2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD) | 979-8-3503-3168-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSCWD57460.2023.10152677

Self-supervised community detection in multiplex networks with graph convolutional autoencoder

Xingyu Liu[1], Junwei Cheng[1,2], Hao Cheng[1], Chaobo He[1,2,*], Qimai Chen[1], and Quanlong Guan[3]

[1]School of Computer Science, South China Normal University, Guangzhou, 510631, China
[2]Pazhou Lab, Guangzhou, 510335, China
[3]College of Information Science and Technology, Jinan University, GuangZhou, 510631, China
Email:[jung, liuxingyu, haocheng]@m.scnu.edu.cn,
hechaobo@foxmail.com, chenqimai@scnu.edu.cn, gql@jnu.edu.cn

***Abstract*—Community detection in multiplex networks has received considerable attention in recent years. However, existing methods that combine graph embedding and downstream tasks still face two challenges. The first is how to fully explore the correlation among the layers in the multiplex networks, and the second is how to make the learned node representation more applicable to the community detection tasks. Aiming at these challenges, we propose a novel self-supervised multiplex community detection model called MGCAE which is based on graph neural networks. To solve the first challenge, we compute the mutual information maximization loss in the self-supervision module. The mutual information includes global representation and common representation of nodes in different layers, and node representation in each layer. For the second challenge, we combine a Bernoulli-Poisson loss and a modularity maximization loss to jointly optimize the reconstruction of the original adjacency matrix, which is in line with the rigorous theory of modularity. We treat graph convolutional autoencoder (GCAE) as the backbone framework and train it by using the unified loss mentioned above. In addition, the model obtains the community detection results in an end-to-end manner, which makes the model independent of downstream tasks and more stable. Experiments on real-world attributed multiplex network datasets demonstrate the effectiveness of our model.**

***keywords*—Community detection; Self-supervised learning; Multiplex networks; Graph convolutional autoencoder.**

I. Introduction

Multiplex networks [1] are widely used to represent the multiple relationships among objects, such as social networks [2] and transportation networks [3], where each layer represents a kind of relationship and has the same nodes and different edges. For example, when the academic network is modeled with a multiplex network, one layer represents the co-authorship, while another is the citation relationship layer. In addition, each author's corresponding research interests and work units can be regarded as nodes' attributes. Such a network is called an attributed multiplex network, which is a more complementary and comprehensive approach for modeling multiple relationships in the real world.

* Corresponding author.

Community detection in multiplex networks focuses on dividing the highly similar and tightly connected nodes into several disjoint communities, which have a wide range of applications, such as social platform networks [4] and protein interaction networks [5]. Most of the existing methods combine graph embedding and downstream tasks, and the representative methods that are based on graph neural networks (GNNs) [6] show superior performance. As a deep nonlinear framework, GNN is a seamless and automatical approach to integrate network topology and node information and also performs well in supervised and semi-supervised learning on graphs. Graph convolutional autoencoder (GCAE) [7] is a variant of GNN, which can learn the accurate node representation.

However, we still face two challenges. The first is how to exploit the interaction of node representation in different layers. A variety of methods separately extract the node embedding from each layer and then aggregate them to apply for the downstream tasks, which ignores the mutual interaction of common representation of nodes in different layers. The second is how to make the learned node representation better used for community detection since node embedding aims to reconstruct the original network rather than detect the community. Therefore, it is urgent to design an integration framework to make node embedding better serve the community detection tasks.

Aiming at these challenges, we propose a graph neural network-based self-supervised method for multiplex community detection. Our major contributions can be summarized as follow:

- We propose a novel self-supervised multiplex community detection method called MGCAE which is based on graph neural networks. We use motifs to enhance the original network structure, which makes it easier for the model to learn the node features of the input network. MGCAE directly obtains self-supervised multiplex community detection results in an end-to-end manner.
- To fully explore the correlation among the layers in the multiplex network, three forms of node representation are

used to compute the mutual information maximization loss in the self-supervision module. To make the learned node embedding more suitable for community detection tasks, we creatively introduce a joint optimization framework based on Bernoulli-Poisson loss and modularity maximization loss to reconstruct the original adjacency matrix. Finally, we train the GCAE by using the unified loss.

