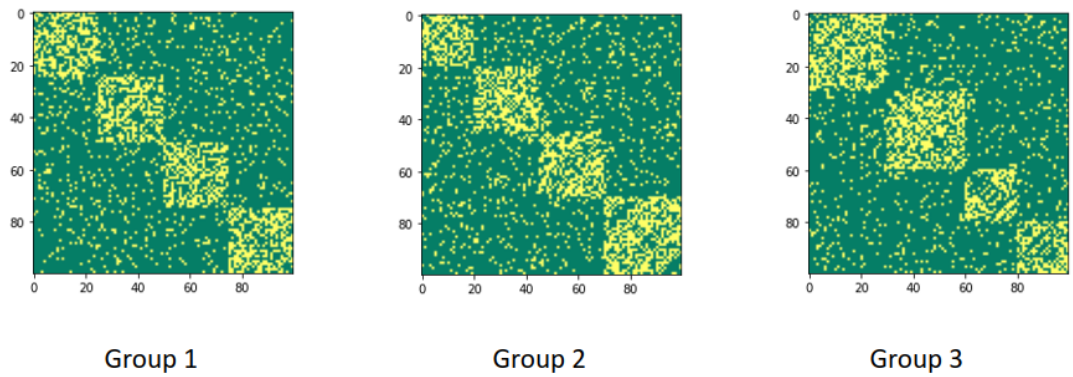


Figure 3: The adjacency matrices correspond to a single layer for each group. G^1 is presented on the right, G^2 in the middle, and G^3 on the left. The intra-block density is set at $\pi_{\text{intra}} = 0.5$. For better visualization, the inter-block connectivity probability is set at $\pi_{\text{inter}} = 0.1$, instead of $\pi_{\text{inter}} = 0.3$ as an experiment.

The actor-based clustering algorithm on multiplex graphs yields an average clustering of vertices. In contrast to MDCSBM, there is no differentiation between blocks of vertices for each group of layers, contributing to the superior NMI/AMI results of MDCSBM over other multiplex algorithms, as shown in Table 2. The mean errors between the predicted and generated parameters of MDCSBM are $mean_{ErroParamMDCSBM} = 0.04$, affirming the improved parameter recovery of MDCSBM compared to the multiplex DCSBM, which results in an error of $mean_{ErroParamMultiDCSBM} = 0.67$.

	MDCSBM	DCSBM	GLouvain	GFSC
NMI	**100**	61.40	77.44	73.80
AMI	**100**	58.74	75.74	72.65

Table 2: The NMI and AMI performances on MDCSBM, multiplex DCSBM, Glouvain and GFSC in variability on block size synthetic datasets.

6.2.2 Variability in Block Distribution

In this experiment, we aim to assess the model's performance regarding variation in block distribution. Therefore, the dataset for this experiment comprises three groups, each containing ten layers. Each graph consists of 100 vertices distributed across four equally sized blocks ($\{25, 25, 25, 25\}$) with the same intra-community probability ($\pi_{intra} = 0.5$). The distinction between layer groups lies in the probability of having an edge between blocks, denoted as $\pi^{G^1}_{inter} = 0.1$, $\pi^{G^2}_{inter} = 0.3$, and $\pi^{G^3}_{inter} = 0.5$. This variability tests the algorithm's capability to identify groups with different $\mathbf{\Pi}$ distributions. Notably, the third group has $\pi_{inter} = \pi_{intra}$, resembling a random graph without communities.

MDCSBM accurately identifies vertex blocks for each group with an estimated parameter error of $mean_{ErroParamMDCSBM} = 0.04$. Additionally, the algorithm recognizes "1 community" for the layers without communities, consistent with a random graph. This result demonstrates the MDCSBM's ability to identify the noisy layer. In contrast, multiplex algorithms exhibit an average effect in the vertex-to-block assignment, especially multiplex DCSBM estimates the vertex-to-block with a higher estimation error ($mean_{ErroParamMultiDCSBM} = 0.71$), as observed previously.

	MDCSBM	DCSBM	GLouvain	GFSC
NMI	**100**	55.54	66.66	53.55
AMI	**100**	55.03	66.66	53.02

Table 3: The NMI and AMI performance for MDCSBM, multiplex DCSBM, Glouvain and GFSC in variability on block size synthetic datasets.

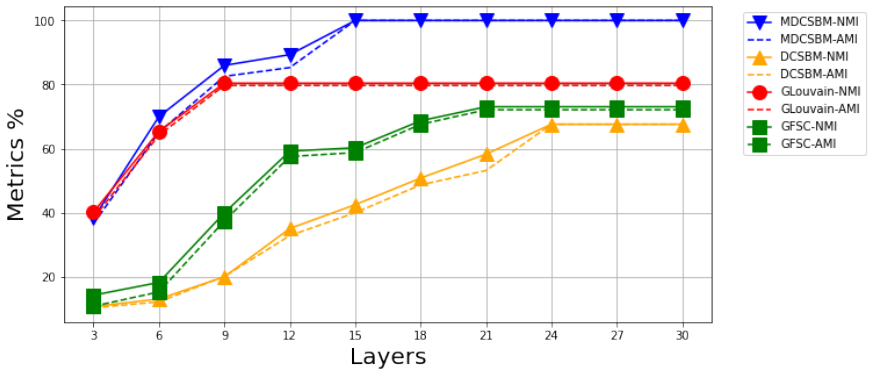


Figure 4: The performance of MDCSBM, DCSBM, Glouvain, and GFSC to find the clusters of vertices regarding the number of layers for each group. The NMI and AMI were used as metrics of performance.

