Multi-view Unsupervised Graph Representation Learning

Jiangzhang Gan[1,2] , **Rongyao Hu**[2] , **Mengmeng Zhan**[1] , **Yujie Mo**[1] , **Yingying Wan**[1] and **Xiaofeng Zhu**[1,3*]

[1]Center for Future Media and School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]School of Mathematical and Computational Science, Massey University Auckland Campus, Auckland 0632, New Zealand
[3]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518000, China

Abstract

Both data augmentation and contrastive loss are the key components of contrastive learning. In this paper, we design a new multi-view unsupervised graph representation learning method including adaptive data augmentation and multi-view contrastive learning, to address some issues of contrastive learning ignoring the information from feature space. Specifically, the adaptive data augmentation first builds a feature graph from the feature space, and then designs a deep graph learning model on the original representation and the topology graph to update the feature graph and the new representation. As a result, the adaptive data augmentation outputs multi-view information, which is fed into two GCNs to generate multi-view embedding features. Two kinds of contrastive losses are further designed on multi-view embedding features to explore the complementary information among the topology and feature graphs. Additionally, adaptive data augmentation and contrastive learning are embedded in a unified framework to form an end-to-end model. Experimental results verify the effectiveness of our proposed method, compared to state-of-the-art methods.

1 Introduction

Unsupervised graph representation learning (UGRL) can output discriminative representations using both structural (*e.g.,* the relationship of data points) and non-structured (*e.g.,* node feature) information of the data, making it easily extract useful information to benefit downstream tasks [Wang *et al.*, 2020a; Gan *et al.*, 2021]. To achieve this, the objective function of the UGRL is often designed to maximize mutual information (MI) between the input and its related information. However, the MI maximization is popular transferred to the Jensen-Shannon divergence maximization, as its computation cost is expensive [Liu *et al.*, 2021]. This results in contrastive learning based UGRL (CL-UGRL). Recently, CL-UGRL shows its superiority in unsupervised representation learning, and has been widely used in various learning tasks, such as clustering and community detection [Jaiswal *et al.*, 2021].

The CL-UGRL method mainly includes three key components, *i.e.,* data augmentation, graph convolutional network (GCN) encoder, and contrastive loss. Specifically, data augmentation aims at creating rational data for contrastive learning by applying certain transformation [Qiu *et al.*, 2020]. GCN encodes the topology structure and node features of the data. The contrastive loss is related to the definitions of anchors, positive samples and negative samples. Moreover, it pushes the anchor embedding similar to positive emedding and dissimilar to negative embedding [Tian *et al.*, 2021]. Although current CL-UGRL methods have achieved success, due to the complexity of the data, previous CL-UGRL methods still have many limitations.

The commonly used graph data augmentation methods are random perturbation, including node dropping, feature masking, subgraph and edge perturbation. For example, to conduct data augmentation, [Zhang *et al.*, 2020] employed node dropping and subgraph, while [You *et al.*, 2020] employed edge perturbation and feature masking. However, random perturbation might destroy the inherent structure and property of the graph, affecting the model performance. To address this issue, [Hassani and Khasahmadi, 2020] used the diffusion method to generate augmented data, which avoids the destruction of the data structure. However, such a method is independent of contrastive learning, and thus fails to adaptively consider the impacts of input graphs.

Compared to unsupervised representation learning methods, the UGRL maintains the graph structure of data points during representation learning, thereby outputting more discriminative representations [Goyal and Ferrara, 2018]. Therefore, the quality of the graph is very important to the performance of the UGRL method. In current UGRL methods, the topology graph and the feature graph are two popular graph structures to represent the relationship between data points. The topology graph usually comes from the real world to reflect the connection of data points, such as social networks and citation networks [Yang *et al.*, 2019]. The feature graph is constructed by the kNN method or the ϵ-graph method to represent the similarity of data points in the feature

