The Multiplex Deep Latent Position Model for the Clustering of nodes in Multiview Networks

Dingge Liang, Marco Corneli, Charles Bouveyron, Pierre Latouche, Junping Yin

The Multiplex Deep Latent Position Model for the Clustering of nodes in Multiview Networks

Dingge Liang[a,c], Marco Corneli[d,e], Charles Bouveyron[d], Pierre Latouche[f], Junping Yin[a,b,c,*]

[a] *Institute of Applied Physics and Computational Mathematics, Beijing 100094, China*
[b] *National Key Laboratory of Computational Physics, Beijing 100088, China*
[c] *Shanghai Zhangjiang Institute of Mathematics, Shanghai 201203, China*
[d] *Université Côte d'Azur, INRIA, CNRS, Laboratoire J.A.Dieudonné, Maasai team, Nice, France*
[e] *Université Côte d'Azur, CEPAM, CNRS, Nice, France*
[f] *Université Clermont Auvergne, Institut Universitaire de France (IUF), CNRS, LMBP UMR 6620, Aubière, France*

Abstract

Multiplex networks capture multiple types of interactions among the same set of nodes, creating a complex, multi-relational framework. A typical example is a social network where nodes (actors) are connected by various types of ties, such as professional, familial, or social relationships. Clustering nodes in these networks is a key challenge in unsupervised learning, given the increasing prevalence of multiview data across domains. While previous research has focused on extending statistical models to handle such networks, these adaptations often struggle to fully capture complex network structures and rely on computationally intensive Markov chain Monte Carlo (MCMC) for inference, rendering them less feasible for effective network analysis. To overcome these limitations, we propose the multiplex deep latent position model (MDLPM), which generalizes and extends latent position models to multiplex networks. MDLPM combines deep learning with variational inference to effectively tackle both the modelling and computational challenges raised by multiplex networks. Unlike most existing deep learning models for graphs that require external clustering algorithms (e.g., k-means) to group nodes based on their latent embeddings, MDLPM integrates clustering di-

*Corresponding author
Email address: yinjp829829@126.com (Junping Yin)

rectly into the learning process, enabling a fully unsupervised, end-to-end approach. This integration improves the ability to uncover and interpret clusters in multiplex networks without relying on external procedures. Numerical experiments across various synthetic data sets and two real-world networks demonstrate the performance of MDLPM compared to state-of-the-art methods, highlighting its applicability and effectiveness for multiplex network analysis.

Keywords: Multiplex network analysis, Node clustering, Deep latent variable models, Graph neural networks

---

1. Introduction and related work

In the realm of network science, multiplex or multiview networks (Kivelä et al., 2014) represent an advanced analytical framework where nodes maintain multiple types of relationships or interactions simultaneously. This complexity is inherent in various real-world networks, such as social media platforms, where individuals can connect through different contexts like professional, familial, or social ties. A long series of statistical methods, such as Arroyo et al. (2021); Sosa and Betancourt (2022); MacDonald et al. (2022) as examples, have been developed to discover the underlying clusters of nodes in multiview networks by learning latent features for the nodes. More recently, deep-learning-based models (Hamilton et al., 2017; Zhang et al., 2018a) have emerged as a promising approach for analyzing multiplex networks and they have shown abilities for representation learning on data with complex structures. Hereafter, we split the existing approaches for node clustering in multiview networks into two categories and briefly review them.

Statistical models for multiview clustering. The latent position model (LPM), originally proposed by Hoff et al. (2002), posits that each node in a network occupies an unknown position in a multivariate continuous latent space, with the likelihood of a link between two nodes modeled as a function of their positions. Handcock et al. (2007) extended LPM to the latent position cluster model (LPCM), incorporating a clustering structure by assuming that node positions are drawn from a Gaussian mixture model. Recently, LPM has been adapted to accommodate multiview data, significantly enhancing its utility for analyzing complex interactions. Key developments include the work of Gollini and Murphy (2016), who extended LPM to multiplex

data by proposing that edge probabilities are functions of a single latent variable. Salter-Townshend and McCormick (2017) introduced a method to simultaneously model network structure and correlations among multiple networks through a multivariate Bernoulli model. Further advancing this field, D'Angelo et al. (2019) and their subsequent study D'Angelo et al. (2023) developed a Bayesian nonparametric model that automatically infers the number of clusters, dynamically adjusting based on the number of actors. Another significant strand of research involves stochastic block models (SBMs, Wang and Wong, 1987; Nowicki and Snijders, 2001), which have been widely used for node clustering (Lee and Wilkinson, 2019). Notable contributions include those by Ishiguro et al. (2010); Barbillon et al. (2017); Paul and Chen (2020), who proposed models that enrich the understanding of latent structures. However, these extended models often rely on Markov chain Monte Carlo (MCMC) methods for inference, which are computationally demanding and scale poorly with the size of the network, rendering these statistical methods impractical for effective applications. The hierarchical block distance model (HBDM, Nakis et al., 2023) employs a multi-scale block structure inspired by stochastic block modeling and latent distance models. HBDM captures homophily and transitivity in graph representation by organizing embedded clusters into a hierarchical tree structure, with the root representing the entire set of latent variable embeddings. This approach approximates the LPM across the hierarchy, providing a framework for multi-scale graph analysis.

Deep learning models for multiview clustering. Recent advances in graph neural networks (GNNs) have notably enhanced network representation learning and have been actively explored in multiview clustering. In this line of methods, CMNA (Chu et al., 2019) utilizes cross-network information to refine inter-vector for network alignment and intra-vector representations for other downstream tasks. Concurrently, mGCN (Ma et al., 2019) and HAN (Wang et al., 2019) employ graph convolutional networks (GCNs, Kipf and Welling, 2016a) and graph attention networks (GATs, Veličković et al., 2018), respectively, to address both inter and intra-network interactions within multiplex networks. Further, DMGI (Park et al., 2020) draws inspiration from deep graph infomax (DGI, Velickovic et al., 2019), which aims at maximizing the mutual information between local graph patches and the global graph representation. This approach facilitates the integration of embeddings from various types of node relations, enhancing the overall model flexibility. De-

spite these advancements, the aforementioned approaches, along with other notable methods (Liu et al., 2017; Zhang et al., 2018b, 2023), primarily focus on network embedding and typically rely on external clustering algorithms like k-means to group the embedded nodes. This reliance separates the clustering process from the generative model, potentially limiting the depth of integration and insight that can be achieved. O2MAC (Fan et al., 2020) represents a significant step forward by employing clustering directly in the attributed multi-view graph. However, it uniquely selects one graph view with the highest informative score as the primary model input. This selection process may lead to information loss by excluding other potentially valuable views, thereby constraining the overall clustering performance.

Main contributions. To address the limitations of existing models while exploring their advantages, we introduce the multiplex deep latent position model (MDLPM) for multiview network data. Within the framework of variational auto-encoders (VAEs, Kipf and Welling, 2016b), MDLPM combines deep learning graph models with statistical modeling, offering a flexible yet comprehensive tool for network analysis. The MDLPM encapsulates several key features:

- **Multiplex LPM-based decoder**: our method features a decoder that models the probability of interactions between node pairs across multiple network views, based on their distances in a joint latent space.
- **Integration of multiple GCNs**: MDLPM employs multiple graph convolutional networks within a variational inference framework. This integration addresses both the modelling and computational challenges in multiview network analysis, ensuring robust performance with complex data structures.
- **End-to-end clustering**: unlike traditional methods that require external clustering algorithms, MDLPM autonomously performs end-to-end clustering. It estimates posterior probabilities for cluster memberships directly, enabling the model to assign nodes to their groups without additional procedures.
- **Hybrid estimation algorithm**: the model incorporates a novel estimation algorithm that combines expectation-maximization, for explicit posterior clustering probabilities, with stochastic gradient descent, for

implicit graph reconstruction. This hybrid approach optimizes both the clustering performance and the fidelity of network representation.

2. The multiplex deep latent position model

In this section, we introduce the generative model we propose for the analysis of multiplex networks.

2.1. Notations

In the following, multiplex networks are modelled as a set of undirected, unweighted graphs $G^{(l)} = (V, E^{(l)})$, $l \in \{1, \cdots, L\}$, with $N = |V|$ nodes and L network views. We emphasize that the set of nodes V remains fixed across views. Then, we introduce an $N \times N$ adjacency matrix $A^{(l)}$, where $A^{(l)}_{ij} = 1$ if there is a link between node i and node j in the l-th view, 0 otherwise. Each node is associated with attribute information, which collectively forms the node feature matrix $X \in \mathbb{R}^{N \times F_v}$. The notation F_v is used to represent the number of dimensions of all the node features. Note that the set of edges $E^{(l)}$ can be associated with an additional covariate information, stored into the edge feature matrix $Y^{(l)} \in \mathbb{R}^{|E^{(l)}| \times F_e^{(l)}}$. The generic entry of $Y^{(l)}$, denoted $y_{ij}^{(l)}$, is a $F_e^{(l)}$-dimensional feature associated with the edge connecting i to j in the l-th view. For instance, $y_{ij}^{(l)}$ could encode the text that author i sent to author j in a communication network view. Our objective is to learn well-represented, latent, joint node embeddings, denoted as Z, within a space of reduced dimensionality P. Simultaneously, we aim at partitioning the nodes into K distinct clusters. Necessary notations are summarized in Table 1.

2.2. Generative model

The generative process of our model is described as follows. First, each node is supposed to be assigned to a cluster via a random variable c_i encoding its cluster membership

$$c_i \overset{\text{i.i.d}}{\sim} \text{Multinomial}(1, \pi), \quad \text{with} \quad \pi \in [0,1]^K, \ \sum_{k=1}^{K} \pi_k = 1. \tag{1}$$

Then, conditionally to its cluster membership, a latent embedding vector z_i is generated

$$z_i | c_{ik} = 1 \sim \mathcal{N}(\mu_k, \sigma_k^2 \mathbf{I}_P), \quad \text{with} \quad \sigma_k^2 \in \mathbb{R}^{+*}, \tag{2}$$

Table 1: List of model parameters and notations

Notation	Description
L	Number of network views
N	Number of nodes
K	Number of clusters
P	Latent space dimension
$A^{(l)}$	Adjacency matrix in $[0,1]^{N\times N}$ in l-th layer
X	Node feature matrix in $\mathbb{R}^{N\times F_v}$
$Y^{(l)}$	Edge feature matrix in l-th layer in $\mathbb{R}^{\lvert E^{(l)}\rvert \times F_e^{(l)}}$
π	Prior cluster probability vector
C	Cluster memberships
Z	Latent node embeddings in $\mathbb{R}^P$
δ_{ik}	Posterior probability that node i is in cluster k

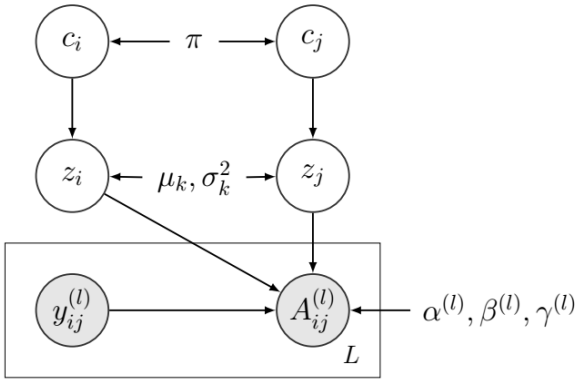


Figure 1: Graphical representation of MDLPM at the (i,j)-th pair level (variational parameters used for inference are not included).

independently for each node, where μ_k and σ_k^2 denote the mean as well as variance of each cluster, and $\mathbf{I}_P$ represents the identity matrix in $\mathbb{R}^P$.