6.2.3 Variability in Number of Layer

In this experiment, we test the scalability of the method regarding the number of layers. Therefore, we keep the number of vertices, groups, and the block distribution constant while we vary the number of layers for each group. Similar to Experiment 6.2.1, we define three groups with an equal number of layers but distinct block divisions for each group. Specifically, we set $\pi_{intra} = 0.5$ and $\pi_{inter} = 0.3$. The block distribution within each group consists of four blocks, as follows: $G^1 = \{25, 25, 25, 25\}$, $G^2 = \{20, 25, 25, 30\}$, and $G^3 = \{30, 30, 20, 20\}$, where G^i represents the ith group.

The result of the following experiments is shown in the figure 4. We can see that the performance of the MDCSBM in retrieving the optimal blocks for each layer augments when the number of layers augments, compared to the other methods. It can be explained by the law of large numbers, which delineates the convergence in probability to the expected value as the number of samples—the layers in our case—increases. As the layers in the multiplex graph augment, the computation time increases linearly due to the number of parameters scaling linearly with the number of layers, given a fixed number of groups and blocks. This contrasts with other tested algorithms, which do not scale well with large datasets.

6.2.4 Variability in Number of Vertices

In this experiment, we assess the scalability of the method regarding the size of each graph. Therefore, we fix the number of layers, the number of groups, and the block distribution of each group, and then we vary the number of vertices of the multiplex graph. We set three groups with an equilibrium of 10 layers for each and different block division such that $G^1 = \{25\%N, 25\%N, 25\%N, 25\%N\}$, $G^2 = \{20\%N, 25\%N, 25\%N, 30\%N\}$, and $G^3 = \{30\%N, 30\%N, 20\%N, 20\%N\}$. The $x\%N$ means x percent from a number of vertices of the layer. The intra-block distribution is $\pi_{intra} = 0.5$ and the inter-distribution is $\pi_{inter} = 0.3$.

In this experiment, we maintain a constant number of layers, groups, and block distribution for each group while varying the graph size by raising its number of vertices. Similar to the previous experiments in 6.2.1, we define three groups with an equal number of layers and distinct block divisions for each.

The obtained result is represented in the figure 5. We can notice that the MDCSBM scales to a large dataset with thousands of nodes. The time complexity varies regarding the graph's size and the block's structure. We study the complexity time of the MDCSBM in the supplementary materials.

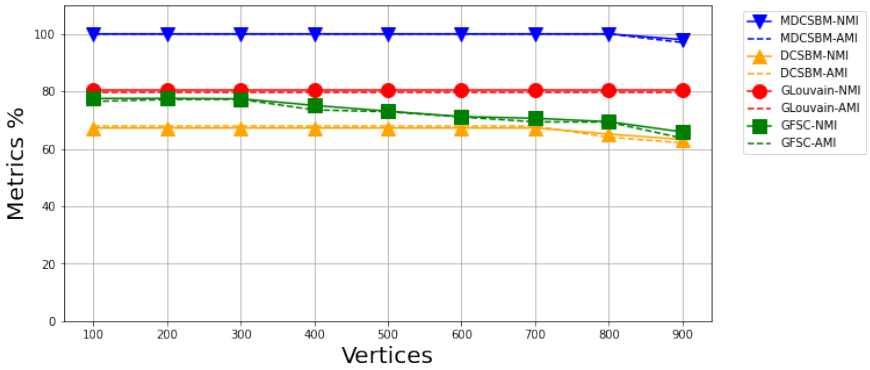


Figure 5: The performance of MDCSBM, DCSBM, Glouvain, and GFSC to find the clusters of vertices regarding the number of vertices in a multiplex graph. The NMI and AMI were used as metrics of performance.

6.2.5 Variability in the Number of Groups

The objective of this experiment is to assess the method's sensitivity with respect to the number of groups within the multiplex graph. Consequently, we keep the number of layers in the multiplex graph fixed at 20, then we vary the number of groups from 2 to 8, ensuring that each group has a distribution distinct from the others.

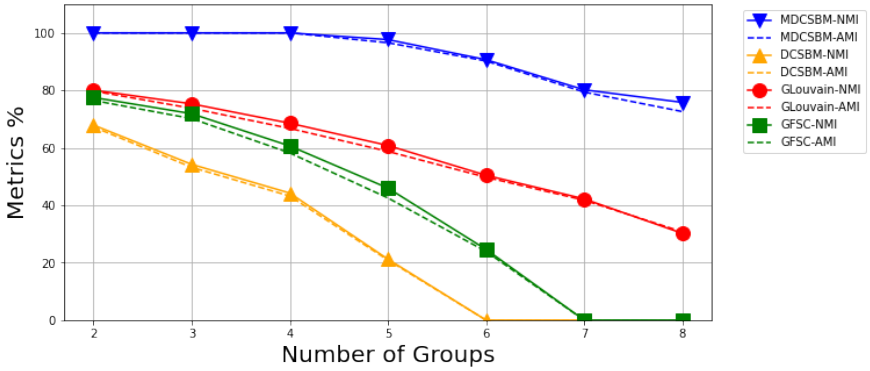


Figure 6: Performance of MDCSBM, DCSBM, GLouvain and GFSC when the number of groups varies in fixed number of layer.

Figure 6 shows the performance of the algorithms in this experiment. The MDCSBM performs better than the others in finding multigroup community detection. However, we can see that for a high number of groups, the performance of MDCSBM decreases; such a result may be explainable by the reduced number of layers for the high number of groups. We can see that this result will be enhanced when each group's layer number increases.

6.2.6 Sensitivity to Block Size

In this experiment, we aim to assess the ability of the method to perform in an unbalanced block size. Therefore, we construct a multiplex graph with 20 layers and 2 groups, such that each group has 10 layers of 100 vertices for each, and each group has two blocks with the same distribution. We fix the size of blocks of one group to be 50 vertex per block, and for the other group, we change the block size from 10 to 50 vertex.

