Redundancy-Free Self-Supervised Relational Learning for Graph Clustering

Siyu Yi, Wei Ju, *Member, IEEE*, Yifang Qin, Xiao Luo, Luchen Liu, Yongdao Zhou, and Ming Zhang

***Abstract*— Graph clustering, which learns the node representations for effective cluster assignments, is a fundamental yet challenging task in data analysis and has received considerable attention accompanied by graph neural networks (GNNs) in recent years. However, most existing methods overlook the inherent relational information among the nonindependent and nonidentically distributed nodes in a graph. Due to the lack of exploration of relational attributes, the semantic information of the graph-structured data fails to be fully exploited which leads to poor clustering performance. In this article, we propose a novel self-supervised deep graph clustering method named relational redundancy-free graph clustering (R^2FGC) to tackle the problem. It extracts the attribute- and structure-level relational information from both global and local views based on an autoencoder (AE) and a graph AE (GAE). To obtain effective representations of the semantic information, we preserve the consistent relationship among augmented nodes, whereas the redundant relationship is further reduced for learning discriminative embeddings. In addition, a simple yet valid strategy is used to alleviate the oversmoothing issue. Extensive experiments are performed on widely used benchmark datasets to validate the superiority of our R^2FGC over state-of-the-art baselines. Our codes are available at https://github.com/yisiyu95/R2FGC.**

***Index Terms*— Deep clustering, graph representation learning, redundancy reduction, relationship preservation (REpre).**

I. Introduction

CLUSTERING, as one of the most classical and fundamental components in machine learning and data mining communities, has attracted significant attention. It serves as a critical preprocessing step in a variety of real-world applications such as community detection [1], anomaly detection [2], domain adaptation [3], and representation learning [4], [5], [6]. The underlying idea of clustering is to assign the samples to different groups such that similar samples are pulled into the same cluster while dissimilar samples are pushed into different clusters. Hence, clustering intuitively reflects the characteristics of the whole dataset, which could provide a priori information for various downstream domains, including computer vision and natural language processing.

Among many challenges therein, how to effectively partition the whole dataset into different clusters remains a fundamental yet open challenge such that the intrinsic distribution information of the dataset can be well-preserved. To achieve this goal, a large number of advanced approaches have been developed over the past decades [7], [8]. The traditional clustering methods such as subspace clustering [8] and spectral clustering [7] aim at projecting the data samples into a low-dimensional space coupled with additional constraint information so that the samples in the latent space can be clearly separated. However, the two-stage training paradigm of the traditional methods is typically suboptimal since the representation learning and clustering are dependent on each other that should be jointly optimized. Moreover, the traditional algorithms have limited model capacity that unavoidably limits their applicability and potential. Recently, benefiting from the strong representation capability of deep learning, massive deep clustering algorithms are proposed to show great potential and advantages over the traditional approaches [9], [10], [11], [12], [13]. The core essence of deep clustering is to group the data samples into different clusters through deep neural networks in an end-to-end fashion. In this way, clustering and representation learning are jointly optimized to learn clustering-friendly representations without manual feature extraction. For example, CC [9] jointly learned effective representations and cluster assignments by leveraging the power of instance- and cluster-level contrastive learning in an end-to-end manner.

With the prevalence of graph-structured data, graph neural networks (GNNs) have been extensively studied and achieved remarkable progress for many promising graph-related tasks and applications [14], [15], [16]. One fundamental problem therein is graph clustering, which divides nodes in a graph into different clusters. GNNs can be well-used for enhancing graph clustering performance to learn effective cluster assignments [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]. Recently, there has been an increasing body of approaches

Manuscript received 6 December 2022; revised 1 June 2023 and 31 August 2023; accepted 4 September 2023. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62306014, Grant 62106008, Grant 62276002, Grant 11871288, and Grant 12131001; in part by the China Postdoctoral Science Foundation under Grant 2023M730057; in part by the Fundamental Research Funds for the Central Universities; in part by the Key Laboratory of Pure Mathematics and Combinatorics, Ministry of Education, China (LPMC); and in part by the Key Laboratory for Medical Data Analysis and Statistical Research of Tianjin, China (KLMDASR). *(Corresponding authors: Wei Ju; Yongdao Zhou; Ming Zhang.)*

Siyu Yi and Yongdao Zhou are with the School of Statistics and Data Science, Nankai University, Tianjin 300071, China (e-mail: siyuyi@mail.nankai.edu.cn; ydzhou@nankai.edu.cn).

Wei Ju, Yifang Qin, Luchen Liu, and Ming Zhang are with the School of Computer Science, Peking University, Beijing 100871, China (e-mail: juwei@pku.edu.cn; qinyifang@pku.edu.cn; liuluchen@pku.edu.cn; mzhang_cs@pku.edu.cn).

Xiao Luo is with the Department of Computer Science, University of California, Los Angeles, Los Angeles, CA 90095 USA (e-mail: xiaoluo@cs.ucla.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3314451.

Digital Object Identifier 10.1109/TNNLS.2023.3314451

on graph clustering. For example, SDCN [18] first incorporated the topological structure knowledge into deep clustering accompanied by autoencoder (AE) [27] and GNN. To better combine node attributes and structure information, DFCN [21] improved the graph AE (GAE) [28] and developed a fusion mechanism to dynamically integrate both sides for robust target distribution generation. Based on AE, AGCC [24] incorporated the attention mechanism to fuse learned node representations and leveraged a self-supervised mechanism to guide the clustering optimization procedure.

Despite the promising achievements of previous methods, a majority of the existing graph clustering approaches still suffer from two key limitations.

1) *Neglect the Exploration of Relational Information:* Most existing GNN-based methods only use message-passing to aggregate neighboring information of the nodes in a graph. The high-order attributive and structural relationships of the non-IID graph-structured data are not well-exploited, which leads that the underlying distribution information cannot be well-revealed for meaningful representations.
2) *Fail to Reduce Redundant Information:* Many clustering methods mainly focus on exploring graph information from multiple perspectives, unavoidably incorporating much redundant information into the learned representations, while the redundancy reduction is not taken into account, which prevents obtaining discriminative representations and excellent clustering performance. As such, it is highly promising to develop an approach that can fully explore the intrinsic relational information among nodes and decrease the redundant information for effective cluster assignments.

Toward this end, this article proposes a novel deep clustering method called relational redundancy-free graph clustering (R^2FGC). The key idea of R^2FGC is to exploit attribute- and structure-level relational information among the nodes from both global and local views in a redundancy-free manner. To achieve the goal effectively, R^2FGC first learns compact representations from an AE and a GAE to explore the attributive and structural information from complementary perspectives. Then, the relational information is extracted based on the learned representations from global and local views. Moreover, to fully benefit from the extracted relationships, we preserve the consistent relationship such that the relational information for the same node is invariant to augmentations, whereas the correlations of the relational distribution for different nodes are reduced for learning discriminative representations. Furthermore, R^2FGC combines the redundancy-free relational learning from both attribute and structure levels with an augmentation-based fusion mechanism to optimize the embedded representations in a self-supervised fashion. Comprehensive experiments are conducted to show that the proposed method can greatly improve the clustering performance compared with the existing state-of-the-art approaches over multiple benchmark datasets. To summarize, the main contributions of our work are as follows.

1) *General Aspects:* This article studies the inherent relational learning for non-IID graph-structured data and explores redundancy-free representations based on relational information for the graph clustering task.
2) *Novel Methodologies:* We propose a novel approach to exploit attribute- and structure-level relational information among the nodes, which aims to extract augmentation-invariant relationships for the same node and decrease the redundant correlations between different nodes. Our R^2FGC is beneficial to obtain effective and discriminative representations for clustering.
3) *Multifaceted Experiments:* We perform extensive experiments on various commonly used datasets to demonstrate the effectiveness of the proposed approach.

II. Related Work

A. Graph Neural Networks

Recent years have witnessed great progress in GNNs and achieved state-of-the-art performance. The concept of GNNs was proposed [29] before 2010 and has become an ever-increasing theme. A general paradigm of GNNs is to iteratively update node representations by aggregating neighboring information based on message-passing [30]. Representative method graph convolutional network (GCN) [31] extended the classical convolutional neural networks to the case of graph-structured data. Subsequent work graph attention network (GAT) [32] further leveraged the attention mechanism [33] to dynamically aggregate the features of neighbors. With the powerful capability of GNNs, the learned graph representations can be used to serve a variety of downstream tasks, such as node classification [31], [34], graph classification [35], [36], [37], and graph clustering [18], [38].

B. Deep Clustering

The goal of deep clustering is to focus on using the excellent representation ability of deep learning to serve the clustering process, which has achieved remarkable progress. The existing methods can be categorized into three main groups based on the training objectives: 1) reconstruction-based methods; 2) self-augmentation-based methods; and 3) spectral-clustering-based methods. The first group uses the AE to reconstruct the original input, which incorporates desired constraints on feature embeddings in the latent space. For instance, DEC [39] iteratively conducted the process of representation learning and clustering assignments via minimizing the Kullback–Leibler (KL) divergence. To preserve important data structure, IDEC [40] introduced AE to improve the clustering so that the local structure of data generating distribution can be maintained. The second group aims to encourage the consistency between original samples and their augmented samples by optimizing the networks. For example, IIC [41] sought to achieve the consistency of assignment probabilities by maximizing the mutual information of paired samples. The third group aims at constructing a robust affinity matrix for effective data partitioning. For instance, RCFE [42] used the idea of rank constraints and clusters data points in a low-dimensional subspace. Li et al. [43] used multiple features to construct affinity graphs for spectral clustering.