Finally, the probability of a connection between nodes i and j within the l-th view, as indicated by the entry $A_{ij}^{(l)}$ in the adjacency matrix, is modelled using a Bernoulli random variable. This variable depends on the distance between the latent positions corresponding to each pair of nodes

$$A_{ij}^{(l)}|z_i, z_j \sim \text{Bernoulli}\left(g(\alpha^{(l)} + {\beta^{(l)}}^\top y_{ij}^{(l)} - \gamma^{(l)}\|z_i - z_j\|^2)\right), \quad \forall l \in \{1, \cdots, L\}, \tag{3}$$

where $\alpha^{(l)}, \gamma^{(l)} \in \mathbb{R}$ and $\beta^{(l)} \in \mathbb{R}^{F_e^{(l)}}$ are free parameters. Furthermore, the function g is the logistic sigmoid function, $y_{ij}^{(l)}$ is the covariate of the edge connecting i with j in view l, if available, and $\|\cdot\|^2$ is the squared Euclidean norm. A graphical representation of the generative model described so far can be seen in Figure 1.

3. Model inference

This section outlines the variational auto-encoding inference procedure and introduces an original estimation method which combines explicit optimization of the posterior clustering probabilities with implicit optimization of the neural network parameters.

3.1. Variational auto-encoding inference

Before getting into the details of the inference process, we first define the set of model parameters as

$$\Theta = \{\pi, \mu := (\mu_1, \cdots, \mu_K), \sigma := (\sigma_1^2, \cdots, \sigma_K^2), \alpha := (\alpha^{(1)}, \cdots, \alpha^{(L)}), \\ \beta := (\beta^{(1)}, \cdots, \beta^{(L)}), \gamma := (\gamma^{(1)}, \cdots, \gamma^{(L)})\}.$$

Moreover, we denote $Z = (z_1, \cdots, z_N)$ and $C = (c_1, \cdots, c_N)$ the set of all node positions and cluster memberships, respectively. The natural inference approach involves maximizing the integrated log-likelihood of the observed data, denoted as $A = (A^{(1)}, \cdots, A^{(L)})$, with respect to Θ (and, possibly, the optional covariate information $Y = (Y^{(1)}, \cdots, Y^{(L)})$, which is omitted to keep the notation uncluttered)

$$\log p(A|\Theta) = \log \int_Z \sum_C p(A, Z, C|\Theta) dZ. \tag{4}$$

Unfortunately, Eq. (4) is not tractable. Therefore, we propose a variational approach to approximate the log-likelihood

$$\log p(A|\Theta) = \mathcal{L}(q(Z, C); \Theta) + D_{KL}(q(Z, C)||p(Z, C|A, \Theta)), \tag{5}$$

where D_{KL} denotes the Kullback-Leibler divergence between the true and approximate posterior distributions of (Z, C) given the data and model parameters.

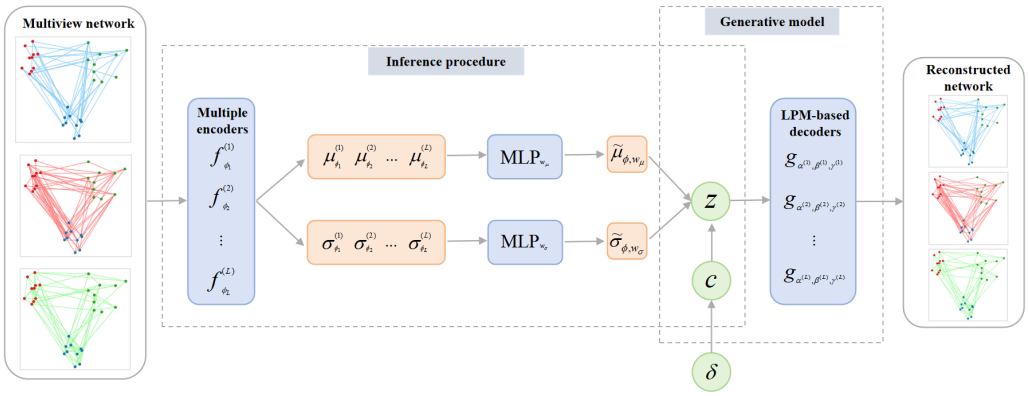


Figure 2: Architecture of MDLPM within the framework of variational auto-encoders.

In order to rely on a tractable family of distributions, we assume that $q(Z, C)$ fully factorizes under the mean-field assumption

$$q(Z, C) = q(Z)q(C) = \prod_{i=1}^{N} q(z_i)q(c_i). \tag{6}$$

To leverage the representational learning capabilities of graph neural networks, we consider multiple GCNs. Thus,

$$[\mu_{\phi_l}^{(l)}, \log \sigma_{\phi_l}^{(l)}] = f_{\phi_l}^{(l)}(\overline{A}^{(l)}, X), \tag{7}$$

where $f_{\phi_l}^{(l)}(\cdot)$ is the GCN for the l-th layer, which is parameterized by ϕ_l, specific to the l-th network view. This layer maps the normalized adjacency matrix, $\overline{A}^{(l)} = \tilde{D}^{(l)^{-\frac{1}{2}}}(A^{(l)} + \mathbf{I}_N)\tilde{D}^{(l)^{-\frac{1}{2}}}$, into a matrix of the variational means ($\mu_{\phi_l}^{(l)} \in \mathbb{R}^{N \times P}$) and vector of the log standard deviations ($\log \sigma_{\phi_l}^{(l)} \in \mathbb{R}^N$), for all nodes. We point out that $\tilde{D}^{(l)}$ is an $N \times N$ diagonal matrix, such that $\tilde{D}_{ii}^{(l)} = \sum_{j=1}^{N}(A_{ij}^{(l)} + \mathbb{I}_{\{i=j\}}), \forall i$. This construction of $\overline{A}^{(l)}$ aligns with the operations detailed in Kipf and Welling (2016a).

Then, to effectively aggregate information from L network views, we introduce two multi-layer perceptron (MLP) layers

$$\tilde{\mu}_{\phi,w_\mu} = \text{MLP}_{w_\mu}\left(\mu_{\phi_1}^{(1)}, \cdots, \mu_{\phi_L}^{(L)}\right), \log \tilde{\sigma}_{\phi,w_\sigma} = \text{MLP}_{w_\sigma}\left(\log \sigma_{\phi_1}^{(1)}, \cdots, \log \sigma_{\phi_L}^{(L)}\right), \tag{8}$$

where the mean vectors and the log standard deviation vectors from L network views are concatenated column-wise. These concatenated vectors then serve as the inputs for two distinct multi-layer perceptron layers, $\text{MLP}_{w_\mu}(\cdot)$ and $\text{MLP}_{w_\sigma}(\cdot)$, respectively.

Finally, given the integrated vectors $\tilde{\mu}_{\phi,w_\mu} \in \mathbb{R}^{N\times P}$ and $\log \tilde{\sigma}_{\phi,w_\sigma} \in \mathbb{R}^N$, we assume

$$q(z_i) = \mathcal{N}\left(z_i; (\tilde{\mu}_{\phi,w_\mu})_i, (\tilde{\sigma}_{\phi,w_\sigma})_i \mathbf{I}_P\right). \tag{9}$$

Moreover, a standard assumption is made for the variational clustering probabilities

$$q(C) = \prod_{i=1}^{N} \text{Multinomial}(c_i; 1, \delta_i), \tag{10}$$

where δ_{ik} represents the variational probability that node i is in cluster k, with $\sum_{k=1}^{K} \delta_{ik} = 1$. An overview of the model architecture is illustrated in Figure 2.

3.2. Optimization

In this part, we focus on maximizing the evidence lower bound (ELBO)

$$\mathcal{L}(A|\Theta) = \int_Z \sum_C q(Z,C) \log \frac{p(A,Z,C|\Theta)}{q(Z,C)} dZ, \tag{11}$$

with respect to the model parameters Θ and the variational parameters $\phi := (\phi_1, \cdots, \phi_L)$.

Proposition 1. *Thanks to Eq.* (6)-(9)-(10)*, the lower bound in Eq.* (11) *can be further developed as*

$$\begin{aligned}
\mathcal{L}(A|\Theta) &= \int_Z \sum_C q(Z,C) \log \frac{p(A|Z,\alpha,\beta,\gamma)p(Z|C,\mu,\sigma^2)p(C|\pi)}{q(Z,C)} dZ \\
&= \mathbb{E}\left[\log p(A|Z,\alpha,\beta,\gamma)\right] + \mathbb{E}\left[\log p(Z|C,\mu,\sigma^2\right] \\
&\quad + \mathbb{E}\left[\log p(C|\pi)\right] - \mathbb{E}\left[\log q(Z|A)\right] - \mathbb{E}\left[\log q(C)\right] \\
&= \mathbb{E}\left[\log p(A|Z,\alpha,\beta,\gamma)\right] + \mathbb{E}\left[\log \frac{p(Z|C,\mu,\sigma^2)}{q(Z)}\right] \\
&\quad + \mathbb{E}\left[\log \frac{p(C|\pi)}{q(C)}\right]
\end{aligned}$$

$$= \frac{1}{2}\,\mathbb{E}\left[\sum_{l=1}^{L}\sum_{i\neq j} A_{ij}^{(l)}\log\eta_{ij}^{(l)} + (1-A_{ij}^{(l)})\log(1-\eta_{ij}^{(l)})\right]$$
$$- \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik} D_{KL}\left(\mathcal{N}\left((\tilde{\mu}_{\phi,w_\mu})_i, (\tilde{\sigma}^2_{\phi,w_\sigma})_i \mathbf{I}_P\right)||\mathcal{N}\left(\mu_k, \sigma_k^2\mathbf{I}_P\right)\right)$$
$$+ \sum_{i=1}^{N}\sum_{k=1}^{K}\delta_{ik}\log(\frac{\pi_k}{\delta_{ik}}),$$

where $\eta_{ij}^{(l)} = g(\alpha^{(l)} + {\beta^{(l)}}^\top y_{ij}^{(l)} - \gamma^{(l)}\|z_i - z_j\|^2)$ *and the expectation is taken with respect to the variational probability* $q(\cdot)$.

Proposition 2. *An explicit optimization of the ELBO with respect to the parameters* $\delta_{ik}, \pi_k, \mu_k$ *and* σ_k *can be performed to obtain the following updates*

$$\hat{\delta}_{ik} = \frac{\pi_k \mathrm{e}^{-D_{KL}^{ik}}}{\sum_{l=1}^{K}\pi_l \mathrm{e}^{-D_{KL}^{il}}}, \tag{12}$$

where $D_{KL}^{ik} = \frac{1}{2}\left\{\log\frac{(\sigma_k^2)^P}{(\tilde{\sigma}^2_{\phi,w_\sigma})_i^P} - P + \frac{(\tilde{\sigma}^2_{\phi,w_\sigma})_i}{\sigma_k^2} + \frac{1}{\sigma_k^2}\|\mu_k - (\tilde{\mu}_{\phi,w_\mu})_i\|^2\right\}$.

Moreover,

$$\hat{\pi}_k = \sum_{i=1}^{N}\delta_{ik}/N, \qquad \hat{\mu}_k = \sum_{i=1}^{N}(\tilde{\mu}_{\phi,w_\mu})_i\delta_{ik}/\sum_{i=1}^{N}\delta_{ik} \tag{13}$$

and

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N}\delta_{ik}\left(P(\tilde{\sigma}^2_{\phi,w_\sigma})_i + \|\mu_k - (\tilde{\mu}_{\phi,w_\mu})_i\|^2\right)}{P\sum_{i=1}^{N}\delta_{ik}}. \tag{14}$$

More detailed derivations are given in the Appendix A.

