Neurocomputing 562 (2023) 126880

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Neighborhood contrastive representation learning for attributed graph clustering

Tong Wang [a,c], Junhua Wu [b], Yaolei Qi [a,c], Xiaoming Qi [a,c], Juwei Guan [c], Yuan Zhang [a,c], Guanyu Yang [a,c,*]

[a] Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, Jiangsu 210096, China
[b] School of Computer Science and Technology, Nanjing Tech University, Jiangsu 211816, China
[c] School of Computer Science and Engineering, Southeast University, Jiangsu 210096, China

ARTICLE INFO

Communicated by D. Wu

Keywords:
Attributed graph clustering
Graph representation learning
Contrastive learning

ABSTRACT

Attributed graph clustering is a fundamental task in graph learning field. Because of the high-dimensional node features and the complex non-Euclidean graph structure, it is challenging for attributed graph clustering methods to exploit graph information. Recent studies on graph contrastive learning (GCL) have achieved promising results. However, existing GCL-based methods neither consider a clustering-friendly node representation nor a clustering-oriented loss function, resulting in inferior performance. To this end, we propose NCAGC, a neighborhood contrastive representation learning method for attributed graph clustering task. Specifically, NCAGC constrains the representation learning of similar nodes by a neighborhood contrast module to ensure the compactness in the latent space, thus facilitating the clustering task. Meanwhile, a contrastive self-expression module is present for learning a discriminative self-expression coefficient matrix, which is crucial for the subsequent subspace clustering. Moreover, the two designed modules are trained and optimized jointly, which benefits the node representation learning and clustering to achieve mutual refinement. Extensive experimental results on four attributed graph datasets demonstrate the superiority of NCAGC compared with 16 state-of-the-art methods, which surpasses the sub-optimal method on Cora dataset by 2.1%, 4.3%, and 3.7% in terms of ACC, NMI, and ARI, respectively. Our code and dataset is available at https://github.com/wangtong627/NCAGC-NeuroCom.

1. Introduction

With the rapid development of information technology, graph-structured data gradually plays a critical role in people's daily life. Attributed graph clustering [1–3], which aims at grouping the given nodes into several disjoint clusters without human annotations, is one of the most fundamental tasks in the graph learning field. In practice, considering the high-dimensional node attributes and the complex non-Euclidean graph structures, it is of great challenge for the attributed graph clustering methods to exploit graph data effectively.

Tackling with graph-structured data, traditional clustering methods [4,5] are limited by their poor expressiveness, which degrades the performance of clustering result. Recently, deep models have rapidly developed and widely used in diverse scenarios [6,7], among them, graph neural network (GNN) [8] has attracted considerable attention, which aims to extract node representation into low-dimension and keep the original information of graph structures and node attributes simultaneously. Owing to the powerful graph representation learning capability, various GNN-based attributed graph clustering methods [9–20] have been proposed. Despite these methods achieve considerable performance, the lack of self-supervised information still limits the performance of representation learning, thus degrading the clustering performance. To this end, Wang et al. develop MSGA [15] by integrating multiscale and self-supervised information in the representation learning process. Even though MSGA proves that using self-supervised learning can effectively enhance the quality of node representation, the self-supervised paradigm in MSGA updating the node representation with the pseudo labels may still affect the convergence. Fortunately, contrastive learning (CL) [21–24], a promising paradigm of self-supervised learning, successfully avoids the aforementioned issues since many graph contrastive learning (GCL) methods [25–28] show promising results exploiting graph data. Nevertheless, existing GCL-based methods still suffer from the issues from the following three

* Corresponding author.
E-mail address: yang.list@seu.edu.cn (G. Yang).

https://doi.org/10.1016/j.neucom.2023.126880
Received 13 November 2022; Received in revised form 14 July 2023; Accepted 3 October 2023
Available online 13 October 2023

aspects when dealing with the attributed graph clustering problem. (1) They merely utilize an instance-level contrast for the graph representation learning. This results in some similar samples being pushed away as negative samples, which is inconsistent with the downstream clustering task. (2) They do not consider design a clustering-oriented loss function. Since the representation learning and clustering of these GCL-based methods are separated, the node representation cannot be optimized according to the clustering task, resulting in suboptimal clustering performance. (3) They rely too much on data augmentation engineering. Because of the complexity of graph-structured data, it is uncertain whether the pre-defined graph data augmentation method is suitable for the corresponding data and downstream tasks, which requires extra domain knowledge.

To get rid of aforementioned issues, in this paper, we develop a generic Neighborhood Contrastive representation learning method for Attributed Graph Clustering, namely NCAGC. To be exact, considering the nodes grouped in the same cluster should have similar representations, we propose a Neighborhood Contrast Module (NCM) with a region-level contrastive loss that minimize the distance of similar nodes. In NCM, all the similar nodes are defined as positive samples, leaving others negative. This way, the learned node representation under the same cluster become more compact and clustering-friendly in the latent feature space, which is more conducive to downstream clustering task. Moreover, we design a Contrastive Self-Expression Module (CSEM) to fully exploit the node representation and construct a self-expression coefficient matrix. In CSEM, by minimizing the contrastive self-expression loss, we can constrain the learning of coefficients from the self-representation layer in an instance-level contrast manner. Therefore, with the guide of CSEM, a more discriminative linear combination coefficient is learned, which is later used for subspace clustering. Besides, both NCM and CSEM perform contrastive learning on the original view, so there is no burden of data augmentation. By iteratively training and optimizing all the modules in a unified framework, the node representation learning and the clustering task can facilitate each other.

The main highlights of this paper are summarized as follows:

1. To the best of our knowledge, NCAGC could be the first method that employs contrastive learning for optimizing the learning of a self-expression matrix. Unlike other self-expression-based clustering methods, NCAGC contrasts the node representation before and after the self-expression network with an instance-level contrastive loss, thus obtaining a more discriminative self-expression coefficient matrix for clustering.
2. NCAGC adopts a region-level contrastive loss for learning more compact and clustering-friendly node representations. This is because nodes under the same cluster should have similar representations beneath the latent space. Besides, we verify the effectiveness of this method in the ablation experiments.
3. Unlike other contrastive learning methods, NCAGC employs contrastive learning in the original view, which eliminates the negative impact of inappropriate data augmentation.
4. Extensive experiments over four attributed graph datasets demonstrate the superiority of our proposed NCAGC in terms of 16 state-of-the-art clustering methods. NCAGC surpasses the sub-optimal method on Cora dataset by 2.1%, 4.3%, and 3.7% in terms of ACC, NMI, and ARI, respectively.

2. Related work

In this section, we briefly introduce recent developments in two related topics: attributed graph clustering and graph contrastive learning.

2.1. Attributed graph clustering

Attributed graph clustering, which aims to divide node samples into different disjoint clusters, is an essential task to explore the information embedded in the network-structured data. The past decade saw an upsurge of attributed graph clustering methods which can be roughly divided into three categories in terms of the input graph information, *i.e.*, (1) the method only using node attributes, (2) the method only using graph structures and (3) the method using both node attributes and graph structures.

The methods in the first category only take the node attributes as input and map them into low-dimensional representation for clustering. Typical methods in this category include K-Means [4], DBSCAN [29]. The methods in the second category encode the graph structures into latent space for characterizing the relationship between nodes, which is later used for clustering. Spectral method [5,30] is the usual method in this category, which aims to divide the input graph into several disconnected subgraphs to ensure the nodes in the same sub-graph have high similarity. After that, numerous graph structure-based methods are proposed such as DeepWalk [31] and deep neural graph learning (DNGR) [32]. Although promising results have been achieved, these traditional shallow methods, such as K-Means and Spectral give discouraging results on complex graph-structured data due to the ability to extract node representation.

