Self-consistent Contrastive Attributed Graph Clustering with Pseudo-label Prompt

Wei Xia, *Graduate Student Member, IEEE,* Qianqian Wang, Quanxue Gao, Ming Yang, and Xinbo Gao, *Senior Member, IEEE*

***Abstract*—Attributed graph clustering, which learns node representation from node attribute and topological graph for clustering, is a fundamental and challenging task for multimedia network-structured data analysis. Recently, graph contrastive learning (GCL)-based methods have obtained impressive clustering performance on this task. Nevertheless, there still remain some limitations to be solved: 1) most existing methods fail to consider the self-consistency between latent representations and cluster structures; and 2) most methods require a post-processing operation to get clustering labels. Such a two-step learning scheme results in models that cannot handle newly generated data, *i.e.*, out-of-sample (OOS) nodes. To address these issues in a unified framework, a Self-consistent Contrastive Attributed Graph Clustering (SCAGC) network with pseudo-label prompt is proposed in this article. In SCAGC, by clustering labels prompt information, a self-consistent contrastive loss, which aims to maximize the consistencies of intra-cluster representations while minimizing the consistencies of inter-cluster representations, is designed for representation learning. Meanwhile, a clustering module is built to directly output clustering labels by contrasting the representation of different clusters. Thus, for the OOS nodes, SCAGC can directly calculate their clustering labels. Extensive experimental results on seven benchmark datasets have shown that SCAGC consistently outperforms 16 competitive clustering methods. The source code could be accessed at https://github.com/xdweixia/SCAGC.**

***Index Terms*—Graph representation learning, node clustering, contrastive learning, unsupervised.**

I. Introduction

IN the multimedia community, network-structured data has penetrated into every corner of life [1]–[3]. Representative examples include shopping networks [4], social networks [5], recommendation systems [6], citation networks [7], *etc*. Real-world scenarios such as these can be modeled as attributed graphs, *i.e.*, topological graphs structure with node attributes (or features). Although graph learning [8]–[13] has brought a powerful and successful revolution in the regime of complex data processing, the methods that can simultaneously deal with non-Euclidean topological graph structure and complex node attribute are not available. To this end, graph neural networks (GNNs) [14]–[17] arises at the historic moment and have made great development in recent years. GNN aims to learn low-dimensional node representation for downstream tasks via simultaneously encoding the topological graph and node attribute. In this article, we will study the attributed graph clustering problem, which is one of the most challenging tasks in the fields of multi-media graph-structured data analysis.

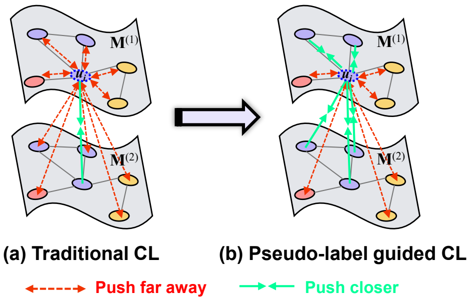


Fig. 1. Our basic idea. Taking a bi-augmentation attributed graph as a showcase, we use $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$ to denote the learned representations under graph augmentation, '○' with different colors to represent different categories. The red lines and green lines represent negative and positive node pairs. **(a)** Traditional contrastive learning, which treats the representations of a node u_i under two different augmentations as positive pairs, while regarding all remaining nodes as negative pairs. In this case, the consistency of clustering labels and representations is ignored, resulting in some true positive pairs being incorrectly mistaken for negative pairs. To conquer this problem, we proposed **(b)** contrastive learning with pseudo-label (clustering label) prompt. With such self-consistency information as supervision, our model can learn more clustering-friendly node representation and produce more correct clustering results.

Attributed graph clustering, *i.e.*, node clustering, aims to divide massive nodes into several disjoint clusters without intense manual guidance [18]–[20]. To date, numerous attributed graph clustering methods have been proposed [21]–[27], among which, most of them are based on graph auto-encoder (GAE) and variational GAE (VGAE) [28]. For example, to learn a robust node representation, the variants of GAE and VGAE are proposed by Pan *et al.* [29], [30], namely adversarially regularized graph auto-encoder (ARGA) and adversarially regularized variational graph auto-encoder (ARVGA). To build a clustering-directed network, inspired by deep embedding clustering (DEC) [31], Wang *et al.* [32] minimized the mismatch between clustering distribution and

The work of W. Xia, Q. Wang and Q. Gao was supported by the National Natural Science Foundation of China under Grant 62176203, the Natural Science Basic Research Plan in Shaanxi Province under Grant 2020JZ-19, the Natural Science Foundation of Shandong Province under Grant ZR202102180986, the Fundamental Research Funds for the Central Universities, and the Innovation Fund of Xidian University. The work of X. Gao was supported by the National Natural Science Foundation of China under Grants No. 62036007, the Special Project on Technological Innovation and Application Development under Grant No.cstc2020jscx-dxwtB0032, and Chongqing Excellent Scientist Project under Grant No. cstc2021ycjh-bgzxm0339. *(Corresponding author: Quanxue Gao).*

Wei Xia, Qianqian Wang, and Quanxue Gao are with School of Telecommunication Engineering, Xidian University, Xi'an 710071, China (e-mail: xd.weixia@gmail.com, qqwang@xidian.edu.cn, qxgao@xidian.edu.cn).

Ming Yang is with the mathematics department of the University of Evansville, Evansville, IN 47722, USA (e-mail: yangmingmath@gmail.com).

Xinbo Gao is with the Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: gaoxb@cqupt.edu.cn).

Manuscript received XXX XX, XXXX; revised XXX XX, XXXX.

target distribution to improve the quality of node representation, and proposed deep attentional embedded graph clustering (DAEGC) approach. Similarly, Bo *et al.* [33] presented a structural deep clustering network (SDCN) to embed the topological structure into deep clustering. SDCN used the traditional auto-encoder to get new node features via encoding node attribute and then used GNN to simultaneously encode topological structure and new node feature to learn the final node representation for clustering. Tu *et al.* [34] proposed a deep fusion clustering network (DFCN), which used a dynamic cross-modality fusion mechanism for obtaining consensus node representation, thereby generating a more robust target distribution for network optimization. Although the aforementioned methods have made encouraging progress, how to mine the highly heterogeneous information embedded in the attribute graph remains to be explored.

Recently, due to its powerful unsupervised representation learning ability, contrastive learning has made vast inroads into the computer vision community [35], [36]. Motivated by this, several recent studies [2], [37]–[43] show promising results on unsupervised graph representation learning (GRL) using approaches related to contrastive loss. These kinds of methods are called graph contrast representation learning methods (GCRL for short in this paper). For example, Velickovic *et al.* [37] proposed deep graph information maximization (DGI) to learn node representation by contrasting the local node-level representation and the global graph-level representation. Similarly, Sun *et al.* [38] proposed to learn graph-level representation by maximizing the mutual information between the graph-level representation and representations of substructures. To well preserve and extract the abundant information hidden in attributed graph data, Peng *et al.* [44] proposed a graphical mutual information maximization approach (GMI) to simultaneously constrains feature and topology-aware mutual information to learn latent representation. Moreover, based on the contrastive loss in SimCLR [35], You *et al.* [40] proposed a new graph contrastive learning network with kinds of graph augmentation approaches (GraphCL) for facilitating node representation learning. More recently, Zhu *et al.* [41] propose graph contrastive learning with adaptive augmentation (GCA). GCA first used adaptive graph augmentation schemes to construct different graph views, then extracted node representation via maximizing the agreement of node representation between graph views with traditional contrastive learning loss.

Despite some achieved commendable results, most existing graph contrastive representation learning-based clustering methods still have the following challenging issues:

1) They failed to consider the self-consistency[1] between latent representations and cluster structures, thus leading to limited presentation performance. Due to ignoring the self-consistency, they made some mistakes about some positive node pairs for negative pairs, as shown in **Figure 1 (a)**, which affect the quality of the learned representation. Benefiting from imprecise clustering labels, such a problem will be effectively alleviated. Thus, more dedicated efforts are pressingly needed.
2) Their objective functions are task-agnostic and need post-processing, *e.g.*, *K*-Means, to get clustering labels, resulting in suboptimal clustering results.
3) They cannot handle out-of-samples, which seriously limits their application in practical engineering. To obtain the clustering label of out-of-sample (OOS) nodes [9], *i.e.*, newly generated nodes, they have to take the existing data and OOS nodes as a whole to retrain the model again. Such a manner is time-consuming and consumes computing resources.

In this paper, we observe that the solutions to the aforementioned issues could be unified into an end-to-end framework, as shown in Figure 2. In brief, the proposed *Self-consistent Contrastive Attributed Graph Clustering with pseudo-label prompt* (SCAGC) aims to learn clustering-friendly node representation and output the clustering results more directly by resorting to a novel self-supervised contrastive learning paradigm with pseudo-label prompt. Specifically, on one hand, with the prompt of pseudo-label, SCAGC treats the representations of intra-cluster nodes as positive pairs and the representations of inter-cluster nodes as negative pairs for node representation learning. On the other hand, to eliminate the influence of post-processing and enable the model to handle OOS nodes, a contrastive clustering module is introduced by considering the similarities of different cluster representations. To summarize, the contributions and novelties of this work are two-fold:

1) From the view of unsupervised learning, in particular, attributed graph clustering, this could be one of the few studies to benefit from clustering-friendly representation with pseudo-label prompt based contrastive learning. Notably, traditional contrastive loss fails to consider the self-consistency between latent representations and cluster structures, which limits performance. Whereas our proposed contrastive loss benefits from pseudo-label prompt information and effectively alleviates the effect of false negative samples. Meanwhile, it is plug-and-play for any deep clustering model. Hence, this work might provide some novel insights into the community of unsupervised/self-supervised learning.
2) From the view of clustering, SCAGC could be the first contrastive attributed graph clustering work without post-processing, and it can directly tackle out-of-sample nodes, which accelerates the implementation of SCAGC in practical engineering.

II. Related Work

In this section, we briefly review the main topic related to this work, *i.e.*, self-supervised learning (including contrastive learning) in clustering.

In recent years, self-supervised learning, as one of the most popular unsupervised learning paradigms, has been widely

[1] Self-consistency means that the consistency of latent representation of different nodes should be consistent with the clustering structures. In other words, for a clustering system, we can know which nodes belong to the same cluster based on the pseudo-label. This clustering structure information should in turn constrain the latent representation so that the representation of nodes in the same cluster has high consistency and the representation of nodes in different clusters has low consistency. In this way, clustering and representation learning are self-consistent, intending to achieve better clustering performance.

studied and applied in various machine learning tasks, especially in representation learning [45], [46]. Based on this, numerous works [31], [47]–[51] have been devoted to combining clustering with self-supervised learning and some of them shown impressive results. For example, Wu *et al*. [48] proposed deep comprehensive correlation mining (DCCM) method for image clustering. The core of DCCM is that it utilizes the calculated pseudo-graph and pseudo-label with high confidence to guide the representation learning module to explore the correlation between samples. To learn more discriminative image representation, Li *et al*. [52] proposed contrastive clustering (CC), which treated label features as representation. However, such a manner may cause the learned representation to lose some sample information. To solve this problem, Niu *et al*. [47] proposed a semantic pseudo-label-based image clustering (SPICE) network, which consists of a representation learning module (for constraining the similarities of different samples) and a clustering head (for constraining the discrepancies of different clusters).

