

Simple Contrastive Graph Clustering

Yue Liu, Xihong Yang, Sihang Zhou¹, Xinwang Liu¹, *Senior Member, IEEE*,
Siwei Wang¹, Ke Liang, Wenxuan Tu, and Liang Li¹

Abstract—Contrastive learning has recently attracted plenty of attention in deep graph clustering due to its promising performance. However, complicated data augmentations and time-consuming graph convolutional operations undermine the efficiency of these methods. To solve this problem, we propose a simple contrastive graph clustering (SCGC) algorithm to improve the existing methods from the perspectives of network architecture, data augmentation, and objective function. As to the architecture, our network includes two main parts, that is, preprocessing and network backbone. A simple low-pass denoising operation conducts neighbor information aggregation as an independent preprocessing, and only two multilayer perceptrons (MLPs) are included as the backbone. For data augmentation, instead of introducing complex operations over graphs, we construct two augmented views of the same vertex by designing parameter unshared Siamese encoders and perturbing the node embeddings directly. Finally, as to the objective function, to further improve the clustering performance, a novel cross-view structural consistency objective function is designed to enhance the discriminative capability of the learned network. Extensive experimental results on seven benchmark datasets validate our proposed algorithm’s effectiveness and superiority. Significantly, our algorithm outperforms the recent contrastive deep clustering competitors with at least seven times speedup on average. The code of SCGC is released at SCGC. Besides, we share a collection of deep graph clustering, including papers, codes, and datasets at ADGC.

Index Terms—Attribute graph clustering, contrastive learning, multilayer perceptrons (MLPs), self-supervised learning.

NOMENCLATURE

Notation	Meaning
$\mathbf{X} \in \mathbb{R}^{N \times D}$	Attribute matrix.
$\mathbf{X}_s \in \mathbb{R}^{N \times D}$	Smoothed attribute matrix.
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Original adjacency matrix.
$\hat{\mathbf{A}} \in \mathbb{R}^{N \times N}$	Adjacency matrix with self-loop.
$\mathbf{I} \in \mathbb{R}^{N \times N}$	Identity matrix.

Manuscript received 25 June 2022; revised 20 January 2023; accepted 22 April 2023. This work was supported in part by the National Key Research and Development Program of China under Project 2020AAA0107100; and in part by the National Natural Science Foundation of China under Project 62006237, Project 61922088, Project 61906020, and Project 61773392. (*Corresponding author: Xinwang Liu.*)

Yue Liu, Xihong Yang, Xinwang Liu, Siwei Wang, Ke Liang, Wenxuan Tu, and Liang Li are with the College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: yueliu@nudt.edu.cn; xinwangliu@nudt.edu.cn).

Sihang Zhou is with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TNNLS.2023.3271871>.

Digital Object Identifier 10.1109/TNNLS.2023.3271871

$\mathbf{D} \in \mathbb{R}^{N \times N}$	Degree matrix.
$\mathbf{L} \in \mathbb{R}^{N \times N}$	Graph Laplacian matrix.
$\tilde{\mathbf{L}} \in \mathbb{R}^{N \times N}$	Symmetric normalized Laplacian matrix.
$\mathbf{Z}^{v_k} \in \mathbb{R}^{N \times d}$	Node embeddings in k th view.
$\mathbf{S} \in \mathbb{R}^{N \times N}$	Cross-view sample correlation matrix.
$\mathbf{Z} \in \mathbb{R}^{N \times d}$	Clustering-oriented node embeddings.

I. INTRODUCTION

GRAPH learning [1], [2], [3], [4] is becoming increasingly crucial in many applications like facial expression recognition [5], video action recognition [6], and the recommendation system [7] for its good hidden correlation exploiting capability. Among all the directions in graph learning, a fundamental and challenging task, that is, deep graph clustering, has recently attracted intensive attention [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

According to the learning mechanism, the existing deep graph clustering methods can be roughly categorized into three classes: generative methods [8], [21], [22], [23], [24], [25], [11], [12], adversarial methods [9], [10], [26], and contrastive methods [13], [14], [15], [16], [17], [18], [27]. In early literature, the generative methods and adversarial methods improve clustering performance by learning cluster-oriented node representations and designing fake sample generation-recognition mechanisms, respectively. However, since most of these methods adopt a clustering guided loss function [28] to force the generated sample embeddings to have the minimum distortion against the prelearned clustering centers [8], [9], [10], [11], [12], [25], [26], their clustering performance is highly dependent on good initial cluster centers, thus leading to manual trial-and-error pretraining. As a consequence, the performance consistency, as well as the implementing convenience, is largely decreased. More recently, thanks to the development of contrastive learning, more consistent and discriminative contrastive loss functions are designed to replace the clustering-guided loss function for network training. As a result, the manual trial-and-error problem is alleviated, and the clustering performance is improved [13], [14], [15], [16], [17], [18], [27]. However, complicated data augmentations and time-consuming graph convolutional operation undermine the efficiency of these methods, making them computational time- and space-consuming (see Section IV-D).

To solve these problems, we propose a simple contrastive graph clustering (SCGC) method to improve the existing methods from the aspect of network architecture, data augmentation, and objective function. To our network architecture, the backbone is designed with a Siamese network whose

subbranch merely consists of the multilayer perceptron (MLP). The neighborhood information aggregation process is conducted independently before network training. In this manner, we filter the high-frequency noise in attributes, thus improving both the clustering performance and training efficiency (see Sections IV-E1 and IV-D). For data augmentation, instead of constructing two different views of the same node with complex modification against graphs, we implement it by designing parameter unshared Siamese encoders and perturbing the node embeddings with Gaussian noise directly. Moreover, we design a neighbor-oriented contrastive objective function to force the cross-view similarity matrix to approximate the self-looped adjacency matrix. By this setting, the network is endowed with the ability to keep the cross-view structural consistency, thus further improving the clustering performance (see Section IV-E1). Benefiting from our simple architecture, SCGC is free from pretraining and outperforms the recent contrastive competitors with at least seven times speedup on average. Meanwhile, we save about 59% GPU memory against other contrastive methods on average (see Section IV-D). The main contributions of this article are summarized as follows.

- 1) We propose a simple, yet effective contrastive deep graph clustering method termed SCGC. Benefiting from its simplicity, SCGC is free from pretraining and saves both time and space for network training.
- 2) A new data augmentation method, which conducts data perturbation only in the enhanced attribute space, is proposed. This fashion is verified to be compatible with the existing contrastive methods.
- 3) We design a novel neighbor-oriented contrastive loss to keep the structural consistency even across views, thus improving the discriminative capability of our network.
- 4) Extensive experimental results on seven benchmark datasets demonstrate the superiority and efficiency of the proposed method against the existing state-of-the-art deep graph clustering competitors.

II. RELATED WORK

A. Deep Graph Clustering

Graph neural networks (GNNs), which possess powerful graph representation learning capability, have achieved impressive performance in knowledge graph and graph learning. Among various directions, deep graph clustering is a fundamental, yet challenging task that aims to reveal the underlying graph structure and divides the nodes into several disjoint groups. According to the learning mechanism, the existing deep graph clustering methods can be roughly categorized into three classes: generative methods [8], [11], [12], [21], [22], [23], [24], [25], adversarial methods [9], [10], [26], and contrastive methods [13], [14], [15], [16], [17], [18], [27], [29], [30], [31], [32]. Our proposed method belongs to the last category. We will review the generative methods and adversarial methods in this section and detail the difference between our proposed method and other contrastive methods in Section II-B. More detailed information about deep graph clustering can be found in the survey paper [33].

The pioneer graph clustering algorithm MGAE [22] embeds nodes into the latent space with GAE [21] and then performs clustering over the learned node embeddings. Subsequently, DAEGC [8] and MAGCN [25] improve the clustering performance of early works with the attention mechanisms [34], [35]. Besides, GALA [24] and AGC [23] enhance the GAE by the symmetric decoder and the high-order graph convolution operation, respectively. In addition, ARGAE [9], [10] and AGAE [26] improve the discriminative capability of samples through adversarial mechanisms [36], [37]. Moreover, SDCN [11] and DFCN [12] verify the effectiveness of the attribute–structure fusion mechanisms to improve the clustering performance.