Benefiting from the ability to capture complex and non-linear node representations, Graph Neural Network (GNN) based clustering methods [9–13,15–18,33,34] have been widely addressed in recent years, which can be grouped in the third category. Typically, these methods apply the traditional clustering to the node embedding extracted by GNN. Kipf et al. make an early attempt that propose graph autoencoder (GAE) and variational graph autoencoder (VGAE) [9] by employing a GCN encoder and a reconstruction decoder to exploit node representation for clustering. For a robust node embedding, Pan et al. propose adversarially regularized graph autoencoder (ARGAE) and adversarially regularized variational graph autoencoder (ARVGAE) [10] by adopting an adversarial training strategy based on [9]. Despite the methods mentioned above achieving promising progress, the inner product decoder they adopted is not learnable, which limits the feature extraction capability, thus affecting the clustering performance. To this end, Salehi et al. extend the previous work [9] and propose graph attention autoencoder (GATE) [11] by utilizing an attention mechanism to reconstruct the node relation. However, these methods generally utilize a two-stage processing pipeline that adopt GNN as a feature extractor and then perform traditional clustering method on the learned embedding. Therefore, the clustering performance is also limited since the overlook of the two process. To get rid of this issue, many end-to-end graph clustering methods, which integrate representation learning and clustering task into a unified framework, have been proposed. Wang et al. design the model by a deep attentional embedding approach called DAEGC [12] to obtain the clustering-oriented node representation by optimizing the clustering distribution and target distribution. Bo et al. further develop a structural deep clustering network (SDCN) [13] that integrates an auto-encoder and a graph convolutional network into a unified framework. Wang et al. propose multi-scale graph attention network (MSGA) [15] which comprises a multi-scale self-expression layer for graph clustering and the pseudo labels to guide the node representation learning. Though the above-mentioned GNN-based attributed graph clustering methods show commendable performance to some extend, they cannot effectively exploit the heterogeneous node embedding, leading to inferior performance. To this end, we introduce the contrastive learning-based method, a promising paradigm of self-supervised learning, for facilitating node representation learning and attributed graph clustering tasks.

2.2. Graph contrastive learning

Recently, contrastive learning (CL) has made considerable achievement [21–23] in representation learning, computer vision, and recommendation systems, owing to its powerful unsupervised learning ability. The goal of CL is to map the raw data into a contrastive representation space where the similarity of positive pairs should be maximized, and that of negative pairs should be minimized [35]. Various works introduce CL into graph learning field, and these works are called graph contrastive learning (GCL) methods. Most of the existing GCL-based methods treat the contrastive instance through the graph augmentation [25–28,36,37]. To be specific, the positive pair is the same instance from two augmented views, while others are defined to be negative. Among them, GRACE [25] proposes a graph contrast representation learning network with graph augmentation approaches and the local node-level instance contrastive loss function. GCA [26] improves the graph augmentation approach with an adaptive method to get better contrast quality, which highlights the important connective structures and adds more noise to the unimportant nodes. Subsequently, DGI [28] make the graph contrast by maximizing the mutual information between the node representation and a summary vector. Based on the InfoNCE loss in [22], GraphCL [27] proposed a new GCL network with various graph augmentation for facilitating the learning of node representation. The above-mentioned works have validated the effectiveness of GCL methods in graph representation learning. however, all of them do not consider neither a clustering-oriented loss function nor node representations, which may limit their performance in clustering.

Different from the above GCL methods, NCAGC makes contributions in the following aspects: (1) Existing works depend on data augmentation to contrast different pairs, however, it is not clear whether the pre-defined data augmentation is suitable for the corresponding tasks, whereas our method adopts another way to contrast without data augmentation. (2) Existing works perform an instance-level contrast on a single node or graph. However, considering that similar samples are pushed away as negative samples, such a contrast schema is not suitable for extracting representations applied to the clustering task. Therefore, our method applies it on a region-level contrast for learning a clustering-friendly node representation. (3) Existing methods aim to learn the node representation with contrastive loss, while NCAGC develops an extra contrastive self-expression loss function for constraining the learning of self-expression coefficient matrix, which is used for subsequent subspace clustering.

3. Methodology

In this section, we formally introduce our proposed NCAGC, where the overall framework is shown in Fig. 1. Specifically, we employ two modules using contrastive learning, *i.e.*, the Neighborhood Contrast Module (NCM) and the Contrastive Self-Expression Module (CSEM). We will describe them in detail in the following.

3.1. Notations

Given an undirected attribute graph $\mathcal{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X})$, where $\mathbf{V} = \{v_1, v_2, \dots, v_N\}$ and $\mathbf{E}$ are the node set and edge set, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is the node attribute matrix with d features and N is the number of nodes and $\mathbf{x}_i \in \mathbb{R}^d$ corresponding to the ith column of matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$. $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix and $\mathbf{A}_{ij} = 1$ iff $(v_i, v_j) \in \mathbf{E}$, i.e. there is an edge between node v_i and v_j, otherwise $\mathbf{A}_{ij} = 0$. $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N]$ is the node representation matrix where $\mathbf{h}_i \in \mathbb{R}^{d^{(j)}}$ corresponding to the ith column of matrix $\mathbf{H} \in \mathbb{R}^{d^{(j)}\times N}$ and $d^{(j)}$ is the dimension of the node representation.

The attributed graph clustering aims to divide the N unlabeled nodes into k disjoint clusters, providing the nodes in the same cluster have high similarity according to their attribute and adjacency information.

3.2. Overall framework

As shown in Fig. 1, our proposed NCAGC contains three joint optimized components: Symmetric Feature Extraction Module, Neighborhood Contrast Module, and Contrastive Self-Expression Module.

- **Symmetric Feature Extraction Module** employs a symmetric GNN to encode the latent node representation from the node attribute and graph structure simultaneously and applies it to downstream clustering tasks.
- **Neighborhood Contrast Module** utilizes a region-level contrastive learning method to improve the quality of extracted node representation by maximizing the similarities of top K nearest neighbor nodes, *i.e.*, positive pairs, and minimizing the similarities of other nodes, *i.e.*, negative pairs.
- **Contrastive Self-Expression Module** is leveraged to help learn a more discriminative self-expression coefficient matrix by contrasting the node representation before and after the reconstruction of the self-expression layer under an instance level.

3.3. Symmetric Feature Extraction Module

As present in Fig. 1, the Symmetric Feature Extraction Module includes a GNN encoder and a symmetric decoder to map the raw node attribute and graph structure into a new low-dimensional space for the attribute graph clustering.

Given the node attribute matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$ and the graph structure matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, and assumed that the encoder network has L layers, the node representation in the ith layer of the encoder $\mathbf{H}^{(i)} \in \mathbb{R}^{d^{(i)}\times N}$ can be formulated as

$$\mathbf{H}^{(i)} = g(\mathbf{H}^{(i-1)}, \mathbf{A}^{(i-1)}|\mathbf{\Omega}^{(i-1)}), \tag{1}$$

where $\mathbf{\Omega}^{(i)}$ is the trainable parameters in the encoder, and $g(\cdot)$ is a nonlinear mapping function that represents the encoding process with arbitrary GNN method. This paper, we employs a two-layer GAT network as the nonlinear mapping encoder. Specifically, the node attribute is considered as the initial node representation, *i.e.*, $\mathbf{H}^{(0)} = \mathbf{X}$, and the output of the encoder is treated as the node representation in the Lth layer of the encoder, *i.e.*, $\mathbf{Z} = \mathbf{H}^{(L)}$.

The decoder is just reversing the encoding process, which means each decoder layer attempts to reverse the process of its corresponding encoder layer. In this way, the node representation can be obtained without supervision. Given the node representation $\mathbf{Z} \in \mathbb{R}^{d^{(L)}\times N}$ and node relation matrix in the Lth layer $\mathbf{A}^{(L)} \in \mathbb{R}^{N\times N}$, and assumed that the decoder network has L layers, the node representation in the ith layer of the decoder $\hat{\mathbf{H}}^{(i)} \in \mathbb{R}^{d^{(i)}\times N}$ can be formulated as

$$\hat{\mathbf{H}}^{(i)} = \hat{g}(\hat{\mathbf{H}}^{(i-1)}, \hat{\mathbf{A}}^{(i-1)}|\hat{\mathbf{\Omega}}^{(i-1)}), \tag{2}$$

where $\hat{\mathbf{\Omega}}^{(i)}$ is the trainable parameter in the decoder and $\hat{g}(\cdot)$ is an nonlinear mapping function in the decoder process. To be consistent with the encoder, the decoder adopts GAT as the nonlinear mapping function. Specifically, the input of the decoder is the node representation after self-expression layer transformation, *i.e.*, $\hat{\mathbf{H}}^{(L)} = \mathbf{ZC}$, where $\mathbf{C} \in \mathbb{R}^{N\times N}$ is the self-expression coefficient matrix. The output of the decoder is treated as the reconstructed node attribute, *i.e.*, $\hat{\mathbf{X}} = \hat{\mathbf{H}}^{(0)}$.