Though both DCCM and SPICE utilized the clustering labels to guild representation learning, SCAGC is significantly different from them in the following aspects. First, they need to manually set some thresholds (or the number of neighbors) to select reliable pseudo-labels for supervising representation learning. However, setting some appropriate thresholds for different databases is challenging, which can limit their application in practice. In contrast, SCAGC directly constrains the self-consistency between pseudo-labels and representations, which is plug-and-play in practice. Second, for instance-level contrastive loss, SPICE regarded the representations of a sample under two different augmentations as a positive pair and left other pairs to be negative. Thus, SPICE failed to consider self-consistency. Third, different from them, our method attempts to deal with attributed graph data rather than image data.

More recently, Zhang *et al*. [50] combined Clustering loss based on the distribution [31] with contrastive loss [35] to learn clustering-favorable representation, and presented supporting clustering with contrastive learning (SCCL). However, similar to cc and SPICE, SCCL only constrained an example and its augmentation should have a similar latent representation, resulting in suboptimal representation. To tackle this problem, Zhong *et al*. [49] proposed a contrastive learning method at the graph level (GCC). Based on the assumption that nodes within a neighbor should belong to the same cluster, GCC first constructed a k-NN similarity graph from the latent representation of the original image and then treated the nodes within the neighborhood as positive samples, and the nodes outside the neighborhood as negative samples to perform contrastive representation learning. Although GCC achieves preliminary image clustering performance, the selection of k has a great impact on clustering according to their paper.

While various interesting contrastive learning-based clustering methods are orthogonal and well complementary to our work, we herein endeavor to investigate a novel self-consistency contrastive learning strategy with clustering labels prompt. To the best of our knowledge, this could be the first of several attributed graph contrastive representation learning methods with self-consistency taking consideration. In addition, it should also be pointed out that SCAGC is also different from GCA [41] and its extension [43] by the following aspects.

1) GCA [41] aims to learn node representations that are insensitive to perturbation on unimportant nodes and edges via their proposed adaptive attributed augmentation scheme. Thus, GCA 1) is task-agnostic; 2) require post-processing operation, *i.e.*, K-Means, to obtain the clustering results; 3) cannot solve out-of-sample problem. **In contrast**, for SCAGC, the node representation learning and clustering interact with each other and jointly evolve in an end-to-end framework. Moreover, the proposed SCAGC can directly predict the clustering labels of OOS.
2) GDCL [43] randomly select negative samples from the clusters which are different from the positive node's cluster to improve contrastive loss for learning robust node representation. However, when obtaining clustering results, GDCL requires K-Means to initialize the clustering centers, leading to unstable performances. Moreover, GDCL only uses clustering labels to construct negative samples, and thus fails to make full use of clustering labels, resulting in suboptimal clustering results. **In contrast**, SCAGC makes good use of the clustering labels to learn node representation, *i.e.*, maximizing the consistencies of intra-cluster nodes and minimizing the consistencies of inter-cluster nodes. Meanwhile, the proposed SCAGC can directly output the clustering labels without any post-processing.

To sum up, the proposed SCAGC is significantly different from the aforementioned related works in terms of both motivation and objective.

III. Methodology

In this section, we first formalize the node clustering task on attributed graphs. Then, the details of all components of SCAGC will be introduced. Finally, the differences between SCAGC and existing works are summarized from the view of technical point.

A. Problem Formalization

Given an arbitrary attributed graph $\mathcal{G} = (\mathbf{U}, \mathbf{E}, \mathbf{X})$, where $\mathbf{U} = \{u_1, u_2, \cdots, u_N\}$ is the vertex set, $\mathbf{E}$ is the edge set, $\mathbf{X} \in \mathbb{R}^{N\times d}$ is the node attribute matrix, N is the number of nodes, and d is the dimension of node attribute matrix. $\mathbf{G} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of $\mathcal{G}$, and $\mathrm{G}_{ij} = 1$ iff $(u_i, u_j) \in \mathbf{E}$, *i.e.*, there is an edge from node u_i to u_j. In this article, we study one of the most representative downstream tasks of GNNs, *i.e.*, node clustering. The target of node clustering is to divide the given N unlabeled nodes into K disjoint clusters $\{\mathbf{C}_{1,} \cdots, \mathbf{C}_k, \cdots, \mathbf{C}_K\}$, such that the node in the same cluster $\mathbf{C}_k$ has high similarity to each other [26], [53].

Out-of-sample nodes refers to newly generated (arrived) data $(\mathbf{X}_{\mathrm{new}}, \mathbf{G}_{\mathrm{new}})$ in data stream.

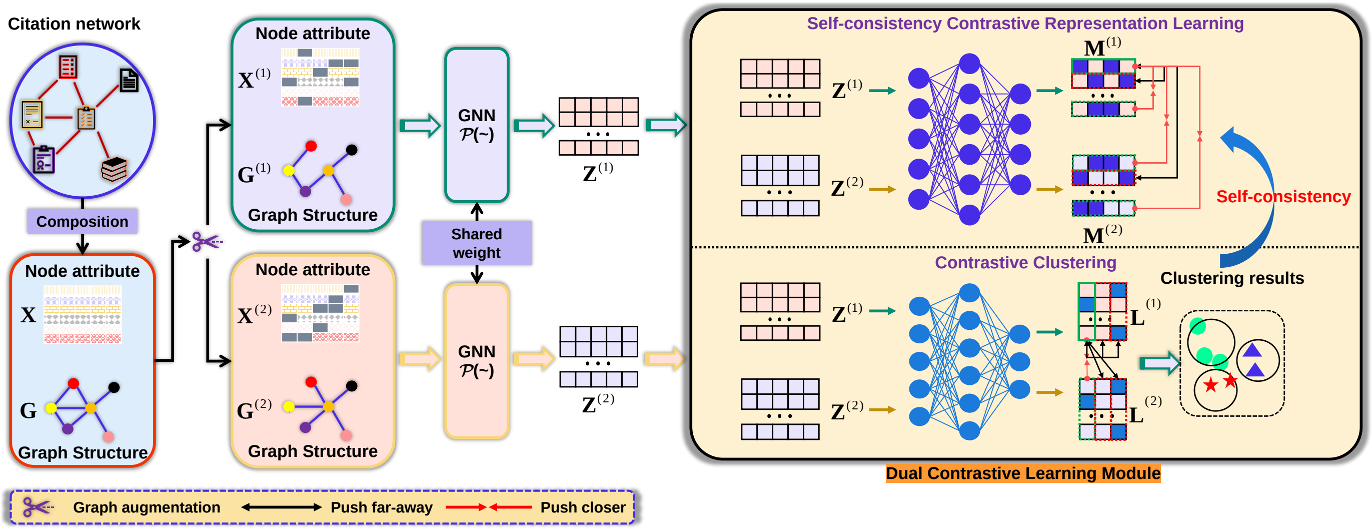


Fig. 2. The pipeline of our work. SCAGC first leverages graph augmentation methods to generate abundant attributed graph views, then, each augmented attributed graph has two compact representations: a clustering assignment probability produced by the clustering module and a low-dimension node representation produced by the graph representation learning module. The two representations interact with each other and jointly evolve in an end-to-end framework. Specifically, the clustering module is trained via contrastive clustering loss to maximize the agreement between representations of the same cluster. The graph representation learning module is trained by using the proposed self-consistent contrastive loss with pseudo labels prompt, where nodes within the same cluster are trained to have consistent representations.

B. Overall Network Architecture

As shown in Figure 2, the network architecture of the proposed SCAGC consists of the following joint optimization components: shared graph convolutional encoder, contrastive clustering module, and self-consistency graph contrastive representation learning module.

- **Shared Graph Convolutional Encoder**: It aims to simultaneously map the augmented node attribute and topological graph structure to a new low-dimensional space for the downstream node clustering task.
- **Self-consistency GCRL Module**: To learn more about discriminative graph representation and utilize the useful information embedded in clustering labels, this module is designed to maximize the consistencies of intra-cluster nodes, *i.e.*, positive pairs, while minimizing the similarities of inter-cluster nodes, *i.e.*, negative pairs.
- **Contrastive Clustering Module**: To directly get clustering labels, this module builds a clustering network by contrasting the representation of different clusters.

C. Shared Graph Convolutional Encoder

Graph contrastive representation has attracted much attention, due to its ability to utilize graph augmentation schemes to generate positive and negative node pairs for representation learning [40], [41]. Specifically, given an arbitrary attributed graph $\mathcal{G}$ with node attribute $\mathbf{X}$ and topological graph $\mathbf{G}$, two stochastic graph augmentation schemes $\mathcal{A}^{(1)} \sim \boldsymbol{\mathcal{A}}$ and $\mathcal{A}^{(2)} \sim \boldsymbol{\mathcal{A}}$ are leveraged to construct two correlated attributed graph views $\{\mathbf{X}^{(1)}, \mathbf{G}^{(1)}\}$ and $\{\mathbf{X}^{(2)}, \mathbf{G}^{(2)}\}$, where $\mathbf{X}^{(v)} = \mathcal{A}^{(v)}(\mathbf{X})$, and $\mathbf{G}^{(v)} = \mathcal{A}^{(v)}(\mathbf{G})$, $v = \{1, 2\}$ is the v-th graph augmentation, $\boldsymbol{\mathcal{A}}$ denotes the set of all kinds of graph augmentation methods, including attribute masking, edge perturbation. To be specific, attribute masking randomly adds noise to node attributes, and edge perturbation randomly adds or drops edges in the topological graphs. The underlying prior of these two graph augmentation schemes is to keep the intrinsic topological structure and node attribute of the attributed graph unchanged. Based on this prior, the learned node representation will be robust to perturbation on insignificant attributes and edges. In this article, we implement the graph augmentations following the setting in GCA [41].

After obtaining two augmented attributed graph views $\{\mathbf{X}^{(1)}, \mathbf{G}^{(1)}\}$ and $\{\mathbf{X}^{(2)}, \mathbf{G}^{(2)}\}$, we utilize a shared two-layer graph convolutional network $\mathcal{P}(\sim)$ to simultaneously encode node attributes and topological graphs of augmented attributed graph views. Thus, we have

$$\overline{\mathbf{Z}}^{(v)} = \mathcal{P}(\mathbf{X}^{(v)},\mathbf{G}^{(v)}|\mathbf{\Omega}^1) = \sigma(\widetilde{\mathbf{D}}_{(v)}^{-\frac{1}{2}}\widetilde{\mathbf{G}}^{(v)}\widetilde{\mathbf{D}}_{(v)}^{-\frac{1}{2}}\mathbf{X}^{(v)}\mathbf{\Omega}^1), \qquad (1)$$

$$\mathbf{Z}^{(v)} = \mathcal{P}(\overline{\mathbf{Z}}^{(v)}, \mathbf{G}^{(v)}|\mathbf{\Omega}^2), \qquad (2)$$

where $\overline{\mathbf{Z}}^{(v)}$ is the 1-st layer's output of shared GNN; $\mathbf{Z}^{(v)} \in \mathbb{R}^{N\times d_1}$ is the node representation under the v-th graph augmentation; $\mathbf{\Omega} = \{\mathbf{\Omega}^1, \mathbf{\Omega}^2\}$ denotes the trainable parameter of graph convolutional encoder; $\widetilde{\mathbf{G}}^{(v)} = \mathbf{G}^{(v)} + \mathbf{I}$; $\widetilde{\mathbf{D}}^{(v)}(i, i) = \sum_j \widetilde{\mathbf{G}}_{ij}^{(v)}$; $\mathbf{I}$ is an identity matrix; $\sigma(\cdot) = \max(0,)$ represents the nonlinear ReLU activation function.