On the other hand, the *implicit optimization* of the encoder parameters ϕ, the two parameters w_μ and w_σ of MLP layers, and decoder parameters α, β, γ, is performed via stochastic gradient descent. In this work, it is implemented using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $5e^{-3}$. In MDLPM, each of the multiple GCN encoders features a first hidden layer with 64 neurons and a second hidden layer with 16 neurons, using

ReLU activation for the initial layer. The output from these encoders is concatenated, resulting in a dimension of $L \times 16$, where L is the number of views. This concatenated output serves as input to two MLP layers, which process it to produce latent embeddings with a final dimension of $P = 16$. Each of the decoders consists of a single-layer neural network that maps these latent embeddings back into a reconstructed graph, utilizing a sigmoid activation function for the output layer. We also point out that during the estimation, a reparameterization trick as in Kingma and Welling (2014) is used for the term $\mathbb{E}\left[\log p(A|Z, \alpha, \beta, \gamma)\right]$.

3.3. Model selection

The ELBO introduced in the previous section allows one to both estimate the (variational) posterior law of (Z, C) as well as the model parameters Θ for a fixed number of clusters K. If this hyper-parameter is changed, the other estimates also might no longer be the same. Therefore, choosing appropriate values for K can be seen as a model selection task. In order to select K we introduce IC2L, a new model selection criterion

$$\begin{aligned} IC2L(K) = \sum_{l=1}^{L} \max_{\alpha^{(l)},\beta^{(l)},\gamma^{(l)}} \log p(A^{(l)}|\hat{Z}, \alpha^{(l)}, \beta^{(l)}, \gamma^{(l)}) - \left(2L + \sum_{l}^{L} F_e^{(l)}\right) \log \frac{N(N-1)}{2} \\ + \max_{\mu,\sigma^2} \log p(\hat{Z}|\hat{C}, \mu, \sigma^2, K, P) - \frac{KP + K}{2} \log(N) \\ + \max_{\pi} \log p(\hat{C}|\pi, K) - \frac{K-1}{2} \log(N), \end{aligned}$$

where $\hat{Z}$ and $\hat{C}$ are the maximum-a-posteriori estimates of Z and C, respectively, obtained from the maximization of the ELBO for a given K and a given P.

This model selection can be obtained similarly to what done (for instance) by Daudin et al. (2008) for stochastic block models. The main intuition is that it can be seen as the counterpart of the BIC criterion, with the complete-data log-likelihood replacing the observed-data log-likelihood. The three penalty terms above contain a first term accounting for the number of model parameters and a second term accounting for the logarithm of the number of observations. In Section 4.5, we conduct several experiments to asses the capacity of IC2L to retrieve the actual number of clusters on simulated datasets. Two additional remarks. First, although MDLPM might benefit from some self-regularization properties due to the use of variational auto-encoding inference, as reported in Kingma et al. (2016); Dai et al. (2017),

we opt here for a standard approach based on asymptotic Bayesian model selection (via IC2L). Second, since IC2L is also dependent on P it might be used to select the number of latent dimensions. However we leave the check for this conjecture for future work and consider here P as given.

4. Numerical experiments

This section demonstrates the effectiveness of our model through empirical evaluations on five synthetic scenarios and the validity of the estimation algorithm proposed.

4.1. Simulation setup

We designed five types of synthetic multiplex networks. In the following, we manipulate these networks by varying the clusters proximity and network similarity, respectively.

Cluster proximity. We first generate three different types of synthetic networks, each of which having three groups of nodes.

- Scenario A consists of three communities, with positions of nodes in a two-dimensional space, generated with the mean of each cluster equal to

$$\begin{cases} \mu_1 = [0, 0] \\ \mu_2 = [1.5 * \zeta_1, 1.5 * \zeta_1] \\ \mu_3 = [-1.5 * \zeta_1, 1.5 * \zeta_1] \end{cases}$$

  respectively. The hyper-parameter $\zeta_1 \in [0.2, 0.95]$ represents the rate of proximity between the clusters, where a larger ζ_1 means that the three clusters are better separated. An example is shown in Figure 3. For three network views, different $\alpha^{(l)}, \beta^{(l)}$ and $\gamma^{(l)}$ are selected

$$\begin{cases} \alpha^{(1)} = -1.5, \beta^{(1)} = 0, \gamma^{(1)} = 0.1 \\ \alpha^{(2)} = -0.2, \beta^{(2)} = 0, \gamma^{(2)} = 0.5 \\ \alpha^{(3)} = 0.2, \beta^{(3)} = 0, \gamma^{(3)} = 1 \end{cases}$$

  Then, edges are generated based on the distance within each node pair, in dimension two. In order to simplify the simulations, we do not consider the edge covariates $Y^{(l)}$, ($\beta^{(l)} = 0$). Finally, three distinct adjacency matrices are built based on Eq. (3), with parameter values depending on the network views.

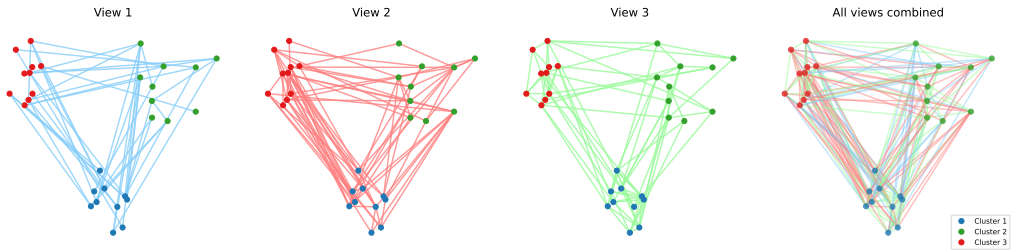


Figure 3: Synthetic network visualization for scenario A, featuring $N = 30$ nodes and a proximity rate ζ_1 of 0.95. The network displays three types of connections and is organized into three distinct clusters of nodes, each represented by a different color.

- Scenario B considers three clusters interacting with each other in a complex way, far from being a community structure. For each network view, different connection probabilities are considered

$$\Pi^{(l)} = \begin{pmatrix} b^{(l)} & a^{(l)} & a^{(l)} \\ a^{(l)} & a^{(l)} & b^{(l)} \\ a^{(l)} & b^{(l)} & a^{(l)} \end{pmatrix}$$

  where

$$\begin{cases} a^{(1)} = 0.25,\ b^{(1)} = 0.01 + (1 - \zeta_2) * (a^{(1)} - 0.01) \\ a^{(2)} = 0.15,\ b^{(2)} = 0.02 + (1 - \zeta_2) * (a^{(2)} - 0.02) \\ a^{(3)} = 0.1,\ b^{(3)} = 0.03 + (1 - \zeta_2) * (a^{(3)} - 0.03) \end{cases}$$

  Three types of edges are then generated based on the connection matrices. Similarly, we introduce a parameter $\zeta_2 \in [0.4, 1.0]$ in order to measure the degree of closeness between clusters, where a larger ζ_2 means less overlap among the three clusters.

- Scenario C considers latent positions created from three circular-structured clusters. Three circles have the same center and the different radius are 1, 5, and 10, respectively. For different network views, links are then generated based on the distance between the node positions, with the same network parameters $\alpha^{(l)}, \beta^{(l)}$ and $\gamma^{(l)}$ as in scenario A.

By varying the values of ζ_1 in scenario A (assortative) and ζ_2 in scenario B (dissortative), we can model the proximity between each cluster and thus test the robustness of our model in both simple and difficult cases. Then,

contrary to standard communities, with strong transitivity (your-friend-is-my-friend effect), scenario C describes the construction of three groups of nodes with little transitivity in each.

Network similarity. We defined two additional parameters ζ_3 in scenario D and ζ_4 in scenario E, to model the similarity among different network views. A larger value for these parameters indicates greater similarity between the views.

- Scenario D consists of three communities, with positions of nodes being generated with the mean of each cluster as in scenario A. Then, for three network views, different $\alpha^{(3)}, \beta^{(3)}$ and $\gamma^{(3)}$ are selected

$$\begin{cases} \alpha^{(1)} = -1.5, \beta^{(1)} = 0, \gamma^{(1)} = 0.1 \\ \alpha^{(2)} = 0.2, \beta^{(2)} = 0, \gamma^{(2)} = 1 \\ \alpha^{(3)} = \zeta_3 * \frac{(\alpha^{(1)}+\alpha^{(2)})}{2} + (1-\zeta_3) * \frac{(\gamma^{(1)}+\gamma^{(2)})}{2}, \\ \quad \beta^{(3)} = 0, \gamma^{(3)} = \zeta_3 * \frac{(\gamma^{(1)}+\gamma^{(2)})}{2} + (1-\zeta_3) * \frac{(\alpha^{(1)}+\alpha^{(2)})}{2} \end{cases}$$

  Similarly, we do not account for edge covariates $Y^{(l)}$, setting $\beta^{(l)} = 0$ for each network view. The hyper-parameter ζ_3, ranging from 0 to 1, quantifies the degree of similarity between the networks.

- Scenario E considers three clusters interacting with each other in a complex way, far from being a community structure. For each network view, different connection probabilities are set as

$$\begin{cases} a^{(1)} = 0.25,\ b^{(1)} = 0.01 \\ a^{(2)} = 0.15,\ b^{(2)} = 0.02 \\ a^{(3)} = 0.1,\ b^{(3)} = \zeta_4 * 0.03 + (1-\zeta_4) * (a1 + a2 + a3 - b1 - b2) \end{cases}$$

  where the parameter $\zeta_4 \in [0, 1]$ measures the degree of similarity between the network views.

4.2. Comparaison with "shallow" multilayer approaches

First, we compared the MDLPM we propose with two established models: SpaceNet, a multiview statistical model based on LPM as described by D'Angelo et al. (2019), and MultiplexSBM, an extension of SBM to multiplex networks, detailed in Barbillon et al. (2017). Since SpaceNet is not an end-to-end clustering method, we performed k-means over the learned node embeddings for clustering.

Table 2: Comparison of clustering performance and computational time.

	Scenario A (with $\zeta_1 = 0.95$)		Scenario B (with $\zeta_2 = 0.9$)		Scenario C	
	ARI	Time(s)	ARI	Time(s)	ARI	Time(s)
SpaceNet	0.5465±0.270	2434.47	0.6693±0.186	2367.54	0.2440±0.075	2522.63
MultiplexSBM	0.5354±0.115	10692.49	1.0000±0.000	275.37	0.6407±0.058	5376.38
MDLPM	0.9508±0.018	143.78	0.9993±0.004	142.17	1.000±0.000	149.45

Settings. In the experiments described below, we simulated 25 distinct networks for scenarios A, B and C. Each network consists of $L = 3$ views, $N = 600$ nodes, and the nodes are spread into $K = 3$ clusters according to the settings defined by the three simulated scenarios.

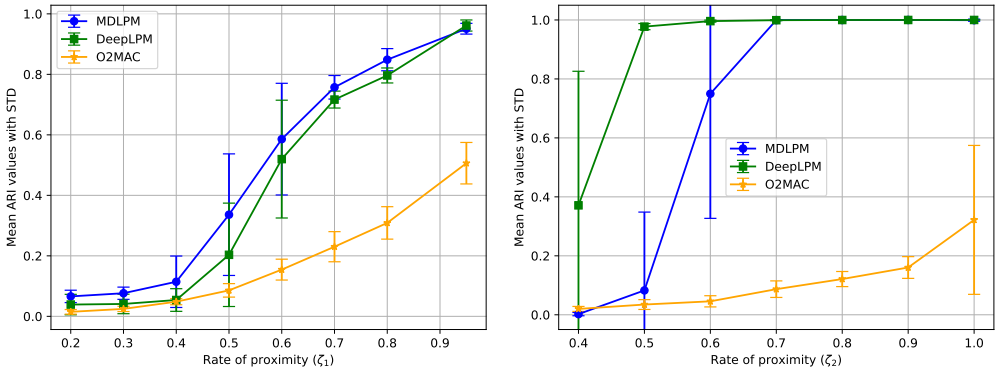
The clustering results and the computational times are shown in Table 2. As it can be seen, MDLPM consistently outperforms SpaceNet across all scenarios, achieving higher adjusted rand indexes (ARIs, Hubert and Arabie, 1985). While MultiplexSBM performs well in scenario B, it struggles with other network structures in scenarios A and C. Additionally, MDLPM is approximately 17 times faster in computation than SpaceNet, which utilizes a time-consuming MCMC inference procedure. The training time for MultiplexSBM is significantly influenced by the number of edges, resulting in slow performance on dense networks, particularly noted in scenarios A and C.