To make sure the node representation can learn more favorable and satisfactory information from both node attributes and graph structures, the node representation reconstruction loss $\mathcal{L}_{rec}$ is defined as

$$\mathcal{L}_{rec} = \min_{\mathbf{\Omega}} \frac{1}{2}\|\mathbf{X} - \hat{\mathbf{X}}\|_F^2. \tag{3}$$

where $\mathbf{\Omega}$ is the trainable parameter in the Symmetric Feature Extraction Module.

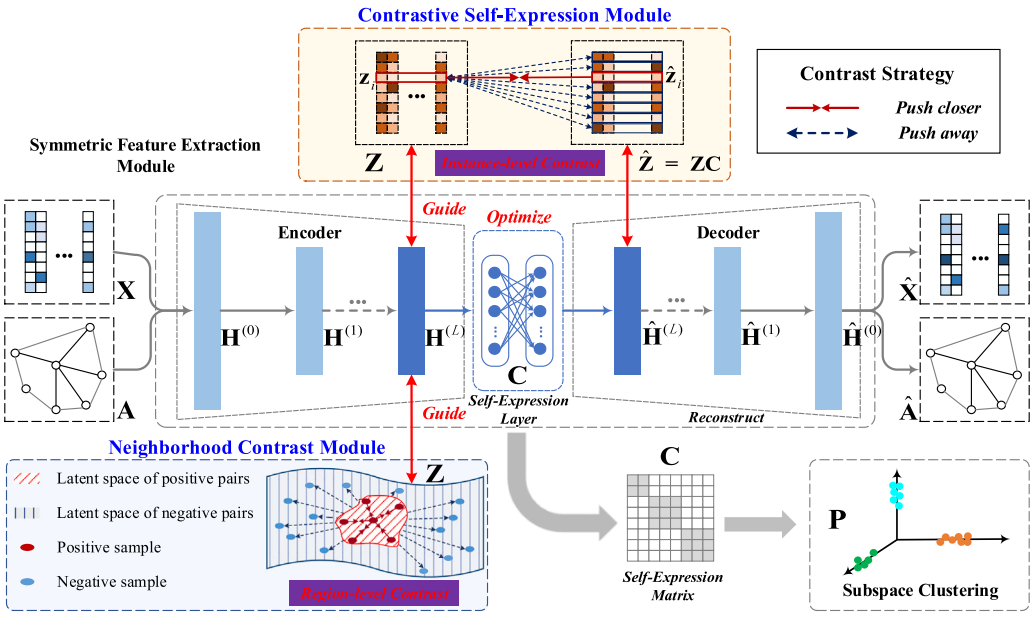


Fig. 1. General Framework of our proposed NCAGC, which consists three modules: (1) Symmetric Feature Extraction Module. (2) Neighborhood Contrast Module. (3) Contrastive Self-Expression Module. $\mathbf{X}$ is the node attribute matrix, $\mathbf{A}$ is the adjacency matrix, $\mathbf{H}^{(L)}$ is the latent node representation in Lth layer of the GNN feature extractor, $\mathbf{C}$ is the self-expression matrix, and $\mathbf{P}$ is the clustering label.

3.4. Neighborhood Contrast Module

The node representation from the Symmetric Feature Extraction Module is not well-designed for the graph clustering task. To obtain node representations that cater to downstream tasks, we propose a Neighborhood Contrast Module.

The goal of attributed graph clustering is to divide the nodes with similar attributes into the same cluster, and the nodes in different clusters have a low similarity. As aforementioned, we can leverage the contrastive learning method, which has been widely adopted in the representation learning field, to 'push closer' the given node representation and similar node representations, meanwhile, 'push away' it from others.

The problem lies in distinguishing whether the node representation is similar to the given or not, in other words, how to define the positive/negative pairs in the contrastive learning process. As the previous works [25–27,36,37] illustrated, graph contrastive learning is executed at instance-level, and positive/negative pairs are constructed by graph data augmentation. However, it is still uncertain what kind of graph data augmentation is suitable for the corresponding clustering tasks since inappropriate augmentation may limit the performance. To get rid of graph data augmentation and consider the node similarity can be described by the distance of their features, we apply a Neighborhood Contrast Module with a region-level contrastive method. In this paper, we obtain the top K similar neighborhoods of the given node by the K-Nearest Neighbor (KNN) method and treat them as positive samples.

To be specific, for the given node representation $\mathbf{z}_i \in \mathbb{R}^{d^{(i)}}$, *i.e.*, the red node in N node samples $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N\}$ and given neighborhood size K in Fig. 2, we can calculate top K similar nodes $\tilde{\mathbf{z}}_j \in \mathbb{R}^{d^{(L)}}$ of the given node $\mathbf{z}_i$ in terms of KNN method and regard them as positive samples, leaving other $N-K$-1 samples to be negative. After that, each positive sample can form a positive pair with the given sample $\mathbf{z}_i$, while each negative sample can form a negative pair with $\mathbf{z}_i$.

The neighborhood contrast loss ℓ_i for a single node $\mathbf{z}_i$ is in the form of

$$\ell_i = \min_{\mathbf{\Omega}} -\log \frac{\sum_{j=1}^{K} \exp(s(\mathbf{z}_i, \tilde{\mathbf{z}}_j))}{\sum_{p\neq i}^{N} \exp(s(\mathbf{z}_i, \mathbf{z}_p))}, \tag{4}$$

where $\mathbf{\Omega}$ is the trainable parameter in the Symmetric Feature Extraction Module, K is the pre-defined neighborhood size, $\tilde{\mathbf{z}}_j$ is the neighborhood

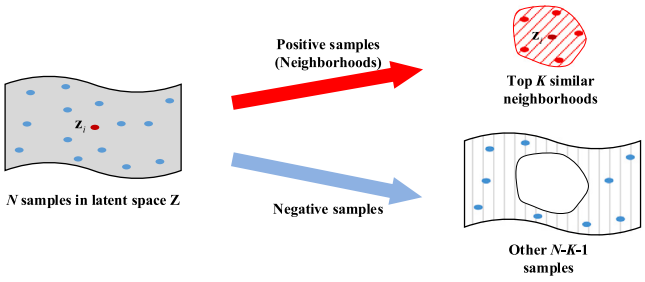


Fig. 2. The selection of positive and negative samples in Contrastive Neighborhood Module.

of $\mathbf{z}_i$ calculated by KNN, $\sum_{j=1}^{K} \exp(s(\mathbf{z}_i, \tilde{\mathbf{z}}_j))$ denotes the total similarity of positive pairs, and the pair-wise similarity $s(\mathbf{a}, \mathbf{b})$ is measured by cosine distance, *i.e.*,

$$s(\mathbf{a}, \mathbf{b}) = \frac{(\mathbf{a})(\mathbf{b})^{\top}}{\|\mathbf{a}\|\|\mathbf{b}\|}, \tag{5}$$

where $\mathbf{a}$ and $\mathbf{b}$ are two arbitrary vectors.

Taking all the nodes in graph $\mathcal{G}$ into account, we have the following neighborhood contrast loss $\mathcal{L}_{nbr}$, *i.e.*,

$$\mathcal{L}_{nbr} = \min_{\mathbf{\Omega}} \sum_{i=1}^{N} \ell_i. \tag{6}$$

3.5. Contrastive Self-Expression Module

Even though the aforementioned modules could output a high-quality node representation $\mathbf{Z}$, there is no guarantee that it can be well used to construct the self-expression coefficient matrix $\mathbf{C}$ in the deep subspace clustering process. Therefore, we propose the Contrastive Self-Expression Module for learning a more discriminative self-expression coefficient matrix.