---

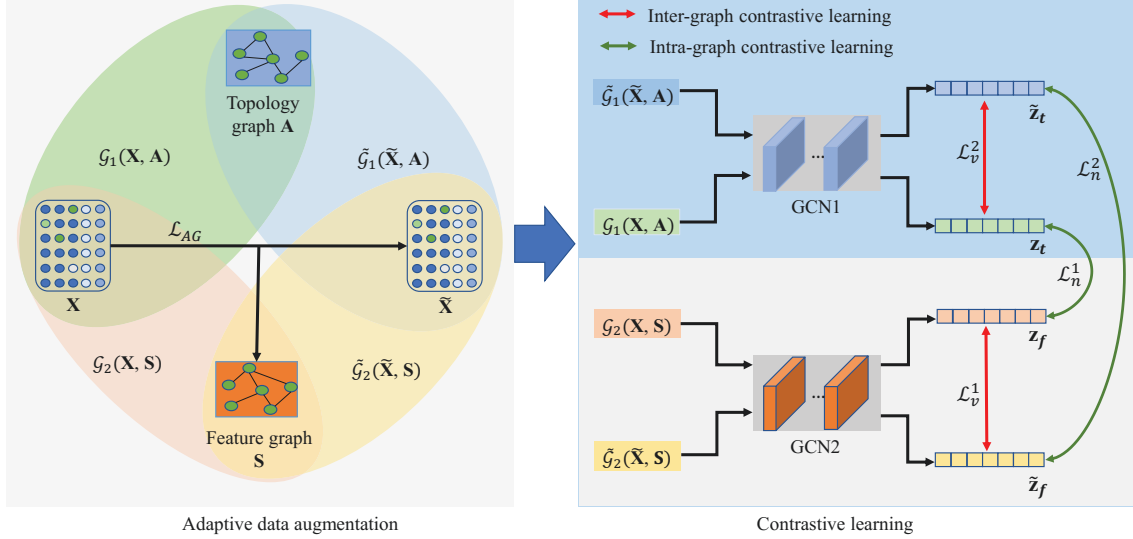*Corresponding author (seanzhuxf@gmail.com).

Figure 1: The flowchart of our method. It first proposes an adaptive data augmentation to generate multi-view information, *i.e.,* $\mathcal{G}_1 = (\mathbf{X}, \mathbf{A})$, $\widetilde{\mathcal{G}}_1 = (\widetilde{\mathbf{X}}, \mathbf{A})$, $\mathcal{G}_2 = (\mathbf{X}, \mathbf{S})$ and $\widetilde{\mathcal{G}}_2 = (\widetilde{\mathbf{X}}, \mathbf{S})$, and then designs two types of contrastive losses, *i.e.,* intra-graph contrastive loss and inter-graph contrastive loss, for conducting multi-view contrastive learning.

space [Liu *et al.*, 2020]. These two kinds of graphs contain different information, which may provide complementary inforamtion to benefit the UGRL. However, existing UGRL usually cannot take full advantage of the information. Therefore, the fusion of two graphs is a way to improve the UGRL.

In this paper, we propose a multi-view CL-UGRL method to address the aforementioned issues by two key components, *i.e.,* adaptive data augmentation and multi-view contrastive learning. Our adaptive data augmentation first generates the feature graph from the feature space and then designs a deep graph learning model to jointly update the feature graph and the new representation based on the original representation. We further combine the feature and topology graphs with the original and new representations to form multi-view information, which is fed into two GCNs to generate multi-view embedding features. Two kinds of contrastive losses are designed on multi-view embedding features to explore the complementary information between the topology and feature graphs. Different from traditional data augmentation independent on contrastive learning, we embed adaptive data augmentation and multi-view contrastive learning in a framework to form an end-to-end model, which jointly conducts data augmentation and multi-view contrastive learning. As a result, data augmentation is iteratively updated by the adjusted multi-view contrastive learning and vice versa.

Compared to previous methods, the main contributions of our method are summarized as follows.

- First, we propose a new data augmentation method, which generates a new graph and new representation, aiming at preserving the intrinsic structure of the data to generate multi-view information for contrastive learning. Furthermore, it is dependent on our multi-view contrastive learning as both of them are embedded in the same framework.

- Second, we propose a new multi-view contrastive learning framework, including two kinds of contrastive losses, which provide complementary information between the feature and topology graphs to strength multi-view contrastive learning.

- Third, we embed adaptive data augmentation and multi-view contrastive learning in a unified framework so that each of them can be iteratively adjusted by the other's update. Extensive experiments on benchmark data sets clearly demonstrate that our method outperforms the state-of-the-art methods on different downstream tasks.

2 Method

In this paper, $\mathcal{G}_1 = (\mathbf{X}, \mathbf{A})$ consists of $\mathbf{A}$ and $\mathbf{X}$, where $\mathbf{X} \in \mathbb{R}^{N \times d}$ and $\mathbf{A} \in \mathbb{R}^{N \times N}$ represent the feature matrix and the topology graph, respectively. N is the number of nodes and d is the dimensions of node features. Moreover, $a_{ij} = 1$ indicates that there is an edge between the i-th node and the j-th node, otherwise $a_{ij} = 0$.