- The extensive experiments on real-world attributed multiplex networks show that our method outperforms state-of-the-art community detection methods.

The rest of this paper is organized as follows. In Section 2 we briefly review the related works. The details on the proposed method MGCAE and the experiment results are given in Section 3 and Section 4, respectively. We finally conclude this paper in Section 5.

II. Related Works

In this section, we briefly review the most relevant works, including self-supervised graph learning and community detection in multiplex networks.

A. Self-supervised graph learning

The lack of labels in real datasets is not conducive to supervised and semi-supervised learning, which rely on manual labels. Instead, self-supervised graph learning extracts the features of nodes without external supervision signal, whose scope of application is more extensive. According to [8], generation-based and contrast-based methods are the most important approaches in the self-supervised graph learning field.

The origin of generation-based methods can be traced back to the autoencoder, which uses the information of nodes and the topology of the network as input. Moreover, the graph convolutional network (GCN) [9] is exploited as the encoder to map the input data to the low-dimensional space to learn node representation, and a decoder is designed to reconstruct the original adjacency matrix. GAE [10] is the most typical representative of generation-based methods, which has a series of derivative works. For example, ARVGA [11] uses GANs [12] to regularize GAE.

Contrast-based methods are based on the idea of mutual information maximization, which is dedicated to strengthening the interaction between similar nodes. The representative contrast-based methods are DeepWalk [13] and node2vec [14], which extract local node features by using the random walk method. Besides, SDNE [15] and LINE [16] learn node embedding through the first-order and second-order proximity between nodes.

B. Community detection in multiplex networks

According to [17], community detection in multiplex networks is a hot research topic recently, and there are several representative methods. Firstly, there are classical methods based on modularity maximization, such as $\mathrm{GN}-Q_M$ [18] and $Louvain-Q_M$ [19]. Secondly, Alimadadi et al. [20] proposed MNLPA that extracts local features in multilayer networks. Moreover, Nie. et al. [21] proposed the constrained Laplacian Rank (CLR), which helps to construct the similarity matrix. Then SwMC and PwMC [22] further introduced hyperparameters according to CLR to learn the weights of each layer of the networks. Finally, graph representation learning methods, such as MNE [23] and PMNE [24], aim to learn informative representations, which are applied to downstream tasks to detect communities. However, these graph representation learning methods are non-goal-oriented such that the captured node features are not necessarily suitable for community detection tasks. In addition, the dependence on downstream tasks makes the performance of the model unstable.

III. Methodology

In this section, we first introduce the overview and definitions of MGCAE and then describe the details of our model.

A. Overview and definitions

Fig. 1 shows the overall framework of our model. MGCAE comprises two components: GCAE and self-supervision. GCAE is composed of a graph encoder and a graph decoder. A two-layer GCN is used as the encoder to learn the node representation. And Bernoulli-Poisson loss and modularity maximization loss are used as reconstruction loss in the graph decoder module. In the self-supervision module, we compute the mutual information maximization loss with three forms of node representation. Finally, we train the GCAE by using the unified loss mentioned above and obtain the community detection results in an end-to-end manner.

Given an undirected attributed network $\mathbf{G} = (\mathbf{A}, \mathbf{X})$, $\mathbf{A} = [a_{uv}]^{N\times N}$ is the adjacency matrix and $\mathbf{X} = [x_{uv}]^{N\times\beta}$ is the attribute matrix, N and β denote the number of nodes and the number of node's attributes, respectively; $a_{uv} = 1$ if node u and node v have linkage, otherwise $a_{uv} = 0$. $x_{uv} = 1$ if node u has the attribute v, otherwise $x_{uv} = 0$. Consisted of $R \geq 1$ layers of attributed networks, an attributed multiplex network is represented by $\mathbf{G}^M = \{\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_R\}$, where $\mathbf{G}_i = (\mathbf{A}_i, \mathbf{X})$ denotes the i-th layer of the network. Note that the nodes and nodes' attributes are the same in different layers, while the adjacency matrices are different.