The performance of the algorithms in this experiment is presented in Figure 6. It is observed that MDCSBM outperforms the others in multi-group community detection. However, with a high number of groups, the

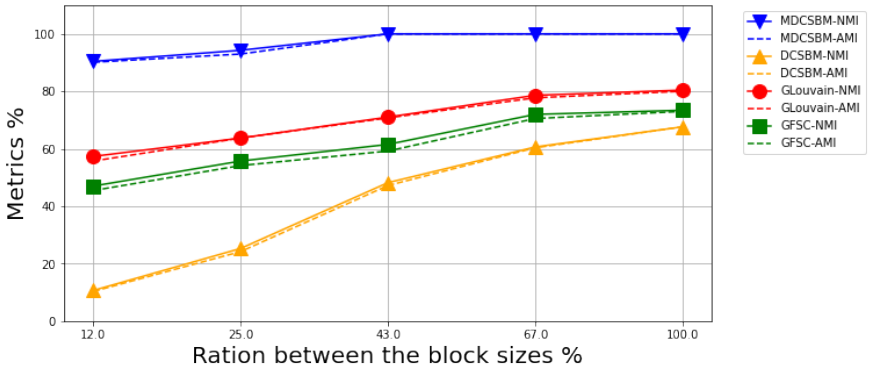


Figure 7: Performance of MDCSBM, DCSBM, GLouvain and GFSC when the size of block variate. The variate group has two blocks; the sensitivity is computed by the fraction between the size of the two blocks.

performance of MDCSBM decreases, potentially due to the reduced number of layers for the high number of groups. This result is expected to improve when the number of layers in each group increases.

6.2.7 Sensitivity to Group Size

This experiment aims to assess the model's performance when the groups have an unbalanced distribution of layers. We consider a multiplex graph with two groups featuring distinct block distributions to achieve this. The number of layers in the first group is fixed at 10, while the number of layers in the second group varies from 1 to 100. Each layer consists of 100 vertices.

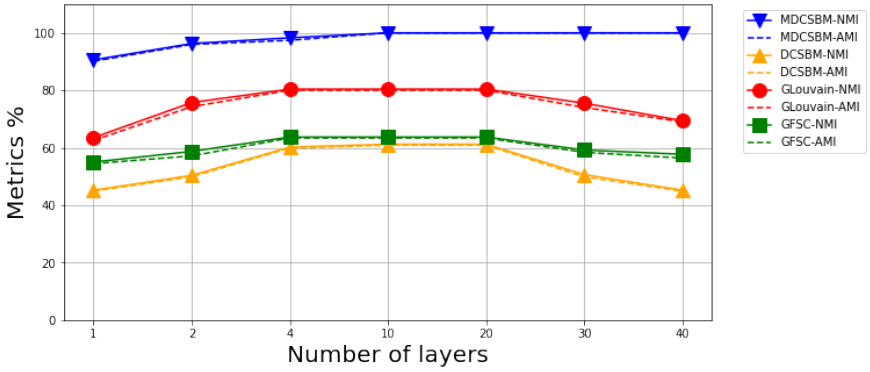


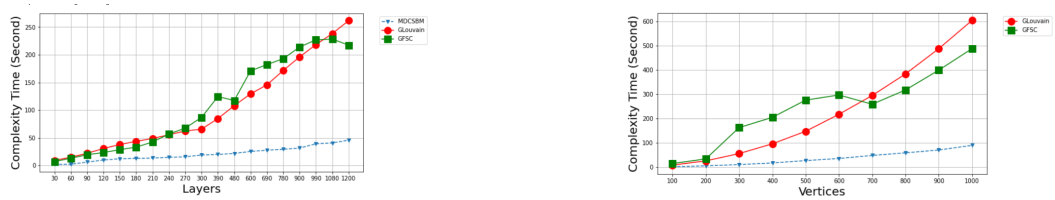
Figure 8: Performance of MDCSBM, DCSBM, GLouvain, and GFSC for different numbers of layers in the group. It varies from 1 to 40 layers.

Figure 8 illustrates various algorithms' performance in this application. Compared to the others, the MDCSBM algorithm demonstrates notable stable performance, even in scenarios with high imbalance in the number of layers. This result is attributed to the effectiveness of the joint clustering approach, which we believe contributes to achieving accurate results.

6.2.8 Time Complexity

To illustrate the model's time complexity and scalability to large datasets, we present the time complexity of the model. To achieve this, we conduct two experiments where the time complexity is evaluated for varying numbers of layers and vertices.

Figure 9 illustrates the time complexity for each case and for each algorithm. Firstly, MDCSBM models scale better in all cases than all other algorithms. Evidently, the time complexity scales linearly with the number



((a)) ((b))

Figure 9: Complexity time regarding a number of layers and vertices. At the right, the time is counted when the number of layers augments. At the left, the time is counted when the number of vertices augments. The time is counted in seconds.

of layers. This behavior is attributed to the increase in variables that linearly correlate with the number of layers. Additionally, the time complexity rises more rapidly with an increase in the number of vertices. This outcome results from the significant growth of the combinatorial solution as the number of vertices increases. Despite this, the modeling approach maintains a reasonable time complexity in scenarios with high combinatorial solutions, owing to our efficient initialization process, which facilitates faster convergence compared to random initialization. Furthermore, our initialization accelerates convergence to a favorable local minimum, achieving speeds over 50 times faster than random initialization.