So far, we have obtained the node representations $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ of two augmented attributed graph views.

D. Self-consistency GCRL Module

Contrastive learning-based graph representation learning has been an effective paradigm for maximizing the similarities of positive pairs while minimizing the similarities of negative pairs to learn discriminative graph representation. For a given attributed graph with N nodes, there are $2N$ augmented nodes. Traditional CL regards the representations of a node under two different augmentations as a positive pair, and leaves other

$2N$-2 pairs to be negative (see Figure 1 (a)). While having promising performance, this assumption runs counter to the criterion of clustering. In clustering, *we hope that the nodes in the same cluster* $\mathbf{C}_k$ *have high similarity to each other while the nodes in different clusters have low similarity to each other*. However, existing methods fail to well consider this criterion, *i.e.*, *neglecting the existence of false-negative pairs*.

In this article, by leveraging pseudo clustering labels $\overrightarrow{\mathbf{L}}$, we can easily get the samples' index of different clusters. As shown in Figure 1 (b), we aim to maximize the consistencies of intra-cluster nodes, *i.e.*, positive pairs, while minimizing the similarities of inter-cluster nodes, *i.e.*, negative pairs. To this end, we first map the node representations $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ to obtain enhanced node representations $\mathbf{M}^{(1)}$ and $\mathbf{M}^{(2)}$ via a shared two-layer fully connected network with parameter ϕ, which also help to form and preserve more information in $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$, where $\mathbf{M}^{(v)} \in \mathbb{R}^{N\times d_2}$, d_2 is the dimension of new node representation. After that, for the i-th node, we propose a new self-supervised contrastive loss function, which is defined as

$$\mathcal{L}_i = -\frac{1}{|\Delta_i|}\sum_{t\in\Delta_i}\sum_{\alpha,\beta=1}^{2}\log\frac{\mathrm{e}^{(\circledS(\mathbf{m}_i^{(\alpha)},\ \mathbf{m}_t^{(\beta)})/\tau_2)}}{\sum\limits_{\alpha',\beta'=1}^{2}\sum\limits_{q\in\nabla_i}\mathrm{e}^{(\circledS(\mathbf{m}_i^{(\alpha')},\ \mathbf{m}_q^{(\beta')})/\tau_2)}}, \quad (3)$$

where $\mathbf{m}_i^{(v)}$ represents the i-th row of node representation $\mathbf{M}^{(v)}$; Δ_i represents the set of nodes that belong to the same cluster as the i-th node, and $|\Delta_i|$ is its cardinality, which can be obtained from the pseudo clustering assignment matrix $\overrightarrow{\mathbf{L}}$; ∇_i is the set of indices of all nodes except the i-th node; τ_2 is the temperature parameter. Given two arbitrary representations $\mathbf{a}$ and $\mathbf{b}$, $\circledS(\mathbf{a},\ \mathbf{b})$ is used to measure the cosine similarity between them, which it is defined as

$$\circledS(\mathbf{a},\ \mathbf{b}) = \frac{\mathbf{a}\cdot\mathbf{b}^{\mathrm{T}}}{\|\mathbf{a}\|\cdot\|\mathbf{b}\|}. \quad (4)$$

Then, taking all nodes into account, the self-supervised contrastive loss is

$$\mathcal{L}_{\mathrm{SGC}} = \min_{\mathbf{\Omega},\ \phi}\sum_{i=1}^{N}\mathcal{L}_i. \quad (5)$$

E. Contrastive Clustering Module

How obtain the clustering labels is crucial for the downstream clustering tasks. Most existing methods directly implement classical clustering algorithms, *e.g.*, K-Means or spectral clustering, on the learned node representation to get clustering results. However, such a strategy executes the node representation and clustering in two separate steps, which limits clustering performance. To this end, we build a clustering network to directly obtain the clustering labels. Specifically, as shown in Figure 2, the clustering network is applied to transform the pattern structures of $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(1)}$ into probability distribution of clustering labels $\widehat{\mathbf{L}}^{(1)}$ and $\widehat{\mathbf{L}}^{(2)}$.

To share the parameters across augmentations, we execute $\widehat{\mathbf{L}}^{(1)}$ and $\widehat{\mathbf{L}}^{(2)}$ through a shared two-layer fully connected network with parameter ψ. Under this setting, we can ensure $\widehat{\mathbf{L}}^{(1)}$ and $\widehat{\mathbf{L}}^{(2)}$ own the same coding scheme. Thus, $\widehat{\mathbf{L}}^{(1)} \in \mathbb{R}^{N\times K}$ is the output of clustering network under the 1-st augmented attributed graph view, and $\widehat{\mathbf{L}}^{(2)}$ for the 2-nd augmented attributed graph view, where K is the number of clusters, $\hat{\boldsymbol{\ell}}_{i,\ k}^{(1)}$ represents the probability that assigning the i-th node to the k-the cluster $\mathbf{C}_k$.

For the obtained assignment matrices $\widehat{\mathbf{L}}^{(1)}$ and $\widehat{\mathbf{L}}^{(2)}$, in the column direction, each column $\hat{\boldsymbol{\ell}}_{,\ k}^{(1)}$ of $\widehat{\mathbf{L}}^{(1)}$ is the representation of the k-th cluster. Thus, *we should push closer the cluster representation of the same class, and also push far away the cluster representation of different class*. That is to say, for the k-th cluster in each augmented attributed graph view, there is only one positive pair $(\hat{\boldsymbol{\ell}}_{,\ k}^{(v)},\ \hat{\boldsymbol{\ell}}_{,\ k}^{(v)})$, and $2K$-2 negative pairs. To this end, motivated by the great success of contrastive learning [35], we leverage the contrastive loss function to implement this constraint. Thus, for the k-th cluster in the 1-st augmentation, we have

$$\mathcal{L}(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)},\hat{\boldsymbol{\ell}}_{,\ k}^{(2)})\text{=-}\log\frac{\mathrm{e}^{(\circledS(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)},\hat{\boldsymbol{\ell}}_{,\ k}^{(2)})/\tau_1)}}{\underbrace{\sum_{j=1,\ j\neq k}^{K}\mathrm{e}^{(\circledS(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)},\hat{\boldsymbol{\ell}}_{,\ j}^{(1)})/\tau_1)}}_{\text{inter-view pairs}}+\underbrace{\sum_{j=1}^{K}\mathrm{e}^{(\circledS(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)},\hat{\boldsymbol{\ell}}_{,\ j}^{(2)})/\tau_1)}}_{\text{intra-view pairs}}}, \quad (6)$$

where τ_1 is parameter to control the softness. Then, taking all positive pairs into account, the contrastive clustering loss $\mathcal{L}_{\mathrm{CC}}$ is defined as

$$\mathcal{L}_{\mathrm{CC}} = \min_{\mathbf{\Omega},\ \psi}\frac{1}{2K}\sum_{k=1}^{K}\left[\mathcal{L}(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)},\hat{\boldsymbol{\ell}}_{,\ k}^{(2)})+\mathcal{L}(\hat{\boldsymbol{\ell}}_{,\ k}^{(2)},\hat{\boldsymbol{\ell}}_{,\ k}^{(1)})\right], \quad (7)$$

Moreover, to avoid trivial solutions, *i.e.*, to make sure that all nodes are evenly assigned into all clusters, we herein introduce a clustering regularizer $\mathcal{R}$, similar to [52], [54], which is defined as

$$\mathcal{R} = \min_{\mathbf{\Omega},\ \psi}\text{-}\sum_{k=1}^{K}[\rho(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)})\log(\hat{\boldsymbol{\ell}}_{,\ k}^{(1)}) + \rho(\hat{\boldsymbol{\ell}}_{,\ k}^{(2)})\log(\hat{\boldsymbol{\ell}}_{,\ k}^{(2)}))], \quad (8)$$

where $\rho(\hat{\boldsymbol{\ell}}_{,\ k}^{(v)}) = \sum_{i=1}^{N}\frac{\hat{\boldsymbol{\ell}}_{i,\ k}^{(v)}}{\|\widehat{\mathbf{L}}^{(v)}\|_1}$.

In the proposed SCAGC training process, when we take the un-augmented attributed graph $(\mathbf{X},\ \mathbf{G})$ as the input of SCAGC, then we can get the clustering assignment matrix $\overrightarrow{\mathbf{L}}$ by discretizing the continuous output probability $\widehat{\mathbf{L}}$.

Remark 1: **Solving out-of-sample nodes**. Recall that most existing GCN based attributed graph clustering methods, *e.g.*, GAE [28], VGAE [28], ARGA [30], ARVGA [30], SDCN [33], DFCN [34], DCRN [55], ITR [56], GraphCL [40] and GCA [41], feed the learned node representation into K-Means to obtain clustering labels. By doing so, node clustering is separated from representation learning, *i.e.*, there is no interaction in the learning process.

For newly generated data $(\mathbf{X}_{\mathrm{new}},\ \mathbf{G}_{\mathrm{new}})$, only by training the whole attributed graph, *i.e.*, $\{(\mathbf{X};\ \mathbf{X}_{\mathrm{new}}),\ (\mathbf{G};\ \mathbf{G}_{\mathrm{new}})\}$, can they obtain the clustering label, which limits their applications in practice. **In contrast**, the proposed SCAGC can handle the out-of-sample problem. Specifically, let the parameterized model trained on the given data be denoted as $\mathcal{F}(\cdot\,|\overline{\mathbf{\Omega}},\ \overline{\phi},\ \overline{\psi})$, where $\overline{\mathbf{\Omega}},\ \overline{\phi},\ \overline{\psi}$ is the trained parameters of SCAGC.