B. Attributed multiplex network data preprocessing

There are high-order linkages among nodes with several common neighbors, and these linkages and nodes constitute motifs. According to the observations of the proposed paper [25], nodes in the same motifs are more likely to exist in the same community. Therefore, we introduce motifs to enhance the topology of the network, which is a further summary of the network structure and is beneficial for the community detection tasks.

For the i-th layer of the network, the neighbors of node u and node v are represented by $\mathcal{N}_i(u)$ and $\mathcal{N}_i(v)$, respectively. If the number of elements in the set $\mathcal{N}_i(u) \cap \mathcal{N}_i(v)$ is greater

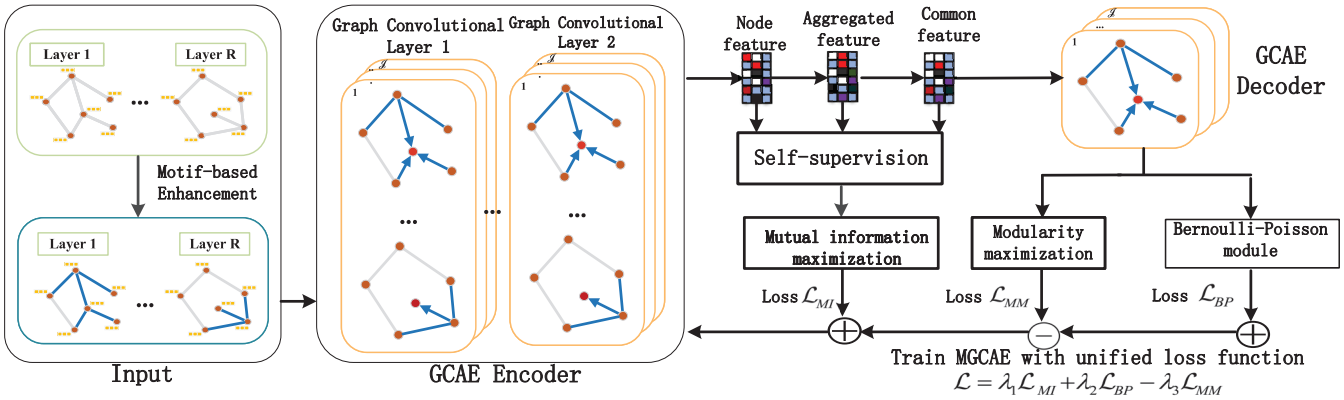


Fig. 1. Overview of MGCAE structure.

than the threshold $\mathcal{E}$, we construct the motif-based hypergraph $\mathcal{G}_i = [\mathcal{G}_{uv}]^{N\times N}$ for i-th layer, that is:

$$\mathcal{G}_i(uv) = \begin{cases} 1, \mathcal{N}_i(u) \cap \mathcal{N}_i(v) > \varepsilon \\ 0, \mathcal{N}_i(u) \cap \mathcal{N}_i(v) \leq \varepsilon \end{cases} \tag{1}$$

We obtain the top k maximal connected subgraphs $\mathcal{G}_i^1, \mathcal{G}_i^2, \ldots, \mathcal{G}_i^k$ of $\mathcal{G}_i$ and use them to enhance the $\mathbf{A}_i$, that is:

$$\mathbf{A}_i \leftarrow \mathbf{A}_i \cup \mathbf{A}_{\mathcal{G}_i^1} \cup \mathbf{A}_{\mathcal{G}_i^2} \cup \mathbf{A}_{\mathcal{G}_i^3} \cup \ldots \cup \mathbf{A}_{\mathcal{G}_i^k} \tag{2}$$

where $\mathbf{A}_{\mathcal{G}_i^k}$ is the adjacency matrix of $\mathcal{G}_i^k$.