The deep subspace clustering method learn the self-expressiveness by introducing a self-expression layer, *i.e.*, a fully connected layer without a bias between the encoder and decoder. Precisely, for a given node representation $\mathbf{z}_i$, the self-expression layer can calculate the linear

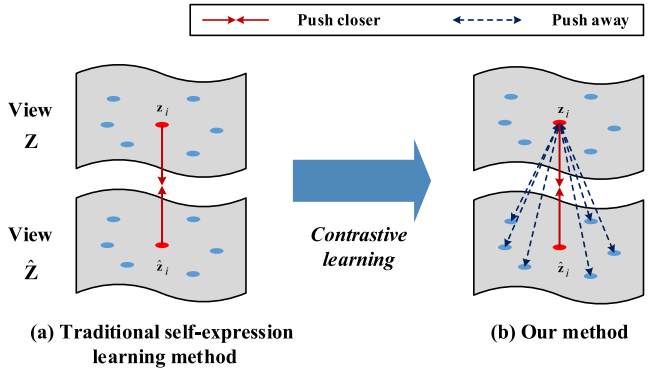


Fig. 3. The contrastive method in self-expression layer.

combination of other nodes $\mathbf{z}_{j,j\neq i}$ to express the reconstructed node representation $\hat{\mathbf{z}}_i$ as

$$\hat{\mathbf{z}}_i = \sum_{i\neq j} c_{ij}\mathbf{z}_j. \tag{7}$$

Existing deep subspace clustering based methods simply optimize $\mathbf{C}$ by minimize the self-expression loss $\mathcal{L}_{se}$, *i.e.*,

$$\mathcal{L}_{se} = \min_{\mathbf{\Omega}} \|\mathbf{Z} - \mathbf{ZC}\|_F^2, \tag{8}$$

where $\mathbf{C} \in \mathbb{R}^{N\times N}$ is the self-expression coefficient matrix which describes the self-expressiveness and consists of each linear combination coefficient c_{ij}.

However, the conventional self-expression loss $\mathcal{L}_{se}$ only considers the corresponding node representation before and after the reconstruction of the self-expression layer, *i.e.*, $\mathbf{z}_i$ and $\hat{\mathbf{z}}_i$, neglecting the information of other nodes, which degrades the learning quality of $\mathbf{C}$ and limits the clustering performance. To this end, we propose the Contrastive Self-Expression Module, which adopts an instance-level contrastive learning method by 'pushing closer' the given node representation $\mathbf{z}_i$ and the corresponding node representation $\hat{\mathbf{z}}_i$, and 'push away' it from other nodes. To our knowledge, this could be the first attempt to improve the quality of the learned self-expression matrix with a contrastive learning strategy.

Specifically, as shown in Fig. 3, we have the attributed graph views before and after the self-expression layer as $\mathbf{Z}$ and $\hat{\mathbf{Z}}$, respectively. For a given node $\mathbf{z}_i$ in $\mathbf{Z}$, it can form a positive pair with the corresponding node $\hat{\mathbf{z}}_i$ in $\hat{\mathbf{Z}}$, leaving other N-1 nodes in $\hat{\mathbf{Z}}$ to be negative pairs. The contrastive self-expression loss ℓ_i for a single node $\mathbf{z}_i$ is in the form of

$$\ell_i = \min_{\mathbf{\Omega},\mathbf{C}} -\log \frac{\exp(s(\mathbf{z}_i,\hat{\mathbf{z}}_i))}{\sum_{j=1}^{N}\exp(s(\mathbf{z}_i,\hat{\mathbf{z}}_j))}, \tag{9}$$

where $\hat{\mathbf{z}}_i = \sum_{i\neq j} c_{ij}\mathbf{z}_j$ denotes the reconstructed node representation in $\hat{\mathbf{Z}}$, $s(\mathbf{a},\mathbf{b})$ is the pair-wise similarity.

Taking all the nodes in graph $\mathcal{G}$ into account, we have the proposed contrastive self-expression loss $\mathcal{L}_{cse}$, *i.e.*,

$$\mathcal{L}_{cse} = \min_{\mathbf{\Omega},\mathbf{C}} \sum_{i=1}^{N} \ell_i. \tag{10}$$

As deep subspace clustering method [38], the self-expression coefficient matrix $\mathbf{C}$ is leveraged to construct the affinity matrix $\mathbf{\Lambda}$ for spectral clustering. Mathematically, we can have the following regularization term $\mathcal{L}_{coef}$, *i.e.*,

$$\mathcal{L}_{coef} = \min_{\mathbf{C}} \|\mathbf{C}\|_p, \tag{11}$$

where $\|\cdot\|_p$ denotes an arbitrary matrix norm, in this paper, we adopt ℓ_2 norm.

3.6. Optimize

Combining Eq. (3), Eq. (6), Eq. (10) and Eq. (11), NCAGC optimizes the following loss function

$$\mathcal{L}_{total} = \min_{\mathbf{\Omega},\mathbf{C}} \mathcal{L}_{rec} + \lambda_1 \cdot \mathcal{L}_{nbr} + \lambda_2 \cdot \mathcal{L}_{cse} + \lambda_3 \cdot \mathcal{L}_{coef}, \tag{12}$$

where λ_1, λ_2, λ_3 are three trade-off parameters that keep the balance of different loss functions. By optimizing Eq. (12), the node representation will be extracted with neighborhood information, meanwhile, the attributed graph clustering and node representation learning process are seamlessly connected. Once the network is trained, we can use the optimized $\mathbf{C}$ to construct $\mathbf{\Lambda}$ for spectral clustering and get clustering label $\mathbf{P}$. In this paper, we adopt the affinity matrix constructing method in [39]. Algorithm 1 shows the optimizing process of the proposed NCAGC.

Algorithm 1 NCAGC

Input: Graph dataset $\mathcal{G}$ with feature matrix $\mathbf{X}$ and adjacency matrix $\mathbf{A}$.
Parameter: Trade-off parameters λ_1, λ_2, λ_3, neighborhood size K, cluster numbers k, training epoches T, iterator t, learning rate lr.

```
Initialize the self-expression matrix C with 1 × 10^−4;
while t < T do
    Calculate the selected neighborhood z̃_j with KNN;
    Joint train the overall framework by Eq. (12);
    Update C from the optimized network;
    Obtain affinity matrix Λ using C;
    Spectral clustering on Λ;
end while
```

Output: Clustering label $\mathbf{P}$.

4. Experiment

In this section, we conduct extensive experiments to fully analyze the effectiveness of our proposed NCAGC and the proposed Neighborhood Contrast Module and Contrastive Self-Expression Module.

4.1. Benchmark datasets

To evaluate the performance of our proposed NCAGC, we conducted the following experiment on four attributed graph datasets (Cora, Citeseer, Wiki, and ACM).

To be specific, Cora [40] and Citeseer [41] are citation networks where nodes denote publications and are connected if cited by another. The node attribute of Cora and Citeseer datasets is described by a spare bag of word feature vectors. Wiki [42] is a webpage network where nodes denote webpages and are connected if linked by one another. TF-IDF weighted vectors describe the node attribute of the Wiki dataset. ACM [43] is a paper network where nodes denote papers, which are connected if they are written by the same author. The node attribute of the ACM dataset is the bag of words of keywords. The details and statistics of these datasets are shown in Table 1.

4.2. Clustering metrics

Similar to the previous work [15], we leverage three widely-used clustering metrics to evaluate the generic and effectiveness of our proposed NCAGC, namely Accuracy (ACC) [44], Normalized Mutual Information (NMI) [45], and Adjusted Rand Index (ARI) [46]. For all these metrics, a better score denotes a better clustering performance.

Table 1
The statistics of the datasets.

Datasets	Nodes	Relation types	Edges	Feature types	Attributes	Classes
Cora	2,708	Citation Network	5,429	Bag of words of keywords	1,433	7
Citeseer	3,327	Citation Network	4,732	Bag of words of keywords	3,703	6
Wiki	2,405	Webpage Network	17,981	TF-IDF	4,973	17
ACM	3,025	Paper Network	29,281	Bag of words of keywords	1,870	3

4.3. Baseline methods

To better assess the clustering performance of our proposed NCAGC over the aforementioned datasets, we choose 16 state-of-the-art basis methods as competitors. They can be generally divided into the following three categories:

(1) Method only using node attributes, which adopts the raw node attribute as input.