The proposed framework is shown in Figure 1. Its key components include adaptive data augmentation and multiple contrastive learning. Specifically, given $\mathbf{X}$ and $\mathbf{A}$, our adaptive data augmentation adaptively learns both the new representations $\widetilde{\mathbf{X}}$ and the feature graph $\mathbf{S}$ of $\mathbf{X}$, followed by generating multi-view information, *i.e.,* $\mathcal{G}_1 = (\mathbf{X}, \mathbf{A})$, $\widetilde{\mathcal{G}}_1 = (\widetilde{\mathbf{X}}, \mathbf{A})$, $\mathcal{G}_2 = (\mathbf{X}, \mathbf{S})$ and $\widetilde{\mathcal{G}}_2 = (\widetilde{\mathbf{X}}, \mathbf{S})$. In particular, two-view information with the same graph, *e.g.,* $\mathcal{G}_1$ and $\widetilde{\mathcal{G}}_1$ (or $\mathcal{G}_2$ and $\widetilde{\mathcal{G}}_2$), is input into the same encoder to output new embedding features $\mathbf{Z}_t$ and $\widetilde{\mathbf{Z}}_t$ (or $\mathbf{Z}_f$ and $\widetilde{\mathbf{Z}}_f$), respectively. Based on resulted multi-view embedding features, we further design the intra-graph contrastive loss and the inter-graph contrastive loss to conduct multi-view contrastive learning.

2.1 Adaptive Data Augmentation

Existing data augmentation strategies are popularly used for Euclidean data, but are difficult for the graph data [Chen *et al.*, 2020]. In the literature, graph contrastive learning usually relies on the contrast between node embedding features in different views [Rai *et al.*, 2021]. To achieve this, data augmentation usually corrupts original data structures to obtain multi-view information by random perturbation, including node dropping, edge perturbation, feature masking, *etc.* However, random perturbation might destroy the intrinsic structure of the data. For example, either the removal or addition of edges might drastically change the identity or even the compound validity of bio-molecule data [You *et al.*, 2020]. Thus, data augmentation should preserve the intrinsic structures of the data, which can help learnt embedding features insensitive to the perturbation on unimportant nodes and edges. Additionally, previous data augmentation is independent on contrastive learning. If these two processes are integrated in the same framework, each of them can be alternatively adjusted by the other. In this way, the weakness of data augmentation can further be improved by contrastive learning, while the updated data augmentation can adjust contrastive learning. However, few literature focused on this.

In this paper, we develop a novel adaptive data augmentation method to dynamically preserve the intrinsic structures of the data in this section and then integrates it with multi-view contrastive learning in a unified framework (Section 2.4). More specifically, the adaptive data augmentation includes two parts, *i.e.,* node feature augmentation and graph structure augmentation, while multi-view contrastive learning is used to explore complementary information between the feature and topology graphs.

Node Feature Augmentation

The method of randomly masking node features recovers masked node features using their remaining features, and is a widely used method of data augmentation [Qiu *et al.*, 2020]. However, it might destroy the intrinsic structure of the data, thereby affecting the generation capability of the model. Hence, an effective data augmentation method is demand.

In traditional machine learning methods, feature selection is widely used to remove redundant features. For example, [Nie *et al.*, 2010] employed a joint $\ell_{2,1}$-norm regularization to select important features while [Wang *et al.*, 2020b] utilized sparse subspace learning to learn feature weights. Motivated by this, we propose to design a new data augmentation method to keep significant feature unchanged while perturbing trivial features. Specifically, we propose to learn the significance of features and mask trivial features. To achieve this, we first define the following objective function:

$$\min_{\mathbf{P}} \ \|\mathbf{X} - \mathbf{XP}\|_F^2 + \|\mathbf{P}\|_F^2 \tag{1}$$

where $\mathbf{P} \in \mathbb{R}^{d \times d}$ is the weight matrix of features. We focus on corrupting the original feature graph at its feature levels. Specifically, we define an indicator vector $\mathbf{m} \in \{0, 1\}^{d \times 1}$ to assign zeros to the features with the least weights in the weight matrix $\mathbf{P}$. As a result, the generated new node representation $\widetilde{\mathbf{X}}$ by the proposed node feature generation is:

$$\widetilde{\mathbf{X}} = [\mathbf{x}_1 \circ \mathbf{m}, ..., \mathbf{x}_N \circ \mathbf{m}] \tag{2}$$

where $\circ$ is the element-wise multiplication.

Graph Structure Augmentation

Edge perturbation perturbs the connectivity of the graph by randomly adding or dropping certain ratio of edges. However, deleting useful edges will reduce the discriminative ability of embedding features, and thus it has been proven to be unfriendly to biological and chemical data [You *et al.*, 2020]. Additionally, the graph structure is invariant during the process of data augmentation. However, the feature graph $\mathbf{S}$ is constructed by the original representations $\mathbf{X}$, which might contain noise or redundancy to affect the quality of $\mathbf{S}$. Eq. (1) is used to output better representation than $\mathbf{X}$, while the new representation can be used to further update $\mathbf{S}$, so that the updated $\mathbf{S}$ can really reflect the intrinsic structure of the data. Graph learning has been designed to dynamically adjust the graph structure, thereby learning the intrinsic structure of the data. For example, [Zhou *et al.*, 2020] designed to use dynamic graph learning for unsupervised feature selection while [Chen *et al.*, 2019] proposed to embed representation learning with graph learning in a unified framework. Inspired by this, we design a graph learning method to highlight the intrinsic structures of the data as follows.