Given the significant gap between SpaceNet, MultiplexSBM and our model (especially in terms of computational times), henceforth we only consider deep learning based competitors that match MDLPM in terms of algorithmic efficiency.

4.3. Comparaison with deep approaches

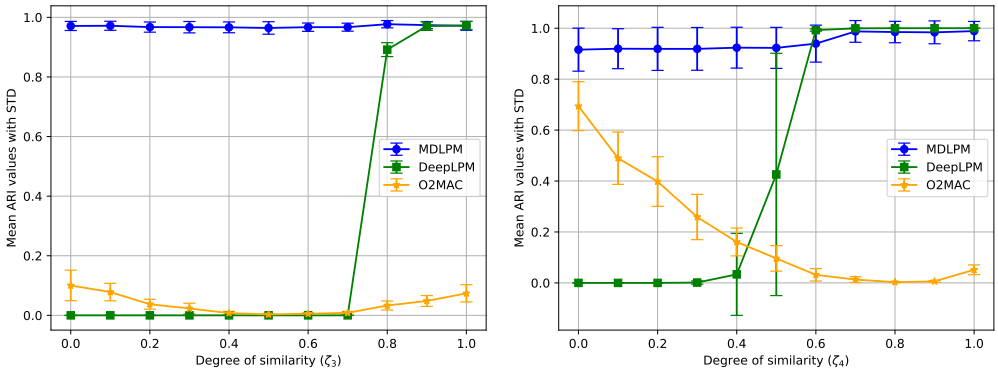
We now aim at bench-marking MDLPM with two deep models: O2MAC (Fan et al., 2020) and DeepLPM (Liang et al., 2024), in the synthetic scenarios A, B and C. It is important to note that DeepLPM is originally designed for single-network analysis. To adapt it for multiview networks, we first aggregate all network views into a single adjacency matrix by summing them up and converting non-zero entries to 1, thus making the matrix binary.

Firstly, Figures 4a and 4b illustrate the evolution of the ARIs as a function of the nuisance parameters (ζ_1 and ζ_2) in scenarios A and B, respectively. In scenario A, our MDLPM consistently outperforms its competitors. DeepLPM has quite similar performances whereas O2MAC is unable to effectively retrieve the three clusters in this situation. In scenario B, both MDLPM and DeepLPM exhibit strong performance for proximity rates ζ_2 greater than 0.6.



(a) Scenario A. (b) Scenario B.

Figure 4: (4a) ARI values with standard deviation for varying proximity rates ζ_1 in Scenario A. (4b) ARI values with standard deviations for varying proximity rates ζ_2 in Scenario B. For each value of ζ_1 and ζ_2, 25 distinct networks are generated and the average ARIs are computed.



(a) Scenario D. (b) Scenario E.

Figure 5: (5a) ARI values with standard deviation for varying similarity rates ζ_3 in scenario D. (5b) ARI values with standard deviations for varying similarity rates ζ_4 in scenario E. 25 distinct networks are generated for each value of ζ_3 and ζ_4, and the average ARIs are calculated.

However, in more challenging conditions where $\zeta_2 \leq 0.6$, DeepLPM outperforms MDLPM. This is due to the fact that summing the network views in input to DeepLPM favorably reduced the data sparsity without modifying the underlying network structure. O2MAC again fails to accurately identify the true clusters. Scenario C sees all models performing well, with MDLPM

Table 3: Comparison of clustering performance in scenario C

Models	DeepLPM	O2MAC	MDLPM
ARI	0.9728±0.095	0.9845±0.009	1.000±0.000

displaying remarkable stability and achieving an ARI of 1, indicating perfect clustering, as shown in Table 3.

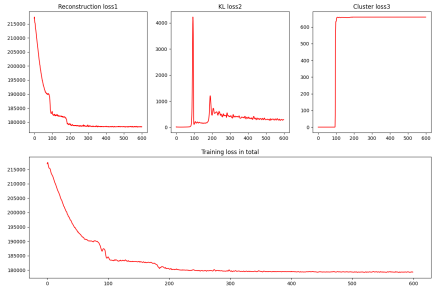
We further compare the three deep models in scenarios D and E to examine how network similarity across different views impacts clustering performance. As shown in Figures 5a and 5b, DeepLPM fails to detect clusters when network similarity degree is below 0.8 in scenario D and 0.6 in scenario E, making it suitable only for multiplex networks with similar structures across views. O2MAC is not suited for networks in scenario D, but outperforms DeepLPM in scenario E when network similarity is smaller than 0.4, indicating it is more suitable for dissimilar network views. MDLPM achieves significantly higher ARI values in both scenarios, outperforming DeepLPM and O2MAC, which demonstrates excellent clustering performance with distinct network structures.

4.4. A detailed illustration example

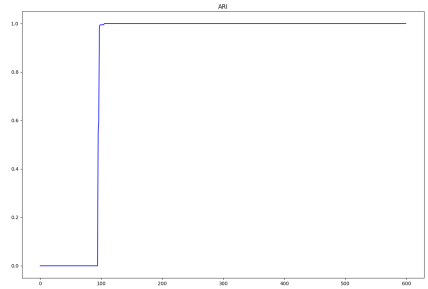
In this subsection, we provide a more detailed illustration for Scenario A, with $\zeta_1 = 0.95$. Figures 6a and 6b track the evolution of the training loss and the clustering performance (ARI values), respectively. Both metrics are observed to converge after 600 epochs, indicating stable model performance. Figure 6c displays the original node positions, with different clusters represented by distinct colors. Finally, we visualize the latent embeddings generated by the proposed MDLPM. These embeddings are mapped to a two-dimensional space using t-SNE (Van der Maaten and Hinton, 2008) for visualization, as shown in Figure 6d. The clear recovering of three clusters in the latent space illustrates the model clustering capability, where the learned embeddings successfully preserve this network structure by retaining the closeness between latent positions.

4.5. Model selection

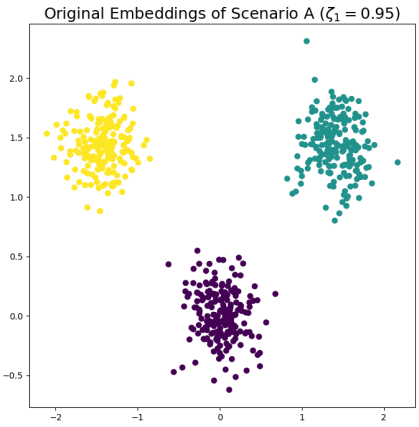
We now highlight the ability of our model selection criteria IC2L to correctly select the number of groups K.



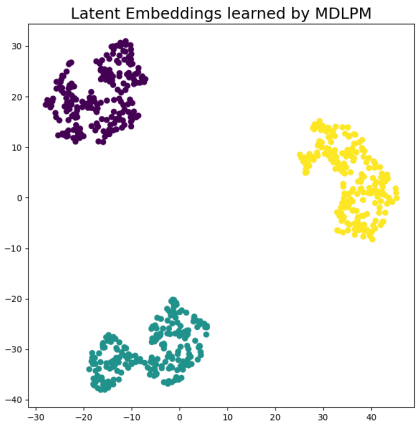
(a) Training loss (negative ELBO).



(b) Clustering ARI.



(c) Original node embeddings.



(d) Embeddings learned by MDLPM.

Figure 6: A detailed illustration of scenario A with $\zeta_1 = 0.95$: (6a) Evolution of the training loss; (6b) Evolution of ARI values. (6c) Original node positions, with clusters represented by distinct colors; (6d) Latent embeddings learned by MDLPM, visualized in two dimensions.

Number of clusters. Letting the number of clusters vary from 2 to 7, we demonstrate in the following how the IC2L criterion can be used to estimate the number of clusters. In this experiment, for each specific cluster value, we produced 25 synthetic networks from scenario A. We then trained MDLPM using a latent embedding dimension of $P = 16$. Firstly, for each network, we selected the largest IC2L value from 5 distinct initializations. Figure 7 displays the results from a random synthetic network, with the standard deviations indicating variance across different initializations. Subsequently,

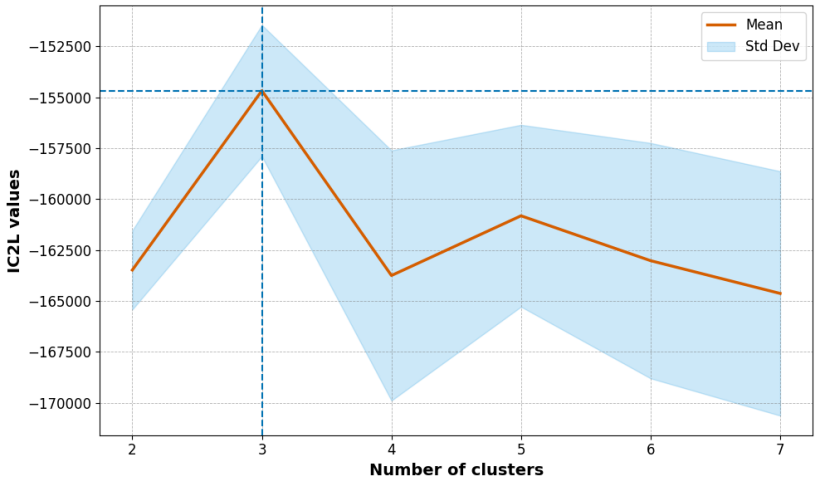


Figure 7: IC2L criterion computed for different number of clusters on one synthetic network in scenario A, for 5 independent initializations. MDLPM was able to estimate $K = 3$ by displaying a maximum of the IC2L value.

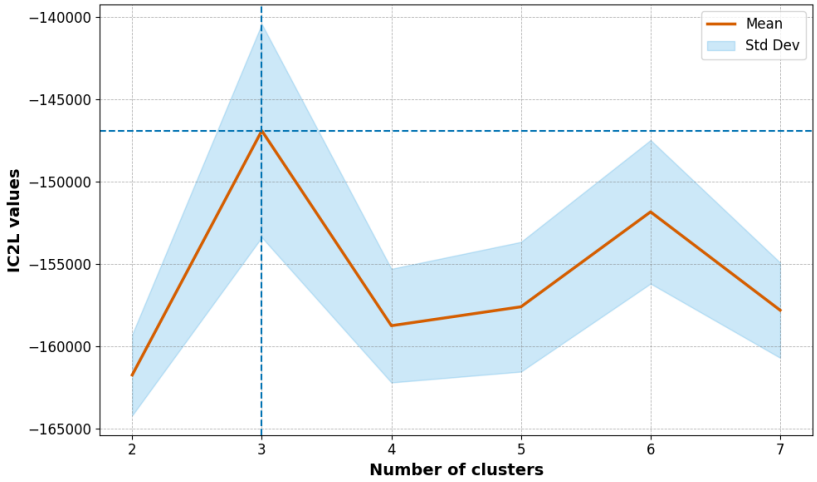


Figure 8: IC2L criterion computed for different number of clusters on 25 synthetic networks in scenario A. GETM was able to estimate $K = 3$ by displaying a maximum of the IC2L value.

we calculated the average IC2L values over the 25 simulated networks. The standard deviations reported in Figure 8 reflect the variance among different networks. Observably, for both the results on a single specific network and

across 25 networks, the largest and average IC2L values achieve their maximum at $K = 3$, thereby recovering the actual value of K in the simulation setting.

5. Unsupervised analysis on real-world networks

In this section, we fit (and test) MDLPM to two real data sets: the Enron network and the DBLP multiplex network.

