Self-Supervised Contrastive Graph Clustering Network via Structural Information Fusion

Xiaoyang Ji, Yuchen Zhou, Haofu Yang, Shiyue Xu, Jiahao Li
Nankai University
Tianjin, China
2110611@mail.nankai.edu.cn

***Abstract*—Graph clustering, a classical task in graph learning, involves partitioning the nodes of a graph into distinct clusters. This task has applications in various real-world scenarios, such as anomaly detection, social network analysis, and community discovery. Current graph clustering methods commonly rely on module pre-training to obtain a reliable prior distribution for the model, which is then used as the optimization objective. However, these methods often overlook deeper supervised signals, leading to sub-optimal reliability of the prior distribution. To address this issue, we propose a novel deep graph clustering method called CGCN. Our approach introduces contrastive signals and deep structural information into the pre-training process. Specifically, CGCN utilizes a contrastive learning mechanism to foster information interoperability among multiple modules and allows the model to adaptively adjust the degree of information aggregation for different order structures. Our CGCN method has been experimentally validated on multiple real-world graph datasets, showcasing its ability to boost the dependability of prior clustering distributions acquired through pre-training. As a result, we observed notable enhancements in the performance of the model.**

***Index Terms*—graph clustering, contrastive learning, information fusion**

I. Introduction

Deep learning methods, renowned for their remarkable representation learning capabilities, have yielded promising outcomes in the realm of deep graph clustering [1]–[3], particularly within various practical graph-based application domains like anomaly detection, social network analysis, and community discovery, etc [4]–[7]. As a classical unsupervised task, graph clustering focuses on how to better classify the individual nodes in a graph into their corresponding clusters as much as possible without supervised signals.

The effectiveness of deep graph clustering methods heavily relies on two critical factors: the optimization objective and the feature extraction technique [8], [9]. Particularly in unsupervised clustering scenarios, where the absence of label guidance poses a challenge, the design of a sophisticated objective function and an intricate feature extraction approach can significantly enhance clustering performance.

During the early stages, deep clustering methods primarily focused on leveraging attribute information within the original feature space, leading to commendable performance in numerous cases [10], [11]. Recent research endeavors have exhibited a strong inclination towards extracting geometric structure information and integrating it with attribute information to further enhance clustering accuracy [12], [13]. However, although existing methods have successfully incorporated both types of information and achieved performance gains, these methods tend to rely on simplistic fusion techniques, neglecting deeper levels of crucial data mining [14]. Consequently, the reliability of prior clustering distributions obtained through pre-training various models remains insufficient.

To tackle these challenges, we propose the CGCN method. The core concept behind this framework revolves around bolstering the reliability of the pre-training's a priori distribution. We introduce a contrastive learning module [15], [16] that facilitates cross-referencing and cross-checking among different pre-training modules, maximizing the retention of vital shared information. Furthermore, we delve into deeper graph structure information, enabling the model to dynamically and adaptively adjust the degree of aggregating different order structure information during training. The primary contributions of this paper can be summarized as follows:

(1) The introduction of a contrast learning mechanism fosters information interoperability among multiple modules during the pre-training process, thereby enhancing the reliability of the priori clustering distribution.

(2) The module design encourages the model to flexibly adjust the degree of information aggregation pertaining to various order structures, allowing for adaptive changes throughout the training process.

(3) We validate the effectiveness of our approach through comprehensive experiments conducted on real-world graph datasets.

II. Related Work

Graph neural networks (GNNs) have gained significant popularity in various graph scenarios [17]–[19], such as graph clustering, knowledge graph [20], link prediction [21], etc.

Among these, attribute graph clustering is a fundamental task that poses significant challenges. It entails the segregation of nodes in an attribute graph into distinct clusters devoid of any human annotation. In early methodologies, self-encoders were employed to acquire node embeddings, which were then subjected to K-means [22] clustering.

Building upon the success of GNNs, the MGAE [23] algorithm was introduced, which employed a graph self-encoder to encode nodes and then performed spectral clustering for node clustering. To develop a clustering-oriented approach,

the DAEGC [24] framework was proposed, incorporating an attention-based graph encoder and clustering alignment loss within deep clustering methods. SDCN [25] demonstrated the effectiveness of integrating structural and attribute information.