Although verified to be effective, since most of these methods adopt a clustering guided loss function [28] to force the learned node embeddings to have the minimum distortion against the prelearned clustering centers, their clustering performance is highly dependent on good initial cluster centers, thus leading to manual trial-and-error pretraining [8], [9], [10], [11], [12], [25], [26]. As a consequence, the performance consistency, as well as the implementing convenience, is largely decreased. Unlike them, our proposed method replaces the clustering-guided loss function by designing a novel neighbor-oriented contrastive loss function, thus getting rid of trial-and-error pretraining.

B. Contrastive Deep Graph Clustering

Contrastive learning has achieved great success on images [38], [39], [40], [41], [42], [43] and graphs [44], [45], [46], [47], [48], [49], [50], [51] in recent years. Inspired by their success, contrastive deep graph clustering methods [13], [14], [15], [16], [17], [18], [27], [52] are increasingly proposed.

Three key factors, that is, data augmentation, network architecture, and objective function, significantly determine the clustering performance of the contrastive methods. According to these factors, we summarize the differences between our proposed SCGC and other contrastive deep graph clustering methods in Table I.

1) *Data Augmentation*: The existing data augmentations in contrastive methods aim to build different views of the same vertex by introducing complex operations over graphs. Specifically, MVGRL [14] and DCRN [16] adopt the graph diffusion matrix as an augmented graph. Besides, SCAGC [17] perturbs the graph topology by randomly adding or dropping edges. In addition, DCRN and SCAGC conduct augmentations on node attributes by the attribute perturbation. Although verified to be effective, these data augmentations are complicated and still entangle the aggregation and transformation during training, thus limiting the efficiency of the contrastive methods. Different from them [14], [16], [17], our SCGC constructs two augmented views of the same vertex by simply designing parameter unshared Siamese encoders and perturbing embeddings directly instead of introducing any complex operations over graphs.

2) *Network Architecture*: To the network architecture, SCAGC [17] and MGCCN [18] both encode nodes with the

TABLE I

DIFFERENCES BETWEEN OUR PROPOSED SCGC AND OTHER CONTRASTIVE DEEP GRAPH CLUSTERING METHODS FROM PERSPECTIVES OF DATA AUGMENTATION, NETWORK ARCHITECTURE, AND OBJECTIVE FUNCTION

Method	Data Augmentation	Network Architecture	Objective Function Form
AGE [13]	No Data Augmentation	Graph Filters+MLP	Cross-Entropy Loss
MVGRL [14]	Graph Diffusion	Un-shared GCNs+ shared MLP	InfoMax Loss
MCGC [15]	No Data Augmentation	Graph Filters	InfoNCE Loss
DCRN [16]	Attribute Perturbation+Graph Diffusion	Auto-Encoder+GCN	MSE Loss to Identity Matrix
SCAGC [17]	Attribute Perturbation+Edge Perturbation	Shared GCN	InfoNCE Loss
MGCCN [18]	No Data Augmentation	Shared GCN	InfoNCE Loss
SCGC (Ours)	Un-shared MLPs+Embedding Perturbation	Graph Filters+Un-shared MLPs	MSE Loss to Adjacent Matrix

shared GCN encoders [53]. Differently, MVGRL [14] adopt two-parameter unshared GCN encoders and the shared MLP as the backbone. In addition, DCRN [16] utilizes the auto-encoder [54] and GCN encoder to embed augmented views into the latent space. However, previous GCN encoders all entangle the transformation and aggregation operation during training, thus leading to high time costs. To solve this issue, AGE [13] decouples these two operations in GCN by a graph Laplacian filter [55] and one MLP. Different from AGE [13], we encode the smoothed node attributes with two separated MLPs, which have the same architecture but unshared parameters.

3) *Objective Function*: Third, for the objective function, MVGRL [14] designs the InfoMax loss [38] to maximize the cross-view mutual information between the node and the global summary of the graph. Meanwhile, AGE [13] designs a pretext task to classify the similar nodes and the dissimilar nodes by the cross-entropy loss. Subsequently, SCAGC [17], MCGC [15], and MGCCN [18] all adopt the infoNCE loss [56] to pull together the positive sample pairs while pushing away the negative sample pairs. Concretely, based on similarity, MCGC defines the positive samples as the k-nearest neighbors of the node while regarding other nodes as negative samples. SCAGC designs the contrastive clustering loss to maximize the agreement between representations of the same cluster. MGCCN pulls close the embeddings of the same node in different GCN layers and pushes away the embeddings of different nodes. In addition, DCRN designs the MSE loss to reduce the redundancy in the feature level and sample level. Different from them, we design a novel neighbor-oriented contrastive loss to keep the structural consistency even across views, thus improving the discriminative capability of our network.

III. METHODOLOGY

A. Notations and Problem Definition

Let $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ be a set of N nodes with C classes and \mathcal{E} be a set of edges. In the matrix form, $\mathbf{X} \in \mathbb{R}^{N \times d}$ and $\mathbf{A} \in \mathbb{R}^{N \times N}$ denote the attribute matrix and the original adjacency matrix, respectively. Then $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$ denotes an undirected graph. The degree matrix is formulated as $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_N) \in \mathbb{R}^{N \times N}$ and $d_i = \sum_{(v_i, v_j) \in \mathcal{E}} a_{ij}$. The graph Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$. With the renormalization trick $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ in GCN [53], the symmetric normalized graph Laplacian matrix is denoted as

$\tilde{\mathbf{L}} = \mathbf{I} - \hat{\mathbf{D}}^{-(1/2)} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-(1/2)}$. The notations are summarized in the Nomenclature.

Deep graph clustering aims to divide the nodes in the graph into several disjoint groups in an unsupervised manner. Concretely, a neural network \mathcal{F} is first trained in an unsupervised manner and encodes the nodes by exploiting node attributes and structural information as follows:

$$\mathbf{E} = \mathcal{F}(\mathbf{A}, \mathbf{X}) \quad (1)$$

where \mathbf{X} and \mathbf{A} denote the attribute matrix and the original adjacency matrix, respectively. Besides, $\mathbf{E} \in \mathbb{R}^{N \times d}$ is the learned node embeddings, where N is the number of samples and d is the number of feature dimensions. After that, a clustering algorithm \mathcal{C} such as K-means [57], spectral clustering [58], or clustering neural network [11] is adopted to divide learned node embeddings \mathbf{E} into k disjoint groups as follows:

$$\Phi = \mathcal{C}(\mathbf{E}) \quad (2)$$

where $\Phi \in \mathbb{R}^{N \times k}$ denotes the cluster membership matrix for all N nodes.

B. Overall Framework

We propose an SCGC algorithm. The framework of SCGC is shown in Fig. 1. It mainly consists of two components: low-pass denoising operation and structural contrastive module (SCM). In the following sections, we will detail low-pass denoising operation, SCM, and the objective function.

C. Low-Pass Denoising Operation

Recent works [13], [59], [60] have demonstrated that the Laplacian filter [55] can achieve the same effect as the graph convolution operation [53]. Motivated by their success, we introduce a low-pass denoising operation to conduct neighbor information aggregation as an independent preprocessing before training. In this manner, the high-frequency noise in attributes will be filtered out efficiently.

Concretely, we introduce a graph Laplacian filter formulated as

$$\mathbf{H} = \mathbf{I} - \tilde{\mathbf{L}} \quad (3)$$

where $\tilde{\mathbf{L}}$ denotes the symmetric normalized graph Laplacian matrix. Subsequently, we stack up t -layer graph Laplacian