$$\min_{\mathbf{S}} \ \sum_{i,j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\| s_{ij} + \lambda \|\mathbf{S}\|_F^2$$
$$\forall i, \mathbf{s}_i \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0 \tag{3}$$

where $\mathbf{1}$ represents the all-one-element vector and λ is a tuning parameter. In this work, Eq. (1) and Eq. (3) update the weight matrix $\mathbf{P}$ and the graph $\mathbf{S}$, respectively. Hence, we integrate Eq. (1) and Eq. (3) in a unified framework as follows:

$$\mathcal{L}_{AG} = \|\mathbf{X} - \mathbf{XP}\|_F^2 + \lambda_1 \sum_{i,j=1}^{N} \|\mathbf{x}_i \mathbf{P} - \mathbf{x}_j \mathbf{P}\| s_{ij} + \lambda_2 \|\mathbf{S}\|_F^2$$
$$\forall i, \mathbf{s}_i \mathbf{1} = 1, s_{i,i} = 0, s_{i,j} \geq 0 \tag{4}$$

where λ_1 and λ_2 are tuning parameters and $\mathbf{P}$ is a trainable parameter. Based on [Jiang *et al.*, 2019], we update $\mathbf{S}$ by:

$$s_{ij} = \frac{exp(ReLU(\mathbf{q}^T|\mathbf{x}_i\mathbf{P} - \mathbf{x}_j\mathbf{P}|))}{\sum_{j=1}^{n} exp(ReLU(\mathbf{q}^T|\mathbf{x}_i\mathbf{P} - \mathbf{x}_j\mathbf{P}|))} \tag{5}$$

where $ReLU(.)$ is the activation function and $\mathbf{q} \in \mathbb{R}^{h \times 1}$ is a trainable parameter.

Given the new representation $\widetilde{\mathbf{X}}$ and feature graph $\mathbf{S}$, multiple-view information (*i.e.,* $\widetilde{\mathcal{G}}_1 = (\widetilde{\mathbf{X}}, \mathbf{A})$, $\mathcal{G}_2 = (\mathbf{X}, \mathbf{S})$ and $\widetilde{\mathcal{G}}_2 = (\widetilde{\mathbf{X}}, \mathbf{S})$) is generated by our proposed adaptive data augmentation method. Different from previous data augmentation, our method iteratively updates both $\mathbf{S}$ and $\widetilde{\mathbf{X}}$, thereby avoiding the drawbacks of previous methods, *e.g.,* sensitive to perturbation on trivial nodes and edges, and destroying the structures of the data.

2.2 GCN Encoder

Given multi-view information (*i.e.,* $\mathcal{G}_1$, $\mathcal{G}_2$, $\widetilde{\mathcal{G}}_1$ and $\widetilde{\mathcal{G}}_2$), two GCN encoders are utilized to generate multi-view embedding features as different graphs might provide complementary information to each other [Kang *et al.*, 2020]. For example, the

topology graph $\mathbf{A}$ contains the similarity from the real world while the feature graph $\mathbf{S}$ gathers the similarity from the feature space. Specifically, given the topology graph $\mathbf{A}$, we input both $\mathcal{G}_1$ and $\widetilde{\mathcal{G}}_1$ into GCN1 to output their corresponding embedding features $\mathbf{Z}_t$ and $\widetilde{\mathbf{Z}}_t$, aiming at simultaneously updating the new representation $\widetilde{\mathbf{X}}$ and the embedding features. In particular, GCN1 is used for both $\mathcal{G}_1$ and $\widetilde{\mathcal{G}}_1$ as they share the same topology graph, with which it is easy to output a high quality $\widetilde{\mathbf{X}}$. Similarly, we input both $\mathcal{G}_2$ and $\widetilde{\mathcal{G}}_2$ into GCN2 to output embedding features $\mathbf{Z}_f$ and $\widetilde{\mathbf{Z}}_f$, as well as to update both $\widetilde{\mathbf{X}}$ and $\mathbf{S}$.

$$\begin{cases} \mathbf{Z}_f = GCN2(\mathbf{X}, \mathbf{A}) = MLP(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}_f^{(0)})\mathbf{W}_f^{(1)}) \\ \mathbf{Z}_t = GCN1(\mathbf{X}, \mathbf{S}) = MLP(\hat{\mathbf{S}}\sigma(\hat{\mathbf{S}}\mathbf{X}\mathbf{W}_t^{(0)})\mathbf{W}_t^{(1)}) \end{cases} \quad (6)$$

$$\begin{cases} \widetilde{\mathbf{Z}}_f = GCN2(\widetilde{\mathbf{X}}, \mathbf{A}) = MLP(\hat{\mathbf{A}}\sigma(\hat{\mathbf{A}}\widetilde{\mathbf{X}}\mathbf{W}_f^{(0)})\mathbf{W}_f^{(1)}) \\ \widetilde{\mathbf{Z}}_t = GCN1(\widetilde{\mathbf{X}}, \mathbf{S}) = MLP(\hat{\mathbf{S}}\sigma(\hat{\mathbf{S}}\widetilde{\mathbf{X}}\mathbf{W}_t^{(0)})\mathbf{W}_t^{(1)}) \end{cases} \quad (7)$$