Benefiting from the breakthroughs of GNNs on graph-structured data, GNNs are capable of organically integrating node attributes and graph structures in a united way and have emerged as a promising way for graph clustering. The basic idea is to group the nodes in the graph into several disjoint clusters. Similar to the deep clustering methods, a majority of the existing graph clustering approaches [18], [22], [23], [24], [44], [45], [46], [47] also continue the paradigm of AE, in which the GAE and the variational GAE (VGAE) are leveraged to operate on graph-structured data. For example, to dynamically learn the importance of the neighboring nodes to the center node, DAEGC [45] used the GAE to capture a compact representation by encoding the graph structures and node attributes. EGAE [23] learned the explainable representations based on the GAE that can also be used for various tasks. Compared with previous methods, our work further explores graph clustering by simultaneously preserving the relational similarity and reducing the redundancy of the learned representations based on both AE and GAE.

C. Self-Supervised Learning (SSL)

Recently, SSL revitalizes and has achieved superior performance across numerous domains. This technique is completely free of the need for explicit labels [48], due to its powerful capability in learning effective representations from unlabeled data. The core procedure of SSL is first designing a domain-specific pretext task and training the networks on the pretext task, such that the learned representations can be more discriminative and applicable. Recently, many SSL approaches have been proposed to marry the power of SSL and deep learning [49], [50], [51], [52] and have shown competitive performance in various downstream application [53], [54], [55], [56], [57]. For example, SimCLR [49] used multiple data augmentations and a learnable nonlinear transformation to train an encoder, such that the model can pull the feature representations from the same samples together. To alleviate the issue of the large batch size of SimCLR, MoCo [50] introduced a moving-averaged encoder to set up a dynamic dictionary for SSL. Furthermore, our proposed R^{2}FGC inherits the advantages of SSL to preserve the consistent relationship and reduce the redundant information among nodes from global and local views for graph clustering.

III. Notations and Problem Definition

In this section, we first briefly give the basic notations and formal terminologies in a graph. Then we introduce the concept of GCN and the problem formalization of graph clustering.

A. Notations

Let $\mathcal{G} = (V, E, \mathbf{X})$ denote an arbitrary undirected graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set with n nodes, E is the edge set, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top \in \mathbb{R}^{n\times d}$ is the node attribute matrix with $\mathbf{x}_i$ corresponding to node i for $i = 1, \ldots, n$, and d is the dimensionality of the node attributes. $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$ denote the adjacency matrix which is generated according to the adjacency relationships in E, and $a_{ij} = 1$ if $(v_i, v_j) \in E$, i.e., there is an edge from node v_i to node v_j, otherwise $a_{ij} = 0$. The adjacency matrix can be normalized by $\mathbf{S} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{A}} = (\tilde{a}_{ij}) = \mathbf{A} + \mathbf{I}$, $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identify matrix for adding self-connections, and $\tilde{\mathbf{D}} = \mathrm{diag}(\tilde{d}_1, \ldots, \tilde{d}_n)$ is the corresponding degree matrix with $\tilde{d}_i = \sum_{j=1}^{n} \tilde{a}_{ij}$.

B. Graph Convolutional Network

GCN generalizes the classical convolutional neural networks to the case of graph-structured data. It uses the graph directly and learns new representations by aggregating the information of a node and its neighbors. In general, a layer of GCN has the form

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{S}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

where $\mathbf{H}^{(0)}$ is the input data, $\sigma(\cdot)$ is an activation function, such as Tanh and ReLU, and $\mathbf{H}^{(l)}$ and $\mathbf{W}^{(l)}$ are the learned embedded representation and the trainable weight matrix in the lth ($l > 0$) layer, respectively.

C. Graph Clustering

Given an unlabeled graph with n nodes, the target of the graph clustering task is to divide these unlabeled nodes into K disjoint clusters $\{C_1, \ldots, C_K\}$ based on a well-learned embedding matrix $\tilde{\mathbf{Z}} \in \mathbb{R}^{n\times d'}$, where d' is the number of dimension of the latent embeddings. The nodes in the same cluster are highly similar and cohesive, while the nodes in different clusters are discriminative and separable.

IV. Proposed Method

In this section, we introduce our proposed method named R^{2}FGC. R^{2}FGC mainly contains four parts, i.e., attribute- and structure-level representation learning module, REpre and de-redundancy module, augmentation-based representation fusion module, and joint optimization module for graph clustering. Fig. 1 shows the framework overview of the proposed R^{2}FGC. In the following, we present the four components and the complexity analysis for R^{2}FGC.

A. Attribute- and Structure-Level Learning Module

AE can reasonably explore the node attribute information, whereas the GAE can effectively capture the topological structure information. To gain a more comprehensive embedding and a better performance on downstream tasks, we consider both AE and GAE to reconstruct the input and learn fusional representation.

The AE module feeds the attribute information into the multilayer perceptrons and extracts the latent representations by minimizing the reconstruction loss between the input raw data and the reconstructed data. The corresponding optimization objective is formalized as

$$\begin{aligned} &\min L_{\mathrm{AE}} = \frac{1}{n}\left|\left|\mathbf{X} - \hat{\mathbf{X}}_{\mathrm{AE}}\right|\right|_F^2 \\ &\text{s.t. } \mathbf{Z}_{\mathrm{AE}} = \phi_e(\mathbf{X}) \\ &\quad \hat{\mathbf{X}}_{\mathrm{AE}} = \phi_d(\mathbf{Z}_{\mathrm{AE}}) \end{aligned} \tag{1}$$

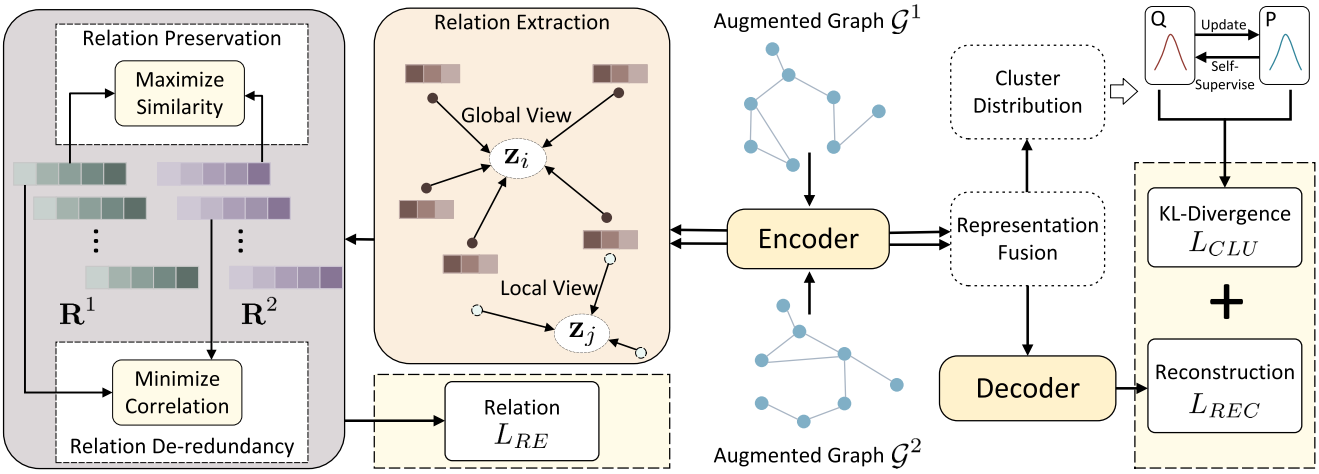


Fig. 1. Framework overview of the proposed method R^{2}FGC. Relational learning and representation fusion are performed to jointly guide the self-supervised graph clustering based on the latent representations from the encoders of AE and GAE. The relationship preservation (REpre) and de-redundancy contribute to exploring inherent node relationship and filter redundancy relationship to learn effective and discriminative representations.

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the input attribute matrix, $\hat{\mathbf{X}}_{\text{AE}} \in \mathbb{R}^{n\times d}$ is the reconstructed data, $||\cdot||_F$ is the Frobenius norm, $\mathbf{Z}_{\text{AE}} \in \mathbb{R}^{n\times d'}$ is the learned latent representation in AE, and ϕ_e and ϕ_d are the encoder and decoder networks, respectively.