- K-means [4] is a classical shallow clustering method, which alternately updates the location of the cluster center and the distance of the sample to the cluster center.

(2) Method only using graph structures, which adopts the raw topological graph structure as input.

- SC [5] is a classical shallow clustering method based on graph theory, which takes the node adjacency matrix as the similarity matrix.

(3) Method using both node attributes and graph structures, which takes node attributes and graph structures as input.

- GAE [9] is an unsupervised graph embedding method with a GCN-based encoder and an inner product decoder.
- VGAE [9] is the variational version of GAE.
- ARGAE [10] is an adversarial regularized unsupervised graph embedding method with generative adversarial methods.
- ARVGAE [10] is the variational version of ARGAE.
- MGAE [14] is a marginalized unsupervised graph clustering method, which employs a mapping loss to optimize the unsupervised graph feature extraction process and adopts spectral clustering to obtain clustering labels.
- GATE [11] employs a node attribute reconstruct loss and graph structure reconstruct loss to optimize node embedding.
- DAEGC [12] is an end-to-end graph clustering method employing attention mechanism in node representation learning and adopts a clustering loss for node clustering.
- SDCN [13] is an end-to-end graph clustering method employing both auto-encoder and graph convolutional network during the node representation process.
- DFCN [47] introduces an information fusion module to enhance the performance of SDCN.
- AGC [48] utilizes adaptive graph convolution layers for different datasets.
- GALA [49] proposes a symmetric graph convolutional autoencoder for graph clustering, which produces a low-dimensional graph representation for clustering.
- MSGA [15] introduces multi-scale self-expression layers and self-supervised method improving the clustering performance.
- GRACE [25] is an unsupervised graph embedding method with graph contrastive learning, which generates two views for comparison using graph augmentation.
- GCA [26] introduces an adaptive graph data augmentation based on GRACE.

4.4. Parameter setting

The proposed NCAGC and baseline methods are implemented on a machine equipped with an NVIDIA RTX 3080 GPU. The deep learning environment for our proposed NCAGC is Pytorch and PyTorch Geometric.

For all baseline methods, the parameters are set according to the corresponding papers for the best performance. For NCAGC, we employ a two-layer GAT network [50] as encoder and a symmetrical decoder with Adam optimizer. The learning rate lr is set as 0.0001 on Cora, Citeseer, and Wiki and 0.0005 on ACM. The dimension of the latent layer for the encoder is searched in the range of $\{256, 512, 1024, 2048\}$. Specifically, the dimension is set as $[1024, 512]$ on Cora, $[1024, 1024]$ on Citeseer, $[1024, 512]$ on Wiki and $[1024, 512]$ on ACM, respectively. We use prelu as an activation function, and the number of neighbors is set as 10. To obtain a balance of different loss functions and get optimal results, we turn the trade-off parameters λ_1, λ_2, λ_3 in the $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. Specially, the trade-off parameters λ_1, λ_2, λ_3 are set as $[10, 10, 10]$ on Cora, $[10, 1, 1]$ on Citeseer, $[10, 1, 10]$ on Wiki and $[100, 100, 1000]$ on ACM, respectively. We train NCAGC for 400 iterations on Cora, 200 on Citeseer and ACM, and 300 on Wiki, respectively. The detailed parameter settings on NCAGC are shown in Table 2.

4.5. Experiment results

To evaluate the performance of NCAGC, we run the aforementioned 16 methods ten times and report the average score to avoid randomness. The clustering performance is summarized in Table 3, where the best score is represented by bold value and the sub-optimal score is underlined.

As shown in Table 3, it can be observed that our proposed NCAGC achieves competitive performance compared with all the baseline methods according to three clustering metrics, which demonstrates the effectiveness of our proposed method. To be specific, we can have the following interesting observations:

- The proposed NCAGC and other GCN-based methods (GAE, ARGAE, GATE, DAEGC, SDCN, DFCN, AGC, GALA, MSGA, GRACE, and GCA) show superiority over K-Means and Spectral method, which demonstrates that the method with both node attributes and graph structures performs better than only using one of them. This is because GCN based method can exploit node attributes and graph structures simultaneously with a multi-layer nonlinear network.
- The proposed NCAGC achieves better clustering performance than some representative graph auto-encoder-based methods (GAE, ARGAE, and GATE). Especially for the Cora dataset, NCAGC surpasses GAE by about 51.9%, 83.3%, and 154.3% and GATE by about 15.9%, 14.4% and 22.3% on ACC, NMI, and ARI, respectively. This result indicates that the contrastive strategy of the Neighborhood Contrast Module employed by considering 'push closer' neighbor nodes, *i.e.*, positive pairs, can help to improve the learned node representation quality.
- In some cases, the clustering performance of the method using graph contrastive learning (GRACE and GCA) is inferior to the clustering-oriented method (DAEGC, SDCN, DFCN, AGC, GALA, MSGA, and NCAGC). This can be concluded that NCAGC integrates node representation learning and clustering into a unified framework, which greatly helps the network learn a clustering-oriented representation. In contrast, GRACE and GCA perform node representation learning and clustering in two separate steps, which limits the performance.

Table 2
Parameter settings on CAGC.

Datasets	Learning rate	Encoder dimension	GNN	Activation function	Neighborhood size	λ_1	λ_2	λ_3
Cora	0.0001	1433 - 1024 - 512	GAT	prelu	10	10	10	10
Citeseer	0.0001	3703 - 1024 - 1024	GAT	prelu	10	100	1	1
Wiki	0.0001	4973 - 1024 - 512	GAT	prelu	10	10	1	10
ACM	0.0005	1870 - 1024 - 512	GAT	prelu	10	100	100	1,000

Table 3
Clustering performance of NCAGC on graph datasets.

Methods	Input	Cora			Citeseer			Wiki			ACM		
		ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
K-Means	Feature	0.492	0.321	0.230	0.540	0.305	0.278	0.417	0.440	0.150	0.673	0.324	0.306
Spectral	Graph	0.367	0.127	0.031	0.239	0.056	0.010	0.220	0.182	0.015	0.368	0.007	0.006
GAE	Both	0.502	0.329	0.217	0.412	0.183	0.189	0.173	0.119	0.160	0.845	0.553	0.594
VGAE	Both	0.559	0.384	0.254	0.447	0.260	0.205	0.286	0.302	0.263	0.847	0.556	0.601
MGAE	Both	0.684	0.511	0.444	0.660	0.412	0.413	0.514	0.485	0.350	0.881	0.621	0.687
ARGAE	Both	0.640	0.449	0.352	0.573	0.350	0.341	0.380	0.344	0.112	0.843	0.545	0.606
ARVGAE	Both	0.638	0.450	0.374	0.544	0.261	0.245	0.386	0.338	0.107	0.845	0.547	0.603
GATE	Both	0.658	0.527	0.451	0.616	0.401	0.381	0.465	0.428	0.316	0.863	0.574	0.624
DAEGC	Both	0.704	0.528	0.496	0.672	0.397	0.410	0.482	0.448	0.330	0.874	0.591	0.633
SDCN	Both	0.712	0.535	0.506	0.659	0.387	0.401	0.385	0.375	0.285	0.904	0.683	0.731
DFCN	Both	0.740	0.561	0.527	0.695	0.439	0.455	0.416	0.400	0.332	0.909	0.694	0.749
AGC	Both	0.689	0.537	0.493	0.670	0.411	0.415	0.477	0.453	0.325	0.893	0.653	0.710
GALA	Both	0.746	0.577	0.532	0.693	**0.441**	0.446	0.525	0.480	**0.366**	0.908	0.688	0.747
MSGA	Both	0.747	0.578	0.519	0.698	0.415	0.433	0.522	0.481	0.323	0.911	0.692	0.752
GRACE	Both	0.671	0.530	0.485	0.651	0.395	0.402	0.475	0.430	0.308	0.881	0.651	0.695
GCA	Both	0.696	0.538	0.491	0.643	0.393	0.405	0.477	0.435	0.315	0.889	0.653	0.698
NCAGA (Ours)	Both	**0.764**	**0.603**	**0.552**	**0.707**	0.438	**0.463**	**0.530**	**0.485**	0.336	**0.917**	**0.711**	**0.769**