Contrastive learning has emerged as a research hot-spot in the field of deep graph clustering. AGE [26] addressed high-frequency noise in node attributes and trained the encoder using adaptive positive and negative sample comparisons. MVGRL [27] generated gradual structural views and compared node embeddings with graph embeddings to enhance learning. Despite the validation of the contrast learning paradigm, several technical issues remain unresolved. GDCL [28] aimed to correct sampling bias in deep graph clustering with contrast depth, while TGC [29] provided a generalized framework for deep node clustering in temporal graphs. To tackle the chanllege of graph scale, a scalable deep graph clustering method called S3GC [30] was proposed, leveraging comparative learning with GNNs.

Although effective, these methods usually ignore the deep structural information and the robust supervised signals form contrastive view. To address this limitation, our paper introduces a novel method called CGCN that utilizes the contrastive learning and higher-order structural information to enhance the prior clustering distribution from the pre-training process. Next, we give the details of the proposed method.

III. Method

A. Overall Framework

Our proposed method, CGCN, is an improvement of the DFCN method [31], a classical deep graph clustering method, by introducing AutoEncoder and Graph AutoEncoder to enable the computation of a priori clustering distributions during the pre-training process. We further introduce contrast learning and higher-order structural information in this session to enhance the reliability of the a priori clustering distribution. The structure of our model is shown in Fig. 1.

B. Autoencoder

We introduce the Autoencoder (AE) as one of the pre-training modules, which is an unsupervised learning model for data reduction and feature extraction. Its goal is to learn a compact representation that can reconstruct the input data while preserving the most important features.

An autoencoder consists of two main components: an Encoder and a Decoder. The Encoder converts the input data into a low-dimensional representation (Encoding), while the Decoder maps the low-dimensional representation back to the original data space (Decoding). In this way, the autoencoder can learn a valid feature representation by minimizing the reconstruction error between the input data and the reconstructed data. Its encoder and decoder are of the form:

$$H = f(X), \qquad \mathbf{Z}_{AE} = g(H) \tag{1}$$

Its loss function is a reconstruction of the sample features:

$$L_{AE} = \|\mathbf{Z}_{AE} - X\|_2^2 \tag{2}$$

where X is the initial node features, H is the output of the encoder. $\mathbf{Z}_{AE}$ is the output of the decoder, i.e. the node embeddings of AE. $f(\cdot)$ and $g(\cdot)$ are functions of the encoder and decoder, and L_{AE} is the loss function used to measure the reconstruction error.

C. Graph Autoencoder

Furthermore, we hereby introduce the module of graph autoencoder (GAE) as an additional pre-training component. The primary objective of the GAE is to simultaneously reconstruct the weighted attribute matrix and the adjacency matrix. The architectural configuration of both the encoder and decoder components is detailed below.

$$\mathbf{Z}^{(l)} = \sigma(\widetilde{\mathbf{A}}\mathbf{Z}^{(l-1)}\mathbf{W}^{(l)}) \tag{3}$$

$$\mathbf{Z}_{GAE} = \sigma(\widetilde{\mathbf{A}}\widehat{\mathbf{Z}}^{(h-1)}\widehat{\mathbf{W}}^{(h)}) \tag{4}$$

$\mathbf{W}^{(l)}$ and $\mathbf{W}^{(h)}$ denote the learnable parameter matrices for the encoder and decoder layers, respectively. The loss function for GAE is defined as follows:

$$L_{IGAE} = L_f + \gamma L_s \tag{5}$$

Such loss function can be divided into two parts, i.e., feature reconstruction L_f and struture reconstruction L_s.

$$L_f = \frac{1}{2N}\|\widetilde{\mathbf{A}}\mathbf{X} - \mathbf{Z}_{GAE}\|_F^2 \tag{6}$$

$$L_s = \frac{1}{2N}\|\widetilde{\mathbf{A}} - \widehat{\mathbf{A}}\|_F^2 \tag{7}$$

$\mathbf{Z}_{GAE} \in \mathbb{R}^{N\times d}$ denotes the node embeddings of GAE. Notably, Eq. (7) employs inner product operations to generate the reconstructed adjacency matrix, leveraging multi-layer representations of the network. By minimizing both Eq. (6) and Eq. (7), the GAE module aims to effectively reduce the reconstruction loss associated with the weighted attribute matrix and the adjacency matrix.