C. The Model

GCAE Encoder: A two-layer GCN is used as the encoder to learn the node representation. We define the number of detected communities as θ, and $\mathbf{H}_i = [h_{uv}]^{N\times\theta}$ denotes the node embedding matrix of the i-th layer of the network. The adjacency matrix $\mathbf{A}_i$ and attribute matrix $\mathbf{X}$ are used as the input of the GCN to obtain the $\mathbf{H}_i$, that is:

$$\mathbf{H}_i = \sigma\left(\widehat{\mathbf{A}}_i \phi\left(\widehat{\mathbf{A}}_i \mathbf{X}\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right) \tag{3}$$

where $\sigma(\cdot)$ and $\phi(\cdot)$ denote the Tanh activation function and the ReLU activation function, respectively; $\widetilde{\mathbf{A}}_i = \mathbf{A}_i + \mathbf{I}_N$, $\widetilde{\mathbf{D}}_i$ is the degree matrix of $\widetilde{\mathbf{A}}_i$ and $\widehat{\mathbf{A}}_i = \widetilde{\mathbf{D}}_i^{-\frac{1}{2}} \widetilde{\mathbf{A}}_i \widetilde{\mathbf{D}}_i^{-\frac{1}{2}}$; $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are the trainable parameters of the first and second layers of GCN, respectively.

Naturally, the adjacency matrix $\mathbf{A}_i$ and the randomly shuffled attribute matrix $\widetilde{\mathbf{X}}$ are used as the input of the GCN to obtain the negative node embedding matrix $\mathbf{M}_i = [m_{uv}]^{N\times\theta}$ of the i-th layer of the network:

$$\mathbf{M}_i = \sigma\left(\widehat{\mathbf{A}}_i \phi\left(\widehat{\mathbf{A}}_i \widetilde{\mathbf{X}}\mathbf{W}^{(0)}\right)\mathbf{W}^{(1)}\right) \tag{4}$$

And then we use average pooling of $\mathbf{H}_i$ to obtain global representation of nodes $\mathbf{S}_i = [s_u]^{1\times\theta}$ at the i-th layer of the network:

$$\mathbf{S}_i = \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}_n \tag{5}$$

where $\mathbf{h}_n$ is the n-th row of $\mathbf{H}_i$.

Similarly, we obtain the node common representation $\mathbf{Z} = [z_{uv}]^{N\times\theta}$ through average pooling of $\mathbf{H}_i$ in another aspect:

$$\mathbf{Z} = \frac{1}{R}\sum_{i=1}^{R} \mathbf{H}_i \tag{6}$$

In the same way, we obtain the negative common representation of nodes $\mathbf{P} = [p_{uv}]^{N\times\theta}$ through average pooling of $\mathbf{M}_i$:

$$\mathbf{P} = \frac{1}{R}\sum_{i=1}^{R} \mathbf{M}_i \tag{7}$$

Finally, we use average pooling of $\mathbf{Z}$ to obtain the common global representation of nodes $\mathbf{E} = [e_u]^{1\times\theta}$:

$$\mathbf{E} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{z}_n \tag{8}$$

where $\mathbf{z}_n$ is the n-th row of $\mathbf{Z}$.

GCAE Decoder: To make the learned node embedding more suitable for community detection tasks, we creatively introduce a joint optimization framework based on Bernoulli-Poisson loss and modularity maximization loss to reconstruct the original adjacency matrix. The Bernoulli-Poisson loss is defined as $\mathcal{L}_{BP}$, and we obtian $\mathcal{L}_{BP}$ through the following formula:

$$\begin{aligned} \mathcal{L}_{BP} = \sum_{i=1}^{R} \Bigg(& - \sum_{\mathbf{A}_i(uv)=1} \log\left(1 - \exp\left(-\mathbf{H}_i(u)\mathbf{H}_i(v)^T\right)\right) \\ & + \sum_{\mathbf{A}_i(uv)=0} \mathbf{H}_i(u)\mathbf{H}_i(v)^T \Bigg) \\ & - \Bigg(\sum_{\mathbf{A}'(uv)=1} \log\left(1 - \exp\left(-\mathbf{Z}(u)\mathbf{Z}(v)^T\right)\right) \\ & + \sum_{\mathbf{A}'(uv)=0} \mathbf{Z}(u)\mathbf{Z}(v)^T \Bigg) \end{aligned} \tag{9}$$

where $\mathbf{H}_i(v)^T$ and $\mathbf{Z}(v)^T$ represent the transpose of $\mathbf{H}_i(v)$ and $\mathbf{Z}(v)$, respectively; $\mathbf{A}' = \sum_{i=1}^{R} \mathbf{A}_i$.