where $\hat{\mathbf{A}} = \mathbf{D_A}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\mathbf{D_A}^{-\frac{1}{2}}$, $\hat{\mathbf{S}} = \mathbf{D_S}^{-\frac{1}{2}}(\mathbf{S} + \mathbf{I})\mathbf{D_S}^{-\frac{1}{2}}$, $\mathbf{D_A}$ and $\mathbf{D_S}$ are the degree matrix of $\mathbf{A} + \mathbf{I}$ and $\mathbf{S} + \mathbf{I}$, respectively, $MLP(.)$ represents the Multi-Layer Perception.

2.3 Multi-view Contrastive Learning

The key idea of contrastive learning is to define semantically similar (positive) and dissimilar (negative) pairs, encouraging the embedding features of similar pairs $(\mathbf{x}, \mathbf{x}^+)$ to be close, and those of dissimilar pairs $(\mathbf{x}, \mathbf{x}^-)$ to be far away. Specifically, the contrastive learning is to achieve the following objective:

$$sim(f(\mathbf{x}), f(\mathbf{x}^+)) \gg sim(f(\mathbf{x}), f(\mathbf{x}^-)) \quad (8)$$

where $f(.)$ represents encoder and $sim(.)$ measures the similarity of embedding features of two nodes. As our proposed adaptive data augmentation generates multi-view information, we design two kinds of contrastive losses (*i.e.,* intra-graph contrastive loss and inter-graph contrastive loss) to explore rich contrastive relations among embedding features, *i.e.,* $\mathbf{Z}_f$, $\mathbf{Z}_t$, $\widetilde{\mathbf{Z}}_f$, and $\widetilde{\mathbf{Z}}_t$.

To do this, we employ two GCNs to generate multi-view embedding features. A intra-graph contrastive loss for the embedding features derived from the same encoder is then built, aiming at separating positive embeddings from negative embeddings within the same graph structure. Meanwhile, an inter-graph contrastive loss for the embedding features from different graph structures can also be built, aiming to separate positive embeddings from negative embeddings across graphs.

Inter-graph Contrastive Learning

The inter-graph contrastive learning is to contrast the embedding features from different graph structures, *i.e.,* the topology graph and the feature graph. Specifically, an inter-graph contrastive loss is designed to pull the embedding features of the same node from different views across graph structures close while pushing the embedding features of one node far away from the embedding features of other nodes from different views across graph structures, *i.e.,*

$$\mathcal{L}_n^1(v_i) = -log \frac{exp(sim(\mathbf{z}_f^{v_i}, \mathbf{z}_t^{v_i}))}{\sum_{j=1}^N exp(sim(\mathbf{z}_f^{v_i}, \mathbf{z}_t^{v_j}))} \quad (9)$$

$$\mathcal{L}_n^2(v_i) = -log \frac{exp(sim(\widetilde{\mathbf{z}}_f^{v_i}, \widetilde{\mathbf{z}}_t^{v_i}))}{\sum_{j=1}^N exp(sim(\widetilde{\mathbf{z}}_f^{v_i}, \widetilde{\mathbf{z}}_t^{v_j}))} \quad (10)$$

where v_i represents the i-th node, $sim(a,b) = exp(\frac{a^T b}{\|a\| \cdot \|b\|})/\vartheta$, ϑ represents the temperature factor that controls the concentration level of the distribution. Besides, $\mathbf{z}_f^{v_i} \in \mathbf{Z}_f$, $\mathbf{z}_t^{v_i} \in \mathbf{Z}_t$, $\widetilde{\mathbf{z}}_f^{v_i} \in \mathbf{Z}_f$, $\widetilde{\mathbf{z}}_t^{v_i} \in \mathbf{Z}_t$. Combining the above two losses, the inter-graph contrastive loss is defined as follows:

$$\mathcal{L}_n = \frac{1}{2N} \sum_{i=1}^N (\mathcal{L}_n^1(v_i) + \mathcal{L}_n^2(v_i)) \quad (11)$$

Intra-graph Contrastive Learning

Intra-graph contrastive loss is used to build the contrast from the same graph structure, which is popular in CL-UGRL [Xie *et al.*, 2021]. In this paper, the intra-graph contrastive loss is designed to pull the embedding features of the same node from different views in the same graph structure close while pushing the embedding features of one node far away from the embedding features of other nodes from both the same view and different views, which come from the same graph structure, *i.e.,*

$$\mathcal{L}_v^1(v_i) = -log \frac{exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_i}))}{\sum_{j=1}^N exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_j}))} \quad (12)$$