D. Structural and Attribute Information Fusion

After pre-training the AE and GAE to obtain their respective node embeddings, we will use these embeddings to generate the final node embeddings as well as the priori clustering distributions. In this subsection, we first present the embedding generation, i.e., structural and attribute information fusion. The initial fusion embeddings $\mathbf{Z}_I$ can be calculated as follows.

$$\mathbf{Z}_I = \delta\mathbf{Z}_{AE} + (1-\delta)\mathbf{Z}_{IGAE} \tag{8}$$

To process the combined information $\mathbf{Z}_I$, we employ an operation similar to graph convolution. This operation allows us to enhance the initial fused embeddings by incorporating the local structure present in the data. The combined information $\mathbf{Z}_I$ is represented as $\mathbf{Z}_I \in \mathbb{R}^{N\times d'}$, where d' represents the dimensionality of the enhanced embeddings.

$$\mathbf{Z}_L = \tilde{\mathbf{A}}\mathbf{Z}_I \tag{9}$$

Here, $\mathbf{Z}^1 \in \mathbb{R}^{N\times d'}$ denotes the locally enhanced representation from the first-order neighborhood. In this way, we

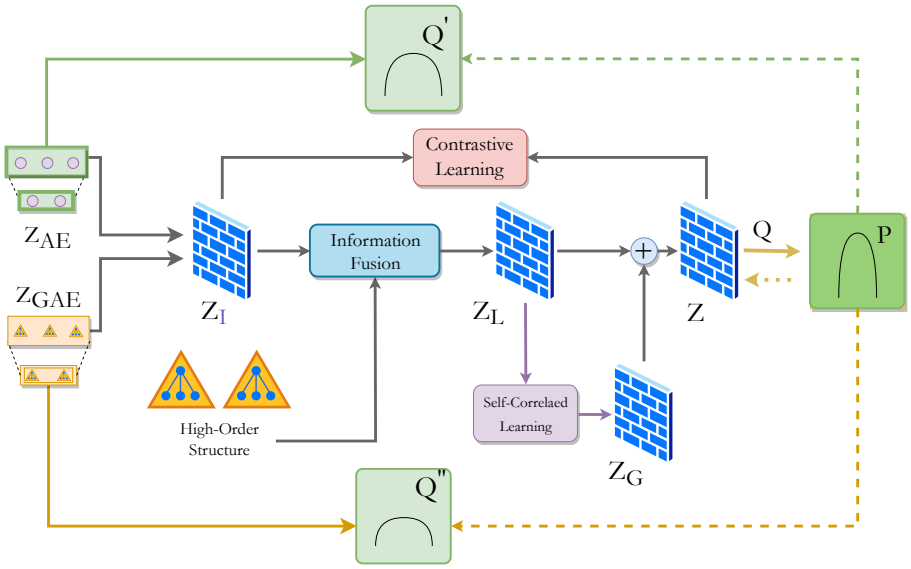


Fig. 1: Overall Framework

can combine different order neighborhood information by the adjacency matrix. Numerous experiments have proved that graph neural networks with more than three layers can bring about over-smoothing phenomenon, so we only select the first two layers of neighborhood information to be combined here.

$$\mathbf{Z}_L = \lambda_1 \mathbf{Z}^1 + \lambda_2 \mathbf{Z}^2 \tag{10}$$

λ_1 and λ_2 are learnable parameters that control the fusion degree of different order neighborhood information.