To compute the modularity maximization loss $\mathcal{L}_{MM}$, we need to construct the modularity matrix first. $\mathbf{B}_i$ and $\mathbf{B}'$ are respectively the modularity matrices based on $\mathbf{A}_i$ and $\mathbf{A}'$, which can be obtained through the following formulas:

$$\mathbf{B}_i = \mathbf{A}_i - \frac{\mathbf{D}_i \mathbf{D}_i^T}{\sum\sum \mathbf{D}_i} \tag{10}$$

$$\mathbf{B}' = \mathbf{A}' - \frac{\mathbf{D}' \left(\mathbf{D}'\right)^T}{\sum\sum \mathbf{D}'} \tag{11}$$

where $\mathbf{D}_i$ is the degree matrix of $\mathbf{A}_i$, and $\mathbf{D}' = \sum_{i=1}^{R} \mathbf{D}_i$.

Then we can obtain $\mathcal{L}_{MM}$ through the following formula:

$$\mathcal{L}_{MM} = \sum_{i=1}^{R} \mathrm{Tr}\left(\mathbf{H}_i^T \mathbf{B}_i \mathbf{H}_i\right) + \mathrm{Tr}\left(\mathbf{Z}_i^T \mathbf{B}' \mathbf{Z}\right) \tag{12}$$

Self-supervision: Generation-based methods and contrast-based methods are the most important approaches in the self-supervised graph learning field. We obtain contrastive generator matrix $\boldsymbol{\mathcal{A}}_i$ of i-th layer of the network through the bi-linear interpolation:

$$\boldsymbol{\mathcal{A}}_i = STACK\left(\rho\left(\mathbf{H}_i \eta_i \mathbf{S}_i + b_i\right), \rho\left(\mathbf{M}_i \omega_i \mathbf{S}_i + y_i\right)\right) \tag{13}$$

where η_i, b_i, ω_i and y_i are the trainable parameters of i-th layer of the network; $\rho(\cdot)$ denotes the sigmoid activation function, $STACK(\cdot)$ is the concatenation function.

In the same way, we obtain contrastive generator matrix $\boldsymbol{\mathcal{A}}^*$ of the whole network through the bi-linear interpolation:

$$\boldsymbol{\mathcal{A}}^* = STACK\left(\rho\left(\mathbf{Z}\mu\mathbf{E} + \gamma\right), \rho\left(\mathbf{P}\delta\mathbf{E} + \tau\right)\right) \tag{14}$$

where μ, γ, δ and τ are the trainable parameters of the whole network.

Finally, the mutual information maximization loss $\mathcal{L}_{MI}$ is measured via the following binary cross entropy function:

$$\begin{aligned} \mathcal{L}_{MI} = -\frac{1}{n^2} \sum_{uv} \big(&\boldsymbol{\mathcal{A}}_{uv} \log \boldsymbol{\mathcal{A}}^*_{uv} \\ &+ \left(1 - \boldsymbol{\mathcal{A}}_{uv}\right) \log\left(1 - \boldsymbol{\mathcal{A}}^*_{uv}\right)\big) \end{aligned} \tag{15}$$

where $\boldsymbol{\mathcal{A}} = \frac{1}{R} \sum_{i=1}^{R} \boldsymbol{\mathcal{A}}_i$.

The unified loss function: In summary, we jointly optimize the GCAE with the unified losses, including the mutual information maximization loss, the modularity maximization loss, and the Bernoulli-Poisson loss. The unified loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{MI} + \lambda_1 \mathcal{L}_{BP} - \lambda_2 \mathcal{L}_{\mathrm{MM}} \tag{16}$$

where λ_1 and λ_2 are the hyperparameters, and their corresponding setup can be seen in Section 4.