However, the node shares the same graph structure but is with different views, which might cause the embedding features in different views similar by Eq. (12). To make the embedding features of the same node in different views distinguishable, we consider adding the diversity constraints into the intra-graph contrasitive loss.

$$\mathcal{L}_v^1(v_i) = -log \frac{exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_i}))}{\sum_{j=1}^N exp(sim(\mathbf{z}_f^{v_i}, \widetilde{\mathbf{z}}_f^{v_j})) + \Theta} \quad (13)$$

where $\Theta = \sum_{j=1}^N exp(sim(\mathbf{z}_f^{v_i}, \mathbf{z}_f^{v_j}))$. Similarly, we define $\mathcal{L}_v^2(v_i)$ between $\mathcal{G}_2$ and $\widetilde{\mathcal{G}}_2$ as:

$$\mathcal{L}_v^2(v_i) = -log \frac{exp(sim(\mathbf{z}_t^{v_i}, \widetilde{\mathbf{z}}_t^{v_i}))}{\sum_{j=1}^N exp(sim(\mathbf{z}_t^{v_i}, \widetilde{\mathbf{z}}_t^{v_j})) + \sum_{j=1}^N exp(sim(\mathbf{z}_t^{v_i}, \mathbf{z}_t^{v_j}))} \quad (14)$$

Combining Eq. (13) with Eq. (14), the intra-graph contrastive loss $\mathcal{L}_v$ is:

$$\mathcal{L}_v = \frac{1}{2N} \sum_{i=1}^N (\mathcal{L}_v^1(v_i) + \mathcal{L}_v^2(v_i)) \quad (15)$$

Finally, combining the inter-graph loss with the intra-graph loss, the objective function of the proposed multi-view contrastive learning is defined as follows.

$$\mathcal{L}_{CL} = (1 - \eta)\mathcal{L}_v + \eta\mathcal{L}_n \quad (16)$$

where η is a non-negative tuning parameter.

The intra-graph contrasitive loss is defined in different views of the same graph structure, while the inter-graph contrastive losses is defined in different views across different graph structures. Hence, Eq. (16) considers the individual graph structures (*e.g.,* the feature graph or the topology graph) by inter-graph contrastive losses as well as the connection between two graphs by inter-graph losses, so that it easily obtains complementary information among different graph structures.

Datasets	GCN	GAT	DGI	GMI	DGCRL	MVGRL	Proposed
Citeseer	70.30 ± 0.50	72.20 ± 0.26	71.80 ± 0.45	72.4 ± 0.26	72.53 ± 0.46	72.56 ± 0.50	72.87 ± 0.40
Cora	81.50± 0.20	83.00± 0.30	82.30± 0.31	83.00± 0.30	83.10± 0.40	82.58 ± 0.50	83.85± 0.35
Wiki-CS	77.19± 0.12	77.65± 0.11	73.53± 0.14	74.85± 0.08	74.89± 0.26	77.52± 0.08	79.56± 0.14
Computers	86.51± 0.19	86.93 ± 0.29	83.95± 0.47	82.21± 0.31	84.71± 0.19	87.52± 0.24	87.98± 0.30

Table 1: Node classification accuracy (%) of all methods on four data sets.

Datasets	Pro-R-F	Pro-R-T	Pro-R-DA	Proposed
Citeseer	72.10±0.20	71.38± 0.40	71.59± 0.37	72.87± 0.40
Cora	83.14± 0.25	82.81± 0.20	81.25± 0.18	83.85± 0.35
Wiki-CS	79.14± 0.07	78.59± 0.06	78.37± 0.10	79.56± 0.14
Computers	86.32± 0.18	87.53± 0.30	85.37± 0.24	87.98± 0.30

Table 2: Node classification accuracy of four methods.

Datasets	Proposed w/o n	Proposed w/o v	Proposed
Citeseer	71.65±0.15	72.10± 0.33	72.87± 0.40
Cora	82.94± 0.18	82.58± 0.04	83.85± 0.35
Wiki-CS	77.59± 0.14	77.58± 0.13	79.56± 0.14
Computers	87.10± 0.14	86.32± 0.24	87.98± 0.30

Table 3: Classification accuracy of three methods on four data sets.

2.4 Overall Objective Function

In order to train an end-to-end CL-UGRL model, we jointly consider multi-view contrastive learning and adaptive data augmentation to have our final objective function as follows.

$$\mathcal{L} = \mathcal{L}_{CL} + \beta \mathcal{L}_{AG} \qquad (17)$$

where β is a tuning parameter. As a result, adaptive data augmentation is updated by multi-view contrastive learning and vice versa. Eq. (17) can be optimized by the standard gradient descent algorithm. We list the pseudo of our method in Appendix. In the downstream task, the mean function is employed to aggregate embedding features outputted by GCN1 and GCN2, *i.e.,* $\mathbf{Z} = (Mean(\mathbf{Z}_t, \widetilde{\mathbf{Z}}_t) \| Mean(\mathbf{Z}_f, \widetilde{\mathbf{Z}}_f))$, where $\|$ denotes the concatenation operator.