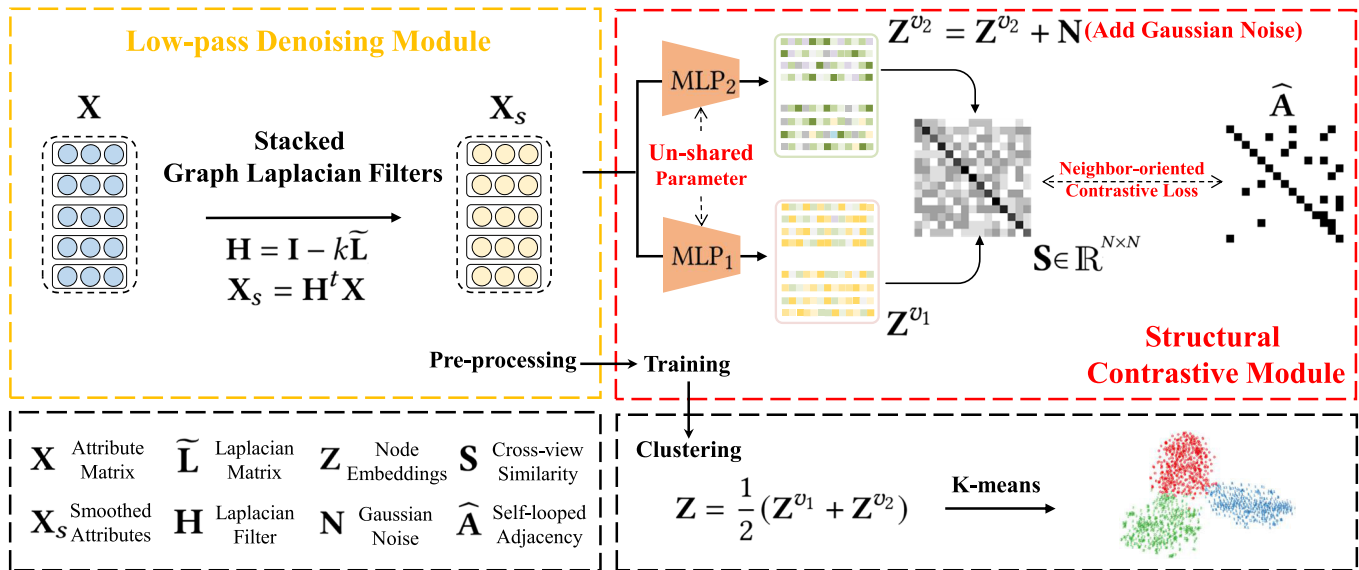


Fig. 1. Illustration of the SCGC algorithm. In our proposed algorithm, we first preprocess the node attributes by the low-pass denoising operation. Then, the SCM encodes the smoothed node attributes with merely two MLPs and constructs augmented views of node embeddings by designing parameter unshared Siamese encoders and perturbing the node embeddings. Moreover, a novel neighbor-oriented contrastive loss is designed to keep the cross-view structural consistency, thus improving the discriminative capability of the network.

filters as follows:

$$\begin{aligned} \mathbf{X}_s &= \left(\prod_{i=1}^t \mathbf{H} \right) \mathbf{X} \\ &= \mathbf{H}^t \mathbf{X} \end{aligned} \quad (4)$$

where \mathbf{X}_s denotes the smoothed attribute matrix. Besides, \mathbf{H}^t denotes the stacked t -layer graph Laplacian filters, which can filter out the high-frequency noise in node attributes.

Through this low-pass denoising operation, high-frequency noise in attributes is filtered out, thus improving the clustering performance and training efficiency (see Sections IV-E1 and IV-D).

D. Structural Contrastive Module

In this section, we design the SCM to keep the structural consistency even across two different views, thus enhancing the discriminative capability of the network.

To be specific, we first encode the smoothed attributes \mathbf{X}_s with the designed parameter unshared MLP encoders and then normalize the learned node embeddings with ℓ^2 -norm as follows:

$$\begin{aligned} \mathbf{Z}^{v1} &= \text{MLP}_1(\mathbf{X}_s), & \mathbf{Z}^{v1} &= \frac{\mathbf{Z}^{v1}}{\|\mathbf{Z}^{v1}\|_2} \\ \mathbf{Z}^{v2} &= \text{MLP}_2(\mathbf{X}_s), & \mathbf{Z}^{v2} &= \frac{\mathbf{Z}^{v2}}{\|\mathbf{Z}^{v2}\|_2} \end{aligned} \quad (5)$$

where \mathbf{Z}^{v1} and \mathbf{Z}^{v2} denote two augmented views of the learned node embeddings. It is worth mentioning that MLP_1 and MLP_2 have the same architecture but unshared parameters, thus \mathbf{Z}^{v1} and \mathbf{Z}^{v2} would contain different semantic information during training.

In addition, we further keep the difference between the two views by simply adding the random Gaussian noise to \mathbf{Z}^{v2}

formulated as

$$\mathbf{Z}^{v2} = \mathbf{Z}^{v2} + \mathbf{N} \quad (6)$$

where $\mathbf{N} \in \mathbb{R}^{N \times d}$ is sampled from the Gaussian distribution $\mathcal{N}(0, \sigma)$. In summary, we construct two augmented views \mathbf{Z}^{v1} and \mathbf{Z}^{v2} by designing parameter unshared encoders and perturbing the node embeddings directly instead of introducing complex operations against graphs, thus improving the training efficiency (see Section IV-D). Besides, recent works [61], [62], [63] have indicated that the complex data augmentations over graphs, like edge adding, edge dropping, and graph diffusion could lead to semantic drift. A similar conclusion is verified through experiments in Section IV-E2.

Subsequently, we design a novel neighbor-oriented contrastive loss to keep cross-view structural consistency. Concretely, we calculate the cross-view sample similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ between \mathbf{Z}^{v1} and \mathbf{Z}^{v2} formulated as

$$\mathbf{S}_{ij} = \mathbf{z}_i^{v1} \cdot \left(\mathbf{z}_j^{v2} \right)^T \quad \forall i, j \in [1, N] \quad (7)$$

where \mathbf{S}_{ij} denotes the cosine similarity between the i th node embedding in the first view and the j th node embedding in the second view. Then, we force the cross-view sample similarity matrix \mathbf{S} to be equal to the self-looped adjacency matrix $\hat{\mathbf{A}} \in \mathbb{R}^{N \times N}$ formulated as

$$\begin{aligned} \mathcal{L} &= \frac{1}{N^2} \sum (\mathbf{S} - \hat{\mathbf{A}})^2 \\ &= \frac{1}{N^2} \left(\sum_i \sum_j \mathbb{1}_{ij}^1 (\mathbf{S}_{ij} - 1)^2 + \sum_i \sum_j \mathbb{1}_{ij}^0 \mathbf{S}_{ij}^2 \right) \end{aligned} \quad (8)$$

where $\mathbb{1}_{ij}^1$ denotes if $\hat{\mathbf{A}}_{ij} = 1$ and $\mathbb{1}_{ij}^0$ denotes if $\hat{\mathbf{A}}_{ij} = 0$. Here, we consider the cross-view neighbors of the same node as the positive samples while regarding other nonneighbor nodes as negative samples. Then we pull together the positive

TABLE II
STATISTICS SUMMARY OF SEVEN DATASETS

Dataset	Type	Sample	Dimension	Edge	Class
CORA	Graph	2708	1433	5429	7
CITeseer	Graph	3327	3703	4732	6
AMAP	Graph	7650	745	119081	8
BAT	Graph	131	81	1038	4
EAT	Graph	399	203	5994	4
UAT	Graph	1190	239	13599	4
CORAFULL	Graph	19793	8710	63421	70

samples while pushing away the negative samples. More precisely, in (8), the first term forces the nodes to agree with their neighbors even across two different views, while the second term minimizes the agreement between the node and its nonneighbors. This neighbor-oriented contrastive objective function enhances the discriminative capability of our network by keeping the cross-view structural consistency, thus improving the clustering performance (see Section IV-E1).

E. Fusion and Clustering

In this section, we first fuse the two augmented views of the node embeddings linearly formulated as

$$\mathbf{Z} = \frac{1}{2}(\mathbf{Z}^{v_1} + \mathbf{Z}^{v_2}) \quad (9)$$

where $\mathbf{Z} \in \mathbb{R}^{N \times d}$ denotes the resultant clustering-oriented node embeddings. Then, we directly perform the K-means algorithm [57] over \mathbf{Z} and obtain the clustering results.

Algorithm 1 Simple Contrastive Graph Clustering

Input: The input graph $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$; The cluster number C ; The iteration number I ; The graph Laplacian filter layer number t ; The Gaussian noise's standard deviation σ .

Output: The clustering result \mathbf{R} .