Then, the self-attention mechanism is applied to automatically assign the weights of the node representation matrix of each layer which is learned through the above process and the final node representation matrix $\boldsymbol{\mathcal{F}} = [f_{uv}]^{N \times \theta}$ is defined as:

$$\boldsymbol{\mathcal{F}} = \alpha_1 \mathbf{H}_1 + \alpha_2 \mathbf{H}_2 + \ldots + \alpha_R \mathbf{H}_R \tag{17}$$

where $a_i \in [a_1, \ldots, a_R]$ represents the weights of the node representation matrix of i-th layer of network which trained by the self-attention mechanism.

Finally, our model obtains the results of community detection in an end-to-end manner, and the community index of node i can be described as follows:

$$\boldsymbol{\ell_i} = \underset{j=1\ldots\theta}{argmax} \boldsymbol{\mathcal{F}}_{ij} \tag{18}$$

IV. Experiments

To validate the effectiveness of the proposed model, we conduct extensive experiments on several real-world attributed multiplex networks, and we introduce experimental setups and evaluation in this section.

A. Experimental setups

1) Datasets: The proposed MGCAE along with the compared methods are tested on the three real-world datasets, i.e., ACM[1], DBLP[2], and IMDB[3]. The details of these datasets are shown in Table I.

2) Baselines:

- GAE-avg [26]: GAE-avg is the application of GAE in multiplex networks.
- MNE [23]: a scalable multiplex network embedding model that represents information from multi-type relations into a unified embedding space.
- PMNE(r) and PMNE(c) [24]: PMNE(r) and PMNE(c) are the "results aggregation" method and "layer co-analysis" method proposed by PMNE, respectively.
- PwMC and SwMC [22]: PwMC is a parameter-weighted multiview clustering method and SwMC is a self-weighted multiview clustering method.

3) Parameter settings and evaluation metrics: Since our model is an autoencoder-related model, it is iteratively trained 100 times on each of the three datasets to prevent overfitting. Moreover, the Adam algorithm is used to optimize the training results, and the learning rate is set to 0.01. To obtain a balance among loss terms in the joint optimization based on (16), we tune λ_1 and λ_2 in the range of $\{0.01, 0.1, 1, 10\}$ and $\left\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\right\}$, respectively. For the baseline methods, we follow the setting in the corresponding papers. Moreover, three representative methods are used to evaluate the results of community detection, which are accuracy (ACC), normalized mutual information (NMI), and average rand index (ARI), and for all methods, the higher scores represent the better performance of community detection. Since all the methods rely on initialization, we repeat them 10 times with random initialization and report the average results.

[1] http://dl.acm.org
[2] https://dblp.uni-trier.de/
[3] https://www.imdb.com/

TABLE I
THE STATISTICS OF THE ATTRIBUTED MULTIPLEX NETWORKS

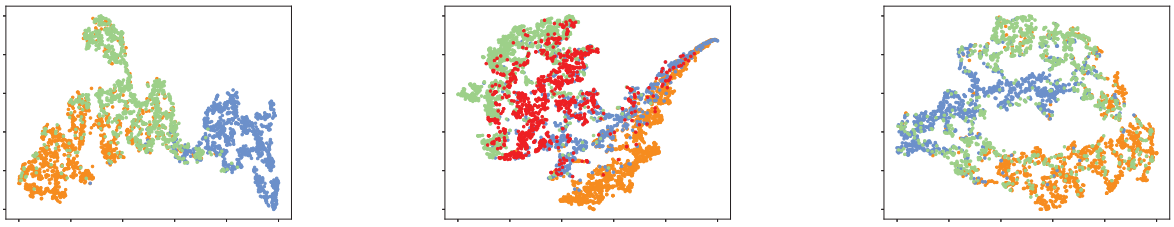
Network	#Node	#Feature	#Edges in each layer	#Community
ACM	3025	1830	co-paper (29,281) co-subject (2,210,761)	3
DBLP	4057	334	co-author (11,113) co-conference (5,000,495) co-term (6,776,335)	4
IMDB	4780	1232	co-actor (98,010) co-director (21,018)	3

TABLE II
COMMUNITY DETECTION RESULTS ON THREE DATASETS. THE BEST RESULTS OF ALL METHODS ARE INDICATED IN BOLD. THE '-' REPRESENTS THE METHOD RUN OUT-OF-MEMORY ON THIS DATASET.