3 Experiments

Data Sets
The used data sets include citation networks data (*i.e.,* Citeseer, Cora), reference network data (*i.e.,* Wiki-CS), and networks data (*i.e.,* Computers).

Comparison Methods
The comparison methods include unsupervised learning methods (*i.e.,* Deep Graph Infomax (DGI) [Veličković *et al.*, 2018], Graphical Mutual Information Maximization (GMI) [Peng *et al.*, 2020], Multi-View Graph Representation Learning (MVGRL) [Hassani and Khasahmadi, 2020] and Deep Graph Contrastive Representation Learning (DGCRL) [Zhu *et al.*, 2020]) and supervised learning methods (*i.e.,* GCN [Kipf and Welling, 2016] and Graph Attention Networks (GAT) [Veličković *et al.*, 2017]).

3.1 Result Analysis
We report the results of node classification of all methods in Table 1. Our method achieves the best performance, followed by MVGRL, DGCRL, GMI and DGI, in terms of unsupervised representation learning methods. Specifically, our method improves by on average 1.02%, 2.28%, 2.95%, and 3.17%, compared to four unsupervised comparison methods, in terms of classification accuracy, on four data sets. Compared to semi-supervised methods (*i.e.,* GCN and GAT) which adopt the label information in the learning process, our method also achieve the superior performance. This demonstrates that our method can output representations that are beneficial for downstream learning tasks. Moreover, it is feasible to generate auxiliary information to mine the information hidden in the graph data through contrastive learning. The reasons is that our adaptive data augmentation method produces important information of the graph data from both the structure and feature levels, while all unsupervised comparison methods adopt the random perturbation method to generate augmented data. Moreover, our method makes full use of the topology and feature information of the graph data.

3.2 Ablation Study
Effectiveness of Aata Augmentation
To verify the effectiveness of our data augmentation method, we propose three new methods, *i.e.,* Proposed-R-F, Proposed-R-T and Proposed-R-DA, based on our method. Specifically, Proposed-R-F removes the first term of Eq. (4), *i.e.,* without considering the adaptive data augmentation of node features. Proposed-R-T removes the second and third terms of Eq. (4), *i.e.,* without considering the adaptive data augmentation of the graph structure. Proposed-R-DA employs random perturbation to conduct data augmentation, *i.e.,* removing the adaptive augmentation module from our method.

We report the results of four methods in Table 2. First, our method improves by on average 1.92%, compared to Proposed-R-DA method, on all data sets. This illustrates that our adaptive augmentation method is effective in preserving the intrinsic structure and significant features of the data. Second, Proposed-R-F and Proposed-R-T improve by on average 1.03% and 0.93%, respectively, compared to Proposed-
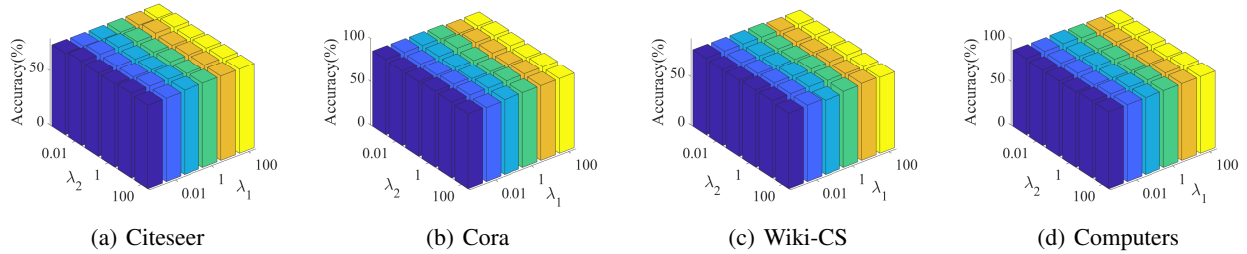
Figure 2: Classification accuracy of our method at different parameter settings (*i.e.,* λ_1 and λ_2) on four data sets.

R-DA, on four data sets. This contributes to the fact that both the node feature and the topology structure are important for the UGRL. This clearly demonstrates the feasibility of the adaptive augmentation of either the node feature or the topology structure.

Effectiveness of Contrastive Losses

To validate the effectiveness of contrastive losses, we conduct experiments to compare our method with its two variants, *i.e.,* Proposed w/o n and Proposed w/o v. Specifically, Proposed w/o n denotes our method in Eq. (4) with only $\mathcal{L}_v$, *i.e.,* removing inter-graph contrastive loss $\mathcal{L}_n$ in Eq. (4). Proposed w/o v denotes our method in Eq. (4) with only $\mathcal{L}_n$. The results are presented in Table 3. The performance of our method degrades without one of them on all data sets. This demonstrates the effectiveness of our contrastive losses. Specifically, our method improves by on average 1.24% and 1.42%, respectively, compared to Proposed w/o n and Proposed w/o v, in terms of node classification. This improvement can be attributed to our multi-view contrastive learning scheme. It also indicates that it is feasible to extract the feature structure and the topology structure of the data.