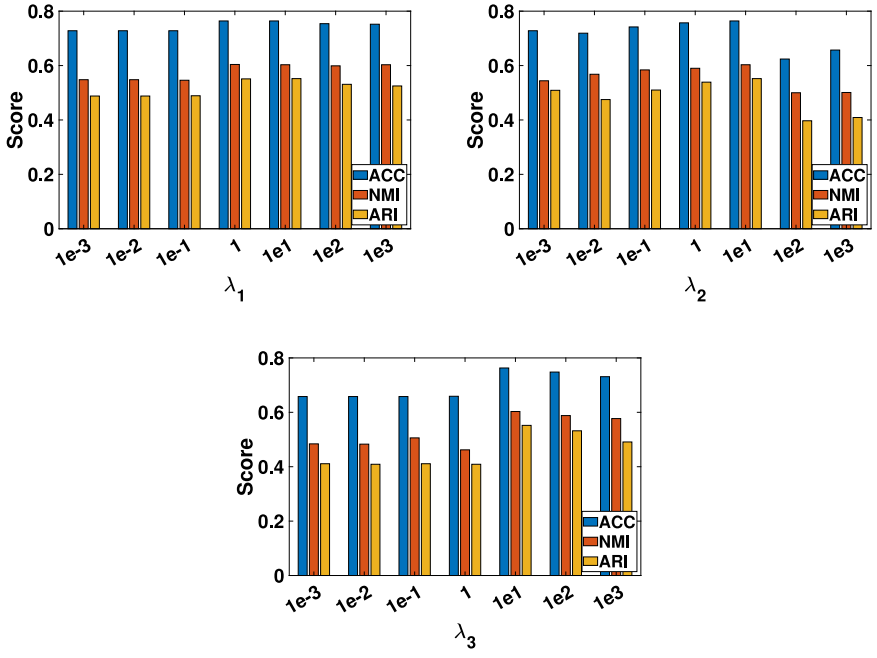


Fig. 4. Influence of individual values on ACC, NMI and ARI of Cora dataset.

- The proposed NCAGC significantly outperforms all the methods in terms of ACC metric and achieves the top 2 scores in terms of NMI and ARI metrics. Taking the clustering performance on the Cora dataset as an example, NCAGC surpasses the sub-optimal method by 2.1%, 4.3%, and 3.7% in terms of ACC, NMI, and ARI, respectively. The remarkable result verifies the powerful clustering ability of NCAGC, which benefits from the incorporation of the Neighborhood Contrast Module and the Contrastive Self-Expression Module.

4.6. Parameter analysis

There are three trade-off parameters in our proposed NCAGC, namely λ_1, λ_2 and λ_3, which denote the weight of neighborhood contrast loss $\mathcal{L}_{nbr}$, contrastive self-expression loss $\mathcal{L}_{cse}$ and regularization of self-expression coefficient $\mathcal{L}_{coef}$, respectively. To this end, we take the Cora dataset as an example and conduct the experiment to verify the effectiveness of different loss functions of NCAGC. Besides, we also investigate the sensitivity of neighborhood size K in the Neighborhood Contrast Module.

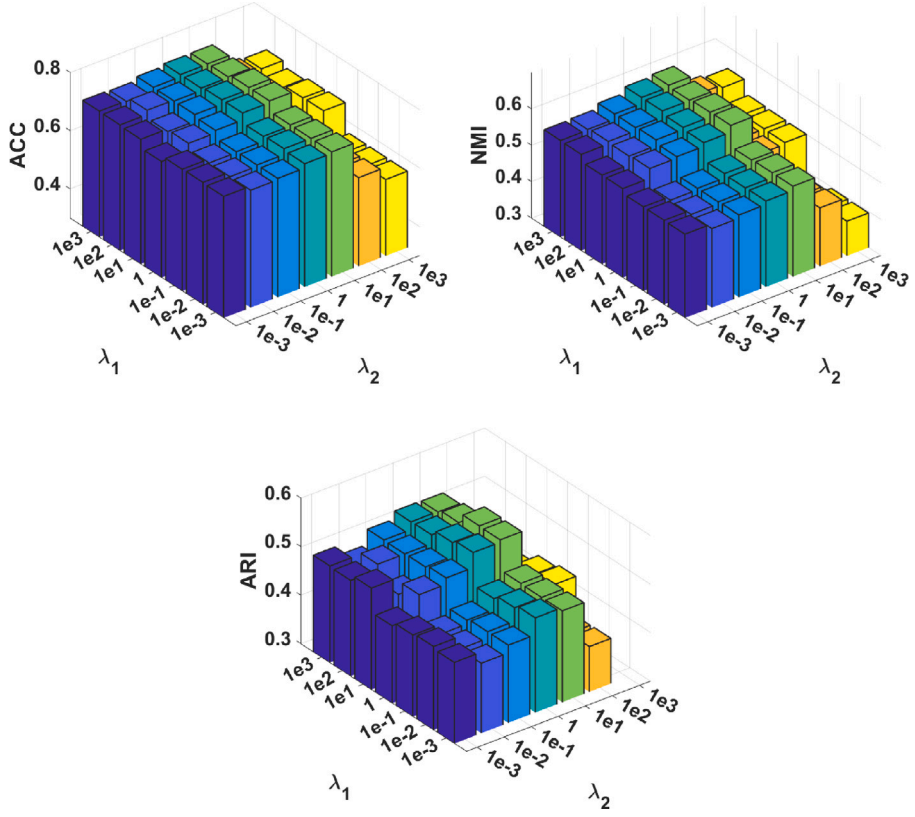


Fig. 5. Influence of λ_1 and λ_2 values on ACC, NMI and ARI of Cora dataset.

(1) The effect of trade-off parameters: We first analyze the influence of individual trade-off parameters via tuning the under-tested parameter from 10^{-3} to 10^3 and fixing other parameters as in Table 2. Fig. 4 shows the changing of the clustering performance by individual parameters. We notice that the score of clustering metrics keeps stable as λ_1 varied. It could be inferred that the contrastive method of the Neighborhood Contrast Module, which 'push closer' the node representation of neighbors has strong robustness. Furthermore, we can observe that big λ_2 and small λ_3 will degrade the clustering performance since λ_2 and λ_3 are related to the learning of the self-expression coefficient, which is crucial to clustering. The result shows that the self-expression layer of NCAGC could learn a more discriminative self-expression coefficient when λ_2 is under 10 or over 10.

Besides, to further analyze the effect of a comprehensive combination of these trade-off parameters, we vary the under-tested parameter from 10^{-3} to 10^3 at the same time. From Fig. 5, we can observe that the clustering performance is slightly improved as λ_2 varies from 10^{-3} to 10^3 and dramatically declines since λ_2 increases to 10^3. The result can be inferred that the contrastive self-expression loss $\mathcal{L}_{cse}$ becomes very large as λ_2 varies, which makes the overall loss function unbalanced and then degrades the performance.

(2) The effect of neighborhood size: To further explore the effectiveness and robustness of our proposed Neighborhood Contrast Module, we vary the size of node neighborhood K in the range of $\{3, 5, 7, 10, 15, 20, 30\}$ while other parameters unchanged and report the clustering result of three datasets in Table 4. We can observe that the clustering results in terms of ACC, NMI, and ARI fluctuated by 0.4%, 0.8%, and 1.2% on the Cora dataset, 0.4%, 0.7%, and 1.1% on Citeseer dataset and 0.8%, 3.1% and 2.3% on ACM dataset. As reported in Table 4, we find NCAGC achieves the best clustering performance when K=10 over several datasets. Besides, this can prove that our proposed Neighborhood Contrast Module can steadily improve performance.

4.7. Ablation study

To validate the effectiveness of different strategies in NCAGC, we carry out three ablation studies on Cora, Citeseer, and Wiki datasets.