In the GAE module, following the improved version in [21], a multilayer GCN is adopted to reconstruct the adjacency matrix and the attribute information. The corresponding reconstruction loss is formalized as

$$\begin{aligned} \min\ & L_{\text{GAE}} = \frac{\alpha}{n}\left|\left|\mathbf{S}-\hat{\mathbf{S}}\right|\right|_F^2 + \frac{1}{n}\left|\left|\mathbf{SX}-\hat{\mathbf{X}}_{\text{GAE}}\right|\right|_F^2 \\ \text{s.t. } & \mathbf{H}_e^{(l+1)} = \sigma\left(\mathbf{SH}_e^{(l)}\mathbf{W}_e^{(l)}\right) \\ & \mathbf{H}_d^{(l+1)} = \sigma\left(\mathbf{SH}_d^{(l)}\mathbf{W}_d^{(l)}\right) \\ & \mathbf{H}_e^{(0)} = \mathbf{X} \end{aligned} \tag{2}$$

where α is a predefined hyperparameter, $\mathbf{S}$ is the normalized adjacency matrix, $\hat{\mathbf{S}}$ is the reconstructed adjacency matrix produced by fusing the respective inner products of the learned latent representation $\mathbf{Z}_{\text{GAE}} \in \mathbb{R}^{n\times d'}$ resulting from the graph encoder and the attribute representations $\hat{\mathbf{X}}_{\text{GAE}}$ (i.e., the reconstructed weighted attribute matrix) resulting from the graph decoder, and $\mathbf{W}_e^{(l)}$ and $\mathbf{W}_d^{(l)}$ are the layer-specific trainable weight matrices in the lth graph encoder and decoder layers, respectively. The detailed fusion mechanism is discussed in Section IV-C, which unites the embedded representations from both AE and GAE to promote latent presentation learning in the graph augmentation fashion.

B. REpre and De-Redundancy Module

In this module, we learn the inherent relational information among the nodes based on augmentations on a given graph. One of the basic ideas is to preserve the similarity of the relational information from two augmented views, while the latent representation of the same node can vary after graph augmentation. Hence, we aim to increase the consistency of the relational information in each positive pair. It allows fine-grained mining of the node relationship. On the other hand, it is necessary to improve the discriminative capability of the resulting representations for graph clustering, and thus, we also decrease the correlation of the relational information in each negative pair. In the following, we first introduce the adopted graph augmentation strategies and relationship extraction methods. Then, we describe the details of the subsequent REpre and relationship de-redundancy (REder).

Based on the given graph, we first construct two different graph views through augmentations, including the following.

1) *Attribute Perturbation:* For each value in the attribute matrix, we disturb it by multiplying a Gaussian random number with a small variance. This strategy performs a slight disturbance on the node features, which would not essentially change the semantic information.
2) *Edge Deletion:* We remove some edges based on the node similarity obtained from the prelearned latent embeddings. For each node, the edges that connect the nodes with low similarity are dropped in a certain proportion. Compared with random deletion, more semantic information can be preserved by referring to the node similarity.
3) *Graph Diffusion:* We transform the adjacency matrix to a diffusion matrix by leveraging graph diffusion [58], which contributes to providing additional local information. Technically, given the transition matrix $\mathbf{T}$, the graph diffusion matrix $\mathbf{U}$ is formulated as

$$\mathbf{U} = \sum_{j=0}^{\infty} \theta_j \mathbf{T}^j$$

where θ_j is the weight coefficient. We adopt the personalized PageRank [59] to characterize graph diffusion, which is a special case. Specifically, $\mathbf{T}$ is chosen as the normalized adjacency matrix $\mathbf{S}$ and $\theta_j = \eta(1-\eta)^j$ with teleport probability $\eta \in (0, 1)$. Then, the resulting diffusion matrix $\mathbf{U}$ has the form

$$\mathbf{U} = \eta(\mathbf{I} - (1-\eta)\mathbf{S})^{-1}. \tag{3}$$

After obtaining two augmented graph views $\mathcal{G}^1 = \{\mathbf{X}^1, \mathbf{S}^1\}$ and $\mathcal{G}^2 = \{\mathbf{X}^2, \mathbf{S}^2\}$, we perform AE and GAE on $\mathbf{X}^1$ and $\mathbf{X}^2$, which generates the attribute-level latent representations $\mathbf{Z}^1_{\mathrm{AE}}, \mathbf{Z}^2_{\mathrm{AE}}$ and the structure-level latent representations $\mathbf{Z}^1_{\mathrm{GAE}}, \mathbf{Z}^2_{\mathrm{GAE}}$. To meticulously characterize the relational information, we explore the similarities of each node to some anchor nodes from both global and local perspectives based on these representations.

1) Extraction of Global Anchors: For capturing the global relationship of a query node $v_i \in V$, the target is to sample diverse anchors from the whole graph nodes. Due to the neighborhood aggregation mechanism in GNNs, we argue that the high-degree nodes may receive more information when passing messages, while the low-degree nodes would receive less information. This may result in poor representations for the nodes with low degrees. Hence, we perform nonuniform sampling on the nodes to balance the qualities of the representations for low- and high-degree nodes. Specifically, we adopt an inverse degree-weighted distribution for sampling anchors, which puts a larger sampling probability on a lower degree node. The sampling weight and probability for each node, respectively, are as follows:

$$w_i = \beta^{\log(\tilde{d}_i+1)}$$
$$p_i = \frac{w_i}{\sum_{v_j \in V} w_j}, \quad \text{for any } v_i \in V$$

where $\beta \in (0, 1)$ is a hyperparameter to control the skewness of the distribution, and $\tilde{d}_i$ is the degree of node v_i. Moreover, quasi-Monte Carlo (QMC) sampling methods usually can achieve a higher convergence rate than Monte Carlo (MC) methods [60]. Hence, based on the defined discrete distribution, we perform multinomial sampling in the QMC fashion [61], [62]. Instead of the uniform random number (the MC fashion), we leverage the randomized 1-D low-discrepancy point set $\{(2i-1)/(2M_1) + \omega \mod 1 \in [0, 1] : \omega \sim U(0, 1), i = 1, \ldots, M_1\}$ to do multinomial sampling on the discrete distribution in each training epoch. Randomization is used to avoid the same sample in different epochs and increase the randomness for extracting more diverse anchors. For each node $v_i \in V$, we denote the index set of the sampled anchors from the global view as A^g_i and $|A^g_i| = M_1$.

2) Extraction of Local Anchors: To fully explore the relational information, besides the global anchor sampling, we also concentrate on the local relational information. Graph diffusion removes the restriction of using only the direct neighbors and alleviates the problem of noisy and often arbitrarily defined edges. It leads that the diffusion matrix $\mathbf{U}$ in (3) can acquire richer structural information in the local view compared with the traditional GNNs. Hence, we leverage graph diffusion to generate the local anchors according to the scores in $\mathbf{U}$. Specifically, the values in the ith row of $\mathbf{U}$ can reflect the influence between node v_i and all the other nodes. We select the nodes with M_2 largest scores in the ith row of $\mathbf{U}$ as the local anchors of node v_i. It makes that the local anchors of v_i share similar semantic information to v_i, which allows us to extract more effective local relationships. For each node $v_i \in V$, we denote the index set of the local anchors as A^l_i and $|A^l_i| = M_2$.

Based on these global- and local-view anchor sets $A^g_i, A^l_i, i = 1, \ldots, n$, we extract the relational information of the nodes in the sense of similarity. We use the AE latent representations $\mathbf{Z}^1_{\mathrm{AE}} = (\mathbf{z}^1_{\mathrm{AE},1}, \ldots, \mathbf{z}^1_{\mathrm{AE},n})^\top, \mathbf{Z}^2_{\mathrm{AE}} = (\mathbf{z}^2_{\mathrm{AE},1}, \ldots, \mathbf{z}^2_{\mathrm{AE},n})^\top$ to illustrate the detailed process. Specifically, given a query node $v_i \in V$, we calculate the similarities between the embedded representation of v_i in $\mathbf{Z}^1_{\mathrm{AE}}$ and the embeddings of these anchors in $\mathbf{Z}^2_{\mathrm{AE}}$ by

$$r^1_g(i, k_g) = \left(\mathbf{z}^1_{\mathrm{AE},i}\right)^\top \mathbf{z}^2_{\mathrm{AE},k_g}, \quad k_g \in A^g_i$$
$$r^1_l(i, k_l) = \left(\mathbf{z}^1_{\mathrm{AE},i}\right)^\top \mathbf{z}^2_{\mathrm{AE},k_l}, \quad k_l \in A^l_i.$$

Similarly, we also calculate the similarities between the embedding of v_i in $\mathbf{Z}^2_{\mathrm{AE}}$ and those of the anchors in $\mathbf{Z}^2_{\mathrm{AE}}$ by

$$r^2_g(i, k_g) = \left(\mathbf{z}^2_{\mathrm{AE},i}\right)^\top \mathbf{z}^2_{\mathrm{AE},k_g}, \quad k_g \in A^g_i$$
$$r^2_l(i, k_l) = \left(\mathbf{z}^2_{\mathrm{AE},i}\right)^\top \mathbf{z}^2_{\mathrm{AE},k_l}, \quad k_l \in A^l_i.$$

Hereafter, let $\mathbf{r}^u_c(i)$ be the relationship vector composed by $r^u_c(i, k)$ with k traversing the whole index set A^c_i of node $v_i, i \in \{1, \ldots, n\}$, and $\mathbf{R}^u_c = (\mathbf{r}^u_c(1), \ldots, \mathbf{r}^u_c(n))^\top$ be the relationship matrix for any $u \in \{1, 2\}, c \in \{g, l\}$.