5.1. Enron email network

We first consider the Enron email dataset, a classical communication network that contains all email exchanges between the firm employees between 1999 and 2002. In the network, nodes represent individuals, and an undirected edge is established between two nodes i and j if the employee i sent at least one email to individual j. For our experiments, we focused on the data from September to December 2001 to form a multiplex network characterized by two types of connections. The first layer represents direct "to" connections, comprising 1,688 links, while the second layer represents "cc" (carbon copy) connections, consisting of 673 links. Each layer of this network contains $N = 148$ nodes, representing the employees.

Model selection. MDLPM was fitted to the Enron email network for different numbers of clusters, ranging between 2 and 7, with fixed dimension ($P = 16$) for the latent space, and we used IC2L for selecting the number K of clusters. The evolution of the IC2L criterion with various cluster numbers is shown in Figure 9. The reported result represents the IC2L values achieved after executing MDLPM 15 times, with standard deviations accounting for variations due to different initializations for each cluster number. Finally, the estimated number of clusters is $K = 4$ displayed by a clear maximum of the IC2L value.

Visualisation and analysis. To compare the clustering results between two multiview deep models: MDLPM and O2MAC, we set the number of clusters to $K = 4$. Then, we employed t-SNE to reduce the latent embeddings to a two-dimensional space for visualization purposes. As illustrated in Figures 10a and 10b, MDLPM effectively separates clusters of nodes in the latent space, with each of the four groups distinctly colored. Central nodes, identified by the largest degree within each cluster, are represented with a larger node size compared to others. In contrast, as shown in Figures 11a and 11b,

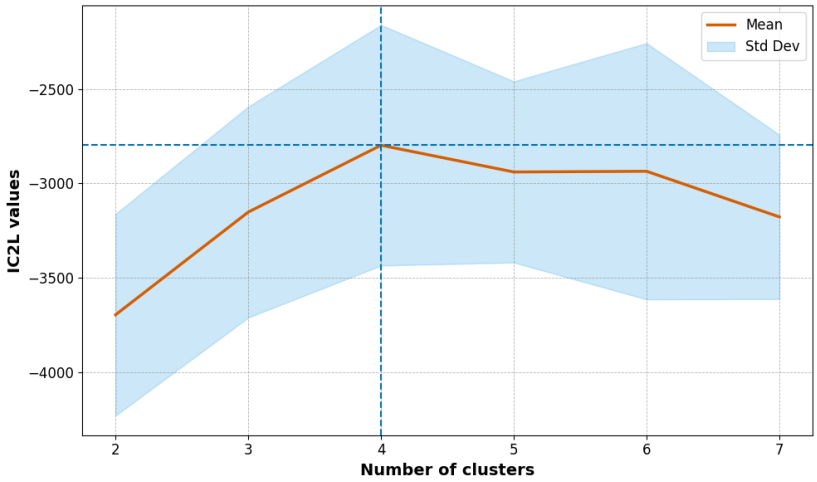
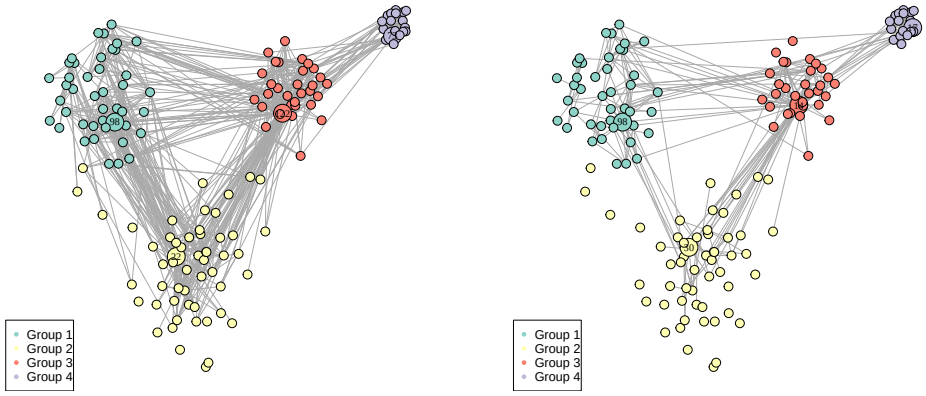


Figure 9: IC2L criterion with different number of clusters on the Enron email network.



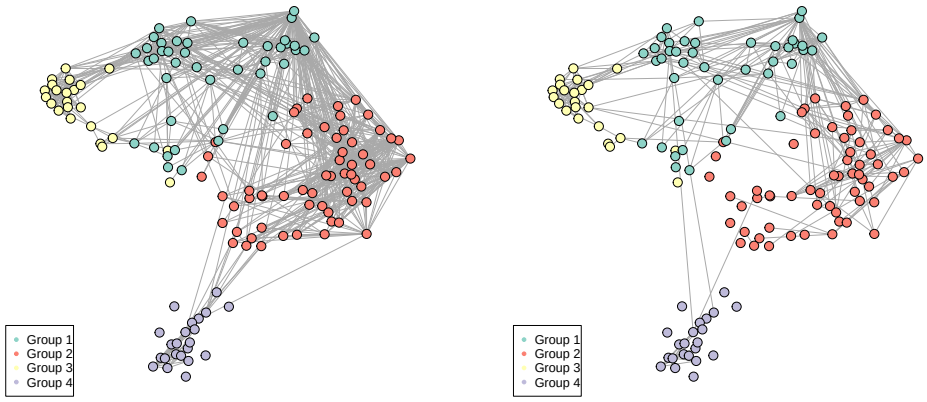
(a) MDLPM layer 1.

(b) MDLPM layer 2.

Figure 10: (10a) Visualization of the first view of the Enron email network as learned by **MDLPM**; (10b) Visualization of the second view. In both, each cluster is distinctly represented by different colors, and central nodes are marked by a larger node size to denote their higher degree within the cluster.

O2MAC primarily displays one clearly defined group in purple, while the remaining embeddings are not very well-separated.

To further delve into the visualization results of MDLPM, we examine the



(a) O2MAC layer 1.

(b) O2MAC layer 2.

Figure 11: (10a) Visualization of the first view of the Enron email network as learned by **O2MAC**; (10b) Visualization of the second view. Each cluster is represented by different colors.

Table 4: Detailed information of central nodes in the Enron network

Network layer 1					
node_id	**firstName**	**lastName**	**email_id**	**status**	**cluster**
98	Mike	Grigsby	mike.grigsby@enron.com	Manager	1
22	Kam	Keiser	kam.keiser@enron.com	Employee	2
122	Sally	Beck	sally.beck@enron.com	Employee	3
73	Drew	Fossum	drew.fossum@enron.com	Vice President	4
Network layer 2					
node_id	**firstName**	**lastName**	**email_id**	**status**	**cluster**
98	Mike	Grigsby	mike.grigsby@enron.com	Manager	1
30	Jonathan	Mckay	jonathan.mckay@enron.com	Director	2
14	Louise	Kitchen	louise.kitchen@enron.com	President	3
17	Kimberly	Watson	kimberly.watson@enron.com	N/A	4

central nodes in each cluster to understand their roles and influence within the network. More detailed information is provided in Table 4.

In the first network view:

- the central node of Group 1 (cyan), node 98, is identified as Mike Grigsby, a manager at Enron. Notably, he was also recognized as one

of the top-10 prestigious employees at Enron in 2001. His role highlights his importance within the network structure.

- node 22 in Group 2 (yellow) is Kam Keiser, noted for being one of the top-10 prestigious employees at Enron in 2002. His extensive communications throughout the network are indicative of his key role.

- node 122 is identified as Sally Beck, the Chief Operating Officer (COO) of Enron. Her responsibilities included overseeing the daily operations of the company, a role that typically reports directly to the Chief Executive Officer (CEO). As a central node in Group 3 (red), the data shows that she handled a significant volume of messages, aligning with the expectations of her position and her central role in managing the company operations.

- the central node of Group 4 (purple), node 73, is identified as Drew Fossum, who served as a vice president at Enron. His prominent position within the network underscores his significant influence and connectivity.

In addition, in the second network view:

- the central node of Group 1 (cyan), node 98, is consistently identified across views as Mike Grigsby, further highlighting his prominent role at Enron.

- node 30, the central node of Group 2 (yellow), is Jonathan Mckay, a director at Enron, indicating his substantial influence within the network.

- the central node of Group 3 (red), node 14, represents Louise Kitchen, who served as the president and chief operating officer of EnronOnline. Her significant role within the company is reflected in her numerous connections across the network, highlighting her deep engagement and significant influence. Additionally, she was recognized as the second most prestigious employee at Enron in 2001, further underscoring her prominent standing within the organization.

- in Group 4 (purple), node 17 is Kimberly Watson. Although specific details of her role at Enron were not fully documented, her work on

pipeline facility planning is similar as Lynn Blair, who handled gas pipeline logistics and held the position of director. This parallel suggests that Watson likely held a significant role at Enron.

Moreover, we analyzed the learned model parameters $\alpha^{(l)}, \gamma^{(l)}$ for each layer l on Enron, $\beta^{(l)} = 0$ since we do not have covariate information in this dataset. The parameters we obtained are $\alpha^{(1)} = 1.4104$, $\alpha^{(2)} = -0.8638$, $\gamma^{(1)} = 1.0378$ and $\gamma^{(2)} = 0.2508$, providing insights into the structure of the two layers in the Enron network. In the first layer, the positive $\alpha^{(1)}$ indicates a relatively dense network structure, and the $\gamma^{(1)}$ shows a strong influence of node distances in the latent space on the likelihood of link formation. In the second layer, the negative $\alpha^{(2)}$ suggests a sparser structure compared to the first. Fewer links are present overall, implying that connections are less frequent. The much lower $\gamma^{(2)}$ indicates that node distances in the latent space have a weaker influence on link formation in this layer. These findings show that the two layers of the Enron network exhibit different structures, with the first layer being more cohesive and structured, while the second layer is more diffuse and less reliant on node proximity for forming connections.

6. DBLP collaboration network

In this section, we analyze a network of scientific authors extracted from the DBLP dataset[1], with authors divided into four research areas: database, data mining, machine learning, and information retrieval. The research area labels are assigned based on the conferences to which the authors submitted their work (Xiao et al., 2019). The network is represented as a multiplex structure with three layers: co-authorship (authors who collaborated on papers), co-conference (authors who published at the same conference), and co-term (authors who published papers with the same terms). This multiplex DBLP network contains $N = 4,057$ nodes (authors), with 11,113 links in the co-authorship layer, 5,000,495 links in the co-conference layer, and 6,772,278 links in the co-term layer. Each node in the dataset has features represented as a bag-of-words for keywords. Using this information, we construct the covariate matrix $y_{ij}^{(l)}$, which represents the cosine similarity between the feature vectors of nodes i and j. For simplicity, we assume the covariate matrix remains the same across all three layers.

[1] `https://dblp.uni-trier.de/`

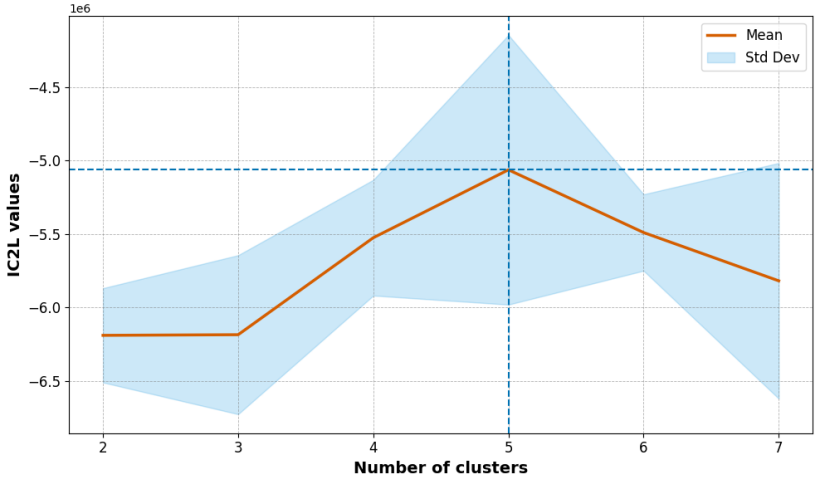


Figure 12: IC2L criterion computed for different number of clusters on DBLP network. MDLPM estimates $K = 5$ by showing the maximum of the IC2L value.