Algorithm 1: Procedure for training SCAGC

```
Input: Attributed graph with node attribute matrix $\mathbf{X}$
       and adjacency matrix $\mathbf{G}$, cluster number $K$,
       hyper-parameters $\tau_1$, $\tau_2$, $\gamma$, learning rate and
       maximum number of iterations $\mathrm{T}_{\max}$.
Output: Clustering label $\overrightarrow{\mathbf{L}}$.
Initialization: initialize the parameters $\boldsymbol{\Omega}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ of
  each component, the clustering assignment matrix $\overrightarrow{\mathbf{L}}$
  by inputting raw attributed graph $(\mathbf{X}, \mathbf{G})$;
// Training SCAGC
for $\mathrm{T} = 1 : \mathrm{T}_{max}$ do
    Sample two stochastic graph augmentation
      schemes $\boldsymbol{A}^{(1)} \sim \boldsymbol{\mathcal{A}}$ and $\boldsymbol{A}^{(2)} \sim \boldsymbol{\mathcal{A}}$;
    Construct the augmented attributed graph views:
      where $\mathbf{X}^{(1)} = \boldsymbol{A}^{(1)}(\mathbf{X})$, $\mathbf{G}^{(1)} = \boldsymbol{A}^{(1)}(\mathbf{G})$,
      $\mathbf{X}^{(2)} = \boldsymbol{A}^{(2)}(\mathbf{X})$, and $\mathbf{G}^{(2)} = \boldsymbol{A}^{(2)}(\mathbf{G})$;
    Obtain variables $\mathbf{Z}^{(1)}$, $\mathbf{Z}^{(2)}$, $\mathbf{M}^{(1)}$, $\mathbf{M}^{(2)}$, $\widehat{\mathbf{L}}^{(1)}$ and $\widehat{\mathbf{L}}^{(2)}$
      by forward propagation;
    Calculate the overall objective with Eq. (9) and
      pseudo clustering label $\overrightarrow{\mathbf{L}}$;
    Update network parameters $\boldsymbol{\Omega}$, $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ via stochastic
      gradient ascent to minimize Eq. (9);
    // Update pseudo clustering label
    if $T \% 5 == 0$ then
        Update the clustering assignment matrix $\overrightarrow{\mathbf{L}}$ by
          mapping raw attributed graph $(\mathbf{X}, \mathbf{G})$;
    end
end
// Obtain clustering results
Obtain the clustering assignment matrix $\overrightarrow{\mathbf{L}}$ by
  mapping raw attributed graph $(\mathbf{X}, \mathbf{G})$;
return: Clustering label matrix $\overrightarrow{\mathbf{L}}$.
```

Next, we send $(\mathbf{X}_{\text{new}}, \mathbf{G}_{\text{new}})$ into $\mathcal{F}(\cdot\,|\overline{\boldsymbol{\Omega}}, \overline{\boldsymbol{\phi}}, \overline{\boldsymbol{\psi}})$. By simply forward propagation, we can calculate the label from the contrastive clustering head directly. The whole process is simple but efficient, requiring no retraining. ■

F. Optimization

Finally, we integrate the aforementioned three sub-modules into an end-to-end optimization framework, the overall objective function of SCAGC can be formulated as

$$\mathcal{L}_{\text{Total}} = \min_{\boldsymbol{\Omega},\,\boldsymbol{\phi},\,\boldsymbol{\psi}} \mathcal{L}_{\text{SGC}} + \mathcal{L}_{\text{CC}} + \gamma \mathcal{R}, \tag{9}$$

where γ is a trade-off parameter. By optimizing Eq. (9), some nodes with correct labels will propagate useful information for graph representation learning, where the latter is used in turn to conduct the subsequent clustering. By this strategy, node clustering and graph representation learning are seamlessly connected, with the aim to achieve better clustering results. We employ Adam optimizer [62] with learning rate η to optimize the proposed SCAGC, *i.e.*, Eq. (9). Algorithm 1 presents the pseudo-code for optimizing the proposed SCAGC.

IV. Experiments

A. Experiment Setup

1) Benchmark Datasets: In this article, we use six real-world attributed graph datasets to evaluate the effectiveness of SCAGC. These datasets cover several domains, *e.g.*, air-traffic network, citation network, academic network, and shopping network. Table I briefly summarizes the statistics of these datasets, the detailed description is as follows:

- UAT[2] dataset is composed of 1, 190 nodes, where each node is an airport, the graph describes the commercial flight relationship between airports. Similar to [57], 239-dimension (dim) one-hot feature of node degrees are used as a node attribute. It has 4 levels as categories, and each reflects passenger traffic at the airport.
- ACM[3] dataset consists of 3, 025 nodes with 3 kinds of research areas of the article. The node attribute consists of an 870-dim one-hot feature of keywords of the article. The graph describes the co-subject relationship of different articles.
- DBLP[4] dataset contains 4, 075 nodes. Each node is an author, and a 334-dim one-hot feature of the keywords of the author is used as a node attribute. The graph characterizes the co-conference relationship, where there exists a link if authors published at the same conference.
- Amazon-Photo[5] dataset and Amazon-Computers[6] dataset are shopping networks, where each node represents a commodity. The graph characterizes the co-purchasing relationship between commodities. The node attribute consists of the bag-of-words comments of commodity, which is encoded as a one-hot feature.
- Cora-Full[2] dataset is composed of 19, 737 nodes with 7 kinds of topic of article. Each node is an article, and 8, 710-dim one-hot feature of keywords of article. The graph describes the article citation relationship, where there exists an edge if one article cited the other.
- Pubmed dataset[2] is a citation network, which is composed of 500-dim TF/IDF weighted word vectors of the paper and their citation relationship. It consists of 19, 717 nodes and is divided into three classes.

2) Baseline Methods: We compare the clustering performance of the proposed SCAGC with 16 state-of-the-art node clustering methods, including the following three categories:

- **Classical**: *K*-means;
- **GCN-based**: GAE [28], VGAE [28], ARGA [30], ARVGA [30], DAEGC [32], SDCN [33], DFCN [34], DCRN [55], CDRS [63], and ITR [56];
- **Contrastive learning-based**: GraphCL [40], GCA [41], GMI [44], SCCL [50], and GCC [6].

Notably, for *K*-means, it takes raw node attribute as input. As other baselines, they take raw node attributes and topo-

[2] https://github.com/yueliu1999/Awesome-Deep-Graph-Clustering/tree/main/dataset

[3] http://dl.acm.org

[4] https://dblp.uni-trier.de/

[5] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_photo.npz

[6] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_computers.npz

TABLE I
STATISTICS OF THE REAL-WORLD EVALUATION DATASETS.

Dataset	# Nodes	# Attribute dimension	# Edges	# Classes	Type	Scale
UAT [57]	1, 190	239	13, 599	4	Air-traffic	Small
ACM [58]	3, 025	1, 870	29, 281	3	Paper relationship	Small
DBLP [59]	4, 057	334	5, 000, 495	4	Author relationship	Small
Amazon-Photo [4]	7, 650	745	119, 081	8	Commodity purchase relationship	Medium
Amazon-Computers [4]	13, 752	767	245, 861	10	Commodity purchase relationship	Large
Pubmed [60]	19, 717	500	44, 438	3	Paper citation relationship	Large
Cora-Full [61]	19, 737	8, 710	63, 421	70	Paper citation relationship	Large

TABLE II
THE CLUSTERING RESULTS ON ACM AND DBLP BENCHMARKS. THE BEST RESULTS IN ALL METHODS AND ALL BASELINES ARE REPRESENTED BY **BOLD** VALUE AND UNDERLINE VALUE, RESPECTIVELY.

Dataset	ACM				DBLP			
Metric	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)
K-Means	67.26 ± 0.75	31.91 ± 0.35	54.47 ± 0.32	30.76 ± 0.62	39.08 ± 0.36	10.11 ± 0.21	38.01 ± 0.37	7.28 ± 0.29
GAE (*NeurIPS*' 16)	82.47 ± 0.92	50.29 ± 1.86	82.65 ± 0.89	54.59 ± 1.99	59.25 ± 0.40	26.37 ± 0.29	59.84 ± 0.32	20.95 ± 0.43
VGAE (*NeurIPS*' 16)	82.85 ± 0.63	50.22 ± 1.24	82.85 ± 0.62	55.56 ± 1.15	62.22 ± 0.83	26.62 ± 1.37	60.70 ± 0.85	25.08 ± 1.23
ARGA (*IEEE TC*' 20)	86.85 ± 0.64	58.05 ± 1.53	86.84 ± 0.60	64.77 ± 1.53	64.60 ± 0.95	28.65 ± 0.63	64.49 ± 0.63	27.44 ± 1.27
ARVGA (*IEEE TC*' 20)	84.84 ± 0.36	52.89 ± 0.84	84.86 ± 0.35	59.67 ± 0.85	64.10 ± 0.96	31.01 ± 0.89	64.36 ± 1.01	25.69 ± 1.51
DAEGC (*IJCAI*' 19)	87.18 ± 0.05	59.32 ± 0.12	87.27 ± 0.05	65.46 ± 0.12	75.87 ± 0.46	42.45 ± 0.58	75.41 ± 0.45	46.80 ± 0.87
SDCN (*WWW*' 20)	89.44 ± 0.26	65.89 ± 0.95	89.40 ± 0.28	71.47 ± 0.67	71.91 ± 0.57	37.80 ± 1.06	71.21 ± 0.73	40.45 ± 1.18
DFCN (*AAAI*' 21)	90.15 ± 0.05	67.98 ± 0.18	90.14 ± 0.05	73.25 ± 0.14	75.42 ± 0.82	43.20 ± 0.74	75.31 ± 0.71	45.07 ± 1.91
DCRN (*AAAI*' 22)	90.51 ± 0.24	68.19 ± 0.31	89.95 ± 0.28	74.52 ± 0.16	76.59 ± 0.32	44.96 ± 0.28	76.03 ± 0.40	47.65 ± 0.39
CDRS (*TNNLS*' 22)	87.95 ± 0.52	59.63 ± 0.55	89.24 ± 0.54	67.51 ± 0.48	69.21 ± 0.21	38.57 ± 0.35	70.54 ± 0.27	39.45 ± 0.34
ITR (*IJCAT*' 22)	88.23 ± 0.07	65.46 ± 0.12	88.74 ± 0.04	68.35 ± 0.06	73.18 ± 0.14	39.46 ± 0.09	72.16 ± 0.05	41.55 ± 0.07
GraphCL (*NeurIPS*' 20)	90.18 ± 0.04	68.24 ± 0.12	90.04 ± 0.05	73.38 ± 0.09	74.90 ± 0.10	45.14 ± 0.14	74.51 ± 0.10	45.86 ± 0.19
GCA (*WWW*' 21)	88.95 ± 0.26	65.33 ± 0.56	89.07 ± 0.26	69.82 ± 0.67	73.90 ± 0.48	41.35 ± 0.79	72.91 ± 0.76	43.65 ± 0.65
GMI (*WWW*' 20)	90.36 ± 0.09	67.92 ± 0.12	89.23 ± 0.08	74.17 ± 0.07	75.23 ± 0.21	43.51 ± 0.15	75.10 ± 0.14	44.09 ± 0.08
SCCL (*NAACL-HLT*' 21)	87.35 ± 1.20	59.43 ± 0.95	87.17 ± 1.32	65.64 ± 1.10	67.73 ± 1.13	36.80 ± 1.56	68.92 ± 0.90	38.23 ± 1.15
GCC (*ICCV*' 21)	89.58 ± 0.09	67.10 ± 0.07	89.14 ± 0.23	68.96 ± 0.14	70.16 ± 0.45	37.20 ± 0.32	70.33 ± 0.50	39.05 ± 0.42
SCAGC	**91.83 ± 0.03**	**71.28 ± 0.06**	**91.84 ± 0.03**	**77.29 ± 0.07**	**79.42 ± 0.02**	**49.05 ± 0.02**	**78.88 ± 0.02**	**54.04 ± 0.03**

logical graph structures as input. For GAE, VGAE, ARGA, ARVGA, SDCN, DFCN, DCRN, GMI, ITR, GraphCL, and GCA, we remain in the same settings as in the papers, their corresponding clustering assignment matrix is obtained by running *K*-means on the extracted node representation. For SCCL and GCC, we change their feature learning structure to GCN, which is consistent with SCAGC, in order to adapt to the attribute graph data.