3) Relationship Preservation: To make the relational information invariant to augmentation, we maximize the proximity of $\mathbf{r}^1_c(i)$ and $\mathbf{r}^2_c(i)$ from both global and local views, i.e., we maximize the attribute-level relational similarities of all the positive pairs under augmentation, which are formulated by

$$R^g_{\mathrm{AE}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{r}^1_g(i)^\top \mathbf{r}^2_g(i)}{\left|\left|\mathbf{r}^1_g(i)\right|\right| \cdot \left|\left|\mathbf{r}^2_g(i)\right|\right|}\right)^2$$

and

$$R^l_{\mathrm{AE}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{r}^1_l(i)^\top \mathbf{r}^2_l(i)}{\left|\left|\mathbf{r}^1_l(i)\right|\right| \cdot \left|\left|\mathbf{r}^2_l(i)\right|\right|}\right)^2.$$

We can similarly obtain the structure-level relational similarities R^g_{GAE} and R^l_{GAE} corresponding to GAE from both the views. This operation helps learn representations that are more reflective of the relationships between the attribute and topological information of all the nodes.

4) Relationship De-Redundancy: In addition, besides preserving the relational similarity under augmentations, the discriminative capability of the latent representation is also important for the downstream graph clustering task. Hence, we decrease the correlations of the relationship vectors for different nodes from both global and local views. It contributes to filtering redundant information and improving the separating capability for better clustering performance. Specifically, we minimize the attribute-level relational correlations of all the negative pairs, which are formulated as follows:

$$C^g_{\mathrm{AE}} = \frac{1}{n(n-1)}\sum_{i,j=1,i\neq j}^{n}\left(\frac{\mathbf{r}^1_g(i)^\top \mathbf{r}^2_g(j)}{\left|\left|\mathbf{r}^1_g(i)\right|\right| \cdot \left|\left|\mathbf{r}^2_g(j)\right|\right|}\right)^2$$

and

$$C^l_{\mathrm{AE}} = \frac{1}{n(n-1)}\sum_{i,j=1,i\neq j}^{n}\left(\frac{\mathbf{r}^1_l(i)^\top \mathbf{r}^2_l(j)}{\left|\left|\mathbf{r}^1_l(i)\right|\right| \cdot \left|\left|\mathbf{r}^2_l(j)\right|\right|}\right)^2.$$

In like manner, we can obtain the corresponding structure-level loss under GAE from global and local views, denoted by C^g_{GAE} and C^l_{GAE}, respectively.

Based on the above discussion, we can capture the augmentation-invariant relational information and conduct redundancy-free relational learning by minimizing the total relationship loss $L_{\mathrm{RE}} = L_{\mathrm{REA}} + L_{\mathrm{REG}}$ with

$$\begin{aligned} L_{\mathrm{REA}} &= C^g_{\mathrm{AE}} + C^l_{\mathrm{AE}} - R^g_{\mathrm{AE}} - R^l_{\mathrm{AE}} \\ L_{\mathrm{REG}} &= C^g_{\mathrm{GAE}} + C^l_{\mathrm{GAE}} - R^g_{\mathrm{GAE}} - R^l_{\mathrm{GAE}}. \end{aligned} \tag{4}$$

The loss L_{RE} takes into account both efficient representation learning and reduction of redundant information upon the relationship extraction of the nodes, which allows for better guidance of downstream tasks.

C. Augmentation-Based Representation Fusion Module

In this section, to obtain fine-grained representations of the nodes, we discuss the fusion mechanism of the attribute- and structure-level latent representations based on augmentations. First, we take a weighted summation of the four parts to fuse the embedded representations from the two levels as follows:

$$\tilde{\mathbf{Z}}_c = \mathbf{W}_1 \odot \left(\mathbf{Z}^1_{\mathrm{AE}} + \mathbf{Z}^2_{\mathrm{AE}}\right) + \mathbf{W}_2 \odot \left(\mathbf{Z}^1_{\mathrm{GAE}} + \mathbf{Z}^2_{\mathrm{GAE}}\right)$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n\times d'}$ are trainable weight matrices to control the importance of the two types of representations, and $\odot$ is the Hadamard product. Based on $\tilde{\mathbf{Z}}_c$, we further blend the embeddings from both global and local views to refine the fused information. From the local view, we adopt the neighborhood aggregation operation on $\tilde{\mathbf{Z}}_c$ to enhance the local information, whereas, from the global view, we use the self-correlation matrix of the nodes characterized by $\tilde{\mathbf{Z}}_c$ to improve the exploitation of the global information, which is normalized by the softmax function. Specifically, the final formula of the fused representation is

$$\tilde{\mathbf{Z}} = \delta \mathbf{S}\tilde{\mathbf{Z}}_c + \mathrm{softmax}\left(\mathbf{S}\tilde{\mathbf{Z}}_c\tilde{\mathbf{Z}}_c^\top \mathbf{S}^\top\right)\mathbf{S}\tilde{\mathbf{Z}}_c \tag{5}$$

where δ is a trainable weight parameter. With $\tilde{\mathbf{Z}}$, we can obtain the reconstructed attribute matrix $\hat{\mathbf{X}}_{\mathrm{AE}}$ in (1) and weighted attribute matrix $\hat{\mathbf{X}}_{\mathrm{GAE}}$ in (2) by feeding $\tilde{\mathbf{Z}}$ into the decoders of AE and GAE, respectively. The reconstructed adjacency matrix is calculated by fusing the self-correlations of the learned representations in GAE, which is formulated as

$$\hat{\mathbf{S}} = \frac{1}{2}\left(\mathbf{Z}^1_{\mathrm{GAE}}\left(\mathbf{Z}^1_{\mathrm{GAE}}\right)^\top + \mathbf{Z}^2_{\mathrm{GAE}}\left(\mathbf{Z}^2_{\mathrm{GAE}}\right)^\top\right) + \hat{\mathbf{X}}_{\mathrm{GAE}}\hat{\mathbf{X}}^\top_{\mathrm{GAE}}.$$

The above fusion process is similar to [21].

In addition, under the neighbor aggregation mechanism, GCN updates node representations by aggregating information from the neighbors. However, when stacking multiple layers, the learned representations would become indistinguishable, seriously degrading the performance, which is the so-called oversmoothing issue [63], [64]. Hence, it is important to balance the message aggregation ability and oversmoothing issue. To alleviate the problem in GAE, we incorporate a novel propagation-regularization (PR) loss to enhance information capturing while alleviating oversmoothing defined as

$$L_{\mathrm{PR}} = \sum_{\mathbf{H}\in\mathcal{E}} \nu(\mathbf{H}, \mathbf{SH})$$

where $\mathcal{E}$ contains the embedding matrix in each layer of both the encoder and the decoder in GAE, and $\nu(\cdot)$ is the metric function, such as the cross entropy, KL divergence, and the Jensen–Shannon divergence. PR simulates a deep GCN by supervision at a low cost, which enables current embeddings to capture further information contained in the deeper layer. Compared with directly increasing the GCN layers, we can more finely balance the information capture ability and the oversmoothing problem by adjusting the weight of the loss.

Thereby, the total reconstruction loss is computed by

$$L_{\mathrm{REC}} = L_{\mathrm{AE}} + L_{\mathrm{GAE}} + \epsilon L_{\mathrm{PR}} \tag{6}$$

where ϵ is the predefined hyperparameter to adjust the influence ratio, and the reconstruction losses L_{AE} and L_{GAE} in AE and GAE are defined in (1) and (2), respectively.

D. Joint Optimization Module for Graph Clustering

Graph clustering is essentially an unsupervised task with no feedback available as reliable guidance. To this end, we perform a clustering layer on the fused representation $\tilde{\mathbf{Z}}$ in (5) and use the soft labels derived by a probability distribution as a self-supervised signal to jointly optimize the redundancy-free relational learning framework for graph clustering.

First, using the Student's t-distribution as a kernel, we calculate the soft cluster assignment probabilities $\mathbf{Q}_1 = (q_{1,ij}), \mathbf{Q}_2 = (q_{2,ij}), \mathbf{Q}_3 = (q_{3,ij}) \in \mathbb{R}^{n\times K}$ upon the latent embeddings $\tilde{\mathbf{Z}}$, $(\mathbf{Z}^1_{\mathrm{AE}}+\mathbf{Z}^2_{\mathrm{AE}})/2$, $(\mathbf{Z}^1_{\mathrm{GAE}}+\mathbf{Z}^2_{\mathrm{GAE}})/2$, respectively, to measure the similarities between the latent representations and cluster centroids, i.e., each value indicates the probability of assigning the ith node to the jth cluster. For example, $q_{1,ij}$ is computed as follows:

$$q_{1,ij} = \frac{\left(1 + \left|\left|\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_j\right|\right|^2\right)^{-1}}{\sum_{k=1}^{K}\left(1 + \left|\left|\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_k\right|\right|^2\right)^{-1}}$$

where $\tilde{\mathbf{Z}} = (\tilde{\mathbf{z}}_1^\top, \ldots, \tilde{\mathbf{z}}_n^\top)^\top$ and $\boldsymbol{\mu}_j$s are the cluster centroids. The $q_{2,ij}$ and $q_{3,ij}$ can be calculated similarly. The $\boldsymbol{\mu}_j$s are initialized by performing k-means on the pretrained fused representation. When the network is well-trained, we adopt the fusion-based assignment matrix $\mathbf{Q}_1$ to measure the cluster assignment probability of all the nodes, i.e.,

$$y_i = \mathrm{argmax}_{j\in\{1,\ldots,K\}} q_{1,ij} \tag{7}$$

where y_i is the predicted cluster of node v_i for $i = 1, \ldots, n$.