Model selection. MDLPM was fitted to the DBLP network for different numbers of clusters, ranging between 2 and 7, with fixed dimension ($P = 16$) for the latent space, and we used IC2L for selecting the number K of clusters. The evolution of the IC2L criterion with various cluster numbers is shown in Figure 12. The reported result represents the IC2L values achieved after executing 5 times for each value of K, with standard deviations accounting for variations dues to different initializations for each cluster number. Finally, the estimated number of clusters is $K = 5$ by displaying a clear maximum of the IC2L value.

Confusion matrix. We first plot the confusion matrix for the predicted labels with $K = 5$ clusters and the actual author research areas, as shown in Figure 13, to examine the distribution of authors across clusters. The introduction of a covariate information, which represents the node's similarity, leads to some interesting groupings. Group 2, Group 4, and Group 5 primarily consist of authors from machine learning, database, and information retrieval, respectively. However, we also observe that Group 1 contains a mix of individuals from information retrieval and authors from other areas, suggesting shared similarities in their research content. Group 3 is particularly notable, as it includes a large proportion of data mining authors along with contributors from other fields. This is likely due to the overlap of com-

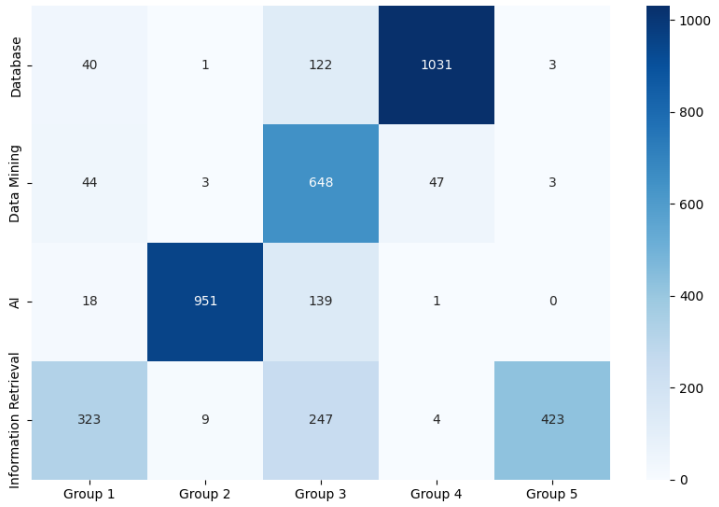


Figure 13: Confusion matrix of predicted clusters compared to research areas.

mon terms in papers related to machine learning, database, and information retrieval, all of which intersect with data mining. This demonstrates that the incorporation of covariate information, such as keyword-based similarity between authors, results in a clustering that is not solely based on research area, allowing MDLPM to capture patterns that would otherwise remain hidden.

Figure 14 further illustrates the distribution of the research areas within each cluster. It provides a visual breakdown of the predominant research areas in each group. For example, Group 4 is dominated by authors from the database field, while Group 2 is primarily composed of machine learning researchers. Interestingly, Group 3 contains a substantial number of data mining authors, along with contributors from other research areas. This highlights the interconnectedness of these fields, further supporting the observation that certain authors may work across multiple domains or share similar research interests based on the content of their publications.

Together, these figures demonstrate that the integration of node features via their similarity, encoded as covariate information, allows MDLPM to capture meaningful patterns that extend beyond predefined research areas, leading to more insightful clustering results.

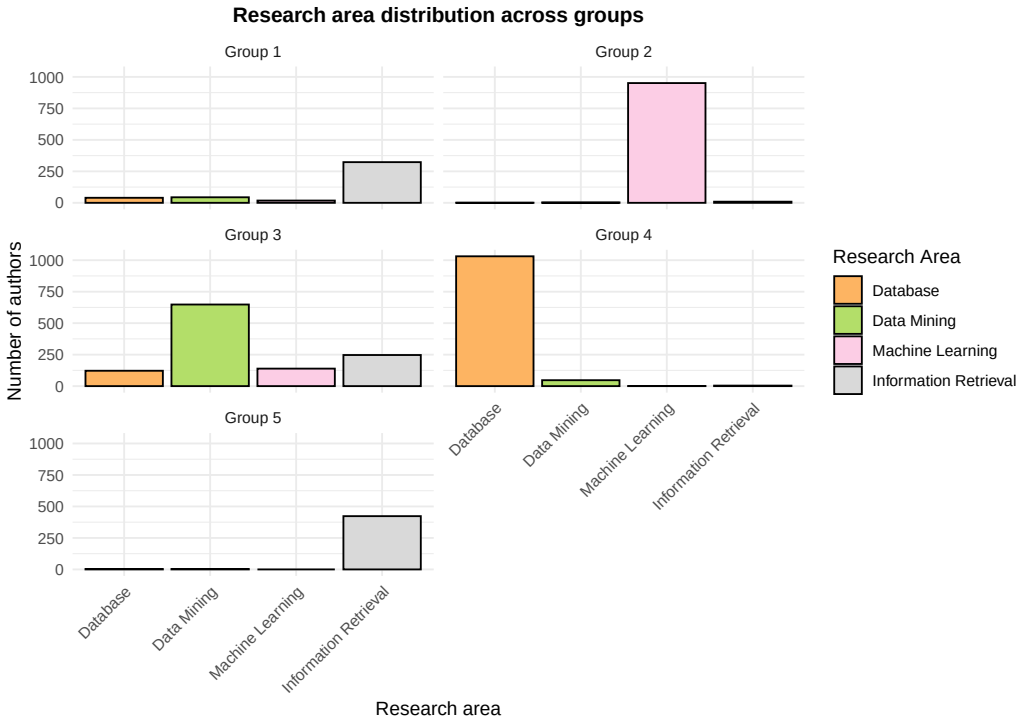
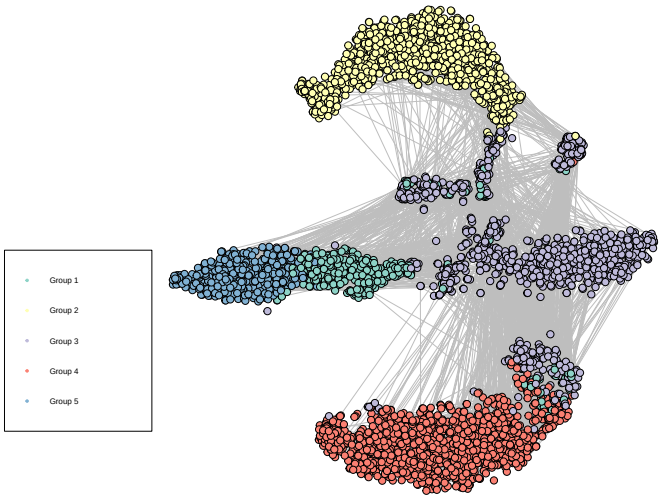


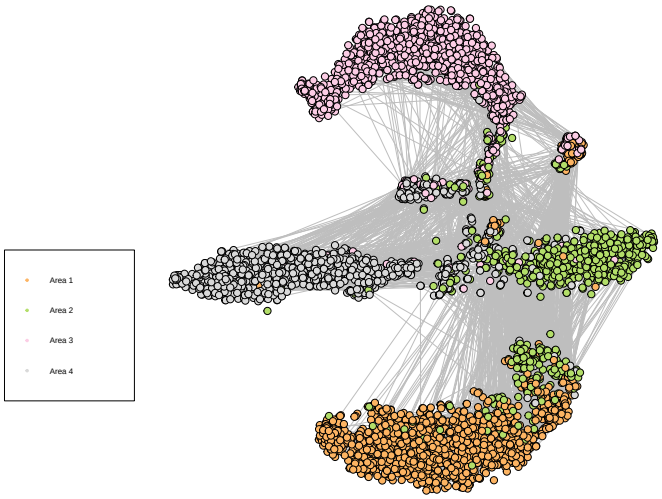
Figure 14: Author's research area distribution across predicted clusters.

Visualisation and analysis. Figure 15a presents the visualization of the first view of the DBLP network learned by MDLPM, with t-SNE applied to the latent embeddings. The node colors represent the different clusters identified by MDLPM. It is worth noting that we display only the embeddings of the first view, as the node positions are shared across the three views, but the connection types differ. The number of links in the second and third views is too large, making them difficult to visualize effectively. Figure 15b displays the embeddings of the first view, but with node colors now based on the author research areas. The inclusion of keyword similarity between node pairs as covariates uncovers more hidden information within the clusters. Notably, groups such as Group 1 and Group 3 now contain a broader mix of research areas, illustrating that in unsupervised problems, the number of clusters cannot be inferred solely from the number of predefined classes (research areas), as is often the case in supervised tasks.

Next, to better understand the clustering results, more analysis on the obtained clusters are performed. Figure 16 presents the number of links



(a) Co-authorship network with learned cluster labels.



(b) Co-authorship network with author research areas.

Figure 15: Embedding visualization of the first view of the DBLP network, with each node indicating an author. (15a) Colors indicate the five clusters learned by MDLPM. (15b) Node color indicates one of the four research areas: Area 1 (Machine Learning), Area 2 (Data Mining), Area 3 (Database), and Area 4 (Information Retrieval).

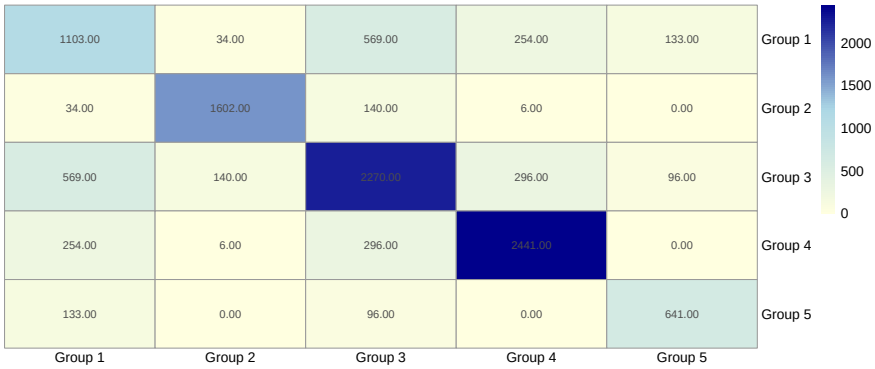
within and between clusters in the three layers of the multiplex DBLP network: (16a) Co-authorship network, (16b) Co-conference network, and (16c) Co-term network. The differences in interaction patterns across these layers reflect distinct network structures, which are further explained by the model parameters $\alpha^{(l)}, \beta^{(l)}, \gamma^{(l)}$ for each layer l from Table 5.

- Co-authorship network (layer 1): in this layer, we observe that most interactions (i.e., co-authorship links) occur within clusters, indicating a clear community structure. For example, Group 3 has a substantial number of internal links (2,270), suggesting strong internal collaboration among authors in this cluster. Other groups, such as Group 1 and Group 4, also show prominent internal links (1,103 and 2,441, respectively). There are relatively few interactions between clusters, reinforcing the notion that co-authorship ties are more concentrated within communities. The model parameter $\alpha^{(1)} = 2.4066$ for this layer reflects a denser structure, $\beta^{(1)} = -0.5285$, which is related to covariate information (keyword similarity), is lower in this layer, showing that keyword similarity plays a lesser role compared to the other two layers. The parameter $\gamma^{(1)} = 1.5497$ suggests that the distance between nodes significantly affects the likelihood of collaboration. Authors who are closer in the latent space are more likely to co-author papers, reinforcing the strong within-cluster community structure.