3) Evaluation Metrics: We leverage four commonly used metrics to evaluate the efficiency of all methods, *i.e.*, accuracy (ACC), normalized mutual information (NMI), average rand index (ARI), and macro F1-score (F1). For these metrics, the higher the value, the better the performance.

4) Implementation Details: The proposed SCAGC and the baseline methods are implemented on a Windows 10 machine with an Intel (R) Xeon (R) Gold 6230 CPU and dual NVIDIA Tesla P100-PCIE GPUs. The deep learning environment consists of PyTorch 1.6.0 platform, PyTorch Geometric 1.6.1 platform, and TensorFlow 1.13.1. To ensure the availability of the initial pseudo clustering assignment matrix $\overrightarrow{\mathbf{L}}$, we pretrain the shared graph convolutional encoder and graph contrastive representation learning module via a classic contrastive learning loss.

For SCAGC, all network parameters are initialized with Xavier initialization [64] and trained by Adam SGD optimizer with learning rate 5×10^{-4}. To avoid over-fitting, we leverage ℓ_2 weight decay technique with 10^{-5} decay rate on all datasets. In SCAGC, there are three trade-off parameters, *i.e.*, τ_1, τ_2, and γ, where τ_1 and τ_2 are temperature parameters for contrastive learning; γ is parameter of the clustering regularizer. Based on empirical values, we turn τ_2 from 0.1 to 0.5 with interval 0.1; we turn τ_1 from 0.6 to 1.0 with interval 0.1; γ is set to 1.0 on all datasets. Thus, for SCAGC, only two parameters need to be fine-tuned, thus, the proposed SCAGC is easy to

TABLE III
THE CLUSTERING RESULTS ON PUBMED BENCHMARK. THE BEST RESULTS IN ALL METHODS AND ALL BASELINES ARE REPRESENTED BY **BOLD** VALUE AND UNDERLINE VALUE, RESPECTIVELY.

Metric	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)
K-Means	59.83± 0.01	31.05± 0.02	58.88 ± 0.01	51.43 ± 0.35
GAE (*NeurIPS*' 16)	62.09 ± 0.81	23.84 ± 3.54	61.37 ± 0.85	20.62 ± 1.39
VGAE (*NeurIPS*' 16)	68.48 ± 0.77	30.61 ± 1.71	67.68 ± 0.89	30.15 ± 1.23
ARGA (*IEEE TC*' 20)	65.26 ± 0.12	24.80 ± 0.17	65.69 ± 0.13	24.53 ± 0.17
ARVGA (*IEEE TC*' 20)	64.25 ± 1.24	23.88 ± 1.05	64.51 ± 1.52	22.82 ± 1.32
DAEGC (*IJCAI*' 19)	68.73 ± 0.03	28.26 ± 0.03	68.23 ± 0.02	29.84 ± 0.04
SDCN (*WWW*' 20)	64.20 ± 1.30	22.87 ± 2.04	65.01 ± 1.21	22.30 ± 2.07
DFCN (*AAAI*' 21)	68.89 ± 0.07	31.43 ± 0.13	68.10 ± 0.07	30.64 ± 0.11
DCRN (*AAAI*' 22)	69.87 ± 0.07	32.20 ± 0.08	68.94 ± 0.08	31.41 ± 0.12
CDRS (*TNNLS*' 22)	69.37 ± 0.04	29.45 ± 0.05	68.19 ± 0.13	27.94 ± 0.05
ITR (*IJCAT*' 22)	70.56 ± 0.02	33.87 ± 0.05	70.31 ± 0.03	31.44 ± 0.07
GraphCL (*NeurIPS*' 20)	68.75 ± 0.17	30.92 ± 0.13	67.56 ± 0.14	30.23 ± 0.08
GCA (*WWW*' 21)	69.51 ± 0.20	31.13 ± 0.18	68.54 ± 0.24	30.85 ± 0.17
GMI (*WWW*' 20)	71.53 ± 0.02	34.21 ± 0.03	70.36 ± 0.05	33.56 ± 0.07
SCCL (*NAACL-HLT*' 21)	68.33 ± 1.21	30.52 ± 1.05	67.80 ± 0.82	29.42 ± 1.35
GCC (*ICCV*' 21)	69.15 ± 0.12	31.26 ± 0.15	68.43 ± 0.17	29.32 ± 0.09
SCAGC	**72.42 ± 0.07**	**35.13 ± 0.05**	**71.55 ± 0.09**	**34.19 ± 0.03**

TABLE IV
THE CLUSTERING RESULTS ON AMAZON-PHOTO AND AMAZON-COMPUTERS BENCHMARKS. THE BEST RESULTS IN ALL METHODS AND ALL BASELINES ARE REPRESENTED BY **BOLD** VALUE AND UNDERLINE VALUE, RESPECTIVELY.

Dataset	Amazon-Photo				Amazon-Computers			
Metric	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)
K-Means	36.53 ± 4.11	19.31 ± 3.75	32.63 ± 1.90	12.61 ± 3.54	36.44 ± 2.64	16.64 ± 4.59	28.08 ± 1.44	2.71 ± 1.98
GAE (*NeurIPS*' 16)	42.03 ± 0.54	31.87 ± 0.51	34.01 ± 0.42	19.31 ± 0.53	43.14 ± 1.74	35.47 ± 1.58	27.06 ± 2.63	19.61 ± 1.85
VGAE (*NeurIPS*' 16)	40.67 ± 0.92	31.46 ± 2.03	38.01 ± 2.67	15.70 ± 1.18	42.44 ± 0.16	37.62 ± 0.23	24.94 ± 0.14	22.16 ± 0.35
ARGA (*IEEE TC*' 20)	57.79 ± 2.26	48.01 ± 1.65	52.56 ± 2.68	34.44 ± 1.58	45.67 ± 0.37	37.21 ± 0.92	40.02 ± 1.29	26.28 ± 1.02
ARVGA (*IEEE TC*' 20)	47.89 ± 1.36	41.37 ± 1.39	42.96 ± 1.46	27.72 ± 1.06	47.16 ± 0.26	38.84 ± 0.96	41.51 ± 0.83	27.27 ± 0.84
DAEGC (*IJCAI*' 19)	60.14 ± 0.93	58.03 ± 1.25	52.37 ± 2.39	43.55 ± 1.76	49.26 ± 0.49	39.28 ± 4.97	33.71 ± 5.76	35.29 ± 1.97
SDCN (*WWW*' 20)	71.43 ± 0.31	64.13 ± 0.10	68.74 ± 0.22	51.17 ± 0.13	54.12 ± 1.13	39.90 ± 1.51	28.84 ± 4.20	31.59 ± 1.08
DFCN (*AAAI*' 21)	73.43 ± 0.61	64.74 ± 1.04	69.96 ± 0.49	52.39 ± 1.01	56.24 ± 0.16	41.83 ± 0.40	33.39 ± 1.11	33.02 ± 0.39
DCRN (*AAAI*' 22)	74.18 ± 0.13	65.33 ± 0.20	70.21 ± 0.18	54.01 ± 0.19	55.19 ± 0.73	40.75 ± 0.65	30.82 ± 0.84	32.07 ± 1.12
CDRS (*TNNLS*' 22)	55.14 ± 1.02	46.57 ± 0.81	45.44 ± 0.70	33.69 ± 0.49	49.05 ± 0.18	36.74 ± 0.14	40.52 ± 0.05	29.87 ± 0.07
ITR (*IJCAT*' 22)	65.96 ± 0.04	62.34 ±0.07	54.32 ± 0.13	45.02 ± 0.12	50.01± 0.29	39.82 ± 0.25	29.75 ± 0.44	31.59 ± 0.51
GraphCL (*NeurIPS*' 20)	66.61 ± 0.56	57.35 ± 0.32	58.52 ± 0.55	45.13 ± 0.44	50.22 ± 0.66	41.78 ± 2.44	32.89 ± 2.16	36.94 ± 3.20
GCA (*WWW*' 21)	71.17 ± 0.27	60.70 ± 0.41	64.12 ± 1.21	49.09 ± 0.62	54.92 ± 0.55	44.36 ± 0.86	40.43 ± 0.45	35.61 ± 0.62
GMI (*WWW*' 20)	73.62 ± 0.07	64.85 ± 0.02	70.17 ± 0.05	52.56 ± 0.03	55.45 ± 0.08	41.23 ± 0.07	31.14 ± 0.07	32.43 ± 0.09
SCCL (*NAACL-HLT*' 21)	70.46 ± 0.22	64.20± 0.28	69.54 ± 0.35	52.55 ± 0.43	52.28 ± 0.69	41.79 ± 0.52	33.38 ± 0.70	34.18 ± 0.83
GCC (*ICCV*' 21)	72.23 ± 0.21	64.45 ± 0.18	69.21 ± 0.15	51.33 ± 0.19	54.35 ± 0.07	44.03 ± 0.06	39.79 ± 0.07	35.23 ± 0.08
SCAGC	**75.25 ± 0.10**	**67.18 ± 0.13**	**72.77 ± 0.16**	**56.86 ± 0.23**	**58.43 ± 0.12**	**49.92 ± 0.08**	**43.14 ± 0.09**	**38.29 ± 0.07**

be implemented.

Moreover, the proposed SCAGC requires the graph augmentation technique for contrastive learning. For convenience, following [41], we leverage the same settings of their proposed adaptive augmentation scheme to generate a two-view attributed graph. Notably, the degree centrality is used as the node centrality function to generate different topology graph views. The output size of the shared graph convolutional encoder is set to 256, the output size of the graph contrastive representation learning sub-network is set to 128, and the output size of the contrastive clustering sub-network is set to be equal to the number of clusters K.

For all baseline methods, we follow the hyper-parameter settings as reported in their articles and run their released code to obtain the clustering results. To avoid the randomness of the clustering results, we repeat each experiment of SCAGC and baseline methods 10 times and report their average values and the corresponding standard deviations.