- 1: Obtain the smoothed attributes \mathbf{X}_s by applying t -layers stacked graph Laplacian filters to the attributes \mathbf{X} in Eq. (3)-(4).
 - 2: **for** $i = 1$ to I **do**
 - 3: Encode \mathbf{X}_s into two augmented views with parameter un-shared Siamese MLP encoders and then normalize them in Eq. (5).
 - 4: Perturb node embeddings by adding the Gaussian noise in Eq. (6).
 - 5: Calculate cross-view sample similarity matrix \mathbf{S} by Eq. (7).
 - 6: Force \mathbf{S} to approach the self-looped adjacency $\hat{\mathbf{A}}$ and calculate the neighbor-oriented contrastive loss \mathcal{L} in Eq. (8).
 - 7: Fuse \mathbf{Z}^{v_1} and \mathbf{Z}^{v_2} to obtain \mathbf{Z} in Eq. (9).
 - 8: Update model by minimizing \mathcal{L} with Adam optimizer.
 - 9: **end for**
 - 10: Obtain \mathbf{R} by performing K-means over \mathbf{Z} .
 - 11: **return** \mathbf{R}
-

F. Objective Function

The optimization objective of the proposed method is the neighbor-oriented contrastive loss \mathcal{L} in (8). We analyze the time complexity of our proposed neighbor-oriented contrastive loss. Given the sample number N and learned feature dimension d , the time complexity of calculating the cross-view sample similarity matrix \mathbf{S} is $\mathcal{O}(N^2d)$. Then, the time complexity of aligning \mathbf{S} with $\hat{\mathbf{A}}$ is $\mathcal{O}(N^2)$. Thus, the whole-time complexity of the proposed loss is $\mathcal{O}(N^2d + N^2) = \mathcal{O}(N^2d)$. We minimize \mathcal{L} with the widely used Adam optimizer [65] during training. The detailed learning process of our proposed SCGC is shown in Algorithm 1.

IV. EXPERIMENT

A. Dataset

To evaluate the effectiveness and efficiency of our proposed SCGC, we conduct extensive experiments on seven benchmark datasets, including CORA [13], CITeseer [13], Brazil Air-Traffic (BAT) [66], Europe Air-Traffic (EAT) [66], USA Air-Traffic (UAT) [66], Amazon Photo (AMAP) [16], and CORAFULL [16]. The brief information of these datasets is summarized in Table II.

B. Experimental Setup

All experimental results are obtained from the desktop computer with the Intel Core i7-6800K CPU, one NVIDIA GeForce RTX 3090 GPU, 64 GB RAM, and the PyTorch deep learning platform.

1) *Training Procedure:* Our network is trained for 400 epochs until convergence by minimizing the contrastive loss in (8) with the Adam optimizer [65]. After optimization, we directly perform the K-means algorithm [57] on the clustering-oriented node embeddings \mathbf{Z} . To avoid the influence of randomness, we conduct ten runs for all compared methods and report the average values with standard deviations of four metrics.

2) *Parameter Settings:* To MCGC [15], we run their source code on merely the graph datasets in Table II for fairness. For other baselines, we reproduce results by adopting their source code with the original settings. In our proposed method, both MLPs consist of a single 500-D embedding layer. The learning rate of the optimizer is set to 1e-3 for CORA/BAT/EAT/UAT, 1e-4 for CORAFULL, 5e-5 for CITeseer, and 1e-5 for AMAP. The layer number t of graph Laplacian filters is set to 2 for CORA/CITeseer/CORAFULL, 3 for BAT/UAT, and 5 for AMAP/EAT. The standard deviation σ of random Gaussian noise is set to 0.01.

3) *Metrics:* To verify the superiority of our SCGC compared with baselines, the clustering performance is evaluated by four widely used metrics, that is, ACC, NMI, ARI, and F1 [67], [68], [69], [70], [71], [72], [73], [74], [75].

C. Performance Comparison

To demonstrate the superiority of our proposed SCGC algorithm, we compare SCGC with 13 baselines. Specifically, K-means [57] is a classic clustering algorithm. Besides, two representative deep clustering methods, that is, AE [54] and

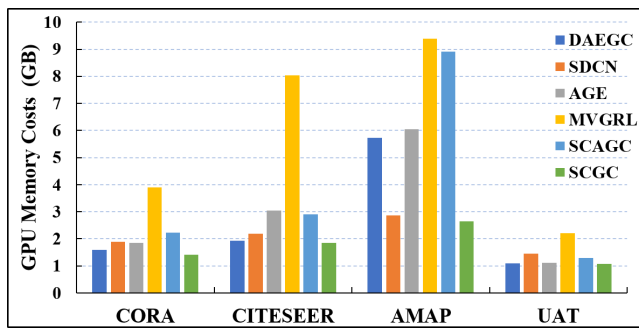


Fig. 2. GPU memory costs of six methods on four datasets.

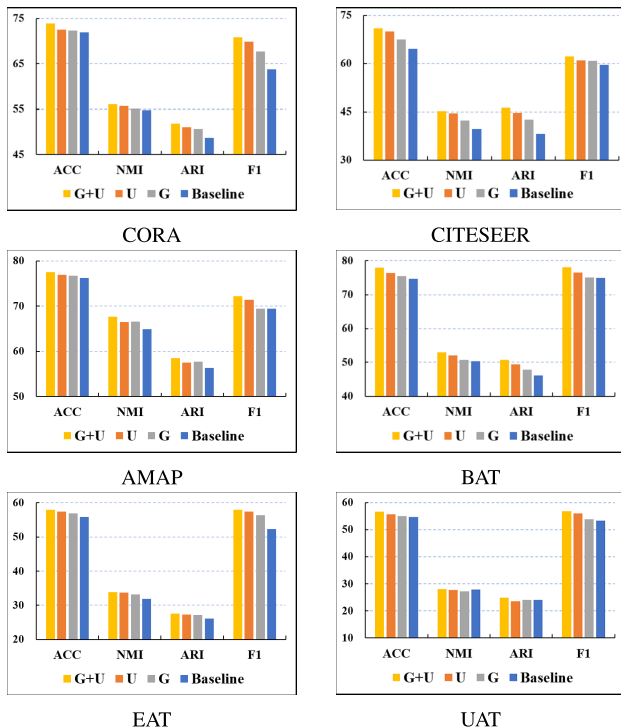


Fig. 3. Ablation studies of unshared MLPs and Gaussian noise on six datasets.

First, we test the training time of our SCGC and nine baselines on six datasets. Concretely, the baselines contain one deep clustering method DEC [28], one spectral-based method SSGC [76], three classical deep graph clustering methods [8], [11], [21], and seven contrastive methods [13], [14], [15], [16], [17], [27], [63]. We run all methods on the desktop platform with the Intel Core i7-6800K CPU, one NVIDIA GeForce RTX 3090 GPU, and 64 GB RAM. For the baselines, we adopt their original source code. For fairness, all methods are trained with 400 epochs. From the results in Table IV, we observe that our method consistently achieves the fastest speed on six datasets. Significantly, SCGC outperforms the recent contrastive deep clustering competitors with at least seven times speedup on average. We summarize two key reasons as follows. 1) The network architecture of SCGC is simple and merely consists of two MLPs. 2) Similar to [13], [49], [59], and [76], our method decouples the GCN [53] and adopts the low-pass denoising operations as an independent preprocessing to conduct neighbor information aggregation, thus simplifying the training process. Besides, the

TABLE VI
ABLATION STUDIES OF LOW-PASS DENOISING OPERATION AND SCM. RESULTS ARE REPORTED WITH MEAN \pm STD UNDER TEN RUNS. BOLD VALUES ARE THE BEST RESULTS