6.2.9 Comparison Between MDCSBM and the Initialization Model

The proposed initialization model, based on spectral representation, can be regarded as a comprehensive approach capable of performing multi-group community detection. To evaluate its performance against the MDCSBM model, we used the test set detailed in 6.2.1. This dataset consists of 3 groups, each containing 10 layers, with each layer comprising 100 vertices organized into four blocks. Vertices within the same block are connected randomly with a probability of $\pi_{\text{intra}} = 0.5$, while connections between vertices in different blocks occur with a probability of $\pi_{\text{inter}} = 0.3$. The distinguishing characteristic of each group lies in the distribution of vertices within the blocks: $G^1 = 25, 25, 25, 25$, $G^2 = 20, 25, 25, 30$, and $G^3 = 30, 30, 20, 20$, where G^i denotes the ith group. Within each group, these values represent the number of vertices in the first, second, third, and fourth blocks, respectively, as depicted in Figure 3.

For this evaluation, we used the MDCSBM algorithm initialized with the proposed spectral model. The experiment was repeated 100 times, and the performance results are summarized in Table 4.

	Model Initialization (Spetrale Representation)	MDCSBM
NMI	75.48+-22.80	**95.60 +- 17.33**
AMI	74.60+-23.82	**95.36+-17.98**

Table 4: The NMI and AMI performance for MDCSBM and model initialization on block size synthetic datasets.

The superior performance of the MDCSBM model compared to the spectral model can be attributed to the way the MDCSBM utilizes the results produced by the spectral model. The spectral model provides a partition that corresponds to a fixed point of its own cost function, serving as an effective initialization. Although this partition does not necessarily represent a fixed point for the MDCSBM, it enables the MDCSBM to further refine and enhance the clustering results, ultimately achieving better overall performance.

6.2.10 Comparison Between Random and Proposed Initialization

The MDCSBM model is highly non-convex, and the resulting outputs are significantly influenced by the initialization of layer-to-group and vertex-to-block values. To address this, we proposed an efficient initialization method based on spectral representation. To illustrate the advantages of this technique, we

present a comparison of performance and computation time between random initialization and our proposed spectral-based initialization model.

The same test set described in 6.2.1 was used for this comparison. This dataset comprises 3 groups, each containing 10 layers, with each layer composed of 100 vertices organized into four blocks. Vertices within the same block are randomly connected with a probability of $\pi_{\text{intra}} = 0.5$, while vertices between different blocks are connected with a probability of $\pi_{\text{inter}} = 0.3$. The distinguishing factor among the groups is the number of vertices in each block: $G^1 = 25, 25, 25, 25$, $G^2 = 20, 25, 25, 30$, and $G^3 = 30, 30, 20, 20$, where G^i represents the ith group. Within each group, the numbers correspond to the vertex count of the first, second, third, and fourth blocks, respectively, as illustrated in Figure 3.

We repeated the experiment 100 times and present the performance results and computation times in Table 5.

	Model Initialization	Random Initialization
NMI	**95.60 +- 17.33**	23.73+-26.40
AMI	**95.36+-17.98**	22.70+-27.38
Time Complexity (Second)	**3.65+-11.16**	130.84+-74.79

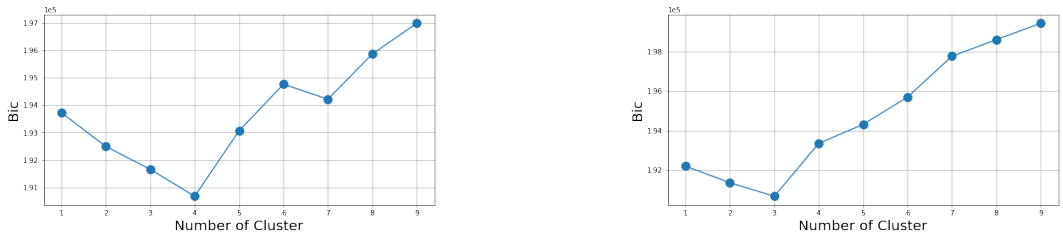
Table 5: The NMI and AMI performance, and time computation for MDCSBM by using the proposed model initialization and random initialization on block size synthetic datasets.

We observed that both the computation time and performance were notably improved when using our proposed initialization method compared to random initialization. This improvement can be attributed to the non-convex nature of the MDCSBM model and its sensitivity to initialization. A robust and well-structured initialization method, such as the one we proposed, is essential for achieving better performance and faster convergence.

6.2.11 Model Selection

The defined MSBM model needs a prior knowledge of number of cluster. Ones can think about formulation that allows to optimize the number of cluster in the inference as presented in Roy et al. (2006); Amini et al. (2024). Such models has been based on Chinese Restaurant Process (CRP) to determine the number of cluster. The CRP modeling in out of the scope of this paper, further enhancement will be reserved for future work. Here in this experiment, we show the results from using he BIC criteria to determine the number of cluster as defined in equation 30. It worth noting to remember that the BIC criteria determine the optimal number of blocks and groups by balancing the fit of the model to the data with a penalization of the model complexity.

In this experiments, we create a multiplex graph of 30 layers. Within this multiplex graph, we set 3 groups of 10 layers for each. We set $\pi_{intra} = 0.5$ and $\pi_{inter} = 0.3$ for all layers. The block distribution within each group consists of four blocks, as follows: $G^1 = \{25, 25, 25, 25\}$, $G^2 = \{20, 25, 25, 30\}$, and $G^3 = \{30, 30, 20, 20\}$, where G^i represents the ith group.



((a)) ((b))

Figure 10: The BIC metric for finding the optimal number of groups and block. At the left, the number of group is fixed to 3, then the BIC is computed for variate number of blocks. At the right, the number of block is fixed to 4, then the BIC is computed for variate number of group.

Figure 10 shows the variation of the Bic regarding the number of blocks when the number of groups is fixed to 3 at the left, and the variation of BIC regarding number of layers when the number of block is fixed to 4 at the right. We can clearly notice that the minimal value of BIC for both experiences indicates the optimal value of cluster as what is attending, showcasing its performance in toy example to define the number of clustering.