B. Node Clustering Performance

Table II, Table III, Table IV, and Table V present the node clustering results of the proposed SCAGC and all baseline methods. It can be observed from these results that:

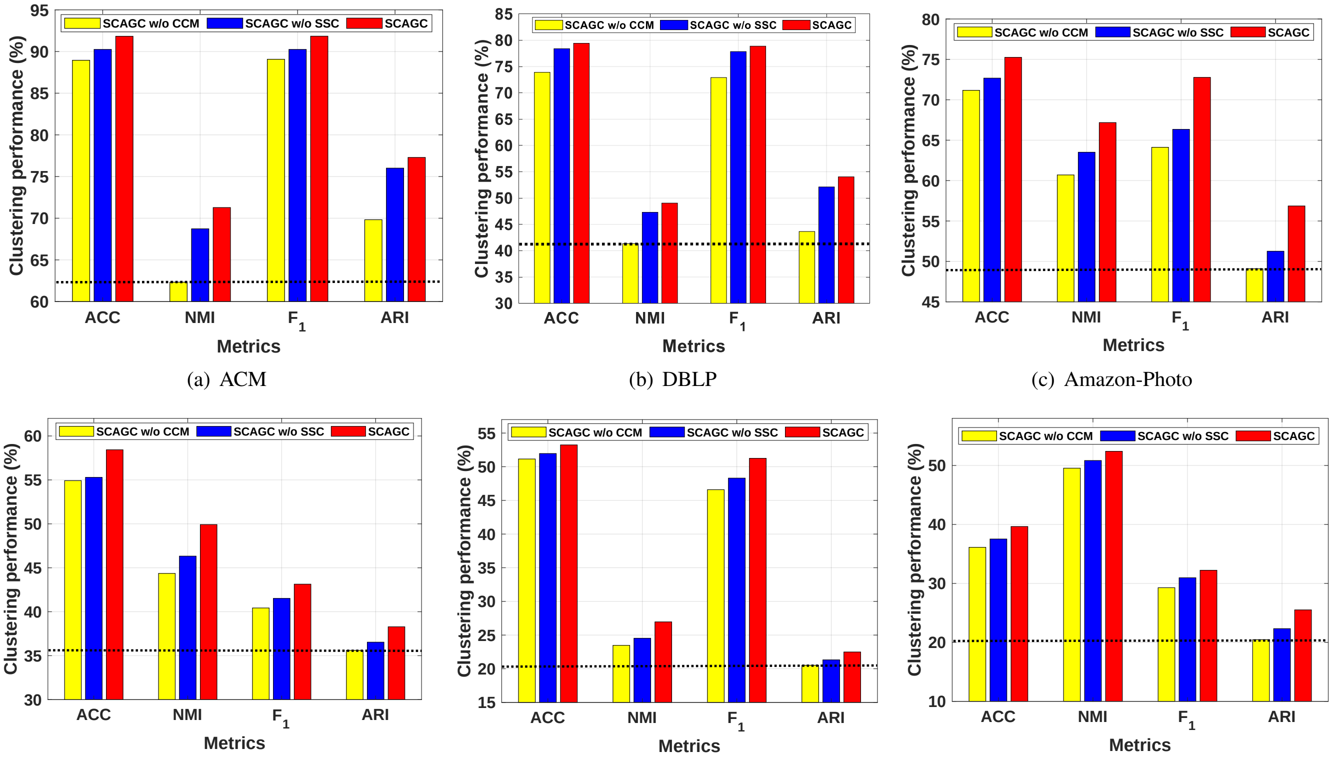
1) The proposed SCAGC significantly and consistently outperforms other comparison methods, which indicates that SCAGC can handle various types of attribute graph data well.
2) The proposed SCAGC and other GCN-based attributed graph clustering methods outperform K-Means. The reason may be that GCN-based attributed graph clustering methods simultaneously explore the information embedded in node attribute and topological graph structure. In contrast, these classical clustering methods only use the node attribute. Moreover, compared with classical clustering methods, GCN-based methods use a multi-layer nonlinear graph neural network as the feature extractor, then map input data into a new subspace to carry out downstream clustering. These results well demonstrate the effectiveness of GCN in processing attributed graph data.
3) The proposed SCAGC achieves much better clustering results than some representative graph auto-encoder (GAE, VGAE, ARGA, ARVGA, CDRS). This is because compared with traditional graph auto-encoder, SCAGC leverages a graph augmentation scheme to generate a useful attributed graph, and takes the relationship between positive pair and negative pair into account. These strategies help to improve the quality of node representation.
4) In some cases, the clustering performance of contrastive learning-based baselines, *i.e.*, GraphCL, GCA, and GCM, are inferior to clustering-directed, *i.e.*, DAEGC, SDCN, DFCN, DCRN, and SCAGC. This is because SCAGC integrates the node clustering and representation into an end-to-end framework, which helps to better explore the cluster structure. In contrast, GraphCL, GCA, and GCM execute the node representation and clustering in two separate steps, which limits their performances.
5) From Tables II-V, we can see that the proposed SCAGC consistently outperforms all baselines on six datasets. Particularly, SCAGC surpasses the closest competitor GCA by 5.95% on ACM and 7.7% on DBLP, in terms of NMI. These remarkable performances verify the clustering ability of SCAGC. And it demonstrates that the contrastive clustering module and self-consistent graph contrastive representation learning module with pseudo-labels prompt is effective at benefiting the node representation learning and clustering.

C. Ablation Studies

In this section, two ablation scenarios are implemented to further verify the effectiveness of the contrastive clustering module and the proposed self-supervised GCRL loss with pseudo-labels prompt.

TABLE V
THE CLUSTERING RESULTS ON UAT AND CORA-FULL BENCHMARKS. THE BEST RESULTS IN ALL METHODS AND ALL BASELINES ARE REPRESENTED BY **BOLD** VALUE AND UNDERLINE VALUE, RESPECTIVELY.

Dataset	UAT				Cora-Full			
Metric	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)
K-Means	42.47 ± 0.15	22.39 ± 0.69	36.12 ± 0.22	15.71 ± 0.76	26.27 ± 1.10	34.68 ± 0.84	22.57 ± 1.09	9.35 ± 0.57
GAE (*NeurIPS*' 16)	48.97 ± 1.52	20.69 ± 0.98	47.95 ± 1.52	18.33 ± 1.79	29.60 ± 0.81	45.82 ± 0.75	25.95 ± 0.75	17.84 ± 0.86
VGAE (*NeurIPS*' 16)	46.32 ± 0.15	16.28 ± 0.20	45.21 ± 0.18	17.45 ± 0.17	32.66 ± 1.29	47.38 ± 1.59	29.06 ± 1.51	20.01 ± 1.38
ARGA (*IEEE TC*' 20)	49.31 ± 0.15	25.44 ± 0.31	50.26 ± 0.16	16.57 ± 0.31	22.07 ± 0.43	41.28 ± 0.25	12.38 ± 0.24	18.85 ± 0.41
ARVGA (*IEEE TC*' 20)	46.12 ± 1.41	15.94 ± 1.63	45.31 ± 1.54	12.77 ± 1.46	29.75 ± 0.69	40.10 ± 0.22	24.62 ± 0.53	16.47 ± 0.38
DAEGC (*IJCAI*' 19)	52.29 ± 0.49	21.33 ± 0.44	50.33 ± 0.64	20.50 ± 0.51	34.35 ± 1.00	49.16 ± 0.73	26.96 ± 1.33	22.60 ± 0.47
SDCN (*WWW*' 20)	52.25 ± 1.91	21.61 ± 1.26	45.59 ± 3.54	21.63 ± 1.49	26.67 ± 0.40	37.38 ± 0.39	22.14 ± 0.43	13.63 ± 2.27
DFCN (*AAAI*' 21)	33.61 ± 0.09	26.49 ± 0.41	25.79 ± 0.29	11.87 ± 0.23	37.51 ± 0.81	51.30 ± 0.41	31.22 ± 0.87	24.46 ± 0.48
DCRN (*AAAI*' 22)	52.15 ± 0.28	24.31 ± 0.25	49.42 ± 0.33	19.67 ± 0.28	38.80 ± 0.60	51.91 ± 0.35	31.68 ± 0.76	25.25 ± 0.49
CDRS (*TNNLS*' 22)	48.75 ± 0.14	21.03 ± 0.08	48.12 ± 0.15	18.49 ± 0.17	33.24 ± 0.71	48.51 ± 0.69	26.15 ± 0.57	20.39 ± 0.73
ITR (*IJCAT*' 22)	50.31 ± 0.32	23.15 ± 0.37	46.54 ± 0.28	19.43 ± 0.27	32.44 ± 0.56	41.96 ± 0.42	29.43 ± 0.47	18.75 ± 0.35
GraphCL (*NeurIPS*' 20)	49.58 ± 0.43	22.14 ± 0.51	49.83 ± 0.46	17.21 ± 0.38	34.09 ± 1.01	42.56 ± 0.82	29.21 ± 0.94	19.85 ± 0.75
GCA (*WWW*' 21)	51.15 ± 0.30	23.47 ± 0.24	46.59 ± 0.29	20.52 ± 0.35	36.12 ± 0.64	49.54 ± 0.73	29.27 ± 0.68	20.45 ± 0.54
GMI (*WWW*' 20)	51.72 ± 0.17	24,47 ± 0.15	50.18 ± 0.18	19.96 ± 0.20	37.40 ± 0.02	51.37 ± 0.05	30.31 ± 0.07	23.55 ± 0.03
SCCL (*NAACL-HLT*' 21)	48.33 ± 0.43	21.72 ± 0.34	49.34 ± 0.27	19.56 ± 0.52	36.47 ± 0.25	48.57 ± 0.16	30.44 ± 0.13	21.25 ± 0.23
GCC (*ICCV*' 21)	50.35 ± 0.13	22.26 ± 0.10	47.26 ± 0.12	18.49 ± 0.14	33.20 ± 0.14	43.27 ± 0.20	28.14 ± 0.16	19.47± 0.18
SCAGC	**53.24 ± 0.12**	**26.96 ± 0.09**	**51.25 ± 0.14**	**22.49 ± 0.13**	**39.65 ± 0.10**	**52.40 ± 0.07**	**32.23 ± 0.11**	**25.51 ± 0.09**



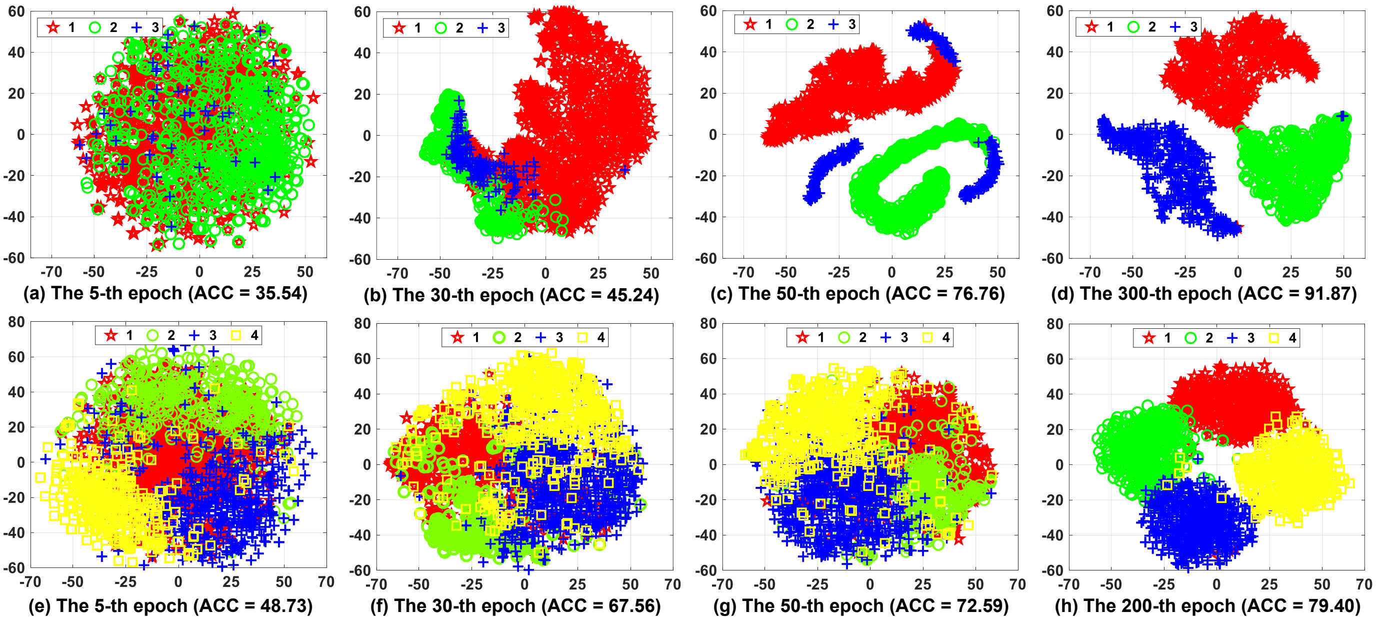
(a) ACM (b) DBLP (c) Amazon-Photo

(d) Amazon-Computers (e) UAT (f) Cora-Full

Fig. 3. Ablation Studies on six datasets, where 'SCAGC w/o CCM' means SCAGC is trained without contrastive clustering module and self-supervised contrastive loss with pseudo-label prompt; 'SCAGC w/o CCM' means SCAGC is trained without self-supervised contrastive loss with pseudo-label prompt.