Next, we introduce an auxiliary confident probability distribution $\mathbf{P} = (p_{ij}) \in \mathbb{R}^{n\times K}$ to improve the confidence of the soft assignment, which is derived from $\mathbf{Q}_1$ and formulated as

$$p_{ij} = \frac{q^2_{1,ij}/\sum_{i=1}^{n} q_{1,ij}}{\sum_{k=1}^{K}\left(q^2_{1,ik}/\sum_{i=1}^{n} q_{1,ik}\right)}.$$

To make the data representation close to cluster centroids and improve cluster cohesion, we minimize the KL divergence loss between $\mathbf{P}$ and $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3$ as follows:

$$L_{\mathrm{CLU}} = \sum_{i=1}^{n}\sum_{j=1}^{K} p_{ij} \log \frac{p_{ij}}{\left(q_{1,ij} + q_{2,ij} + q_{3,ij}\right)/3}. \tag{8}$$

Algorithm 1 R^{2}FGC

Input: Attribute matrix $\mathbf{X}$; adjacency matrix $\mathbf{A}$; cluster number K; hyperparameters M_1, M_2; Maximum iterations I_{max};

Output: Clustering result $\mathbf{y}$;

```
Initialize the parameters in AE, GAE, the fusion part, and the cluster centroids;
for i = 1 to I_max do
    Obtain {X^1, S^1} and {X^2, S^2} by augmentation;
    Update Z^1_AE, Z^2_AE and Z^1_GAE, Z^2_GAE by encoding X^1 and X^2 in AE and GAE;
    Calculate R^1_g, R^2_g, R^1_l, R^2_l based on AE and GAE in Section IV-B;
    Update Z̃_c, Z̃, Ŝ in Section IV-C and obtain X̂_AE, X̂_GAE in Section IV-A;
    Calculate Q_1, Q_2, Q_3, and P in Section IV-D;
    Calculate the losses L_RE, L_REC, L_CLU in (4), (6), (8), respectively;
    Conduct the backpropagation and update the whole network in the proposed R^2FGC by minimizing (9);
end for
Obtain the clustering result y with the fused representation Z̃ by (7);
return y;
```

Using the confident distribution $\mathbf{P}$, the process self-supervises the cluster assignment without any label guidance. We integrate the latent representations from AE, GAE, and the fusion mechanism in the self-supervised clustering procedure to obtain more accurate clustering results.

To sum up, the total loss L in the whole framework of R^{2}FGC is composed of the relationship loss, the reconstruction loss, and the self-supervised clustering loss, i.e.,

$$L = L_{\mathrm{RE}} + L_{\mathrm{REC}} + \kappa L_{\mathrm{CLU}} \tag{9}$$

where κ is a predefined hyperparameter to balance the weight of the clustering loss. The training process of our proposed R^{2}FGC is summarized in Algorithm 1.

E. Computational Complexity Analysis

For the scalability of large-scale datasets, we adopt the mini-batch stochastic gradient descent to optimize our method. Assume that the batch size is B and the dimensions of each layer of AE and GAE are $\bar{d}_1, \ldots, \bar{d}_{L_1}$ and $\tilde{d}_1, \ldots, \tilde{d}_{L_2}$, respectively. Given a graph with n nodes and $|E|$ edges, the dimension of the original attributes is d. The time complexities of AE and GAE are $O(n\sum_{i=1}^{L_1}\bar{d}_i\bar{d}_{i-1})$ and $O(|E|\sum_{i=1}^{L_2}\tilde{d}_i\tilde{d}_{i-1})$ with $\bar{d}_0 = \tilde{d}_0 = d$, respectively. For each batch, the complexity of the relationship learning module is $O(B(B+d')(M_1+M_2))$ based on d'-dimensional latent representations. Moreover, we perform the representation fusion and PR in $O(B^2d' + B\log B)$ time and conduct the self-supervised clustering in $O(BK + B\log B)$ time with K classes in the task. Hence, the total computational complexity of our method R^{2}FGC is $O(n\sum_{i=1}^{L_1}\bar{d}_i\bar{d}_{i-1} + |E|\sum_{i=1}^{L_2}\tilde{d}_i\tilde{d}_{i-1} + n(B+d')(M_1+M_2) + n(Bd'+K))$, which is linearly related to the numbers of nodes and edges.

TABLE I
DESCRIPTION OF THE BENCHMARK DATASETS

Dataset	Type	Samples	Dimension	Classes
ACM	Graph	3025	1870	3
AMAP	Graph	7650	745	8
CITE	Graph	3327	3703	6
DBLP	Graph	4057	334	4
HHAR	Record	10299	561	6

V. EXPERIMENTS

In this section, we first introduce the experimental settings and then conduct experiments to validate the effectiveness of R^{2}FGC. We aim to answer the following research questions.

1) *RQ1:* Compared with the state-of-the-art methods, does our method R^{2}FGC achieve better performance for self-supervised graph clustering?
2) *RQ2:* How do different components of the proposed method contribute to the clustering performance?
3) *RQ3:* How do the hyperparameters in R^{2}FGC affect the final clustering performance?
4) *RQ4:* How is the convergence of the proposed model under different datasets?
5) *RQ5:* Is there any supplementary analysis that can illustrate the superiority of R^{2}FGC?

A. Experimental Settings

1) Datasets: For comparison, we perform the proposed method R^{2}FGC on five commonly used benchmark datasets. Four of them are graph datasets, including a paper network ACM,[1] a shopping network AMAP,[2] a citation network CITE,[3] and an author network DBLP[4]; another is a nongraph dataset, i.e., a record dataset HHAR [65]. Following [18], for the nongraph data, the adjacency matrix is generated by the undirected k-nearest neighbor graph. Table I briefly summarizes the information of these benchmark datasets.

2) Compared Methods: To illustrate the superiority of our proposed R^{2}FGC, we compare its clustering performance with some state-of-the-art clustering methods, which are divided into four categories, i.e., the classical shallow clustering method k-means, the AE-based methods, the GCN-based methods, and the combination of AE and GCN. The AE-based methods contain AE [27], DEC [39], and IDEC [40]. They convert the raw data into low-dimensional codes to learn feature representations by AE and then perform clustering over the learned latent embeddings. The GCN-based methods include GAE, VGAE [28], DAEGC [45], and ARGA [46]. They adopt the GCN encoder to learn the node content and topological information for clustering. In addition, some methods combine AE and GCN to boost

[1] http://dl.acm.org/
[2] https://github.com/shchur/gnn- benchmark/raw/master/data/npz/amazon_electronics photo.npz
[3] http://citeseerx.ist.psu.edu/index
[4] https://dblp.uni-trier.de

TABLE II

CLUSTERING PERFORMANCE ON FIVE BENCHMARK DATASETS (MEAN±STD). THE BEST RESULTS IN ALL THE METHODS AND ALL THE BASELINES ARE HIGHLIGHTED IN **BOLD** AND UNDERLINE, RESPECTIVELY