Dataset	Metric	(w/o) L & SCM	(w/o) SCM	(w/o) L	L+SCM
CORA	ACC	33.80 \pm 2.71	56.42 \pm 4.96	57.81 \pm 0.82	73.88\pm0.88
	NMI	14.98 \pm 3.43	44.51 \pm 3.69	34.65 \pm 1.25	56.10\pm0.72
	ARI	08.60 \pm 1.95	32.70 \pm 4.22	28.70 \pm 1.27	51.79\pm1.59
	F1	30.26 \pm 4.46	47.75 \pm 7.31	52.29 \pm 0.90	70.81\pm1.96
CITESEER	ACC	39.32 \pm 3.17	57.48 \pm 3.94	65.59 \pm 0.86	71.02\pm0.77
	NMI	16.94 \pm 3.22	37.95 \pm 2.30	39.03 \pm 0.66	45.25\pm0.45
	ARI	13.43 \pm 3.02	34.00 \pm 2.76	38.66 \pm 0.50	46.29\pm1.13
	F1	36.08 \pm 3.53	44.90 \pm 3.97	60.75 \pm 1.25	64.80\pm1.01
AMAP	ACC	27.22 \pm 0.76	29.58 \pm 1.85	45.32 \pm 1.69	77.48\pm0.37
	NMI	13.23 \pm 1.33	14.87 \pm 3.08	29.58 \pm 2.98	67.67\pm0.88
	ARI	05.50 \pm 0.44	05.47 \pm 1.35	19.26 \pm 2.02	58.48\pm0.72
	F1	23.96 \pm 0.51	26.01 \pm 2.88	41.59 \pm 0.36	72.22\pm0.97
BAT	ACC	40.23 \pm 1.19	56.11 \pm 2.44	70.46 \pm 0.38	77.97\pm0.99
	NMI	26.92 \pm 2.39	34.79 \pm 1.93	48.47 \pm 0.37	52.91\pm0.68
	ARI	9.520 \pm 1.42	24.13 \pm 2.27	45.11 \pm 0.36	50.64\pm1.85
	F1	34.45 \pm 2.10	54.35 \pm 3.13	68.47 \pm 0.52	78.03\pm0.96
EAT	ACC	32.23 \pm 0.56	51.03 \pm 2.03	53.86 \pm 1.12	57.94\pm0.42
	NMI	11.02 \pm 1.21	32.27 \pm 0.71	29.17 \pm 1.63	33.91\pm0.49
	ARI	02.20 \pm 0.40	24.03 \pm 1.07	23.87 \pm 2.24	27.51\pm0.59
	F1	23.49 \pm 0.92	47.99 \pm 1.74	52.70 \pm 0.17	57.96\pm0.46
UAT	ACC	42.47 \pm 0.15	39.79 \pm 2.50	46.45 \pm 0.79	56.58\pm1.62
	NMI	22.39 \pm 0.69	12.99 \pm 1.70	21.66 \pm 0.94	28.07\pm0.71
	ARI	15.71 \pm 0.76	07.04 \pm 1.15	15.35 \pm 0.64	24.80\pm1.85
	F1	36.12 \pm 0.22	38.09 \pm 4.19	43.72 \pm 1.24	55.52\pm0.87

TABLE VII
PERFORMANCE COMPARISONS OF DIFFERENT AUGMENTATIONS ON SIX DATASETS. ALL RESULTS ARE REPORTED WITH MEAN \pm STD UNDER TEN RUNS. THE BOLD VALUES INDICATE THE BEST RESULTS

Dataset	Metric	Drop	Add	Diffusion	Ours
CORA	ACC	72.03 \pm 0.93	72.02 \pm 0.91	72.94 \pm 0.75	73.88\pm0.88
	NMI	54.85 \pm 0.95	54.73 \pm 1.12	55.83 \pm 1.14	56.10\pm0.72
	ARI	49.71 \pm 1.65	49.66 \pm 1.70	51.44 \pm 1.40	51.79\pm1.59
	F1	70.07 \pm 0.80	69.86 \pm 0.96	69.67 \pm 1.71	70.81\pm1.96
CITESEER	ACC	67.74 \pm 0.51	68.31 \pm 0.56	68.15 \pm 0.49	71.02\pm0.77
	NMI	42.86 \pm 0.59	42.94 \pm 0.84	43.12 \pm 0.53	45.25\pm0.45
	ARI	43.28 \pm 0.65	43.69 \pm 0.82	43.40 \pm 0.92	46.29\pm1.13
	F1	63.38 \pm 0.3	63.54 \pm 1.17	63.02 \pm 1.57	64.80\pm1.01
AMAP	ACC	76.94 \pm 0.36	76.78 \pm 0.37	77.15 \pm 0.56	77.48\pm0.37
	NMI	66.82 \pm 0.90	66.41 \pm 1.30	67.43 \pm 0.85	67.67\pm0.88
	ARI	58.07 \pm 0.91	57.81 \pm 0.62	58.42 \pm 0.98	58.48\pm0.72
	F1	71.65 \pm 0.47	72.14 \pm 1.05	71.23 \pm 0.75	72.22\pm0.97
BAT	ACC	57.25 \pm 2.08	67.63 \pm 1.04	69.01 \pm 2.47	77.94\pm0.99
	NMI	29.23 \pm 3.98	42.51 \pm 1.20	44.89 \pm 2.23	52.91\pm0.68
	ARI	24.28 \pm 2.32	37.47 \pm 1.93	39.62 \pm 3.27	50.64\pm1.85
	F1	56.46 \pm 2.86	67.30 \pm 0.76	68.36 \pm 2.81	78.03\pm0.96
EAT	ACC	51.03 \pm 2.03	57.37 \pm 0.26	54.74 \pm 1.43	57.94\pm0.42
	NMI	32.27 \pm 0.71	32.62 \pm 0.39	29.25 \pm 1.50	33.91\pm0.49
	ARI	24.03 \pm 1.07	26.27 \pm 0.74	24.42 \pm 1.58	27.51\pm0.59
	F1	47.99 \pm 1.74	57.43 \pm 0.6	53.98 \pm 1.82	57.96\pm0.46
UAT	ACC	51.54 \pm 1.00	53.61 \pm 0.11	55.13 \pm 1.20	56.58\pm1.62
	NMI	18.81 \pm 2.13	24.59 \pm 1.07	25.52 \pm 1.80	28.07\pm0.71
	ARI	16.69 \pm 2.57	18.10 \pm 0.86	22.68 \pm 2.06	24.80\pm1.85
	F1	49.89 \pm 1.31	52.16 \pm 1.04	54.08 \pm 1.31	55.52\pm0.87

preprocessing time of the proposed method is also important. The preprocessing time is additionally tested with the same device. From the results in Table V, it is observed that the preprocessing does not take a long time (about 11.28% additional time) in our proposed method.

Second, we conduct experiments to test GPU memory costs of SCGC and five baselines including two classical deep graph clustering methods [8], [11] and three contrastive methods [13], [14], [17] on four datasets. From the results in Fig. 2, two conclusions are obtained as follows. 1) SCGC achieves comparable memory costs as the classical deep graph clustering methods including DAEGC [8] and SDCN [11]. 2) Compared to the contrastive methods [13], [14], [17], our proposed method saves about 59% GPU memory on

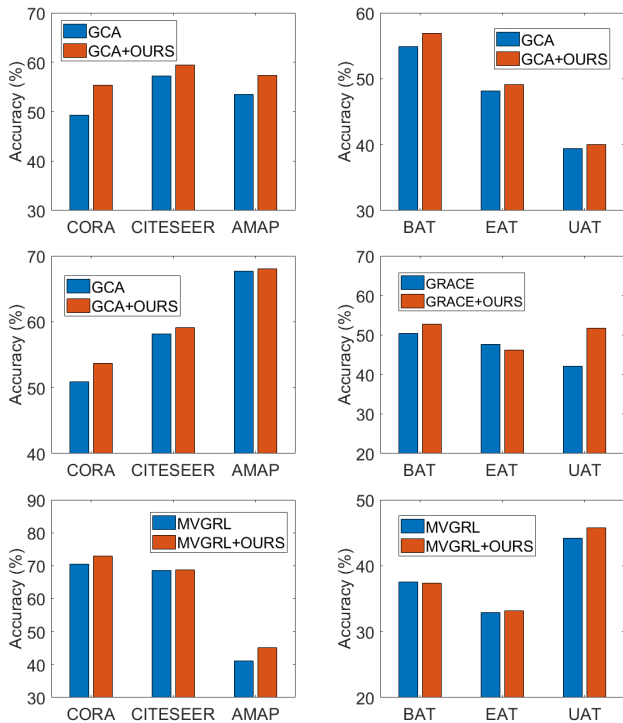


Fig. 4. Experimental results of transferring our augmentation fashion to GCA [77], GRACE, and MVGRL [14].

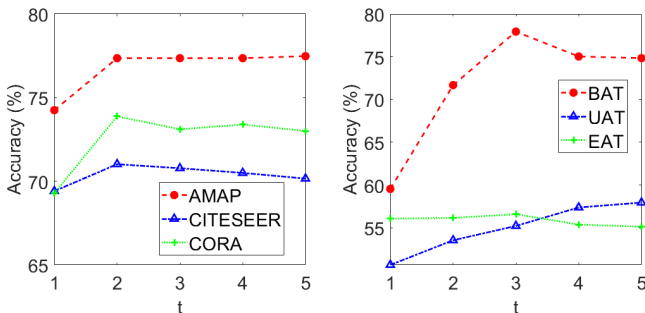


Fig. 5. Sensitivity analysis of the layer number t of graph Laplacian filters on six datasets.

average. We summarize two reasons as follows. 1) The Siamese MLP encoders in SCGC are light. 2) Our proposed method merely applies data augmentations in the latent space instead of introducing complex space-consuming operations over graphs [14], [17].