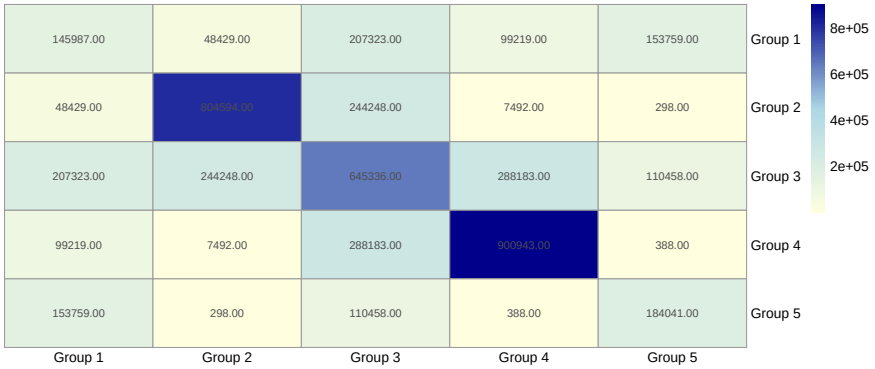
- Co-conference network (layer 2): this layer reveals a more complex structure, where we still observe some community patterns, but certain clusters exhibit stronger connections to other groups. For instance, Group 3 shows a substantial number of links within its cluster (645,336), but also significant interactions with Group 1 (207,323) and Group 4 (288,183). Group 4 has a high number of internal links (900,543), indicating a dense cluster, but it also maintains notable cross-cluster links, particularly with Group 3 and Group 1. This sug-

Table 5: Model parameters of each network layer

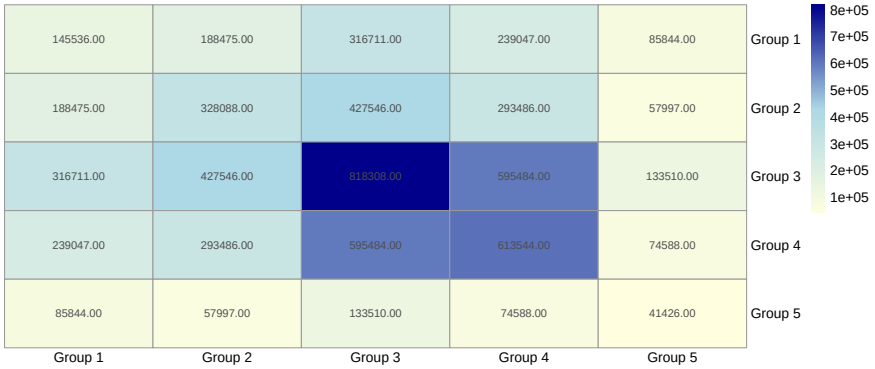
	Co-authorship network	Co-conference network	Co-term network
$\alpha^{(l)}$	2.4066	0.7490	-0.2085
$\beta^{(l)}$	-0.5285	3.3120	3.5475
$\gamma^{(l)}$	1.5497	0.0183	0.0043



(a) Co-authorship network (layer 1).



(b) Co-conference network (layer 2).



(c) Co-term network (layer 3).

Figure 16: Number of links within each cluster in the multiplex DBLP network.

gests that co-conference participation encourages both within-cluster and cross-cluster collaboration. The model parameter $\alpha^{(2)} = 0.7490$ is lower than in the co-authorship network, indicating a slightly less dense structure, while $\beta^{(2)} = 3.3120$ highlights the importance of covariate information, showing that keyword similarity significantly influences the connections in this layer. The parameter $\gamma^{(2)} = 0.0183$ is much lower than in the co-authorship layer, meaning that node distances in the latent space have a minimal impact on link formation in this layer.

- Co-term network (layer 3): the structure in this layer is distinctly different from the previous two. While there are still strong internal connections (e.g., Group 3 with 818,308 internal links and Group 4 with 613,544), the interaction between different clusters is much more pronounced. For example, Group 3 and Group 2 share a considerable number of links (427,546), indicating a significant overlap in terms between these clusters. This suggests that the co-term network reflects a broader thematic overlap among research areas, where authors from different clusters share common terms and keywords, even though they may not directly collaborate. The model parameters $\alpha^{(3)} = -0.2085$ and $\gamma^{(3)} = 0.0043$ further emphasize the distinctive structure of this layer. The low $\alpha^{(3)}$ value suggests a more dispersed structure compared to the co-authorship and co-conference networks. The parameter $\gamma^{(3)}$ is the smallest among all layers, indicating that node distance in the latent space has very little effect on link formation. Additionally, the high $\beta^{(3)} = 3.5475$ reflects the significant role of keyword similarity, further reinforcing the observation that co-term interactions are driven by shared research interests rather than spatial proximity in the latent space.

Overall, the three views demonstrate varying levels of community structure. In the co-authorship network, interactions are primarily within clusters, spatial proximity in the latent space plays a significant role, showing strong community structures. On the other hand, the co-conference and co-term networks, rely more on covariate information (keyword similarity) to form links, leading to more cross-cluster interactions and broader thematic overlaps.

Furthermore, as shown in Figure 15, nodes associated with Area 4 (information retrieval) are primarily divided into Group 1 and Group 5 in the

Table 6: Detailed information of top-5 nodes in Group 1 and 5

Group 1				
node_id	**firstName**	**lastName**	**degree**	**research area**
2951	Jiawei	Han	45	2
3363	Christos	Faloutsos	42	2
471	Qiang	Yang	35	3
1558	Wei-Ying	Ma	34	4
547	Drew	Fossum	29	1
Group 5				
node_id	**firstName**	**lastName**	**degree**	**research area**
4	Azin	Ashkan	4	4
647	Gavin	Jancke	4	4
911	Daniel	C. Robbins	4	4
1137	Prasenjit	Majumder	4	4
1299	Kazuko	Kuriyama	4	4

latent space. To explore the differences between these two clusters, we examined the node degrees for both groups, focusing on the top-5 highest degree nodes, as detailed in Table 6. As we can see, in Group 1, the top-5 authors have notably high node degrees. Additionally, Group 1 consists of authors from diverse research areas, including database (Area 1), data mining (Area 2), and machine learning (Area 3). This suggests that Group 1 represents a cross-disciplinary collaboration network, with authors actively collaborating across various domains. In contrast, Group 5 consists solely of authors from the information retrieval field (Area 3), with all top-5 nodes having a degree of 4. This indicates a more specialized research community, with fewer cross-domain interactions compared to Group 1. The lower node degrees in Group 5 further highlight the specific nature of this group, which is more homogeneous in its research focus.

We also investigated Group 3, which contains a majority of authors from data mining, along with a number of authors from the other three research areas. To explore the interdisciplinary connections within this group, we examined the top author from each research area, based on node degree, as shown in Table 7.

- For author Weiguo Fan, his work focuses on the intersection of informa-

Table 7: Detailed information of top-1 nodes in Group 3 from 4 areas

Group 3				
node_id	**firstName**	**lastName**	**degree**	**research area**
1162	Weiguo	Fan	20	4
1100	Alon	Y. Halevy	18	1
3366	Aoying	Zhou	13	2
1323	Bruce	G. Buchanan	9	3

tion retrieval and data mining, with papers such as "Mining the web for answers to natural language questions" and "Intelligent fusion of structural and citation-based evidence for text classification". These works illustrate his use of data mining techniques to address challenges in information retrieval, highlighting his relevance to both areas.

- Alon Y. Halevy has published papers like "Efficiently ordering query plans for data integration" and "Learning to map between ontologies on the semantic web", where the term "data" frequently appears. His work, often centered on data integration and knowledge representation, shares strong thematic overlaps with data mining, particularly in processing and extracting insights from large datasets.

- Aoying Zhou has published works such as "Mining functional dependency rules of relational databases" and "Incremental mining of schema for semi-structured Data". The keyword "mining" features prominently in his research, underlining his close ties to the data mining field, making him a central figure in Group 3 in the data mining area.

- Finally, Bruce G. Buchanan has contributed to knowledge-based systems, with papers like "Learning intermediate concepts in constructing a hierarchical knowledge base" and "Partial compilation of strategic knowledge". His work on hierarchical knowledge structures shares similarities with data mining, where extracting and organizing knowledge from large datasets is a core objective.

In summary, the interdisciplinary nature of Group 3 is evident in how these authors from distinct research areas incorporate data mining techniques and themes into their work. This suggests that Group 3 serves as a hub for

cross-domain collaboration, with data mining as the unifying theme that connects authors from machine learning, information retrieval and database.

7. Conclusion

We introduced the multiplex deep latent position model (MDLPM), a novel approach for performing node clustering on multiview network data in an end-to-end manner. By integrating multiple GCN encoders with multiplex LPM-based decoders, our model combines the interpretability of statistical models with the robust representation learning capabilities of graph neural networks. An original estimation procedure was developed that merges explicit optimization via variational inference, with implicit optimization using stochastic gradient descent. Numerical experiments demonstrated the clustering performance of MDLPM compared to state-of-the-art methods. Real-world applications, exemplified by our analysis of the Enron network and the DBLP network, underscore the practical utility of our model for unsupervised analysis. While the current focus is on identifying and analyzing non-overlapping clusters, future research will also consider extending MDLPM to more effectively handle scenarios involving overlapping clusters.

8. Acknowledgement

This work was supported by the National Postdoctoral Researcher Support Program of China [grant number GZC20242191]; the Major Program of National Natural Science Foundation of China [grant numbers 12292980, 12292984, 12031016]; the National Key R&D Program of China [grant numbers 2023YFA1009000, 2023YFA1009004, 2020YFA0712203, 2020YFA0712201]; the Beijing Natural Science Foundation [grant number BNSF-Z210003]; and the Department of Science, Technology and Information of the Ministry of Education [grant number 8091B042240].

Appendix A. Derivatives of the ELBO

We perform the explicit optimization of the ELBO with respect to the parameters $\delta_{ik}, \pi_k, \mu_k$ and σ_k by calculating the derivatives of the ELBO.

Under the equality constraint $\sum_{k=1}^{K} \delta_{ik} = 1, \forall k$, we use the method of Lagrange multipliers. Firstly, we introduce a Lagrange multiplier λ_i

$$\tilde{\mathcal{L}} = \mathcal{L} - \sum_{i=1}^{N} \lambda_i \left(\sum_{k=1}^{K} \delta_{ik} - 1 \right), \tag{A.1}$$

then, we derive $\tilde{\mathcal{L}}$ according to δ_{ik}

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \delta_{ik}} = \log \pi_k - \log \delta_{ik} - \frac{\delta_{ik}}{\delta_{ik}} - D_{KL}^{ik} - \lambda_i = 0, \tag{A.2}$$

thus, we have

$$\log \delta_{ik} = \log \pi_k - 1 - D_{KL}^{ik} - \lambda_i \tag{A.3}$$

$$\delta_{ik} = \mathrm{e}^{\{\log \pi_k - 1 - D_{KL}^{ik} - \lambda_i\}} = \frac{\mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}}}{\mathrm{e}^{\{1+\lambda_i\}}}. \tag{A.4}$$

By using the constraint on $\sum_{k=1}^{K} \delta_{ik}$, we can get

$$\sum_{k=1}^{K} \delta_{ik} = \frac{\sum_{k=1}^{K} \mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}}}{\mathrm{e}^{\{1+\lambda_i\}}} = 1 \tag{A.5}$$

$$\log \sum_{k=1}^{K} \mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}} = \log \mathrm{e}^{\{1+\lambda_i\}} \tag{A.6}$$

$$\lambda_i = \log \sum_{k=1}^{K} \mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}} - 1. \tag{A.7}$$

After putting the value of λ_i into Eq. A.4

$$\delta_{ik} = \frac{\mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}}}{\mathrm{e}^{\{1+\log \sum_{k=1}^{K} \mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}} - 1\}}} = \frac{\mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}}}{\sum_{k=1}^{K} \mathrm{e}^{\{\log \pi_k - D_{KL}^{ik}\}}}. \tag{A.8}$$