Dataset	Metric	k-Means	AE	DEC	IDEC	GAE	VGAE	DAEGC	ARGA	SDCN	DFCN	AGCC	R^{2}FGC (Ours)
ACM	ACC	67.31±0.71	81.83±0.08	84.33±0.76	85.12±0.52	84.52±1.44	84.13±0.22	86.94±2.83	86.29±0.36	90.45±0.18	90.90±0.20	90.38±0.38	**92.43±0.18**
	NMI	32.44±0.46	49.30±0.16	54.54±1.51	56.61±1.16	55.38±1.92	53.20±0.52	56.18±4.15	56.21±0.82	68.31±0.25	69.40±0.40	68.34±0.89	**72.42±0.53**
	ARI	30.60±0.69	54.64±0.16	60.64±1.87	62.16±1.50	59.46±3.10	57.72±0.67	59.35±3.89	63.37±0.86	73.91±0.40	74.90±0.40	73.73±0.90	**78.72±0.47**
	F1	67.57±0.74	82.01±0.08	84.51±0.74	85.11±0.48	84.65±1.33	84.17±0.23	87.07±2.79	86.31±0.35	90.42±0.19	90.80±0.20	90.39±0.39	**92.45±0.18**
AMAP	ACC	27.22±0.76	48.25±0.08	47.22±0.08	47.62±0.08	71.57±2.48	74.26±3.63	76.44±0.01	69.28±2.30	53.44±0.81	76.88±0.80	75.25±1.21	**81.28±0.05**
	NMI	13.23±1.33	38.76±0.30	37.35±0.05	37.83±0.08	62.13±2.79	66.01±3.40	65.57±0.03	58.36±2.76	44.85±0.83	69.21±1.00	68.37±1.39	**73.88±0.17**
	ARI	5.50±0.44	20.80±0.47	18.59±0.04	19.24±0.07	48.82±4.57	56.24±4.66	59.39±0.02	44.18±4.41	31.21±1.23	58.98±0.84	58.32±2.38	**66.25±0.36**
	F1	23.96±0.51	47.87±0.20	46.71±0.12	47.20±0.11	68.08±1.76	70.38±2.98	69.97±0.02	64.30±1.95	50.66±1.49	71.58±0.31	70.04±1.63	**75.29±0.32**
CITE	ACC	39.32±3.17	57.08±0.13	55.89±0.20	60.49±1.42	61.35±0.80	60.97±0.36	64.54±1.39	61.07±0.49	65.96±0.31	69.50±0.20	68.08±1.44	**70.60±0.45**
	NMI	16.94±3.22	27.64±0.08	28.34±0.30	27.17±2.40	34.63±0.65	32.69±0.27	36.41±0.86	34.40±0.71	38.71±0.32	43.90±0.20	40.86±1.45	**45.39±0.37**
	ARI	13.43±3.02	29.31±0.14	28.12±0.36	25.70±2.65	33.55±1.18	33.13±0.53	37.78±1.24	34.32±0.70	40.17±0.43	45.50±0.37	41.82±2.03	**47.07±0.30**
	F1	36.08±3.53	53.80±0.11	52.62±0.17	61.62±1.39	57.36±0.82	57.70±0.49	62.20±1.32	58.23±0.31	63.62±0.24	64.30±0.20	60.47±1.57	**65.28±0.12**
DBLP	ACC	38.65±0.65	51.43±0.35	58.16±0.56	60.31±0.62	61.21±1.22	58.59±0.06	62.05±0.48	64.83±0.59	68.05±1.81	76.00±0.80	73.45±2.16	**80.95±0.20**
	NMI	11.45±0.38	25.40±0.16	29.51±0.28	31.17±0.50	30.80±0.91	26.92±0.06	32.49±0.45	29.42±0.92	39.50±1.34	43.70±1.00	40.36±2.81	**50.82±0.32**
	ARI	6.97±0.39	12.21±0.43	23.92±0.39	25.37±0.60	22.02±1.40	17.92±0.07	21.03±0.52	27.99±0.91	39.15±2.01	47.00±1.50	44.40±3.79	**56.34±0.42**
	F1	31.92±0.27	52.53±0.36	59.38±0.51	61.33±0.56	61.41±2.23	58.69±0.07	61.75±0.67	64.97±0.66	67.71±1.51	75.70±0.80	71.84±2.02	**80.54±0.19**
HHAR	ACC	59.98±0.02	68.69±0.31	69.39±0.25	71.05±0.36	62.33±1.01	71.30±0.36	76.51±2.19	63.30±0.80	84.26±0.17	87.10±0.10	86.54±1.79	**88.91±0.05**
	NMI	58.86±0.01	71.42±0.97	72.91±0.39	74.19±0.39	55.06±1.39	62.95±0.36	69.10±2.28	57.10±1.40	79.90±0.09	82.20±0.10	82.21±1.78	**83.39±0.07**
	ARI	46.09±0.02	60.36±0.88	61.25±0.51	62.83±0.45	42.63±1.63	51.47±0.73	60.38±2.15	44.70±1.00	72.84±0.09	76.40±0.10	75.58±1.85	**78.52±0.08**
	F1	58.33±0.03	66.36±0.34	67.29±0.29	68.63±0.33	62.64±0.97	71.55±0.29	76.89±2.18	61.10±0.90	82.58±0.08	87.30±0.10	85.79±2.48	**89.23±0.06**

the embedded representations for clustering, which contains SDCN [18], DFCN [21], and AGCC [24]. These methods integrate GCN with AE from different perspectives to jointly train the clustering network.

3) Training Procedure: The training of our method R^{2}FGC includes two phases. First, following [21], the AE and GAE are pretrained independently for 30 epochs to minimize their respective reconstruction loss functions. Both the subnetworks are integrated into a united framework for another 100 epochs to obtain the initial representations and cluster centroids. Then, we train the whole network for at least 300 epochs until convergence to minimize the total loss in (9). Following the compared methods, to alleviate the adverse influence of the randomness, we repeat the experiment ten times to evaluate our method and report the mean values and the standard deviations (i.e., mean±std) of the considered metric values. We implement our method using PyTorch 1.8.0 and PyTorch Geometric 1.7.2, which can easily train GNNs for a variety of applications associated with graph-structured data.

4) Parameter Settings: For a fair comparison, we adopt the same parameter setting for AE and GAE as [21], i.e., the layers of the encoder (/decoder) for AE and GAE are set to 4 and 3, respectively; the dimensions of the encoder (/decoder) for AE are set to 128, 256, 512, and 20 in turn; the dimensions of the encoder (/decoder) for GAE are set to 128, 256, and 20 in turn. The network is trained with the Adam optimizer. The learning rate is set to $5e^{-5}$ for ACM, $1e^{-4}$ for DBLP, and $1e^{-3}$ for AMAP, CITE, and HHAR, respectively. The hyperparameters M_1 and M_2 are set to {256, 8}. Moreover, the parameters $\alpha, \eta, \beta, \epsilon, \kappa$ are set to 0.1, 0.2, 0.8, $5e^3$, and 10, respectively. The optimization stops when the validation loss comes to a plateau.

5) Evaluation Metrics: To evaluate the clustering performance of each compared method, we adopt four widely used evaluation metrics following [18], i.e., accuracy (ACC), normalized mutual information (NMI), average rand index (ARI), and macro $F1$-score ($F1$). For each metric, a larger value implies a better clustering result.

B. Performance Comparison (RQ1)

The experimental results of our method and 11 compared methods on five benchmark datasets are reported in Table II, in which the bold and underlined values indicate the best results in all the methods and all the baselines, respectively. From these results, we have the following observations.

1) Compared with shallow clustering method k-means, these deep graph clustering methods clearly show preferable performance. It indicates that the strong capability for learning representation of deep neural network methods enables exploiting more meaningful information from graph-structured data for clustering.
2) The purely AE-based methods (AE, DEC, and IDEC) perform worse than the methods combining AE and GCN (SDCN, DFCN, and AGCC) in most cases. The reason may be that the AE-based methods only leverage the attribute information to learn the latent representation, which overlooks the structure-level semantic information. Similarly, the purely GCN-based methods (GAE, VGAE, DAEGC, and ARGA) also show inferior performance than SDCN, DFCN, and AGCC in most circumstances. It indicates that integrating AE into GCN can capture the attribute and structure information more effectively from complementary views.

3) Our method R^{2}FGC achieves the best clustering performance compared with all the baselines in terms of the four considered metrics over all the datasets. For both graph and nongraph data, our approach represents a significant improvement over the baselines. For example, compared with the best results among all the baselines, for the ACM dataset, our method relatively improves 1.68%, 4.35%, 5.10%, and 1.82% on ACC, NMI, ARI, and $F1$; for the AMAP dataset, our method improves 5.72%, 6.75%, 11.55%, and 5.18% on ACC, NMI, ARI, and $F1$; and for the DBLP dataset, our method improves 6.51%, 16.29%, 19.87%, and 6.39% on ACC, NMI, ARI, and $F1$, respectively.
4) The reasons for the superiority of our method R^{2}FGC are that 1) R^{2}FGC extracts the inherent relational information based on AE and GAE from both local and global views under augmentation, which allows for better exploration of both attribute and structure information; 2) Under augmentation, R^{2}FGC preserves the consistent relationship among the nodes but not the latent representations, which expects to learn more essential representations of the semantic information; 3) R^{2}FGC decreases the redundant relationship among the nodes for learning discriminative and meaningful representations, which can better serve the graph clustering; 4) R^{2}FGC couples AE and GAE together in the representation fusion mechanism to fully integrate and refine the attribute and structure information; and 5) R^{2}FGC also brings the PR to mitigate the possible over-smoothing problem caused by GAE to promote the clustering performance. With the addition of relationship extraction, REpre, and de-redundancy strategies, R^{2}FGC outperforms all the baselines upon the fusion mechanism of AE and GAE and the regularization method of alleviating oversmoothing.

C. Ablation Study (RQ2)

In this section, to further investigate the validity of our proposed method, we conduct some ablation experiments to study the contribution of each component of R^{2}FGC. We mainly focus on the influence of global-view relationship extraction (gloRE), local-view relationship extraction (locRE), REpre, REder, and PR. In addition, we make some discussion on the proposed global sampling strategy.

1) Effects of gloRE and locRE: In the relationship extraction module, we explore the inherent relationship from both global and local views. The former view learns the global relationship of the nodes and the latter concerns the neighbor relationship. We perform some ablation experiments to verify the respective effectiveness of the global- and local-view strategies. Specifically, we consider the following two cases.