1) Effect of Contrastive Clustering Module: To better illustrate the effectiveness of the contrastive clustering module, we compare the clustering results of SCAGC and SCAGC without contrastive clustering module and self-consistent contrastive loss with pseudo-label prompt (**termed SCAGC w/o CCM**) on six datasets. Note that, in this scenario, SCAGC w/o CCM is trained using traditional contrastive loss [35], [41], *i.e.*, SCAGC w/o CCM is clustering-agnostic. As shown in Figure 3 (a-f), the clustering performance of SCAGC (see the red bar) are substantially superior to SCAGC w/o CCM (see the yellow bar). This is because SCAGC can better extract node representation benefiting from the contrastive clustering module. While in the absence of the specific clustering task, SCAGC w/o CCM fails to explore the cluster structure, resulting in a quick drop in the performance of SCAGC.

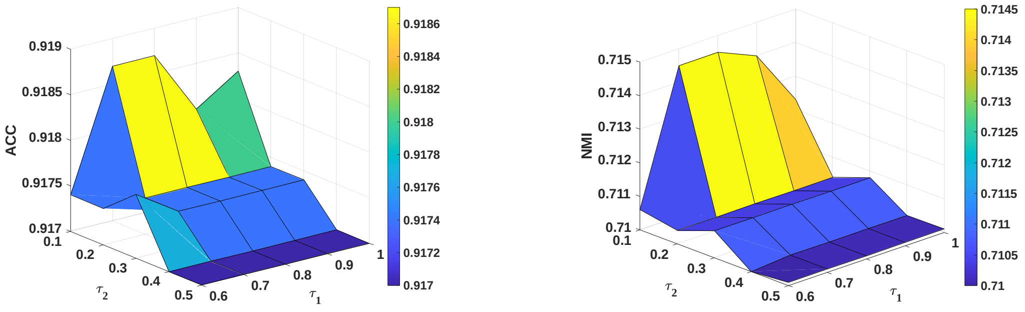
2) Importance of the Proposed Self-consistent GCRL Loss: To this end, we compare the clustering performances of SCAGC and SCAGC without self-consistent GCRL loss with



(a) The 5-th epoch (ACC = 35.54) (b) The 30-th epoch (ACC = 45.24) (c) The 50-th epoch (ACC = 76.76) (d) The 300-th epoch (ACC = 91.87)

(e) The 5-th epoch (ACC = 48.73) (f) The 30-th epoch (ACC = 67.56) (g) The 50-th epoch (ACC = 72.59) (h) The 200-th epoch (ACC = 79.40)

Fig. 4. The t-SNE visualizations on the ACM (a-d) and IMDB (e-h) datasets with the increasing number of iterations.



(a) ACC (b) NMI

Fig. 5. The clustering performance vs. τ_2 and τ_1 on ACM dataset.

pseudo-label prompt (**termed SCAGC w/o SSC**) on six datasets. Notably, in this scenario, SCAGC w/o SSC is trained by replacing Eq. (3) with a standard contrastive loss [35], [41]. As reported in Figure 3 (a-f), SCAGC (see red bar) always achieves the best performance in terms of all four metrics. These results demonstrate that taking into account self-consistency guides the learning of latent representation, thus, considering the self-consistency between cluster structure and latent representations is a promising method for unsupervised clustering tasks.

TABLE VI
OUT-OF-SAMPLE NODE CLUSTERING ON CORA-FULL DATASET. THE BEST RESULTS IN ALL METHODS AND ALL BASELINES ARE REPRESENTED BY **BOLD** VALUE AND UNDERLINE VALUE, RESPECTIVELY.

Metric	ACC (↑)	NMI (↑)	F_1 (↑)	ARI (↑)
DCRN	38.80 ± 0.60	51.91 ± 0.35	31.68 ± 0.76	25.25 ± 0.49
GraphCL	34.09 ± 1.01	42.56 ± 0.82	29.21 ± 0.94	19.85 ± 0.75
GCA	36.12 ± 0.64	49.54 ± 0.73	29.27 ± 0.68	20.45 ± 0.54
SCAGC-T	42.19 ± 0.32	53.40 ± 0.21	33.47 ± 0.14	26.38 ± 0.08
SCAGC-OOS	**40.04 ± 0.05**	**51.96 ± 0.03**	**32.76 ± 0.02**	**25.61 ± 0.06**
SCAGC	39.65 ± 0.10	52.40 ± 0.07	32.23 ± 0.11	25.51 ± 0.09

D. Model Analysis

1) *The Effect of Handling Out-of-sample Data:* Taking the large-scale datasets Cora-Full as an example, we verify the effectiveness of SCAGC to handle out-of-sample nodes. To this end, as reported in Remark 1, we randomly sample 50% of the sub-attribute graph as training data to train SCAGC and the rest as newly generated data for testing. SCAGC-T and SCAGC-OOS are denoted as clustering results of training and testing, respectively. As shown in Table VI, compare to the top three algorithms, SCAGC-OOS achieves the best clustering results. These results indicate that the trained SGCMC enjoys a promising generalization ability to the out-of-sample data in real attributed graph data.

2) *Visualizations of Clustering Results:* By simultaneously exploiting the good property of GCRL and taking advantage of the clustering labels, SCAGC ought to learn a discriminative node representation and desirable clustering label at the same time. To illustrate how SCAGC achieves the goal, as shown in Figure 4, we implement t-SNE [65] on the learned $\mathbf{M}$ at four different training iterations on ACM and DBLP datasets, where different colors indicated different clustering labels predicted by SCAGC. As observed, the cluster assignments

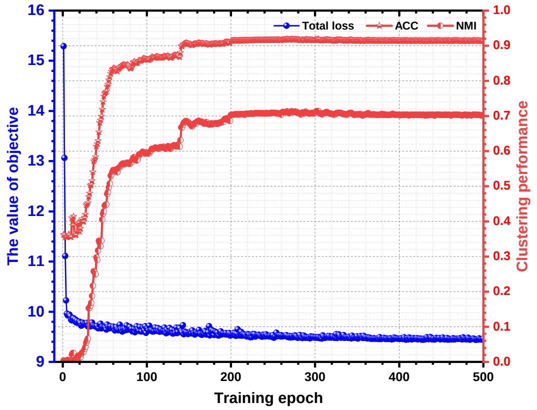


Fig. 6. The convergence of SCAGC on ACM dataset.

become more reasonable, and different clusters scatter and gather more distinctly. These results indicate that the learned node representation becomes more compact and discriminative with the increasing number of iterations.

3) Sensitivity Analysis: We also make experiments to verify the sensitivity on hyper-parameters in the proposed SCAGC, namely τ_1 and τ_2. To this end, we turn τ_2 from 0.1 to 0.5 with interval 0.1, and turn τ_1 from 0.6 to 1.0 with interval 0.1. The results on the ACM dataset are shown in Figure 5. One can observe that the fluctuation ranges of both ACC and NMI do not exceed 0.5% when we tune the hyper-parameters τ_1 and τ_2. These results clearly verify that the self-supervised contrastive learning solution offered by SCAGC to attributed graph clustering is relatively stable and effective.

4) Convergence Analysis: Taking ACM dataset as an example, we investigate the convergence of SCAGC. We record the objective values and clustering results of SCAGC with iteration and plot them in Figure 6. As shown in Figure 6, the objective values (see the blue line) decrease a lot in the first 100 iterations, then continuously decrease until convergence. Moreover, the ACC of SCAGC continuously increases to a maximum in the first 200 iterations, and generally maintain stable to slight variation. The curves in terms of NMI metric has a similar trend. These observations clearly indicate that SCAGC usually converges quickly.

V. Conclusion and Future Work

To conclude, we propose a novel self-consistent contrastive attributed clustering (SCAGC) approach, which can directly predict the clustering labels of the unlabeled attributed graphs and handle out-of-sample nodes. We also propose a new self-consistent contrastive loss based on imprecise clustering labels to improve the quality of node representation. We believe that the proposed SCAGC will help facilitate the exploration of the attributed graph where labels are time and labor-consuming to acquire. In the future, motivated by explainable AI (XAI), like Peng *et al.* [66], we will also study explainable contrastive attributed graph clustering.

Acknowledgment

The authors would like to thank the editors and reviewers for their helpful comments and suggestions. Also, deserving of recognition is Prof. Xi Peng, Mr. Yanqiao Zhu, and Mr. Yichen Xu, for providing the datasets and codes of their methods.

References

[1] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang, "Understanding negative sampling in graph representation learning," in *ACM SIGKDD*, pp. 1666–1676, 2020.

[2] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "GCC: graph contrastive coding for graph neural network pre-training," in *ACM SIGKDD*, pp. 1150–1160, 2020.

[3] Z. Peng, M. Luo, J. Li, L. Xue, and Q. Zheng, "A deep multi-view framework for anomaly detection on attributed networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 6, pp. 2539–2552, 2022.

[4] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," in *NeurIPS Workshop on Relational Representation Learning*, 2018.

[5] J. Piao, G. Zhang, F. Xu, Z. Chen, and Y. Li, "Predicting customer value with social relationships via motif-based graph attention networks," in *WWW*, pp. 3146–3157, 2021.

[6] C. Huang, H. Xu, Y. Xu, P. Dai, L. Xia, M. Lu, L. Bo, H. Xing, X. Lai, and Y. Ye, "Knowledge-aware coupled graph neural network for social recommendation," in *AAAI*, pp. 4115–4122, 2021.

[7] S. Wan, S. Pan, J. Yang, and C. Gong, "Contrastive and generative graph convolutional networks for graph-based semi-supervised learning," in *AAAI*, pp. 10049–10057, 2021.

[8] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, and L. Wang, "Learning a joint affinity graph for multiview subspace clustering," *IEEE Trans. Multim.*, vol. 21, no. 7, pp. 1724–1736, 2019.

[9] Y. Han, L. Zhu, Z. Cheng, J. Li, and X. Liu, "Discrete optimal graph clustering," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1697–1710, 2020.

[10] D. Shi, L. Zhu, Y. Li, J. Li, and X. Nie, "Robust structured graph clustering," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 11, pp. 4424–4436, 2020.

[11] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: incomplete multi-view clustering via contrastive prediction," in *IEEE CVPR*, pp. 11174–11183, 2021.

[12] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Trans. Multim.*, vol. 24, pp. 2461–2472, 2022.