Consequently, we propose the incorporation of a self-correlation mechanism to leverage non-local relationships within the initial information fusion space among samples. To this end, we first calculate the normalized self-correlation matrix denoted as $\mathbf{S} \in \mathbb{R}^{N \times N}$, as described by Eq. (11).

$$\mathbf{S}_{ij} = \frac{e^{(\mathbf{Z}_L \mathbf{Z}_L^{\mathbf{T}})_{ij}}}{\sum_{k=1}^{N} e^{(\mathbf{Z}_L \mathbf{Z}_L^{\mathbf{T}})_{ik}}} \tag{11}$$

Following this, considering S as the coefficient, we take into account the global inter-sample correlations and perform a recombination.

$$\mathbf{Z}_G = \tilde{\mathbf{S}} \mathbf{Z}_L \tag{12}$$

Lastly, we employ skip connections to facilitate the smooth transmission of information within the fusion mechanism.

$$\mathbf{Z}_{final} = \lambda_b \mathbf{Z}_G + \mathbf{Z}_L \tag{13}$$

Here, the scaling parameter λ_b is introduced to adjust the contribution of the cross-modal dynamic fusion mechanism. This mechanism effectively considers both local and global levels of sample correlations. By intricately integrating and refining the information obtained from the Autoencoder (AE) and Graph Autoencoder (GAE), this method facilitates a more accurate learning of consensus latent representations.

E. Priori Clustering Distribution

To provide more reliable guidance during the training of the clustering network, the model initially employs a robust clustering embedding $\tilde{\mathbf{Z}} \in \mathbb{R}^{N \times d'}$. This embedding integrates information from both the Autoencoder (AE) and Improved Graph Autoencoder (IGAE) to facilitate target distribution generation. The generation process can be formulated as follows.

$$q_{ij} = \frac{(1 + \|\tilde{z}_i - u_j\|^2 / v)^{-\frac{v+1}{2}}}{\sum_{j'} (1 + \|\tilde{z}_i - u_{j'}\|^2 / v)^{-\frac{v+1}{2}}} \tag{14}$$

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'} (q_{ij'}^2 / \sum_i q_{ij'})} \tag{15}$$

In the initial stage, the similarity between the i-th sample ($\tilde{z}_i$) and the j-th pre-calculated clustering center (u_j) in the fused embedding space is computed using the Student's t-distribution as a kernel. This process is represented by Equation (14), where v denotes the degree of freedom for the Student's t-distribution and q_{ij} represents the probability of assigning the i-th node to the j-th center, indicating a soft assignment.

The soft assignment matrix $\mathbf{Q} \in \mathbb{R}^{N \times K}$ reflects the distribution of all samples. In the subsequent step, Equation (15) is introduced to encourage all samples to move closer to the cluster centers, thereby enhancing the confidence of cluster assignment. Specifically, $0 \leq p_{ij} \leq 1$ denotes an element of the generated target distribution $\mathbf{P} \in \mathbb{R}^{N \times K}$, indicating the probability of the i-th sample belonging to the j-th cluster center.

Using the iteratively generated target distribution, the soft assignment distribution of the AE and GAE modules is calculated separately by applying Equation (14) to the latent embeddings of each module. The soft assignment distributions of the AE and GAE are denoted as Q' and Q'', respectively.

TABLE I: Dataset Description.

Datasets	Nodes	Edges	Clusters	Feature
DBLP	4,058	3,528	4	334
CITE	3,327	4,732	6	3,703
ACM	3,025	13,128	3	1,870

To achieve integration of network training and enhanced representation capacity for each component, we introduce a triplet clustering loss that utilizes adapted KL-divergence. The proposed loss function can be defined as follows.

$$L_{KL} = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{(q_{ij} + q_{ij}^{'} + q_{ij}^{''})/3} \tag{16}$$

In this formulation, we aim to align the summation of the soft assignment distributions of the two modules, and the fused representations with the robust target distribution. As the target distribution is generated without human guidance, we refer to the loss function as the triplet clustering loss and the corresponding training mechanism as the triplet self-supervised strategy.

F. Loss Function

The overall learning objective consists of two main parts, i.e., the reconstruction loss of AE and IGAE, and the clustering loss which is correlated with the target distribution:

$$L = L_{AE} + L_{IGAE} + L_C + \lambda L_{KL} \tag{17}$$

λ is a pre-defined hyper-parameter which balances the importance of reconstruction and clustering.