1) *R^{2}FGC w/o gloRE:* R^{2}FGC without considering the gloRE.
2) *R^{2}FGC w/o locRE:* R^{2}FGC without considering the locRE.

The corresponding results are displayed in Table III. From the comparison of R^{2}FGC w/o gloRE and R^{2}FGC w/o locRE, for the ACM dataset, locRE has a greater effect than the global-view one on clustering in terms of ACC, NMI, ARI, and $F1$, while for the CITE and DBLP datasets, the gloRE may have a more prominent contribution. As for the AMAP and HHAR datasets, R^{2}FGC w/o gloRE and R^{2}FGC w/o locRE show close metric values, which indicates that global- and local-view extractions almost play equal roles. Moreover, R^{2}FGC consistently shows better performance than R^{2}FGC w/o gloRE and R^{2}FGC w/o locRE over the five considered datasets. Hence, these results illustrate that both the views are necessary and important for achieving good clustering performance.

2) Effects of gloRE, locRE, and PR: In addition, REpre is used to learn the effective representations by preserving the consistent relationship information, whereas the REder conduces to reduce the confusing information, which benefits obtaining discriminative embeddings. Moreover, we adopt PR to relieve the oversmoothing issue. Hence, we also explore their respective efficiencies in the ablation experiments, i.e., four cases are considered as follows.

1) *R^{2}FGC w/o REpre:* R^{2}FGC without considering REpre.
2) *R^{2}FGC w/o REder:* R^{2}FGC without considering REder.
3) *R^{2}FGC w/o REpre and REder:* R^{2}FGC without both REpre and de-redundancy.
4) *R^{2}FGC w/o PR:* R^{2}FGC without adopting the PR trick.

The corresponding results are also shown in Table III. Comparing R^{2}FGC w/o REpre with R^{2}FGC w/o REder, it is observed that REpre outperforms REder for the ACM dataset; REder shows more significant power to improve the clustering performance for the AMAP, CITE, and DBLP datasets; these two strategies have almost equal impact on the HHAR dataset. Moreover, by contrasting with R^{2}FGC w/o REpre & REder, both R^{2}FGC w/o REpre and R^{2}FGC w/o REder give better clustering results with higher metric values, which implies that both REpre and REder possess the capability to promote the effect of graph clustering. In addition, by comparing R^{2}FGC w/o PR and R^{2}FGC w/o REpre & REder, for the ACM, AMAP, and HHAR datasets, the oversmoothing issue has a more significant impact on clustering performance, whereas for the CITE and DBLP datasets, the relationship extraction is more important for good performance. For example, R^{2}FGC on the ACM dataset has 1.99% relative improvement over R^{2}FGC w/o PR in terms of NMI; R^{2}FGC on the CITE dataset obtains 6.28% improvement over R^{2}FGC w/o REpre & REder in terms of ARI. Hence, these results demonstrate that all the proposed components in R^{2}FGC are efficient for reaching informative representation and good performance for graph clustering.

3) Discussion on Global Sampling Strategy: To further illustrate the effectiveness of the QMC inverse degree-weighted distribution sampling for extracting global anchors, we perform experiments to compare it with two MC cases on the AMAP, DBLP, and HHAR datasets, i.e., we consider the following three cases.

1) *R^{2}FGC With QMC Global Sampling (Ours):* R^{2}FGC with considering QMC inverse degree-weighted distribution sampling, i.e., the low-discrepancy point set is used in multinomial sampling.

TABLE III

ABLATION STUDY ON FIVE BENCHMARK DATASETS (MEAN±STD). THE RESULTS SHOW THE CONTRIBUTIONS OF GLORE, LOCRE, REPRE, REDER, AND PR IN THE PROPOSED METHOD AND THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**

Dataset	Model	ACC	NMI	ARI	F1
ACM	R^{2}FGC w/o gloRE	92.39±0.14	72.39±0.44	78.72±0.37	92.40±0.14
	R^{2}FGC w/o locRE	92.21±0.16	72.17±0.53	78.53±0.43	92.19±0.16
	R^{2}FGC w/o REpre	92.13±0.23	71.88±0.67	78.09±0.56	92.27±0.21
	R^{2}FGC w/o REder	92.35±0.17	72.34±0.55	78.62±0.46	92.41±0.18
	R^{2}FGC w/o REpre & REder	92.05±0.16	71.31±0.49	77.81±0.41	92.11±0.16
	R^{2}FGC w/o PR	91.93±0.18	71.01±0.51	77.44±0.47	91.95±0.18
	R^{2}FGC (Ours)	**92.43±0.18**	**72.42±0.53**	**78.72±0.47**	**92.45±0.18**
AMAP	R^{2}FGC w/o gloRE	81.21±0.07	73.79±0.18	66.04±0.45	75.14±0.45
	R^{2}FGC w/o locRE	81.23±0.06	73.81±0.22	66.15±0.51	75.03±0.48
	R^{2}FGC w/o REpre	81.24±0.06	73.81±0.19	66.20±0.42	75.21±0.33
	R^{2}FGC w/o REder	80.85±0.45	73.06±0.69	66.00±0.43	74.88±0.46
	R^{2}FGC w/o REpre & REder	80.75±0.07	72.87±0.23	65.83±0.36	74.51±0.40
	R^{2}FGC w/o PR	80.39±0.61	72.50±0.82	65.53±0.67	74.35±0.82
	R^{2}FGC (Ours)	**81.28±0.05**	**73.88±0.17**	**66.25±0.36**	**75.29±0.32**
CITE	R^{2}FGC w/o gloRE	69.84±0.53	44.47±0.32	45.61±0.41	64.77±0.32
	R^{2}FGC w/o locRE	70.25±0.51	44.99±0.40	46.65±0.37	64.93±0.43
	R^{2}FGC w/o REpre	70.03±0.58	44.37±0.64	46.03±0.57	64.73±0.48
	R^{2}FGC w/o REder	68.84±0.45	43.06±0.47	44.59±0.51	64.37±0.38
	R^{2}FGC w/o REpre & REder	68.20±0.41	42.97±0.32	44.29±0.41	64.30±0.29
	R^{2}FGC w/o PR	69.70±0.57	44.36±0.43	45.89±0.39	64.97±0.29
	R^{2}FGC (Ours)	**70.60±0.45**	**45.39±0.37**	**47.07±0.30**	**65.28±0.12**
DBLP	R^{2}FGC w/o gloRE	80.15±0.40	49.74±0.52	55.34±0.26	79.73±0.37
	R^{2}FGC w/o locRE	80.72±0.27	50.73±0.41	56.23±0.52	80.21±0.25
	R^{2}FGC w/o REpre	80.82±0.25	50.56±0.40	56.04±0.52	80.31±0.24
	R^{2}FGC w/o REder	79.63±0.44	49.41±0.49	55.65±0.68	79.69±0.30
	R^{2}FGC w/o REpre & REder	78.95±0.23	48.77±0.40	55.27±0.32	79.65±0.22
	R^{2}FGC w/o PR	80.73±0.14	49.59±0.26	55.88±0.25	80.23±0.16
	R^{2}FGC (Ours)	**80.95±0.20**	**50.82±0.32**	**56.34±0.42**	**80.54±0.19**
HHAR	R^{2}FGC w/o gloRE	88.82±0.05	83.38±0.04	78.43±0.07	89.13±0.05
	R^{2}FGC w/o locRE	88.87±0.05	83.32±0.06	78.45±0.07	89.18±0.05
	R^{2}FGC w/o REpre	88.87±0.05	83.32±0.07	78.45±0.08	89.18±0.05
	R^{2}FGC w/o REder	88.83±0.05	83.37±0.04	78.44±0.06	89.15±0.05
	R^{2}FGC w/o REpre & REder	88.79±0.04	83.01±0.04	78.05±0.05	88.85±0.04
	R^{2}FGC w/o PR	87.98±0.03	82.84±0.08	77.45±0.05	88.30±0.03
	R^{2}FGC (Ours)	**88.91±0.05**	**83.39±0.07**	**78.52±0.08**	**89.23±0.06**

2) *R^{2}FGC With MC Global Sampling:* R^{2}FGC with considering MC inverse degree-weighted distribution sampling, i.e., the uniform random numbers are used.
3) *R^{2}FGC With SRS:* R^{2}FGC with considering simple random sampling for drawing global anchors.

The results are depicted in Fig. 2. Comparing the three strategies, R^{2}FGC with QMC global sampling shows better performance over the three considered datasets in terms of the average ACC, NMI, ARI, and $F1$ scores. Moreover, R^{2}FGC with MC global sampling outperforms R^{2}FGC with SRS, which implies that inverse degree-weighted distribution sampling is indeed effective to avoid poor representations. In addition, from the error bars in Fig. 2, we can also find that R^{2}FGC with QMC global sampling leads to smaller variances for the metric values, which benefits from the high convergence rate of the QMC sampling strategy. The sampled global anchor set is a better representation of the target distribution, which motivates the subsequent representation learning to have a better and more stable performance. In this way, our proposed sampling method guarantees good robustness to relationship extraction and thus to clustering performance.