7 Conclusion

Throughout this paper, we have introduced the Mixture Degree-Corrected Stochastic Block Model (MDCSBM) for multi-group community detection in multiplex graphs. The MDCSBM defines the existing groups that share a similar community structure. Therefore, for each identified group, a distinct DCSBM is derived to ascertain the community structure of each vertex. We have devised an Expectation-Maximization (EM) framework for estimating layer-to-group assignment variables, followed by a Variational EM technique for estimating vertex-to-block assignments. A novel centroid methodology has been proposed to initialize layer-to-group and vertex-to-block variables, enhancing the model's convergence.

This model has been formulated to refine the estimation of the generating model underlying multiplex graphs. It significantly contributes to a better comprehension of community structures within multiplex graphs characterized by multigroups of community memberships. While the current presentation exclusively addresses unweighted graphs, potential extensions encompassing the incorporation of weights through alternative probability distributions such as Gaussian or Poisson distributions exist. Such extensions would undoubtedly enrich the model's applicability in capturing the intricacies of diverse real-world scenarios.

References

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Nazanin Afsarmanesh and Matteo Magnani. Finding overlapping communities in multiplex networks. *arXiv preprint arXiv:1602.03746*, 2016.

Melissa Ailem, François Role, and Mohamed Nadif. Co-clustering document-term matrices by direct maximization of graph modularity. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pp. 1807–1810, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450337946. doi: 10.1145/2806416.2806639. URL `https://doi.org/10.1145/2806416.2806639`.

Arash Amini, Marina Paez, and Lizhen Lin. Hierarchical stochastic block model for community detection in multiplex networks. *Bayesian Analysis*, 19(1):319–345, 2024.

Jesús Arroyo, Avanti Athreya, Joshua Cape, Guodong Chen, Carey E. Priebe, and Joshua T. Vogelstein. Inference for multiple heterogeneous networks with a common invariant subspace, 2020.

Pierre Barbillon, Sophie Donnet, Emmanuel Lazega, and Avner Bar-Hen. Stochastic block models for multiplex networks: an application to a multilevel network of researchers. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 180(1):295–314, 2017.

Punam Bedi and Chhavi Sharma. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6:n/a–n/a, 02 2016. doi: 10.1002/widm.1178.

Michele Berlingerio, Fabio Pinelli, and Francesco Calabrese. Abacus: frequent pattern mining-based community discovery in multidimensional networks, 2013. URL `https://arxiv.org/abs/1303.2025`.

Peter Bickel, David Choi, Xiangyu Chang, and Hai Zhang. Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *The Annals of Statistics*, 41(4), August 2013. ISSN 0090-5364. doi: 10.1214/13-aos1124. URL `http://dx.doi.org/10.1214/13-AOS1124`.

Alain Celisse, J. J. Daudin, and Laurent Pierre. Consistency of maximum-likelihood and variational estimators in the stochastic block model, 2012.

Marco Corneli. *Dynamic stochastic block models, clustering and segmentation in dynamic graphs*. Theses, Université Panthéon-Sorbonne - Paris I, November 2017. URL https://theses.hal.science/tel-01926276.

Marco Corneli, Pierre Latouche, and Fabrice Rossi. Exact icl maximization in a non-stationary temporal extension of the stochastic block model for dynamic networks. *Neurocomputing*, 192:81–91, June 2016. ISSN 0925-2312. doi: 10.1016/j.neucom.2016.02.031. URL http://dx.doi.org/10.1016/j.neucom.2016.02.031.

Caterina De Bacco, Eleanor A. Power, Daniel B. Larremore, and Cristopher Moore. Community detection, link prediction, and layer interdependence in multilayer networks. *Physical Review E*, 95(4), April 2017. ISSN 2470-0053. doi: 10.1103/physreve.95.042317. URL http://dx.doi.org/10.1103/PhysRevE.95.042317.

Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X*, 5:011027, Mar 2015. doi: 10.1103/PhysRevX.5.011027. URL https://link.aps.org/doi/10.1103/PhysRevX.5.011027.

Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Generalized louvain method for community detection in large networks. In *2011 11th international conference on intelligent systems design and applications*, pp. 88–93. IEEE, 2011.

Vishnu Manasa Devagiri, Veselka Boeva, Shahrooz Abghari, Farhad Basiri, and Niklas Lavesson. Multi-view data analysis techniques for monitoring smart building systems. *Sensors*, 21(20), 2021. ISSN 1424-8220. doi: 10.3390/s21206775. URL https://www.mdpi.com/1424-8220/21/20/6775.

Nathan Eagle and Alex (Sandy) Pentland. Reality Mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, 2006.

Nada Elgendy and Ahmed Elragal. Big data analytics: A literature review paper. volume 8557, pp. 214–227, 08 2014. ISBN 978-3-319-08975-1. doi: 10.1007/978-3-319-08976-8_16.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010a. ISSN 0370-1573. doi: https://doi.org/10.1016/j.physrep.2009.11.002. URL https://www.sciencedirect.com/science/article/pii/S0370157309002841.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, feb 2010b. doi: 10.1016/j.physrep.2009.11.002. URL https://doi.org/10.1016%2Fj.physrep.2009.11.002.

Zaynab Hammoud and Frank Kramer. Multilayer networks: aspects, implementations, and application in biomedicine. *Big Data Analytics*, 5, 07 2020. doi: 10.1186/s41044-020-00046-0.

Beibei Han, Yingmei Wei, Qingyong Wang, and Shanshan Wan. Dual adaptive learning multi-task multi-view for graph network representation learning. *Neural Networks*, 162:297–308, 2023.