L_C is a loss function that introduces the contrastive learning, which aims to enhance the constrative signals for the model training. In particular, L_C can be divided into two parts: pre-training embedding alignment and training embedding alignment.

$$L_C = \alpha L_{pre} + \beta L_{train} \tag{18}$$

$$L_{pre} = \|\mathbf{Z}_{GAE} - \mathbf{Z}_{AE}\|_2^2 \tag{19}$$

$$L_{train} = \|\mathbf{Z}_{final} - \mathbf{Z}_{AE}\|_2^2 \tag{20}$$

By incorporating these constraints into the contrast mechanism, we aim to encourage the model to acquire more trustworthy contrast signals throughout the pre-training and training stages. This ensures that the optimization process progresses in a more accurate direction, leading to improved model performance.

IV. Experiments

A. Datasets and Baselines

We conduct experiments on several real-world datasets, i.e., DBLP, CITE, and ACM. As shown in Table I, these datasets have different edge numbers and feature dimensions.

To showcase the effectiveness of our proposed method, we conduct a comparative analysis with several state-of-the-art (SOTA) methods. These methods include AE [32], DEC [33], and IDEC [34], which are autoencoder-based clustering approaches that learn representations for clustering by training an autoencoder. Additionally, GAE and VGAE [35], ARGA [36], and DAEGC [24] are representative graph convolutional network-based methods. These approaches embed the clustering representation with structural information using Graph Convolutional Networks (GCN). Furthermore, SDCN [25] and DFCN [31] are hybrid methods that leverage both AE and GCN modules for clustering.

B. Experimental Settings

For comparison baseline methods, we use their default parameters. For our model, in the part of loss function, we use two parameters α and β to control the pre-training loss and training loss which introduce comparative learning, respectively. For different data sets, the values of these two hyper-parameters are different, which will be introduced in detail in the parameter sensitivity experiment.

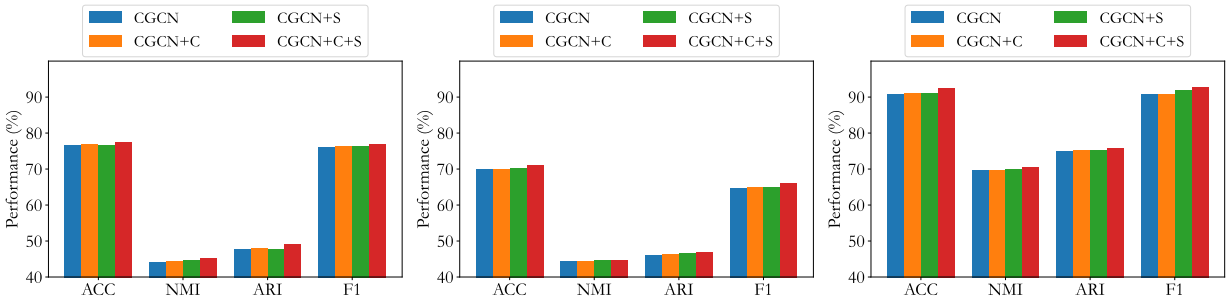
C. Node Clustering Performance

The clustering performance of our proposed CGCN method and the baseline methods on three benchmark datasets is summarized in Table II. Based on the results, we make the following observations:

* CGCN consistently outperforms the compared methods in most cases. K-means performs clustering on raw data, while AE, DEC, and IDEC only utilize node attribute representations for clustering, neglecting the structural information. SDCN and DFCN rely on simplistic fusion techniques, which do not fully capture the deeper levels of important data patterns. As a result, the reliability of the prior clustering distributions obtained through pre-training various models remains insufficient.
* GCN-based techniques like GAE, VGAE, ARGA, and DAEGC lack the same level of effectiveness demonstrated by CGCN. These approaches inadequately exploit the information inherent in the data and may be subject to constraints related to oversmoothing. In contrast, CGCN integrates attribute-based representations acquired through AE within the comprehensive clustering framework. This integration enables concurrent exploration of graph structure and node attributes, enabled by a fusion module that facilitates consensus representation learning. Consequently, CGCN delivers a substantial enhancement in clustering performance in comparison to existing GCN-based methods.
* CGCN achieves better clustering results than the strongest baseline method, DFCN, in the majority of cases, particularly on the DBLP, ACM, and CITE datasets. For example, on the DBLP dataset, CGCN achieves a 1.71% improvement in accuracy (ACC), a 3.43% improvement in normalized mutual information (NMI), a 4.46% improvement in adjusted Rand index (ARI), and a 1.58% improvement in F1 score compared to DFCN. This improvement is attributed to CGCN's contrast learning mechanism, which promotes information interoperability among multiple modules during pre-training, enhancing