(1) Ablation study on Neighborhood Contrast Module: Particularly, we compare NCAGC and NCAGC without using Neighborhood Contrast Module (**termed NCAGC w/o NBR**) by removing neighborhood contrast loss $\mathcal{L}_{nbr}$. In this scenario, the total loss function of NCAGC w/o NBR can be described by $\mathcal{L}_{rec} + \mathcal{L}_{cse} + \mathcal{L}_{coef}$, where $\mathcal{L}_{rec}$ is the loss of node representation reconstruction, $\mathcal{L}_{cse}$ is the loss of self-expression employed instance contrastive learning method and $\mathcal{L}_{coef}$ is the regularization term. The result of the ablation study on the Neighborhood Contrast Module is shown in Table 5; comparing with NCAGC w/o NBR, we can observe that NCAGC outperforms Cora, Citeseer and ACM datasets by 2.8%, 3.5% and 1.6% according to ACC, by 4.3%, 5.0% and 4.3% according to NMI and by 6.7%, 7.2% and 2.9% according to ARI. This is because using the Neighborhood Contrast Module can better extract node representation and help to improve the clustering performance.

(2) Ablation study on Contrastive Self-Expression Module: Furthermore, we evaluate the effectiveness of Contrastive Self-Expression Module by using self-expression loss $\mathcal{L}_{se}$ instead of contrastive self-expression loss $\mathcal{L}_{cse}$ in NCAGC (**termed NCAGC w/o CSE**). To be specific, the traditional self-expression loss $\mathcal{L}_{se}$ can be presented by Eq. (8). In this scenario, the total loss function of NCAGC w/o CSE can be described by $\mathcal{L}_{rec} + \mathcal{L}_{nbr} + \mathcal{L}_{se} + \mathcal{L}_{coef}$, where $\mathcal{L}_{rec}$ is the loss of node representation reconstruction, $\mathcal{L}_{nbr}$ is the neighborhood contrast loss, $\mathcal{L}_{se}$ is the loss of self-expression, and $\mathcal{L}_{coef}$ is the regularization term. The result of the ablation study on the Contrastive Self-Expression Module is shown in Table 5. We can observe that it helps improve the clustering performance on Cora, Citeseer, and ACM datasets by 11.3%, 9.8%, and 25.6% in terms of ACC, by 10.4%, 10.6%, and 77.7% in terms of NMI and by 17.7%, 21.5%, and 117.8% in terms of ARI. This result demonstrates that the contrastive self-expression loss $\mathcal{L}_{cse}$ taking

Table 4
Parameter analysis on the size of neighborhood.

Size of K	Cora			Citeseer			ACM		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
$K = 3$	0.762	0.601	0.550	0.704	0.433	0.457	0.912	0.697	0.756
$K = 5$	0.761	0.598	0.547	0.705	0.434	0.458	0.911	0.692	0.753
$K = 7$	0.762	0.597	0.550	0.706	0.437	0.460	0.911	0.694	0.755
$K = 10$	0.764	0.603	0.552	0.707	0.437	0.462	0.917	0.711	0.769
$K = 15$	0.763	0.601	0.552	0.706	0.436	0.461	0.909	0.690	0.750
$K = 20$	0.761	0.602	0.546	0.704	0.437	0.460	0.916	0.702	0.766
$K = 30$	0.762	0.602	0.550	0.704	0.434	0.458	0.909	0.690	0.749

Table 5
Ablation study of the model.

Strategies	Cora			Citeseer			ACM		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
NCAGC w/o NBR	0.742	0.578	0.517	0.683	0.417	0.432	0.902	0.682	0.747
NCAGC w/o CSE	0.685	0.546	0.469	0.643	0.395	0.380	0.730	0.400	0.353
NCAGC w/o ATT	0.736	0.560	0.501	0.703	0.426	0.455	0.907	0.681	0.745
NCAGC	0.764	0.603	0.552	0.707	0.438	0.463	0.917	0.711	0.769

the negative sample information into account greatly helps to learn a more discriminative self-expression matrix and improve the clustering performance.

Besides, we find that the Contrastive Self-Expression Module could bring more impressive clustering performance compared with Neighborhood Contrast Module. This is because the contrastive self-expression loss $\mathcal{L}_{cse}$ can constrain the learning of the self-expression matrix directly. However, the neighborhood contrast loss $\mathcal{L}_{nbr}$ is used to constrain the learning of node representation and then help to learn a better self-expression matrix indirectly.

(3) Ablation study on attention mechanism: Since the feature extractor in Symmetric Feature Extraction Module is flexible. We also conduct the ablation study about the effect of the attention mechanism by using GCN instead of GAT for the feature extraction process (**termed NCAGC w/o ATT**). As reported in Table 5, we can observe that NCAGC using GAT as the feature extractor surpasses NCAGC w/o ATT on Cora, Citeseer, and ACM datasets by 3.6%, 0.5%, and 1.1% in terms of ACC, by 7.6%, 2.6% and 4.3% in terms of NMI and by 10.0%, 1.5%, and 3.1% in terms of ARI. And it demonstrates that GAT can better encode the node attribute and optimize the node relation with the attention mechanism, which significantly helps to improve the quality of node representation.

5. Conclusion

This paper proposes an end-to-end framework for attributed graph clustering called NCAGC. To learn a clustering-friendly node representation, we design a Neighborhood Contrast Module by maximizing the similarity of neighbor nodes. Besides, for a more discriminative self-expression matrix, we develop a Contrastive Self-Expression Module by contrasting the node representation before and after the reconstruction of the self-expression layer. We show that our proposed method achieves state-of-the-art performance compared with 16 clustering methods over four attributed graph datasets.

CRediT authorship contribution statement

Tong Wang: Methodology, Software, Writing – original draft. **Junhua Wu:** Supervision, Writing – review & editing. **Yaolei Qi:** Writing – review & editing. **Xiaoming Qi:** Writing – review & editing. **Juwei Guan:** Validation. **Yuan Zhang:** Validation. **Guanyu Yang:** Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] H. Cai, V.W. Zheng, K.C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1616–1637.
[2] J. Cheng, Q. Wang, Z. Tao, D. Xie, Q. Gao, Multi-view attribute graph convolution networks for clustering, in: International Joint Conference on Artificial Intelligence, IJCAI, 2020, pp. 2973–2979.
[3] Q. Gao, W. Xia, Z. Wan, D. Xie, P. Zhang, Tensor-SVD based graph learning for multi-view subspace clustering, in: AAAI Conference on Artificial Intelligence, AAAI, 2020, pp. 3930–3937.
[4] J.A. Hartigan, M.A. Wong, Algorithm AS 136: A k-means clustering algorithm, J. R. Stat. Soc. 28 (1) (1979) 100–108.
[5] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Neural Information Processing Systems, NeurIPS, 2001, pp. 849–856.
[6] W. Huang, L. Zhang, H. Wu, F. Min, A. Song, Channel-equalization-HAR: A light-weight convolutional neural network for wearable sensor based human activity recognition, IEEE Trans. Mob. Comput. (2022) 1, http://dx.doi.org/10.1109/TMC.2022.3174816.
[7] W. Huang, L. Zhang, S. Wang, H. Wu, A. Song, Deep ensemble learning for human activity recognition using wearable sensors via filter activation, ACM Trans. Embed. Comput. Syst. 22 (1) (2023) 15:1–15:23.
[8] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, ICLR, 2017.
[9] T.N. Kipf, M. Welling, Variational graph auto-encoders, CoRR abs/1611.07308, 2016.
[10] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, Adversarially regularized graph autoencoder for graph embedding, in: J. Lang (Ed.), International Joint Conference on Artificial Intelligence, IJCAI, 2018, pp. 2609–2615.
[11] A. Salehi, H. Davulcu, Graph attention auto-encoders, in: International Conference on Tools with Artificial Intelligence, ICTAI, 2020, pp. 989–996.
[12] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, C. Zhang, Attributed graph clustering: A deep attentional embedding approach, in: S. Kraus (Ed.), International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 3670–3676.
[13] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, P. Cui, Structural deep clustering network, in: The Web Conference, WWW, 2020, pp. 1400–1410.
[14] C. Wang, S. Pan, G. Long, X. Zhu, J. Jiang, MGAE: Marginalized graph autoencoder for graph clustering, in: Conference on Information and Knowledge Management, CIKM, 2017, pp. 889–898.