Qiuyi Han, Kevin S. Xu, and Edoardo M. Airoldi. Consistent estimation of dynamic and multi-layer block models, 2015.

Zhao Kang, Guoxin Shi, Shudong Huang, Wenyu Chen, Xiaorong Pu, Joey Tianyi Zhou, and Zenglin Xu. Multi-graph fusion for multi-view spectral clustering, 2019.

Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1), January 2011. ISSN 1550-2376. doi: 10.1103/physreve.83.016107. URL http://dx.doi.org/10.1103/PhysRevE.83.016107.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pp. 381–388. AAAI Press, 2006. ISBN 9781577352815.

Alec Kirkley and M. E. J. Newman. Representative community divisions of networks. *Communications Physics*, 5(1), February 2022. ISSN 2399-3650. doi: 10.1038/s42005-022-00816-3. URL http://dx.doi.org/10.1038/s42005-022-00816-3.

Alec Kirkley, Alexis Rojas, Martin Rosvall, and Jean-Gabriel Young. Compressing network populations with modal networks reveal structural diversity. *Communications Physics*, 6(1), June 2023. ISSN 2399-3650. doi: 10.1038/s42005-023-01270-5. URL http://dx.doi.org/10.1038/s42005-023-01270-5.

Patricio La Rosa, Terrence Brooks, Elena Deych, Berkley Shands, F. Prior, Linda Larson-Prior, and William Shannon. Gibbs distribution for statistical analysis of graphical data with a sample application to fcmri brain images. *Statistics in medicine*, 35, 11 2015. doi: 10.1002/sim.6757.

Clement Lee and Darren J. Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1), dec 2019. doi: 10.1007/s41109-019-0232-2. URL https://doi.org/10.1007%2Fs41109-019-0232-2.

Wenzhe Li, Sungjin Ahn, and Max Welling. Scalable mcmc for mixed membership stochastic blockmodels, 2015.

Yixuan Li, Kun He, Kyle Kloster, David Bindel, and John Hopcroft. Local spectral clustering for overlapping community detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):1–27, 2018.

Matteo Magnani, Obaida Hanteer, Roberto Interdonato, Luca Rossi, and Andrea Tagarelli. Community detection in multiplex networks. *ACM Computing Surveys (CSUR)*, 54(3):1–35, 2021.

Domenico Mandaglio, Alessia Amelio, and Andrea Tagarelli. Consensus community detection in multilayer networks using parameter-free graph pruning. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*, pp. 193–205. Springer, 2018.

Anastasia Mantziou, Simon Lunagomez, and Robin Mitra. Bayesian model-based clustering for populations of network data, 2023.

Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010. doi: 10.1126/science.1184819. URL https://www.science.org/doi/abs/10.1126/science.1184819.

Mohamed Nadif and Gerard Govaert. Model-based co-clustering for continuous data. In *2010 Ninth International Conference on Machine Learning and Applications*, pp. 175–180, 2010. doi: 10.1109/ICMLA.2010.33.

Teng Niu, Shiai Zhu, Lei Pang, and Abdulmotaleb El Saddik. Sentiment analysis on multi-view social data. In *MultiMedia Modeling: 22nd International Conference, MMM 2016, Miami, FL, USA, January 4-6, 2016, Proceedings, Part II 22*, pp. 15–27. Springer, 2016.

Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Trans. Intell. Syst. Technol.*, 8(2), oct 2016. ISSN 2157-6904. doi: 10.1145/2915921. URL https://doi.org/10.1145/2915921.

Subhadeep Paul and Yuguo Chen. Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electronic Journal of Statistics*, 10(2), January 2016. ISSN 1935-7524. doi: 10.1214/16-ejs1211. URL http://dx.doi.org/10.1214/16-EJS1211.

Xinyu Que, Fabio Checconi, Fabrizio Petrini, and John A Gunnels. Scalable community detection with the louvain algorithm. In *2015 IEEE International Parallel and Distributed Processing Symposium*, pp. 28–37. IEEE, 2015.

Daniel M Roy, Charles Kemp, Vikash Mansinghka, and Joshua Tenenbaum. Learning annotated hierarchies from relational data. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper_files/paper/2006/file/663fd3c5144fd10bd5ca6611a9a5b92d-Paper.pdf.

Liangxun Shuo and Bianfang Chai. Discussion of the community detection algorithm based on statistical inference. *Perspectives in Science*, 7:122–125, 2016. ISSN 2213-0209. doi: https://doi.org/10.1016/j.pisc.2015.11.020. URL https://www.sciencedirect.com/science/article/pii/S2213020915000658. 1st Czech-China Scientific Conference 2015.

Natalie Stanley, Saray Shai, Dane Taylor, and Peter Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE Transactions on Network Science and Engineering*, 3, 07 2015. doi: 10.1109/TNSE.2016.2537545.

Andrea Tagarelli, Alessia Amelio, and Francesco Gullo. Ensemble-based community detection in multilayer networks. *Data Mining and Knowledge Discovery*, 31:1506–1543, 2017.

Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data mining and knowledge discovery*, 25:1–33, 2012.

Toni Vallè s-Català, Francesco A. Massucci, Roger Guimerà, and Marta Sales-Pardo. Multilayer stochastic block models reveal the multilayer structure of complex networks. *Physical Review X*, 6(1), mar 2016. doi: 10.1103/physrevx.6.011036. URL https://doi.org/10.1103%2Fphysrevx.6.011036.

Jan van der Laan, Edwin de Jonge, Marjolijn Das, Saskia Te Riele, and Tom Emery. A Whole Population Network and Its Application for the Social Sciences. *European Sociological Review*, 39(1):145–160, 06 2022. ISSN 0266-7215. doi: 10.1093/esr/jcac026. URL https://doi.org/10.1093/esr/jcac026.

Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2019.

Hao Wang, Yan Yang, and Bing Liu. Gmc: Graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1116–1129, 2020. doi: 10.1109/TKDE.2019.2903810.

James D. Wilson, John Palowitch, Shankar Bhamidi, and Andrew B. Nobel. Community extraction in multilayer networks with heterogeneous community structure, 2017.

Tianyu Xia, Yijun Gu, and Dechun Yin. Research on the link prediction model of dynamic multiplex social network based on improved graph representation learning. *IEEE Access*, 9:412–420, 2020.

Jean-Gabriel Young, Alec Kirkley, and M. E. J. Newman. Clustering of heterogeneous populations of networks. *Physical Review E*, 105(1), January 2022. ISSN 2470-0053. doi: 10.1103/physreve.105.014312. URL http://dx.doi.org/10.1103/PhysRevE.105.014312.

Xi-Nian Zuo, Jeffrey Anderson, Pierre Bellec, Rasmus Birn, Bharat Biswal, Janusch Blautzik, John Breitner, Randy Buckner, Vince Calhoun, Francisco Castellanos, Antao Chen, Bing Chen, Jiangtao Chen, Xu Chen, Stanley Colcombe, William Courtney, Cameron Craddock, Adriana Di Martino, Haoming Dong, and Michael Milham. An open science resource for establishing reliability and reproducibility in functional connectomics. *Scientific Data*, 1, 09 2015. doi: 10.1038/sdata.2014.49.

Katharina A. Zweig. *Network Representations of Complex Systems*, pp. 109–148. Springer Vienna, Vienna, 2016. ISBN 978-3-7091-0741-6. doi: 10.1007/978-3-7091-0741-6_5. URL https://doi.org/10.1007/978-3-7091-0741-6_5.

A Identifiability

The identifiability of the parameters for uni layer Bernoulli SBM has been proved in Celisse et al. (2012). The proof has been extended to a multiplex graph for pillar division Barbillon et al. (2017). We extend this analysis for multiplex DCSBM with multi groups.

Theorem A.1. *Let assume that there is K groups and every group has the same number of blocks $Q^k = Q^{k'} = Q\ \forall k, k' \in \{1, ..., Q\}^2$. Assume for any $q \in \{1, ..., Q\}, k \in \{1, ..., K\}$, $\alpha_q^k > 0, \beta^k > 0$. Let $\mathbf{\Pi} \in]0, 1[^{K*Q\times K*Q}$ diagonal block that contains matrices Π^k at diagonal as follow:*

$$\begin{bmatrix} \mathbf{\Pi^1} & ... & 0 \\ : & ... & : \\ 0 & ... & \mathbf{\Pi^K} \end{bmatrix}$$

Let also $\boldsymbol{\alpha}$ be a $K*Q \times K*Q$ matrix, which is the diagonilization of $[\alpha_1^1, ...\alpha_Q^1, ...\alpha_Q^K]$ vector, and $\boldsymbol{\beta}$ be a $K*Q \times K*Q$ matrix, which is the diagonilization of $[\beta^1, ...\beta^1, \beta^2...\beta^K]$ vector, where β^i is repeated Q times, $\forall i \in \{1, ..., K\}$. Assume that the elements of $\boldsymbol{r} = \mathbf{\Pi}.\boldsymbol{\alpha}.\boldsymbol{\beta}$ are distinct. Then the MDCSBM parameters are identifiable under equivalent solutions.

Proof. We consider the degree node heterogeneity parameter constant and extend the proof from Celisse et al. (2012) to the MDCSBM model as follows. For any group k, $r_{q,k}$ is the probability for a giving member from block q in group k to have a connection with another in the same group $r_{q,k} = \sum_{l=1}^{Q} \beta_k \pi_{ql}^k \alpha_l^k$. Let $\mathbf{R}$ be $Q*K$ square matrix such that $R_{i,q,k} = (r_{q,k})^i$ for $i \in 0, ..., Q*k-1$. R is a Vandermonde matrix that is invertible by assumptions.

Therefore, for any $i = 0, ..., (2Q-1)*K$, let set

$$\mu_i = \sum_{q,k} \alpha_{q,k} (r_{q,k})^i \tag{31}$$

and M is a $k(Q+1) \times KQ$ matrix such that

$$M_{ij} = \mu_{i+j} \tag{32}$$

For any $i = 0, ..., Q*k$, we define the Q*K square matrix M^i by removing line i from the matrix. In hence,

$$M^Q = R\boldsymbol{\alpha}R^T \tag{33}$$

where $\boldsymbol{\alpha}$ is $Q*K$ matrix as defined previously, where all $\alpha_q^k \neq 0$. Because R being invertible, then $det(M) > 0$. Let us define

$$B(X, \theta) = \sum_{i=0}^{Q\times K} (-1)^{i+Q*K} det(M^i(\theta))X^i \tag{34}$$

B is of degree $Q \times K$. For $V^i(\theta) = (1, r_i(\theta), ..., (r_i(\theta))^Q)$, then

$$B(r_i(\theta), \theta) = det(M(\theta), V_i(\theta)) \tag{35}$$

The column of M are linearly combinations of V_i, then $B(r_i(\theta), \theta) = 0$ for any i. It means that B can be factorized as follow:

$$B(x, \theta) = det(M^{Q\times K}) \prod_{i=0}^{KQ-1} (x - r_i(\theta)) \tag{36}$$