E. Ablation Studies

1) *Effectiveness of Low-Pass Denoising Operation and SCM*: In this section, we conduct ablation studies to verify the effectiveness of two components in our network, that is, low-pass denoising operation and SCM. Here, we denote the low-pass denoising operation as L for short. In Table VI, “L + SCM” denotes our proposed SCGC. Besides, “(w/o) L,” “(w/o) SCM,” and “(w/o) L and SCM” denote SCGC without L, SCM, and both of them, respectively. From these results, we have three observations as follows. 1) The low-pass denoising operation could improve the performance of the baseline by filtering out the high-frequency noise in node

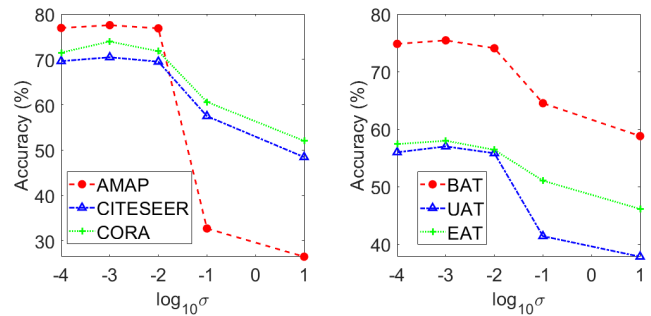


Fig. 6. Sensitivity analysis of the standard deviation σ of Gaussian noise on six datasets.

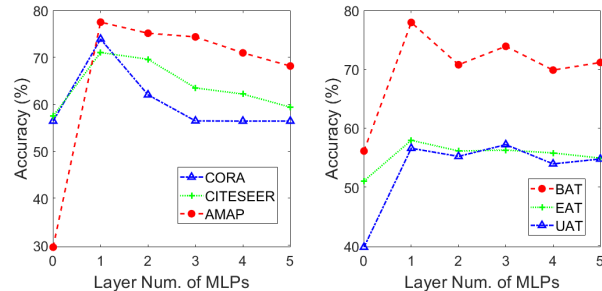


Fig. 7. Analysis of the layer number of MLPs in our SCGC.

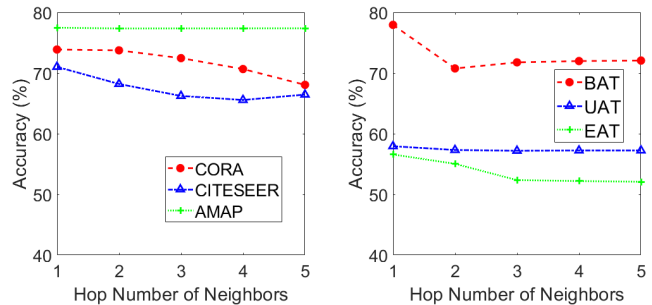


Fig. 8. Analyses of hop number of neighbors in our proposed contrastive loss on six datasets.

attributes. Concretely, the loss-pass denoising “L + SCM” outperforms “(w/o) L” average 10.77% ACC on six datasets. 2) Through our proposed SCM, the discriminative of samples is enhanced, thus achieving better performance compared to the baseline. We observe that the “L + SCM” outperforms “(w/o) SCM” average 17.78% ACC. 3) Our method consistently outperforms other variants by a large margin. Overall, the aforementioned observations have verified the effectiveness of the low-pass denoising operation and SCM in our proposed SCGC.

2) *Effectiveness of the Proposed Data Augmentation*: In our proposed SCGC, we construct augmented views of the same node by designing parameter unshared Siamese MLP encoders and adding Gaussian noise to node embeddings instead of introducing complex operations over graphs. To verify the effectiveness of this new data augmentation fashion, we first conduct expensive ablation experiments in Fig. 3. Here, we denote “U,” “G,” and “U + G,” as the strategy of setting parameter unshared MLPs, adding Gaussian noise to node embeddings, and both of them, respectively. From these results, we have two findings as follows. 1) These two

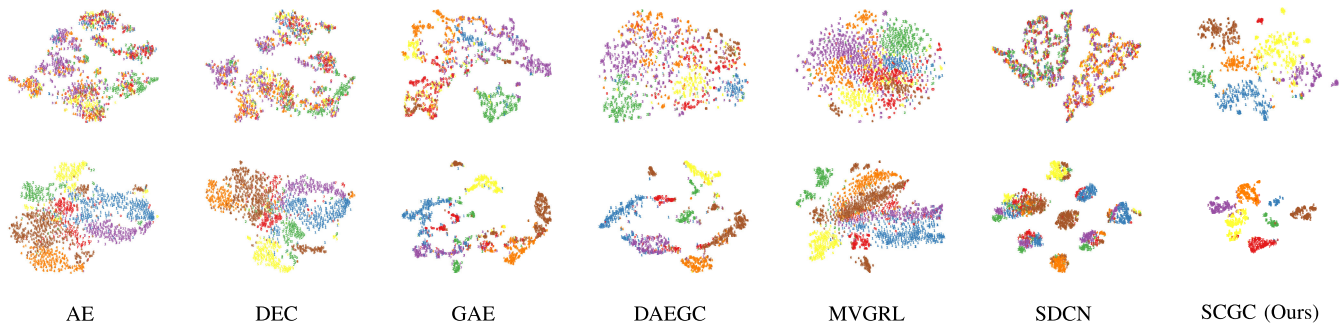


Fig. 9. 2-D visualization on two datasets. The first row and second row correspond to CORA and AMAP, respectively.

simple strategies, which both aim to construct two different views of the same node, improve the clustering performance. 2) The combination of these two strategies achieves the best performance. In summary, we have verified the effectiveness of our proposed data augmentation fashion through these experimental results.

In addition, we compare our data augmentation fashion with other classical graph data augmentations including edge dropping [17], edge adding [17], and graph diffusion [14], [16], [52]. Concretely, in Table VII, the data augmentation in SCGC is replaced by randomly dropping 10% edges (“Drop”), or randomly adding 10% edges (“Add”) or, graph diffusion (“Diffusion”) with 0.20 teleportation rate. From the results, we have two observations as follows. 1) The clustering performance is harmed by random edge dropping and adding, which may lead to semantic drift [61]. 2) Graph diffusion could achieve comparable performance on CORA and AMAP datasets while cannot compare with ours on other datasets. It indicates that graph diffusion might change the underlying semantics of graphs [63]. Overall, expensive experiments have demonstrated the effectiveness of our proposed data augmentation method.

To further verify the effectiveness and compatibility of our proposed data augmentation, we conduct experiments to transfer our augmentation fashion to other contrastive methods, including MVGRL [14], GRACE, and GCA [77]. Specifically, in these methods, we modify the original data augmentation fashion by setting unshared Siamese encoders and adding Gaussian noise to node embeddings. For the parameter settings and objective functions, we keep them consistent with the original literature. From the results in Fig. 4, we have two conclusions as follows. 1) Our proposed augmentation fashion is compatible with the existing contrastive methods. 2) Except for the performance of GRACE on EAT dataset and MVGRL on the BAT dataset, the clustering performance of these contrastive methods could be improved by our proposed augmentation.

F. Sensitivity Analysis of Hyperparameters

For the hyperparameters, we provide the search space as follows. For the learning rate, we search it in $[1e^{-5}, 1e^{-3}]$. Besides, for the layer number t , we search it in $\{2, 3, 4, 5\}$. Moreover, the standard deviation σ of the random Gaussian noise is set to 0.01. In this section, we analyze the sensitivity of the hyperparameters.

1) *Sensitivity Analysis of Hyperparameter t* : We conduct experiments to investigate the influence of the layer number t of graph Laplacian filters on our proposed SCGC. As shown in Fig. 5, we have two observations as follows. 1) SCGC could achieve promising performance when $t \in [2, 3]$. 2) Our proposed model becomes insensitive to t when $3 < t \leq 5$.

2) *Sensitivity Analysis of Hyperparameter σ* : Besides, we investigate the robustness of our proposed method SCGC to hyperparameter σ , which controls the Gaussian noise to the node embeddings \mathbf{Z}^{v_2} . From the results in Fig. 6, two conclusions are obtained as follows. 1) Our SCGC is robust to σ when $\sigma \in [0.001, 0.1]$. 2) The clustering performance decreases drastically when $\sigma > 0.1$. The reason is that too much noise would lead to node embedding semantic drift. We set σ to 0.01 in our model.