Dataset	ACM			DBLP			IMDB		
Method	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
GAE-avg	0.6810	0.4775	0.4456	0.5563	0.3102	0.2431	0.4453	0.0401	0.0488
MNE	0.6337	0.3562	0.3358	-	-	-	0.3876	0.0011	0.0012
PMNE(r)	0.6455	0.4156	0.3257	0.3846	0.0732	0.0925	0.4534	0.0014	0.0107
PMNE(c)	0.6798	0.4885	0.4420	-	-	-	**0.4859**	0.0315	0.0298
PwMC	0.4335	0.0326	0.0423	0.3251	0.0254	0.0179	0.2357	0.0019	0.0071
SwMC	0.3733	0.0885	0.0203	**0.6556**	0.3634	0.3764	0.3025	0.0078	0.0011
MGCAE	**0.7201**	**0.6094**	**0.4993**	0.3932	**0.4550**	**0.3812**	0.3933	**0.0954**	**0.0671**

TABLE III
ABLATION STUDY ON THE COMMUNITY DETECTION TASKS. MI, BP, AND MM DENOTE THE MUTUAL MAXIMIZATION ERROR, BERNOULLI-POISSON ERROR, AND MODULARITY MAXIMIZATION ERROR. THE BEST RESULTS ARE INDICATED IN BOLD.

Dataset	ACM			DBLP			IMDB		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
MI	0.6061	0.4497	0.3962	0.2955	0.3331	0.2374	0.3856	0.0101	0.0138
MI+BP	0.6188	0.4807	0.4134	0.3740	0.3860	0.3352	0.3990	0.0551	0.0474
MI+BP+MM	**0.7201**	**0.6094**	**0.4993**	**0.3932**	**0.4550**	**0.3812**	**0.3933**	**0.0954**	**0.0671**



(a) ACM (b) DBLP (c) IMDB

Fig. 2. Visualization of the node embedding learned by MGCAE on the ACM, DBLP, IMDB datasets. Different colors represent different communities.

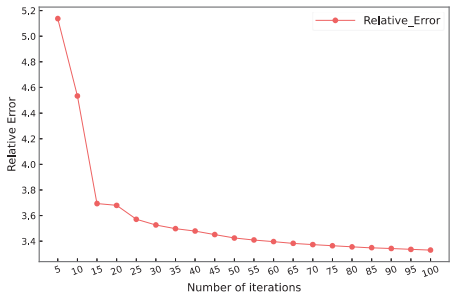


Fig. 3. Convergence analysis of MGCAE on the ACM dataset.

B. Evaluation

1) Performance of community detection: Table II shows the results of all methods tested on the above three datasets, and our method mostly outperforms baselines. After analysis, we can know that our method averagely improves by 8.93% of NMI and 2.56% of ARI compared with the other method with the best results, respectively. Therefore, the experimental data validate the performance of the model guided by maximizing mutual information. It also demonstrates the feasibility of unifying Bernoulli Poisson loss and modularity loss to calculate reconstruction loss. Specifically, our method outperforms PwMC and SwMC in most cases. The reason is that they are both linear models, while we leverage the deep nonlinear feature learning framework of GCN. Secondly, our method is superior to GAE-avg in most cases. It is because GAE-avg uses a simple average aggregation method to fuse the node representation of each layer, which ignores the relationship among layers. Instead, our method exploits complementary and comprehensive mutual information among the different

layers of the multiplex network. This is also the reason why PMNE(r) performs worse than our model and PMNE(c) performs similarly to our model. Nodes in the IMDB network rarely have high confidence, which leads to the failure of mining the associated information among nodes. Therefore, all methods perform worst on the IMDB dataset. Finally, Fig. 3 illustrates the convergence analysis experimental results on the ACM dataset, which show that the performance of our model tends to converge as the number of iterative training increases.