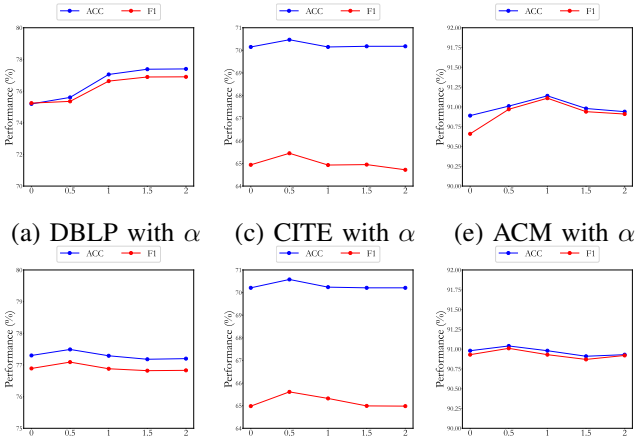
TABLE II: Node Clustering Performance. We bold the best results and add underline to the second best results.

Datasets	Mteric	Kmeans	AE	DEC	IDEC	GAE	VGAE	ARGA	DAEGC	SDCNQ	SDCN	DFCN	CGCN	Improv.
DBLP	ACC	38.7±0.7	51.4±0.4	58.2±0.6	60.3±0.6	61.2±1.2	58.6±0.1	61.6±1.0	62.1±0.5	65.7±1.3	68.1±1.8	76.0±0.8	**77.3±0.2**	+1.71%
	NMI	11.5±0.4	25.4±0.2	29.5±0.3	31.2±0.5	30.8±0.9	26.9±0.1	26.8±1.0	32.5±0.5	35.1±1.1	39.5±1.3	43.7±1.0	**45.2±0.4**	+3.43%
	ARI	7.0±0.4	12.2±0.4	23.9±0.4	25.4±0.6	22.0±1.4	17.9±0.1	22.7±0.3	21.0±0.5	34.0±1.8	39.2±2.0	47.0±1.5	**49.1±0.5**	+4.46%
	F1	31.9±0.3	52.5±0.4	59.4±0.5	61.3±0.6	61.4±2.2	58.7±0.1	61.8±0.9	61.8±0.7	65.8±1.2	67.7±1.5	75.7±0.8	**76.9±0.6**	+1.58%
CITE	ACC	39.3±3.2	57.1±0.1	55.9±0.2	60.5±1.4	61.4±0.8	61.0±0.4	56.9±0.7	64.5±1.4	61.7±1.1	66.0±0.3	69.5±0.2	**70.3±0.1**	+1.16%
	NMI	16.9±3.2	27.6±0.1	28.3±0.3	27.2±2.4	34.6±0.7	32.7±0.3	34.5±0.8	36.4±0.9	34.4±1.2	38.7±0.3	43.9±0.2	**44.7±0.2**	+1.83%
	ARI	13.4±3.0	29.3±0.1	28.1±0.4	25.7±2.7	33.6±1.2	33.1±0.5	33.4±1.5	37.8±1.2	35.5±1.5	40.2±0.4	45.5±0.3	**46.5±0.1**	+2.20%
	F1	36.1±3.5	53.8±0.1	52.6±0.2	61.6±1.4	57.4±0.8	57.7±0.5	54.8±0.8	62.2±1.3	57.8±1.0	63.6±0.2	64.3±0.2	**65.1±0.1**	+1.25%
ACM	ACC	67.3±0.7	81.8±0.1	84.3±0.8	85.1±0.5	84.5±1.4	84.1±0.2	86.1±1.2	86.9±2.8	87.0±0.1	90.5±0.2	90.9±0.2	**91.5±0.1**	+0.67%
	NMI	32.4±0.5	49.3±0.2	54.5±1.5	56.6±1.2	55.4±1.9	53.2±0.5	55.7±1.4	56.2±4.2	58.9±0.2	68.3±0.3	69.4±0.4	**70.3±0.2**	+1.30%
	ARI	30.6±0.7	54.6±0.2	60.6±1.9	62.2±1.5	59.5±3.1	57.7±0.7	62.9±2.1	59.4±3.9	65.3±0.2	73.9±0.4	74.9±0.4	**75.6±0.2**	+0.94%
	F1	67.6±0.7	82.0±0.1	84.5±0.7	85.1±0.5	84.7±1.3	84.2±0.2	86.1±1.2	87.1±2.8	86.8±0.1	90.4±0.2	90.8±0.2	**91.1±0.2**	+0.34%