3) *Sensitivity Analysis of Layer Number of MLPs*: We analyze the layer number of MLPs in our proposed method in this section. In Fig. 7, it is worth mentioning that we directly perform the clustering algorithm on the smoothed attributes when the layer number of MLPs is equal to zero. From these results, we conclude as follows. 1) The MLP encoders are effective in improving clustering performance. 2) Our method could achieve the best performance when the layer number of MLPs equals one.

4) *Sensitivity Analysis of the Hop Number in the Neighbor-Oriented Contrastive Loss*: In this section, we conduct experiments by adopting different hop number neighbors as the positive samples in our contrastive loss. As shown in Fig. 8, the one-hop neighbors are the best choice. Besides, the clustering performance is not sensitive to the hop number when the hop number is between 3 and 5.

G. Visualization Analysis

To show the superiority of SCGC intuitively, we visualize the distribution of learned embeddings of SCGC and six compared baselines on CORA and AMAP datasets via the t -SNE algorithm [78]. As shown in Fig. 9, visible results demonstrate that SCGC better reveals the intrinsic clustering structure compared with other baselines.

V. CONCLUSION

In this article, we propose a contrastive deep graph clustering method termed SCGC to improve the existing methods from the perspectives of network architecture, data augmentation, and objective function. As to the architecture, our

network mainly includes two parts, that is, preprocessing and network backbone. Concretely, a simple low-pass denoising operation conducts neighbor information aggregation as an independent preprocessing. Through this operation, we filter out the high-frequency noise in attributes in an efficient manner, thus improving the clustering performance. Besides, only two MLPs are included as the backbone. For data augmentation, we construct different graph views by setting parameter unshared encoders and perturbing the node embeddings instead of introducing complex operations over graphs. Furthermore, we propose a novel neighbor-oriented contrastive loss to keep cross-view structural consistency, thus enhancing the discriminative capability of the network. Benefiting from the simplicity of SCGC, it is free from pretraining and saves both time and space for network training. Significantly, our algorithm outperforms the recent contrastive deep clustering competitors with at least seven times speedup on average. Extensive experimental results on seven datasets have demonstrated the effectiveness and superiority of SCGC. Besides, this study makes deep graph clustering methods more possible to be applied in many domains, like community detection [79], document mining [80], metagenomic binning [81], single-cell RNA sequencing [82], knowledge graph [83], [84], and so on.

The limitations of our proposed SCGC are summarized as follows. 1) SCGC relies on the predefined number of clusters and cannot recognize the cluster number automatically. 2) SCGC is suitable for the middle graph with about 20 000 samples and cannot handle large-scale graph data. For the limitations mentioned above, we plan to do some further potential research as follows. 1) To make our method get rid of the predefined cluster number, the density-based clustering method or reinforcement learning can be potential solutions. 2) To make the method suitable for large-scale graph data, we need to design the memory and time-efficient sampling method and clustering method. 3) To make the method to be suitable for both heterophily and homophily graphs.

REFERENCES

- [1] W. Jin et al., "Adversarial attacks and defenses on graphs," *ACM SIGKDD Explor. Newsl.*, vol. 22, no. 2, pp. 19–34, Jan. 2021.
- [2] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 66–74.
- [3] B. Wu, X. Yang, S. Pan, and X. Yuan, "Adapting membership inference attacks to GNN for graph classification: Approaches and implications," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2021, pp. 1421–1426.
- [4] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *Proc. AAAI*, 2021, vol. 35, no. 11, pp. 10049–10057.
- [5] Y. Xie, T. Chen, T. Pu, H. Wu, and L. Lin, "Adversarial graph representation adaptation for cross-domain facial expression recognition," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1255–1264.
- [6] Y. Luo, Z. Huang, Z. Wang, Z. Zhang, and M. Baktashmotlagh, "Adversarial bipartite graph learning for video domain adaptation," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 19–27.
- [7] Z. Li, Q. Xu, Y. Jiang, X. Cao, and Q. Huang, "Quaternion-based knowledge graph network for recommendation," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 880–888.
- [8] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: A deep attentional embedding approach," 2019, *arXiv:1906.06532*.
- [9] S. Pan, R. Hu, S. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020.
- [10] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2609–2615.
- [11] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proc. Web Conf.*, Apr. 2020, pp. 1400–1410.
- [12] W. Tu et al., "Deep fusion clustering network," 2020, *arXiv:2012.09600*.
- [13] G. Cui, J. Zhou, C. Yang, and Z. Liu, "Adaptive graph encoder for attributed graph embedding," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 976–985.
- [14] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.
- [15] W. Xia, S. Wang, M. Yang, Q. Gao, J. Han, and X. Gao, "Multi-view graph embedding clustering network: Joint self-supervision and block diagonal representation," *Neural Netw.*, vol. 145, pp. 1–9, Jan. 2022.
- [16] Y. Liu et al., "Deep graph clustering via dual correlation reduction," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7603–7611.
- [17] W. Xia, Q. Gao, M. Yang, and X. Gao, "Self-supervised contrastive attributed graph clustering," 2021, *arXiv:2110.08264*.
- [18] L. Liu, Z. Kang, L. Tian, W. Xu, and X. He, "Multilayer graph contrastive clustering network," 2021, *arXiv:2112.14021*.
- [19] C. Wang, S. Pan, C. P. Yu, R. Hu, G. Long, and C. Zhang, "Deep neighbor-aware embedding for node clustering in attributed graphs," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108230.
- [20] Z. Kang, Z. Liu, S. Pan, and L. Tian, "Fine-grained attributed graph clustering," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, 2022, pp. 370–378.
- [21] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.
- [22] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: Marginalized graph autoencoder for graph clustering," in *Proc. 2017 ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 889–898.
- [23] X. Zhang, H. Liu, Q. Li, and X.-M. Wu, "Attributed graph clustering via adaptive graph convolution," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4327–4333.
- [24] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, "Symmetric graph convolutional autoencoder for unsupervised graph representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6518–6527.
- [25] J. Cheng, Q. Wang, Z. Tao, D. Xie, and Q. Gao, "Multi-view attribute graph convolution networks for clustering," in *Proc. 29th Int. Conf. Int. Joint Conf. Artif. Intell.*, 2021, pp. 2973–2979.
- [26] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, "Adversarial graph embedding for ensemble clustering," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3562–3568.
- [27] H. Zhao, X. Yang, Z. Wang, E. Yang, and C. Deng, "Graph debiased contrastive learning with joint representation clustering," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 3434–3440.
- [28] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
- [29] Y. Liu et al., "Hard sample aware network for contrastive deep graph clustering," 2022, *arXiv:2212.08665*.
- [30] X. Yang, Y. Liu, S. Zhou, S. Wang, X. Liu, and E. Zhu, "Contrastive deep graph clustering with learnable augmentation," 2022, *arXiv:2212.03559*.
- [31] Y. Yang et al., "Self-supervised heterogeneous graph pre-training based on structural clustering," 2022, *arXiv:2210.10462*.
- [32] W. Xia, T. Wang, Q. Gao, M. Yang, and X. Gao, "Graph embedding contrastive multi-modal representation learning for clustering," *IEEE Trans. Image Process.*, vol. 32, pp. 1170–1183, 2023.
- [33] Y. Liu et al., "A survey of deep graph clustering: Taxonomy, challenge, and application," 2022, *arXiv:2211.12875*.
- [34] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–16.
- [35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
- [36] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–22.
- [37] H. Wang et al., "GraphGAN: Graph representation learning with generative adversarial nets," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–12.