[13] Q. Wang, X. He, X. Jiang, and X. Li, "Robust bi-stochastic graph regularized matrix factorization for data clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 390–403, 2022.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[15] D. Valsesia, G. Fracastoro, and E. Magli, "Learning localized representations of point clouds with graph-convolutional generative adversarial networks," *IEEE Trans. Multim.*, vol. 23, pp. 402–414, 2021.

[16] W. Nie, M. Ren, A. Liu, Z. Mao, and J. Nie, "M-GCN: multi-branch graph convolution network for 2d image-based on 3d model retrieval," *IEEE Trans. Multim.*, vol. 23, pp. 1962–1976, 2021.

[17] B. Fatemi, L. E. Asri, and S. M. Kazemi, "SLAPS: self-supervision improves structure learning for graph neural networks," in *NeurIPS*, pp. 22667–22681, 2021.

[18] J. Wen, K. Yan, Z. Zhang, Y. Xu, J. Wang, L. Fei, and B. Zhang, "Adaptive graph completion based incomplete multi-view clustering," *IEEE Trans. Multim.*, vol. 23, pp. 2493–2504, 2021.

[19] Q. Wang, J. Cheng, Q. Gao, G. Zhao, and L. Jiao, "Deep multi-view subspace clustering with unified and discriminative learning," *IEEE Trans. Multim.*, vol. 23, pp. 3483–3493, 2021.

[20] W. Xia, Q. Wang, Q. Gao, X. Zhang, and X. Gao, "Self-supervised graph convolutional network for multi-view clustering," *IEEE Trans. Multim.*, to be published, doi: 10.1109/TMM.2021.3094296.

[21] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: marginalized graph autoencoder for graph clustering," in *CIKM*, pp. 889–898, 2017.

[22] X. Zhang, H. Liu, Q. Li, and X. Wu, "Attributed graph clustering via adaptive graph convolution," in *IJCAI*, pp. 4327–4333, 2019.

[23] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, "Symmetric graph convolutional autoencoder for unsupervised graph representation learning," in *IEEE ICCV*, pp. 6518–6527, 2019.

[24] J. Cheng, Q. Wang, Z. Tao, D. Xie, and Q. Gao, "Multi-view attribute graph convolution networks for clustering," in *IJCAI*, pp. 2973–2979, 2020.

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

[26] W. Xia, Q. Wang, Q. Gao, X. Zhang, and X. Gao, "Self-supervised graph convolutional network for multi-view clustering," *IEEE Trans. Multim.*, vol. 24, pp. 3182–3192, 2022.

[27] Z. Lin and Z. Kang, "Graph filter-based multi-view attributed graph clustering," in *IJCAI*, pp. 2723–2729, 2021.

[28] T. N. Kipf and M. Welling, "Variational graph auto-encoders," in *NeurIPS Workshop on Bayesian Deep Learning*, 2016.

[29] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI*, pp. 2609–2615, 2018.

[30] 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, 2020.

[31] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, vol. 48, pp. 478–487, 2016.

[32] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: A deep attentional embedding approach," in *IJCAI*, pp. 3670–3676, 2019.

[33] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *WWW*, pp. 1400–1410, 2020.

[34] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, and J. Cheng, "Deep fusion clustering network," in *AAAI*, pp. 9978–9987, 2021.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, pp. 1597–1607, 2020.

[36] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE CVPR*, pp. 9726–9735, 2020.

[37] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *ICLR*, 2019.

[38] F. Sun, J. Hoffmann, V. Verma, and J. Tang, "Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *ICLR*, 2020.

[39] R. Zhang, C. Lu, Z. Jiao, and X. Li, "Deep contrastive graph representation via adaptive homotopy learning," *CoRR*, vol. abs/2106.09244, 2021.

[40] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *NeurIPS*, 2020.

[41] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Graph contrastive learning with adaptive augmentation," in *WWW*, pp. 2069–2080, 2021.

[42] M. Jin, Y. Zheng, Y. Li, C. Gong, C. Zhou, and S. Pan, "Multi-scale contrastive siamese networks for self-supervised graph representation learning," in *IJCAI*, pp. 1477–1483, 2021.

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

[44] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, "Graph representation learning via graphical mutual information maximization," in *WWW*, pp. 259–270, 2020.

[45] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, 2021.

[46] S. Albelwi, "Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging," *Entropy*, vol. 24, no. 4, p. 551, 2022.

[47] C. Niu and G. Wang, "SPICE: semantic pseudo-labeling for image clustering," *CoRR*, vol. abs/2103.09382, 2021.

[48] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, and H. Zha, "Deep comprehensive correlation mining for image clustering," in *IEEE ICCV*, pp. 8149–8158, 2019.

[49] H. Zhong, J. Wu, C. Chen, J. Huang, M. Deng, L. Nie, Z. Lin, and X. Hua, "Graph contrastive clustering," in *IEEE ICCV*, pp. 9204–9213, IEEE, 2021.

[50] D. Zhang, F. Nan, X. Wei, S. Li, H. Zhu, K. R. McKeown, R. Nallapati, A. O. Arnold, and B. Xiang, "Supporting clustering with contrastive learning," in *NAACL-HLT*, pp. 5419–5430, Association for Computational Linguistics, 2021.

[51] J. Lv, Z. Kang, X. Lu, and Z. Xu, "Pseudo-supervised deep subspace clustering," *IEEE Trans. Image Process.*, vol. 30, pp. 5252–5263, 2021.

[52] Y. Li, P. Hu, J. Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *AAAI*, pp. 8547–8555, 2021.

[53] G. Cui, J. Zhou, C. Yang, and Z. Liu, "Adaptive graph encoder for attributed graph embedding," in *ACM SIGKDD*, pp. 976–985, 2020.

[54] Y. Mao, X. Yan, Q. Guo, and Y. Ye, "Deep mutual information maximin for cross-modal clustering," in *AAAI*, pp. 8893–8901, 2021.

[55] Y. Liu, W. Tu, S. Zhou, X. Liu, L. Song, X. Yang, and E. Zhu, "Deep graph clustering via dual correlation reduction," in *AAAI*, pp. 7603–7611, 2022.

[56] W. Tu, S. Zhou, X. Liu, Y. Liu, Z. Cai, E. Zhu, C. Zhang, and J. Cheng, "Initializing then refining: A simple graph attribute imputation network," in *IJCAI, pages = 3494–3500, year = 2022*.

[57] J. Wu, J. He, and J. Xu, "Demo-net: Degree-specific graph neural networks for node and graph classification," in *ACM SIGKDD*, pp. 406–415, 2019.

[58] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *ACM SIGKDD*, pp. 990–998, 2008.

[59] S. Pan, J. Wu, X. Zhu, C. Zhang, and Y. Wang, "Tri-party deep network representation," in *IJCAI*, pp. 1895–1901, 2016.

[60] G. Namata, B. London, L. Getoor, B. Huang, and U. Edu, "Query-driven active surveying for collective classification," in *10th International Workshop on Mining and Learning with Graphs*, vol. 8, p. 1, 2012.

[61] A. Bojchevski and S. Günnemann, "Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking," in *ICLR*, 2018.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

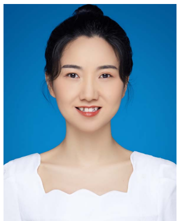
[63] P. Zhu, J. Li, Y. Wang, B. Xiao, S. Zhao, and Q. Hu, "Collaborative decision-reinforced self-supervision for attributed graph clustering," *IEEE Trans. Neural Networks Learn. Syst.*, 2022. doi:10.1109/TNNLS.2022.3171583.

[64] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS* (Y. W. Teh and D. M. Titterington, eds.), vol. 9, pp. 249–256, 2010.

[65] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[66] X. Peng, Y. Li, I. W. Tsang, H. Zhu, J. Lv, and J. T. Zhou, "XAI beyond classification: Interpretable neural clustering," *J. Mach. Learn. Res.*, vol. 23, pp. 6:1–6:28, 2022.

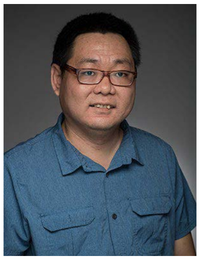
Wei Xia received the B.Eng. degree in Communication Engineering from Lanzhou University of Technology, Lanzhou, China, in 2018. He is currently pursuing a Ph.D. degree in communication and information system at Xidian University, Xi'an, China. His research interests include multi-modal learning, representation learning, unsupervised and self-supervised learning.



Qianqian Wang received the B.Eng. degree in communication engineering from Lanzhou University of Technology, Lanzhou, China, in 2014, the Ph.D. degree from Xidian University, Xi'an, China, in 2019. She was a Visiting Scholar with Northeastern University, Boston, MA, USA, from 2017 to 2018. She is currently a Lecturer with the School of Telecommunications Engineering, Xidian University. Her research interests include pattern recognition, PCA, multi-view clustering, and partial multi-view clustering.

This article has been accepted for publication in IEEE Transactions on Multimedia. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3213208

Quanxue Gao received the B. Eng. degree from Xi'an Highway University, Xi'an, China, in 1998, the M.S. degree from the Gansu University of Technology, Lanzhou, China, in 2001, and the Ph.D. degree from Northwestern Polytechnical University, Xi'an China, in 2005. He was an associate research with the Biometrics Center, The Hong Kong Polytechnic University, Hong Kong from 2006 to 2007. From 2015 to 2016, he was a visiting scholar with the department of computer science, The University of Texas at Arlington, Arlington USA. He is currently a professor with the School of Telecommunications Engineering, Xidian University, and also a key member of State Key Laboratory of Integrated Services Networks. He has authored around 80 technical articles in refereed journals and proceedings, including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, CVPR, AAAI, and IJCAI. His current research interests include pattern recognition and machine learning.



Ming Yang received his BS in Math from Jilin University, Changchun, China in 2007 and his Ph.D. in Math from Texas A&M University-College Station, USA, in 2012. Currently, he is an assistant professor in the mathematics department of the University of Evansville, IN, USA. His research interests are machine learning, image processing and tensor decomposition. He has published several research papers in top-tier journals, including SIAM Journal on Imaging Sciences, IEEE Signal Processing Letters, Alcohol, Journal of Dynamics and Differential Equations, Linear and Multilinear Algebra, PLoS ONE, Applied Sciences-Basel.

Xinbo Gao received the B.Eng., M.Sc. and Ph.D. degrees in electronic engineering, signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow at the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a postdoctoral research fellow at the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. Since 2001, he has been at the School of Electronic Engineering, Xidian University. He is currently a Cheung Kong Professor of the Ministry of Education of P. R. China, a Professor of Pattern Recognition and Intelligent System of Xidian University and a Professor of Computer Science and Technology at Chongqing University of Posts and Telecommunications. His current research interests include Image processing, computer vision, multimedia analysis, machine learning and pattern recognition. He has published six books and around 300 technical articles in refereed journals and proceedings. Prof. Gao is on the Editorial Boards of several journals, including Signal Processing (Elsevier) and Neurocomputing (Elsevier). He served as the General Chair/Co-Chair, Program Committee Chair/Co-Chair, or PC Member for around 30 major international conferences. He is a Fellow of the Institute of Engineering and Technology and a Fellow of the Chinese Institute of Electronics.