(a) DBLP (b) CITE (c) ACM

Fig. 2: Ablation Study.



(a) DBLP with α (c) CITE with α (e) ACM with α

(b) DBLP with β (d) CITE with β (f) ACM with β

Fig. 3: Parameter sensitivity study.

the reliability of the prior clustering distribution. Furthermore, CGCN allows for flexible adjustment of the degree of information aggregation related to different order structures, enabling adaptive changes throughout the training process.

D. Ablation Study

To further validate the effectiveness of our proposed method, we conducted ablation studies. In these studies, we compared the performance of DFCN with two modified versions: DFCN+C and DFCN+S. DFCN+C represents the introduction of only the contrastive learning mechanism, while DFCN+S denotes adjusting the degree of information aggregation related to various order structure information.

Across all three datasets, we consistently observed that both DFCN+C and DFCN+S outperformed the baseline DFCN. The improvement in clustering performance was more significant when the contrastive learning mechanism was introduced, compared to solely adjusting the degree of information aggregation related to various order structure information. Moreover, when both mechanisms were employed simultaneously in DFCN+C+S, even better results were achieved. These findings collectively highlight that our proposed method can leverage more comprehensive information to enhance the generalization capabilities of deep clustering networks.

E. Parameter Sensitivity

The introduction of two hyperparameters, α and β, in CGCN allows us to control the proportion of dissimilarity between Z_{AE} and $\tilde{\mathbf{A}}$ and between Z_{AE} and Z_{IGAE} during training. To investigate the impact of these parameters on all datasets, we conducted experiments by fixing the value of one parameter and varying the value of the other parameter from 0 to 2 in increments of 0.5. The results, as shown in the figures, provide the following insights:

* The hyper-parameters α and β have a significant effect on improving the clustering performance, and the adjustment of their ratio yields different clustering outcomes.
* The method exhibits stable performance changes within the range of values for α and β.

* The optimal ratios of the two hyper-parameters vary for different datasets. For the ACM dataset, the best model performance is achieved when α:β = 0.5:1. For the CITE dataset, the optimal parameters are α:β = 0.5:0.5. Lastly, for the DBLP dataset, the optimal parameters are α:β = 2:0.5. These findings suggest that the optimal ratio of the hyper-parameters depends on the specific characteristics and complexities of each dataset.

V. Conclusion

In order to overcome the limitations of neglecting deep contrast and structural information in depth graph clustering, we present a novel method called Contrastive Graph Convolutional Network (CGCN). Our approach integrates contrast signals and deep structural information into the pre-training phase. Experimental evaluations conducted on various real graph datasets demonstrate that our proposed CGCN method enhances the reliability of the prior clustering distributions obtained from pre-training, leading to improved model performance. Moving forward, our future research endeavors will focus on developing scalable graph clustering frameworks to accommodate larger-scale graph datasets.

References

[1] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *TKDE*, 2018.

[2] L. Yue, X. Jun, Z. Sihang, W. Siwei, G. Xifeng, Y. Xihong, L. Ke, T. Wenxuan, L. X. Wang *et al.*, "A survey of deep graph clustering: Taxonomy, challenge, and application," *arXiv preprint arXiv:2211.12875*, 2022.