Let assume the $\boldsymbol{\theta} = (\mathbf{\Pi}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ and $\boldsymbol{\theta'} = (\mathbf{\Pi'}, \boldsymbol{\alpha'}, \boldsymbol{\beta'})$ are two sets of parameters such that for any multiplex $\mathcal{G}$ graph with multi-group model, $\mathcal{L}(\mathcal{G}; \boldsymbol{\theta}) = \mathcal{L}(\mathcal{G}; \boldsymbol{\theta'})$. Therefore, we get, $\mu_i(\boldsymbol{\theta}) = \mu_i(\boldsymbol{\theta'})$, that means that $M^i(\theta) = M^i(\theta')$ for any i. The $B(;\theta) = B(;\theta')$ because it dependents on the determinant of M, which leads to $r_i(\theta) = r_i(\theta')$. Ths $R(\theta) = R(\theta')$, and

$$\boldsymbol{\alpha}(\theta) = (R(\theta)^T)^{-1} M^{Q,K} R(\theta) = \boldsymbol{\alpha}(\theta') \tag{37}$$

Therefore $\boldsymbol{\alpha} = \boldsymbol{\alpha'}$. The same steps can be applied to proof the identifiability of $\boldsymbol{\beta}$ where the matrix diagonal $\boldsymbol{\alpha}$ is replaced by diagonal matrix of $\boldsymbol{\beta}$ where every $\beta_k, \forall k \in \{1, ..., K\}$ will be repeated Q times before set β_{k+1}. It leads to a matrix with the same dimension $Q \times K$.

For $\mathbf{\Pi}$, let's define

$$U_{ij} = R(\boldsymbol{\theta})\boldsymbol{\beta}(\boldsymbol{\theta})\boldsymbol{\alpha}(\boldsymbol{\theta})\mathbf{\Pi}\boldsymbol{\alpha}(\boldsymbol{\theta})\boldsymbol{\beta}(\boldsymbol{\theta})(R(\boldsymbol{\theta}))^T$$

From previously, $R(\theta) = R(\theta')$, $\boldsymbol{\alpha}(\boldsymbol{\theta}) = \boldsymbol{\alpha}(\boldsymbol{\theta}')$ and $\boldsymbol{\beta}(\boldsymbol{\theta}) = \boldsymbol{\beta}(\boldsymbol{\theta}')$ then

$$U(\boldsymbol{\theta}) = U(\boldsymbol{\theta}') \rightarrow \mathbf{\Pi} = \mathbf{\Pi}' \tag{38}$$

□

B Consistency of Maximum likelihood

The asymptotic consistency of the maximum likelihood estimator of Bernoulli uni layer SBM has been studied in Celisse et al. (2012); Bickel et al. (2013). The proof of the consistency of MDCSBM is straightforward from the proof of MSBM model, which can derived from the proof of uni-layer SBM. Let us assume that the following assumptions hold:

Assumption B.1. *For every $q \neq q'$,there exists $w \in \{1, ..., Q^k\}$ such that $\pi^k_{qw} \neq \pi^k_{q'w}$, or $\pi^k_{wq} \neq \pi^k_{wq'}$*

Assumption B.2. *There exists $\zeta > 0$ such that $\forall (q, w) \in \{1, ..., Q^k\}$, $\pi^k_{qw} \in]0, 1[\rightarrow \pi_{qw} \in [\zeta, 1 - \zeta]$*

Assumption B.3. *There exists $\gamma \in 1/Q^k$ such that $\forall q \in \{1, ..., Q^k\}$, $\alpha^k_q \in]0, 1[\rightarrow \alpha_q \in [\gamma, 1 - \gamma]$*

Assumption B.4. *There exists $\xi \in 1/K$ such that $\forall k \in \{1, ..., K\}$, $\beta^k \in]0, 1[\rightarrow \beta^k \in [\xi, 1 - \xi]$*

Theorem B.5. *Let (Θ, d) and (Ψ, d') denote metric spaces and let $M_n : \Theta \times \Psi \rightarrow \mathcal{R}$ be a random function and $M : \Theta \rightarrow \mathcal{R}$ a deterministic function such that for every $\epsilon > 0$*

$$sup_{d(\theta,\theta_0)} M(\theta) < M(\theta_0) \tag{39}$$

$$sup_{(\theta,\psi)\in\Theta\times\Psi} |M_n(\theta, \psi) - M(\theta)| := ||M_n - M||_{\Theta\times\Psi} \rightarrow 0 \tag{40}$$

and $(\hat{\theta}\hat{\psi}) = \underset{\theta,\psi}{argmax} M_n(\theta, \psi)$, then

$$d(\hat{\theta}, \theta_0) \rightarrow 0 \tag{41}$$

The proof can be performed by the same steps by taking

$$M_n(\pi, \alpha, \beta) = \frac{1}{N(N-1)L * K} \sum_{l=1}^{L} log\Big(\sum_k \sum_{z_{[n]}} \beta_k \prod_{i\neq j} \mathbf{Bernoulli}(\pi^k_{z^k_i, z^k_j}) \prod_i \alpha^k_{z^k_i}\Big) \tag{42}$$

$$M(\pi) = \max_{a_{i,j}\in\mathcal{A}} \sum_k \sum_{q,w} \beta^{*k} \alpha^{*k}_q \alpha^{*k}_w \sum_{q',w'} a^{*k}_{qq'} a^{*k}_{ww'} [\pi^{*k}_{q,w} log(\pi^k_{q',w'}) + (1 - \pi^{*k}_{q,w}) log(1 - \pi^{*k}_{q',w'})] \tag{43}$$

where $\mathbf{Bernoulli}(\pi)$ is the Bernoulli distribution with parameter π, and $\beta^*, \alpha^* and \ \ \pi^*$ denote the true parameters respectively, with

$$\mathcal{A} = \{(a^k_{ij})_{1\leq q,w\leq Q^k}, a^k_{qw} \geq 0, \sum_w a^k_{qw} = 1\} \tag{44}$$