2) Ablation Study: We compare the performance of the mutual maximization error (MI), modularity maximization error (MM), and Bernoulli-Poisson error (BP) on attributed multiplex networks on the datasets, and present experimental results in Table III. Firstly, introducing BP on top of MI can improve the model's performance on unsupervised tasks, which can be observed by comparing MI with MI+BP. Moreover, the joint error (MI+BP+MM) can further improve the discriminative ability of node embedding for community detection tasks.

3) Visualization of node embedding: The visualization of the node embedding learned by MGCAE on the ACM, DBLP, and IMDB multiplex networks is presented in Fig. 2. Different colors represent different communities. It is obvious that the edge of each community is clear, which proves that MGCAE can divide the nodes into communities well.

V. Conclusion

We propose a novel framework namely MGCAE for self-supervised community detection in attributed multiplex networks which is based on graph neural networks. Firstly, we compute the mutual information maximization loss in the self-supervision module and then combine modularity maximization loss and Bernoulli Poisson loss to optimize the reconstruction of the original network. We treat graph convolutional autoencoder as the backbone framework and train it by using the unified loss mentioned above. In addition, the model obtains the community detection results in an end-to-end manner. We evaluate the proposed MGCAE on three real-world datasets and the results demonstrate the effectiveness of our model.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62077045, in part by the Humanity and Social Science Youth Foundation of Ministry of Education of China under Grant 19YJCZH049, and in part by the Natural Science Foundation of Guangdong Province of China under Grant 2019A1515011292.

References

[1] F. Battiston, V. Nicosia, and V. Latora, "The new challenges of multiplex networks: Measures and models," *The European Physical Journal Special Topics*, vol. 226, no. 3, pp. 401–416, 2017.

[2] M. Wang, C. Wang, J. X. Yu, and J. Zhang, "Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 998–1009, 2015.

[3] C. Von Ferber, T. Holovatch, Y. Holovatch, and V. Palchykov, "Public transport networks: empirical analysis and modeling," *The European Physical Journal B*, vol. 68, no. 2, pp. 261–275, 2009.

[4] L. Tang and H. Liu, "Community detection and mining in social media," *Synthesis lectures on data mining and knowledge discovery*, vol. 2, no. 1, pp. 1–137, 2010.

[5] H. Mahmoud, F. Masulli, S. Rovetta, and G. Russo, "Community detection in protein-protein interaction networks using spectral and graph approaches," in *International meeting on computational intelligence methods for bioinformatics and biostatistics*. Springer, 2013, pp. 62–75.

[6] J. Bruna and X. Li, "Community detection with graph neural networks," *stat*, vol. 1050, p. 27, 2017.

[7] C. He, Y. Zheng, J. Cheng, Y. Tang, G. Chen, and H. Liu, "Semi-supervised overlapping community detection in attributed graph with graph convolutional autoencoder," *Information Sciences*, vol. 608, pp. 1464–1479, 2022.

[8] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and P. Yu, "Graph self-supervised learning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[10] ——, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[11] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," *arXiv preprint arXiv:1802.04407*, 2018.

[12] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[13] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[15] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[16] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[17] X. Huang, D. Chen, T. Ren, and D. Wang, "A survey of community detection methods in multilayer networks," *Data Mining and Knowledge Discovery*, vol. 35, no. 1, pp. 1–45, 2021.

[18] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[20] F. Alimadadi, E. Khadangi, and A. Bagheri, "Community detection in facebook activity networks and presenting a new multilayer label propagation algorithm for community detection," *International Journal of Modern Physics B*, vol. 33, no. 10, p. 1950089, 2019.

[21] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[22] F. Nie, J. Li, X. Li *et al.*, "Self-weighted multiview clustering with multiple graphs." in *IJCAI*, 2017, pp. 2564–2570.

[23] H. Zhang, L. Qiu, L. Yi, and Y. Song, "Scalable multiplex network embedding." in *IJCAI*, vol. 18, 2018, pp. 3082–3088.

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

[25] P.-Z. Li, L. Huang, C.-D. Wang, and J.-H. Lai, "Edmot: An edge enhancement approach for motif-aware community detection," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 479–487.

[26] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang, "One2multi graph autoencoder for multi-view graph clustering," in *Proceedings of The Web Conference 2020*, 2020, pp. 3070–3076.