- [38] R. D. Hjelm et al., "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent.*, 2018.
- [39] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.
- [40] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.
- [41] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," 2021, *arXiv:2103.03230*.
- [42] H. Zhong et al., "Graph contrastive clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9204–9213.
- [43] X. Yang, X. Hu, S. Zhou, X. Liu, and E. Zhu, "Interpolation-based contrastive learning for few-label semi-supervised learning," 2022, *arXiv:2202.11915*.
- [44] P. Velickovic, W. Fedus, and W. L. Hamilton, "Deep graph infomax," in *Proc. ICLR*, 2019, p. 4.
- [45] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. NIPS*, vol. 33, 2020, pp. 5812–5823.
- [46] J. Xia, L. Wu, J. Chen, B. Hu, and S. Z. Li, "SimGRACE: A simple framework for graph contrastive learning without data augmentation," 2022, *arXiv:2202.03104*.
- [47] S. Thakoor et al., "Large-scale representation learning on graphs via bootstrapping," in *Proc. Int. Conf. Learn. Represent.*, 2021, p. 109631.
- [48] P. Bielak, T. Kajdanowicz, and N. V. Chawla, "Graph barlow twins: A self-supervised representation learning framework for graphs," 2021, *arXiv:2106.02466*.
- [49] Y. Hu, H. You, Z. Wang, Z. Wang, E. Zhou, and Y. Gao, "Graph-MLP: Node classification without message passing in graph," 2021, *arXiv:2106.04051*.
- [50] K. Liang, Y. Liu, S. Zhou, X. Liu, and W. Tu, "Relational symmetry based knowledge graph contrastive learning," 2022, *arXiv:2211.10738*.
- [51] Y. Zhu, Y. Xu, Q. Liu, and S. Wu, "An empirical study of graph contrastive learning," 2021, *arXiv:2109.01116*.
- [52] Y. Liu, S. Zhou, X. Liu, W. Tu, and X. Yang, "Improved dual correlation reduction network," 2022, *arXiv:2202.12533*.
- [53] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 3861–3870.
- [54] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3861–3870.
- [55] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn.*, 1995, pp. 351–358.
- [56] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
- [57] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *J. Roy. Stat. Soc. C, Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 2010.
- [58] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
- [59] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6861–6871.
- [60] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3585–3594.
- [61] M. Sun, J. Xing, H. Wang, B. Chen, and J. Zhou, "MoCL: Data-driven molecular fingerprint via knowledge-aware contrastive learning from molecular graph," 2021, *arXiv:2106.04509*.
- [62] P. Trivedi, E. Singh Lubana, Y. Yan, Y. Yang, and D. Koutra, "Augmentations in graph contrastive learning: Current methodological flaws & towards better practices," 2021, *arXiv:2111.03220*.
- [63] N. Lee, J. Lee, and C. Park, "Augmentation-free self-supervised learning on graphs," 2021, *arXiv:2112.02472*.
- [64] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.
- [65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [66] N. Mrabah, M. Bouguessa, M. Fawzi Touati, and R. Ksantini, "Rethinking graph auto-encoder models for attributed graph clustering," 2021, *arXiv:2107.08562*.
- [67] S. Zhou et al., "Multiple kernel clustering with neighbor-kernel subspace segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1351–1362, Apr. 2020.
- [68] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6073–6082, Dec. 2018.
- [69] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, "Dynamic affinity graph construction for spectral clustering using multiple features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6323–6332, Dec. 2018.
- [70] L. Li et al., "Local sample-weighted multiple kernel clustering with consensus discriminative graph," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 15, 2022, doi: [10.1109/TNNLS.2022.3184970](https://doi.org/10.1109/TNNLS.2022.3184970).
- [71] P. Zhu, J. Li, Y. Wang, B. Xiao, S. Zhao, and Q. Hu, "Collaborative decision-reinforced self-supervision for attributed graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 18, 2022, doi: [10.1109/TNNLS.2022.3171583](https://doi.org/10.1109/TNNLS.2022.3171583).
- [72] H. Zhang, P. Li, R. Zhang, and X. Li, "Embedding graph auto-encoder for graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 25, 2022, doi: [10.1109/TNNLS.2022.3158654](https://doi.org/10.1109/TNNLS.2022.3158654).
- [73] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, and D. Tao, "Tensorized bipartite graph learning for multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5187–5202, Apr. 2023.
- [74] W. Xia, Q. Wang, Q. Gao, X. Zhang, and X. Gao, "Self-supervised graph convolutional network for multi-view clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 3182–3192, 2022.
- [75] J. Zhang et al., "Multiple kernel clustering with dual noise minimization," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 3440–3450.
- [76] H. Zhu and P. Koniusz, "Simple spectral graph convolution," in *Proc. ICLR*, 2020, pp. 2069–2080.
- [77] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *Proc. Web Conf.*, Apr. 2021, pp. 2069–2080.
- [78] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 1–15, 2008.
- [79] 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 *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 3974–3981.
- [80] B. Chiu, S. K. Sahu, D. Thomas, N. Sengupta, and M. Mahdy, "Autoencoding keyword correlation graph for document clustering," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4637–4645.
- [81] H. Xue, V. Mallawaarachchi, Y. Zhang, V. Rajan, and Y. Lin, "RepBin: Constraint-based graph representation learning for metagenomic binning," in *Proc. AAAI*, 2022, pp. 4671–4679.
- [82] Z. Yu, Y. Lu, Y. Wang, F. Tang, K.-C. Wong, and X. Li, "ZINB-based graph embedding autoencoder for single-cell RNA-seq interpretations," in *Proc. AAAI*, 2022.
- [83] K. Liang et al., "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multimodal," 2022, *arXiv:2212.05767*.
- [84] L. Meng et al., "SARF: Aliasing relation assisted self-supervised learning for few-shot relation reasoning," 2023, *arXiv:2304.10297*.



Yue Liu is currently pursuing the master's degree with the National University of Defense Technology (NUDT), Changsha, China.

He has published several peer-reviewed papers, including International Conference on Learning Representations (ICLR), AAAI Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), ACM International Conference on Multimedia (ACM MM), and ACM International Conference on Research and Development in Information Retrieval (SIGIR). His current research interests include self-supervised learning, knowledge graph, and deep graph clustering. More information can be found at <https://yueliu1999.github.io/>.



Xihong Yang is recommended for admission to the National University of Defense Technology (NUDT), Changsha, China, as a master's student with excellent grades and competition awards. He is working hard to pursue his master's degree.

His current research interests include semisupervised learning, self-supervised learning, and graph neural networks.



Sihang Zhou received the Ph.D. degree from the School of Computer, National University of Defense Technology (NUDT), Changsha, China, in 2019.

He is now a Lecturer with the College of Intelligence Science and Technology, NUDT. He has published more than 40 peer-reviewed papers, including IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON MEDICAL IMAGING (TMI), *Information Fusion*, *Medical*

Image Analysis, AAAI Conference on Artificial Intelligence (AAAI), International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), and so on. His current research interests include machine learning and medical image analysis.



Xinwang Liu (Senior Member, IEEE) received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013.

He is now a Professor with the School of Computer, NUDT. He has published more than 60 peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), International Conference on Machine Learning (ICML), Neural Information Processing Systems (NeurIPS), IEEE/CVF International Conference on Computer Vision (ICCV), IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), AAAI Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), and so on. His current research interests include kernel learning and unsupervised feature learning.

Dr. Liu serves as an Associate Editor of *Information Fusion Journal*. More information can be found at <https://xinwangliu.github.io/>.



Siwei Wang is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), Changsha, China.

His current research interests include kernel learning, unsupervised multiple-view learning, scalable clustering, and deep unsupervised learning.

Mr. Wang has published several papers and served as a PC member/Reviewer in top journals and conferences such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON CYBERNETICS (TCYB), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), International Conference on Machine Learning (ICML), IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), European Conference on Computer Vision (ECCV), IEEE/CVF International Conference on Computer Vision (ICCV), AAAI Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), and so on.



Ke Liang received the B.Sc. degree from Beihang University (BUAA), Beijing, China, in 2017, and the M.Sc. degree from Pennsylvania State University (PSU), State College, PA, USA, in 2021. He is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), Changsha, China.

His current research interests include graph representation learning, multimodal representation learning, and medical image processing.



Wenxuan Tu is currently pursuing the Ph.D. degree with the College of Computer, National University of Defense Technology (NUDT), Changsha, China.

He has published several papers in highly regarded journals and conferences such as AAAI Conference on Artificial Intelligence (AAAI), International Conference on Machine Learning (ICML), IEEE TRANSACTIONS ON MULTIMEDIA (MM), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), *Information Sciences*, and so on. His research interests include unsupervised graph learning, deep graph

clustering, and image semantic segmentation.



Liang Li received the bachelor's degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2018, and the master's degree from the National University of Defense Technology (NUDT), Changsha, China, in 2020, where he is currently pursuing the doctoral degree.

His current research interests include multiple-view learning, multiple kernel learning, scalable clustering, and incomplete clustering.