[15] T. Wang, J. Wu, Z. Zhang, W. Zhou, G. Chen, S. Liu, Multi-scale graph attention subspace clustering network, Neurocomputing 459 (2021) 302–314.
[16] W. Xia, Q. Wang, Q. Gao, X. Zhang, X. Gao, Self-supervised graph convolutional network for multi-view clustering, IEEE Trans. Multim. 24 (2022) 3182–3192.
[17] W. Xia, S. Wang, M. Yang, Q. Gao, J. Han, X. Gao, Multi-view graph embedding clustering network: Joint self-supervision and block diagonal representation, Neural Netw. 145 (2022) 1–9.
[18] H. Xu, W. Xia, Q. Gao, J. Han, X. Gao, Graph embedding clustering: Graph attention auto-encoder with cluster-specificity distribution, Neural Netw. 142 (2021) 221–230.
[19] W. Xia, Q. Gao, Q. Wang, X. Gao, C. Ding, D. Tao, Tensorized bipartite graph learning for multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2023) 5187–5202.
[20] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, D. Tao, Enhanced tensor RPCA and its application, IEEE Trans. Pattern Anal. Mach. Intell. 43 (6) (2021) 2133–2140.
[21] K. He, H. Fan, Y. Wu, S. Xie, R.B. Girshick, Momentum contrast for unsupervised visual representation learning, in: Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 9726–9735.
[22] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, Vol. 119, ICML, 2020, pp. 1597–1607.
[23] W.V. Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, L.V. Gool, SCAN: Learning to classify images without labels, in: European Computer Vision Conference, Vol. 12355, ECCV, 2020, pp. 268–285.
[24] G. Sheng, Q. Wang, C. Pei, Q. Gao, Contrastive deep embedded clustering, Neurocomputing 514 (2022) 13–20.
[25] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Deep graph contrastive representation learning, CoRR abs/2006.04131, 2020.
[26] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in: J. Leskovec, M. Grobelnik, M. Najork, J. Tang, L. Zia (Eds.), The web conference, 2021, pp. 2069–2080.
[27] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, in: Annual Conference on Neural Information Processing Systems, NeurIPS, 2020.
[28] P. Velickovic, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep graph infomax, CoRR abs/1809.10341, 2018.
[29] M. Ester, H. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: International Conference on Knowledge Discovery and Data Mining, KDD, 1996, pp. 226–231.
[30] W. Chen, G. Feng, Spectral clustering: A semi-supervised approach, Neurocomputing 77 (1) (2012) 229–242.
[31] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: online learning of social representations, in: International Conference on Knowledge Discovery and Data Mining, KDD, 2014, pp. 701–710.
[32] S. Cao, W. Lu, Q. Xu, Deep neural networks for learning graph representations, in: AAAI Conference on Artificial Intelligence, AAAI, 2016, pp. 1145–1152.
[33] L. Liu, Z. Kang, J. Ruan, X. He, Multilayer graph contrastive clustering network, Inform. Sci. 613 (2022) 256–267.
[34] H. Liao, J. Hu, T. Li, S. Du, B. Peng, Deep linear graph attention model for attributed graph clustering, Knowl.-Based Syst. 246 (2022) 108665.
[35] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: Conference on Computer Vision and Pattern Recognition, CVPR, 2006, pp. 1735–1742.
[36] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, in: Annual Conferenceon Neural Information Processing Systems, NeurIPS, 2020.
[37] W. Xia, T. Wang, Q. Gao, M. Yang, X. Gao, Graph embedding contrastive multimodal representation learning for clustering, IEEE Trans. Image Process. 32 (2023) 1170–1183.
[38] P. Ji, T. Zhang, H. Li, M. Salzmann, I.D. Reid, Deep subspace clustering networks, in: Annual Conference on Neural Information Processing Systems, NeurIPS, 2017, pp. 24–33.
[39] P. Ji, M. Salzmann, H. Li, Efficient dense subspace clustering, in: Winter Conference on Applications of Computer Vision, 2014, pp. 461–468.
[40] A. McCallum, K. Nigam, J. Rennie, K. Seymore, Automating the construction of internet portals with machine learning, Inf. Retr. 3 (2) (2000) 127–163.
[41] C.L. Giles, K.D. Bollacker, S. Lawrence, CiteSeer: An automatic citation indexing system, in: International Conference on Digital Libraries, 1998, pp. 89–98.
[42] C. Yang, Z. Liu, D. Zhao, M. Sun, E.Y. Chang, Network representation learning with rich text information, in: Q. Yang, M.J. Wooldridge (Eds.), International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 2111–2117.
[43] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, B. Wang, One2Multi graph autoencoder for multi-view graph clustering, in: The Web Conference, WWW, 2020, pp. 3070–3076.
[44] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Trans. Knowl. Data Eng. 17 (12) (2005) 1624–1637.
[45] P.A. Estévez, M. Tesmer, C.A. Perez, J.M. Zurada, Normalized mutual information feature selection, IEEE Trans. Neural Netw. 20 (2) (2009) 189–201.
[46] L.J. Hubert, P. Arabie, Comparing partitions, J. Classification 2 (1985) 193–218.
[47] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, J. Cheng, Deep fusion clustering network, in: AAAI Conference on Artificial Intelligence, AAAI, 2021, pp. 9978–9987.
[48] X. Zhang, H. Liu, Q. Li, X. Wu, Attributed graph clustering via adaptive graph convolution, in: International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 4327–4333.
[49] J. Park, M. Lee, H.J. Chang, K. Lee, J.Y. Choi, Symmetric graph convolutional autoencoder for unsupervised graph representation learning, in: International Conference on Computer Vision, ICCV, 2019, pp. 6518–6527.
[50] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, ICLR, 2018.

Tong Wang received the Master degree in computer technology in Nanjing Tech University, Nanjing, China. He is currently pursuing the Ph.D. degree in computer science and technology in Southeast University, Nanjing, China. His research interests include pattern recognition, computer vision and medical image analysis.

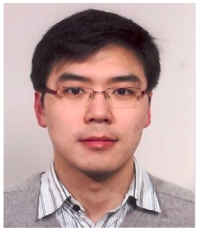
Junhua Wu received the Ph.D. degree from Southeast University, Nanjing, China. She is currently an associate professor with the School of Computer Science and Technology, Nanjing Tech University. Her research interests include software engineering, code generation and natural language processing.

Yaolei Qi received the B.Eng. degree from Southeast University, Nanjing China, in 2019. He is now working toward the Ph.D. degree in College of Computer Science and Technology, Southeast University, Nanjing China. His research interest include image processing, computer vision, machine learning and deep learning.

XiaoMing Qi received the B.S degree from Nanjing University of Science & Technology, Nanjing, China, in 2019. He is working toward the Ph.D. degree in Southeast University, Nanjing, China. His research interests include deep learning, medical image analysis, and federated learning.

Juwei Guan received the B.S. degree in software engineering from the Qingdao University of Technology in 2017 and received the M.S. degree in computer technology with the Jiangxi University of Finance and Economics in 2021. He is currently pursuing the Ph.D. degree in computer science and technology with the Southeast University. His current research interests include image processing, object detection, multi-domain learning and deep learning.

Yuan Zhang received the B.Eng. degree in Electronics and Engineering from Northwestern Polytechnical University, Xian'an, China, in 2019 and M.Sc. degree in Biomedical Engineering from Imperial College London, London, UK in 2020. She is currently working toward the Ph.D degree in Computer Science and Technology from Southeast University. Her research interests include medical image analysis and computer vision, machine learning and deep learning.



Guanyu Yang B.S. (2002), M.S. (2005), Biomedical Engineering, Southeast University, Ph.D. (2008), Signal and Image Processing, Leiden University Medical Center (LUMC, Leiden University, The Netherlands), Postdoctoral Fellowship (2009–2011). After returning to China in 2011, he joined Southeast University as a faculty member. Now he is the Vice Dean of School of Computer Science and Engineering, Professor, Doctoral Supervisor and Head of Department of Imaging Science and Technology. 2018, he is a member of Medical Imaging Professional Committee of Chinese Society of Graphics, Senior Member of IEEE.