Finally, we obtain

$$\hat{\delta}_{ik} = \frac{\pi_k \mathrm{e}^{-D_{KL}^{ik}}}{\sum_{l=1}^{K} \pi_l \mathrm{e}^{-D_{KL}^{il}}}. \tag{A.9}$$

Similarly, since $\sum_{k=1}^{K} \pi_k = 1, \forall k$, we introduce another Lagrange multiplier τ

$$\tilde{\mathcal{L}} = \mathcal{L} - \tau \left(\sum_{k=1}^{K} \pi_k - 1 \right), \tag{A.10}$$

then, we derive $\tilde{\mathcal{L}}$ according to π_k

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \pi_k} = \sum_{i=1}^{N} \frac{\delta_{ik}}{\pi_k} - \tau = 0, \tag{A.11}$$

next, we use the equality constraint to solve τ

$$\sum_{k=1}^{K} \sum_{i=1}^{N} \delta_{ik} = \sum_{k=1}^{K} \pi_k \tau \tag{A.12}$$

$$\tau = N, \tag{A.13}$$

and finally, we have

$$\hat{\pi}_k = \sum_{i=1}^{N} \delta_{ik} / N. \tag{A.14}$$

Lastly, we need to calculate the derivatives for μ_k and σ_k^2. We start by deriving $\tilde{\mathcal{L}}$ according to μ_k

$$\frac{\partial \tilde{\mathcal{L}}}{\partial \mu_k} = -\frac{1}{2} \sum_{i=1}^{N} \delta_{ik} \{ \frac{1}{\sigma_k^2} \left(2\mu_k - 2(\tilde{\mu}_{\phi, w_\mu})_i \right) \} = 0, \tag{A.15}$$

then, we obtain

$$\mu_k \sum_{i=1}^{N} \delta_{ik} = \sum_{i=1}^{N} (\tilde{\mu}_{\phi, w_\mu})_i \delta_{ik} \tag{A.16}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} (\tilde{\mu}_{\phi, w_\mu})_i \delta_{ik}}{\sum_{i=1}^{N} \delta_{ik}}, \tag{A.17}$$

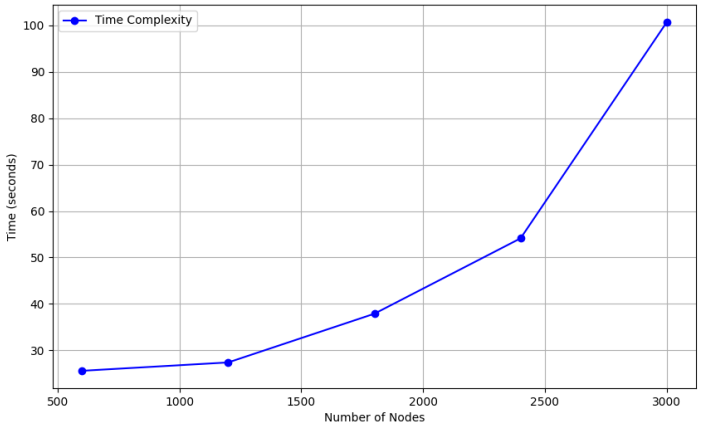


Figure B.17: Computational complexity versus number of nodes.

and finally for σ_k^2, we have

$$\begin{aligned}
\frac{\partial \tilde{\mathcal{L}}}{\partial \sigma_k^2} = -\frac{1}{2}\sum_{i=1}^{N}\delta_{ik}\{\frac{P}{\sigma_k^2} - \frac{1}{\sigma_k^4}\left(P(\tilde{\sigma}^2_{\phi,w_\sigma})_i + \|\mu_k - (\tilde{\mu}_{\phi,w_\mu})_i\|^2\right)\} = 0 \\
P\sum_{i=1}^{N}\frac{\delta_{ik}}{\sigma_k^2} = \sum_{i=1}^{N}\frac{\delta_{ik}}{\sigma_k^4}\left(P(\tilde{\sigma}^2_{\phi,w_\sigma})_i + \|\mu_k - (\tilde{\mu}_{\phi,w_\mu})_i\|^2\right) \\
P\sum_{i=1}^{N}\delta_{ik}\sigma_k^2 = \sum_{i=1}^{N}\delta_{ik}\left(P(\tilde{\sigma}^2_{\phi,w_\sigma})_i + \|\mu_k - (\tilde{\mu}_{\phi,w_\mu})_i\|^2\right) \\
\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N}\delta_{ik}\left(P(\tilde{\sigma}^2_{\phi,w_\sigma})_i + \|\mu_k - (\tilde{\mu}_{\phi,w_\mu})_i\|^2\right)}{P\sum_{i=1}^{N}\delta_{ik}}.
\end{aligned} \tag{A.18}$$

Appendix B. Computational time

The computational time for the simulated network with 600 nodes and 75,292 edges across three network views, is roughly 0.04s per epoch, resulting in a total of 25.59s for 600 epochs on a GeForce RTX 4070 Ti GPU. We also evaluated larger datasets, with node counts ranging from 600 to 3,000. The corresponding computation times are displayed in Figure B.17. As seen, the

largest dataset we analyzed consisted of 3,000 nodes and the training time was approximately 100.67s. For the Enron network, which consists of 148 nodes and 2,361 edges across two network views, the training takes about 0.15 seconds per epoch, around 180 seconds for 1,200 epochs on the same GPU. For the DBLP network, consisting of 4,057 nodes, with 11,113 links in the first layer, 5,000,495 links in the second layer, and 6,772,278 links in the third layer, the training process takes approximately 1.30 seconds per epoch, totaling 1,058 seconds for 800 epochs on the same GPU. This finding demonstrates the scalability of our approach and its capability to handle larger datasets, while still maintaining a reasonable training time.

Appendix C. Implementation details

In this paper, SpaceNet and MultiplexSBM are implemented using the *spaceNet* and *sbm* packages available on CRAN. DeepLPM and O2MAC are conducted using their respective Python code, which is available on GitHub: `https://github.com/ldggggg/deepLPM`, `https://github.com/googlebaba/WWW2020-O2MAC/tree/master`.

References

Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C.E., Vogelstein, J.T., 2021. Inference for multiple heterogeneous networks with a common invariant subspace. Journal of Machine Learning Research 22, 1–49.

Barbillon, P., Donnet, S., Lazega, E., Bar-Hen, A., 2017. 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, 295–314.

Chu, X., Fan, X., Yao, D., Zhu, Z., Huang, J., Bi, J., 2019. Cross-network embedding for multi-network alignment, in: The world wide web conference, pp. 273–284.

Dai, B., Wang, Y., Aston, J., Hua, G., Wipf, D., 2017. Hidden talents of the variational autoencoder. arXiv preprint arXiv:1706.05148 .

Daudin, J.J., Picard, F., Robin, S., 2008. A mixture model for random graphs. Statistics and computing 18, 173–183.

D'Angelo, S., Alfò, M., Fop, M., 2023. Model-based clustering for multidimensional social networks. Journal of the Royal Statistical Society Series A: Statistics in Society 186, 481–507.

D'Angelo, S., Murphy, T.B., Alfò, M., 2019. Latent space modelling of multidimensional networks with application to the exchange of votes in eurovision song contest. The Annals of Applied Statistics 13, 900–930.

Fan, S., Wang, X., Shi, C., Lu, E., Lin, K., Wang, B., 2020. One2multi graph autoencoder for multi-view graph clustering, in: proceedings of the web conference 2020, pp. 3070–3076.

Gollini, I., Murphy, T.B., 2016. Joint modeling of multiple network views. Journal of Computational and Graphical Statistics 25, 246–265.

Hamilton, W.L., Ying, R., Leskovec, J., 2017. Inductive representation learning on large graphs, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 1025–1035.

Handcock, M.S., Raftery, A.E., Tantrum, J.M., 2007. Model-based clustering for social networks. Journal of the Royal Statistical Society: Series A (Statistics in Society) 170, 301–354.

Hoff, P.D., Raftery, A.E., Handcock, M.S., 2002. Latent space approaches to social network analysis. Journal of the american Statistical association 97, 1090–1098.

Hubert, L., Arabie, P., 1985. Comparing partitions. Journal of classification 2, 193–218.

Ishiguro, K., Iwata, T., Ueda, N., Tenenbaum, J., 2010. Dynamic infinite relational model for time-varying relational data analysis. Advances in Neural Information Processing Systems 23.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M., 2016. Improved variational inference with inverse autoregressive flow. Advances in neural information processing systems 29, 4743–4751.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes. International Conference on Learning Representations (ICLR) .

Kipf, T.N., Welling, M., 2016a. Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations (ICLR-17).

Kipf, T.N., Welling, M., 2016b. Variational graph auto-encoders, in: NeurIPS Workshop on Bayesian Deep Learning (NeurIPS-16 BDL).

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A., 2014. Multilayer networks. Journal of complex networks 2, 203–271.

Lee, C., Wilkinson, D.J., 2019. A review of stochastic block models and extensions for graph clustering. Applied Network Science 4, 1–50.

Liang, D., Corneli, M., Bouveyron, C., Latouche, P., 2024. Clustering by deep latent position model with graph convolutional network. Advances in Data Analysis and Classification , 1–34.

Liu, W., Chen, P.Y., Yeung, S., Suzumura, T., Chen, L., 2017. Principled multilayer network embedding, in: 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE. pp. 134–141.

Ma, Y., Wang, S., Aggarwal, C.C., Yin, D., Tang, J., 2019. Multi-dimensional graph convolutional networks, in: Proceedings of the 2019 siam international conference on data mining, SIAM. pp. 657–665.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-sne. Journal of machine learning research 9.

MacDonald, P.W., Levina, E., Zhu, J., 2022. Latent space models for multiplex networks with shared structure. Biometrika 109, 683–706.

Nakis, N., Çelikkanat, A., Lehmann, S., Mørup, M., 2023. A hierarchical block distance model for ultra low-dimensional graph representations. IEEE Transactions on Knowledge and Data Engineering .

Nowicki, K., Snijders, T.A.B., 2001. Estimation and prediction for stochastic blockstructures. Journal of the American statistical association 96, 1077–1087.

Park, C., Kim, D., Han, J., Yu, H., 2020. Unsupervised attributed multiplex network embedding, in: Proceedings of the AAAI conference on artificial intelligence, pp. 5371–5378.

Paul, S., Chen, Y., 2020. A random effects stochastic block model for joint community detection in multiple networks with applications to neuroimaging. The Annals of Applied Statistics 14, 993–1029.

Salter-Townshend, M., McCormick, T.H., 2017. Latent space models for multiview network data. The annals of applied statistics 11, 1217.

Sosa, J., Betancourt, B., 2022. A latent space model for multilayer network data. Computational Statistics & Data Analysis 169, 107432.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2018. Graph attention networks. International Conference on Learning Representations (ICLR-18) .

Velickovic, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D., 2019. Deep graph infomax. ICLR (Poster) 2, 4.

Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S., 2019. Heterogeneous graph attention network, in: The world wide web conference, pp. 2022–2032.

Wang, Y.J., Wong, G.Y., 1987. Stochastic blockmodels for directed graphs. Journal of the American Statistical Association 82, 8–19.

Xiao, W., Houye, J., Chuan, S., Bai, W., Peng, C., P., Y., Yanfang, Y., 2019. Heterogeneous graph attention network. WWW .

Zhang, D., Yin, J., Zhu, X., Zhang, C., 2018a. Network representation learning: A survey. IEEE transactions on Big Data 6, 3–28.

Zhang, H., Qiu, L., Yi, L., Song, Y., 2018b. Scalable multiplex network embedding., in: IJCAI, pp. 3082–3088.

Zhang, P., Zhang, Y., Wang, J., Yin, B., 2023. Mvma-gcn: Multi-view multi-layer attention graph convolutional networks. Engineering Applications of Artificial Intelligence 126, 106717.