[3] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, and F. Sun, "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multimodal," *arXiv*, 2022.

[4] D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, P. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[5] J. Ma, Y. Liu, M. Liu, and M. Han, "Curriculum contrastive learning for fake news detection," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4309–4313.

[6] S. Cavallari, V. W. Zheng, H. Cai, K. C.-C. Chang, and E. Cambria, "Learning community embedding with community detection and node embedding on graphs," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 377–386.

[7] M. Liu and Y. Liu, "Inductive representation learning in temporal networks via mining neighborhood and community influences," in *SIGIR*, 2021.

[8] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[9] 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.

[10] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning deep representations for graph clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1, 2014.

[11] 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.

[12] S. Wang, J. Yang, J. Yao, Y. Bai, and W. Zhu, "An overview of advanced deep graph node clustering," *IEEE Transactions on Computational Social Systems*, 2023.

[13] N. Mrabah, M. Bouguessa, M. F. Touati, and R. Ksantini, "Rethinking graph auto-encoder models for attributed graph clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[14] Y. Liu, K. Liang, J. Xia, X. Yang, S. Zhou, M. Liu, X. Liu, and S. Z. Li, "Reinforcement graph clustering with unknown cluster number," *arXiv preprint arXiv:2308.06827*, 2023.

[15] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 10, 2021, pp. 8547–8555.

[16] E. Pan and Z. Kang, "Multi-view contrastive graph clustering," *Advances in neural information processing systems*, vol. 34, pp. 2148–2159, 2021.

[17] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014.

[18] M. Liu, K. Liang, D. Hu, H. Yu, Y. Liu, L. Meng, W. Tu, S. Zhou, and X. Liu, "Tmac: Temporal multi-modal graph learning for acoustic event classification," in *ACM MM*, 2023.

[19] H. Yu, C. Ma, M. Liu, X. Liu, Z. Liu, and M. Ding, "Guardfl: Safeguarding federated learning against backdoor attacks through attributed client graph clustering," *arXiv preprint arXiv:2306.04984*, 2023.

[20] L. Meng, K. Liang, B. Xiao, S. Zhou, Y. Liu, M. Liu, X. Yang, and X. Liu, "Sarf: Aliasing relation assisted self-supervised learning for few-shot relation reasoning," *arXiv preprint arXiv:2304.10297*, 2023.

[21] M. Liu, K. Liang, B. Xiao, S. Zhou, W. Tu, Y. Liu, X. Yang, and X. Liu, "Self-supervised temporal graph learning with temporal and structural intensity alignment," *arXiv preprint arXiv:2302.07491*, 2023.

[22] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[23] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "Mgae: Marginalized graph autoencoder for graph clustering," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 889–898.

[24] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: A deep attentional embedding approach," *arXiv preprint arXiv:1906.06532*, 2019.

[25] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proceedings of the web conference 2020*, 2020, pp. 1400–1410.

[26] G. Cui, J. Zhou, C. Yang, and Z. Liu, "Adaptive graph encoder for attributed graph embedding," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 976–985.

[27] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *International conference on machine learning*. PMLR, 2020, pp. 4116–4126.

[28] H. Zhao, X. Yang, Z. Wang, E. Yang, and C. Deng, "Graph debiased contrastive learning with joint representation clustering." in *IJCAI*, 2021, pp. 3434–3440.

[29] M. Liu, Y. Liu, K. Liang, S. Wang, S. Zhou, and X. Liu, "Deep temporal graph clustering," *arXiv preprint arXiv:2305.10738*, 2023.

[30] F. Devvrit, A. Sinha, I. Dhillon, and P. Jain, "S3gc: Scalable self-supervised graph clustering," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3248–3261, 2022.

[31] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, and J. Cheng, "Deep fusion clustering network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9978–9987.

[32] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[33] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*. PMLR, 2016, pp. 478–487.

[34] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation." in *Ijcai*, vol. 17, 2017, pp. 1753–1759.

[35] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[36] S. Pan, R. Hu, S.-f. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2475–2487, 2019.