3.3 Parameters Sensitivity Analysis

We evaluate the effectiveness of our method at different parameters' settings, *i.e.,* λ_1 and λ_2 in Eq. (4) and summarize the classification results of our method at different settings (*e.g.,* λ_1 and $\lambda_2 \in \{10^{-2}, 10^{-1}, ..., 10^2\}$) in Figure 2. Obviously, our method is not sensitive to the setting of either λ_1 or λ_2 as different settings make little changes on the classification accuracy, *e.g.,* $\leq 1\%$ in our experiments. This shows that our method easily achieves good performance by setting the ranges of λ_1 and λ_2 as [0.001,0.1].

4 Conclusion

In this paper, we designed a novel CL-UGRL method to embed data augmentation and multi-view contrastive learning in a unified framework. To do this, this paper first proposed an adaptive data augmentation method to maintain the intrinsic structure of the data, and then designed contrastive losses to explore complementary information among two different graph structures. As a result, data augmentation and multi-view contrastive learning are iteratively adjusted to preserve the intrinsic of the data as well as extract complementary information. Experimental results on four real data sets demonstrated the effectiveness of our method, compared to state-of-the-art comparison methods.

References

[Chen *et al.*, 2019] Yu Chen, Lingfei Wu, and Mohammed J Zaki. Deep iterative and adaptive learning for graph neural networks. *arXiv preprint arXiv:1912.07832*, 2019.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[Gan *et al.*, 2021] Jiangzhang Gan, Ziwen Peng, Xiaofeng Zhu, Rongyao Hu, Junbo Ma, and Guorong Wu. Brain functional connectivity analysis based on multi-graph fusion. *Medical image analysis*, 71:102057, 2021.

[Goyal and Ferrara, 2018] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126, 2020.

[Jaiswal *et al.*, 2021] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

[Jiang *et al.*, 2019] Bo Jiang, Ziyan Zhang, Doudou Lin, Jin Tang, and Bin Luo. Semi-supervised learning with graph learning-convolutional networks. In *CVPR*, pages 11313–11320, 2019.

[Kang *et al.*, 2020] Zhao Kang, Guoxin Shi, Shudong Huang, Wenyu Chen, Xiaorong Pu, Joey Tianyi Zhou, and Zenglin Xu. Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*, 189:105102, 2020.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Liu *et al.*, 2020] Zhonghua Liu, Zhihui Lai, Weihua Ou, Kaibing Zhang, and Ruijuan Zheng. Structured optimal graph based sparse feature extraction for semi-supervised learning. *Signal Processing*, 170:107456, 2020.

[Liu *et al.*, 2021] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[Nie *et al.*, 2010] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. *Advances in neural information processing systems*, 23, 2010.

[Peng *et al.*, 2020] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, pages 259–270, 2020.

[Qiu *et al.*, 2020] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, pages 1150–1160, 2020.

[Rai *et al.*, 2021] Nishant Rai, Ehsan Adeli, Kuan-Hui Lee, Adrien Gaidon, and Juan Carlos Niebles. Cocon: Cooperative-contrastive learning. In *CVPR*, pages 3384–3393, 2021.

[Tian *et al.*, 2021] Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*, 2021.

[Veličković *et al.*, 2017] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[Veličković *et al.*, 2018] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[Wang *et al.*, 2020a] Feng Wang, Huaping Liu, Di Guo, and Fuchun Sun. Unsupervised representation learning by invariancepropagation. *arXiv preprint arXiv:2010.11694*, 2020.

[Wang *et al.*, 2020b] Zheng Wang, Feiping Nie, Lai Tian, Rong Wang, and Xuelong Li. Discriminative feature selection via a structured sparse subspace learning module. In *IJCAI*, pages 3009–3015, 2020.

[Xie *et al.*, 2021] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021.

[Yang *et al.*, 2019] Liang Yang, Zesheng Kang, Xiaochun Cao, Di Jin, Bo Yang, and Yuanfang Guo. Topology optimization based graph convolutional network. In *IJCAI*, pages 4054–4061, 2019.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *NIPS*, 33:5812–5823, 2020.

[Zhang *et al.*, 2020] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. Motif-driven contrastive learning of graph representations. *arXiv preprint arXiv:2012.12533*, 2020.

[Zhou *et al.*, 2020] Peng Zhou, Liang Du, Xuejun Li, Yi-Dong Shen, and Yuhua Qian. Unsupervised feature selection with adaptive multiple graph learning. *Pattern Recognition*, 105:107375, 2020.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.