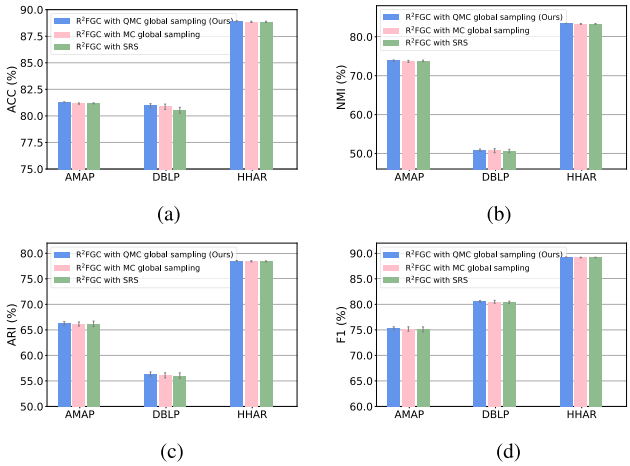


Fig. 2. Performance comparisons with respect to different global sampling strategies on the AMAP, DBLP, and HHAR datasets. (a) ACC. (b) NMI. (c) ARI. (d) $F1$.

D. Parameter Sensitivity Analysis (RQ3)

In this section, we examine the sensitivity of the proposed R^{2}FGC to hyperparameters. For gloRE and locRE in Section IV-B, we need to predefine the numbers of the

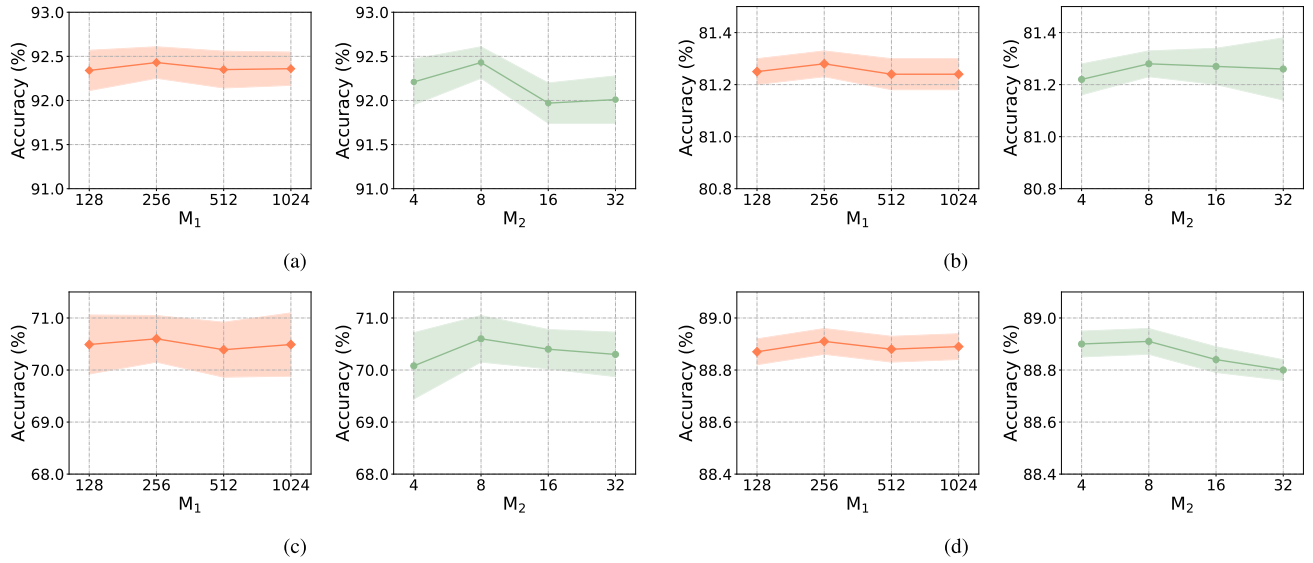


Fig. 3. Performance comparisons with respect to different amounts of global anchors M_1 and local anchors M_2 on (a) ACM, (b) AMAP, (c) CITE, and (d) HHAR datasets.

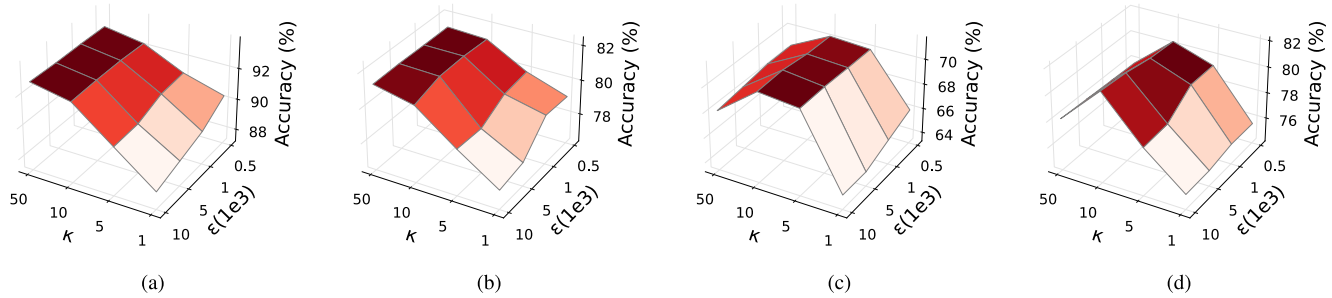


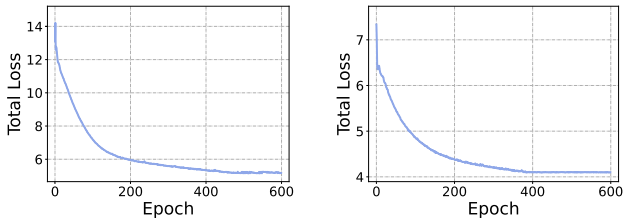
Fig. 4. Performance comparisons with respect to different loss weight parameters κ and ϵ on (a) ACM, (b) AMAP, (c) CITE, and (d) DBLP datasets.

global and local anchors M_1 and M_2 for sampling. Hence, we investigate the effect of varying M_1 and M_2 on the ACM, AMAP, CITE, and HHAR datasets. For each dataset, we consider $M_1 = \{128, 256, 512, 1024\}$ and $M_2 = \{4, 8, 16, 32\}$. When M_1 is varied, we fix M_2 to its optimal setting as in Section V-A, and vice versa. In addition, we explore the impact of two loss weight parameters ϵ and κ on the ACM, AMAP, CITE, and DBLP datasets. We vary ϵ across $\{5e^2, 1e^3, 5e^3, 1e^4\}$ and κ across $\{1, 5, 10, 50\}$. The results are depicted in Figs. 3 and 4, respectively.

1) Performance of Different Amounts of Global Anchors: From Fig. 3, it can be seen that the average accuracies for the considered datasets are relatively stable as M_1 changes. It may be due to that with the QMC multinomial sampling, the drawn anchors can well mimic the defined inverse degree-weighted distribution even if M_1 is small. It helps solve the problem caused by varying qualities of the learned representations for the nodes with different degrees. On the other hand, we draw different samples in different training epochs based on the randomization strategy, which increases the diversity of the samples to catch a broad relationship, even with a small number of global anchors. Therefore, the clustering performance is robust to the number of global anchors based on the proposed sampling strategy.

2) Performance of Different Amounts of Local Anchors: As for M_2, it can be found that on the four datasets, as M_2 increases, it promotes the clustering performance first and then shows a weakening tendency. The possible reason may be that small M_2 cannot well collect the neighboring information, whereas large M_2 may absorb nodes involved in other clusters, which can disturb the extraction of local relationship. Hence, a moderate number of local anchors is preferable, and a well-designed deterministic sampling is desirable to avoid the intake of inconsistent information from other nodes.

3) Performance of Different Amounts of Loss Weights: As shown in Fig. 4, when κ is small, increasing ϵ leads to a decrease in model performance. This is because large ϵ enhances the information aggregation ability of the nodes, which is equivalent to a deep GCN and thus increases the risk of oversmoothing, while small κ means a low self-supervision ability, which results in poor cohesion and insufficient discrimination in node representations. With the increase in κ, better representation cohesion is achieved, and increasing ϵ appropriately is promising to improve the performance by balancing the strength of neighbor aggregation in GCN and the weakness of oversmoothness. However, when ϵ becomes excessively large, there may be a slight decline in performance



(a) (b)

Fig. 5. Curves of the training loss against the number of epochs on (a) AMAP and (b) HHAR datasets.

due to the oversmoothing issue on some datasets. In addition, when ϵ is fixed, increasing κ results in an increasing trend in model performance on the ACM and AMAP datasets. However, on the CITE and DBLP datasets, excessively large κ leads to a decreasing trend. One possible reason is that CITE and DBLP represent citation networks and author networks, respectively, where different articles and individuals may belong to distinct disciplines or communities. Forcing strong cohesion in these cases may lead to suboptimal results. Overall, we recommend to set κ around 10 and ϵ around $5e^3$ for satisfying performance. When dealing with a new dataset, a small-scale hyperparameter tuning around the recommended values is needed due to the dataset's specific characteristics.

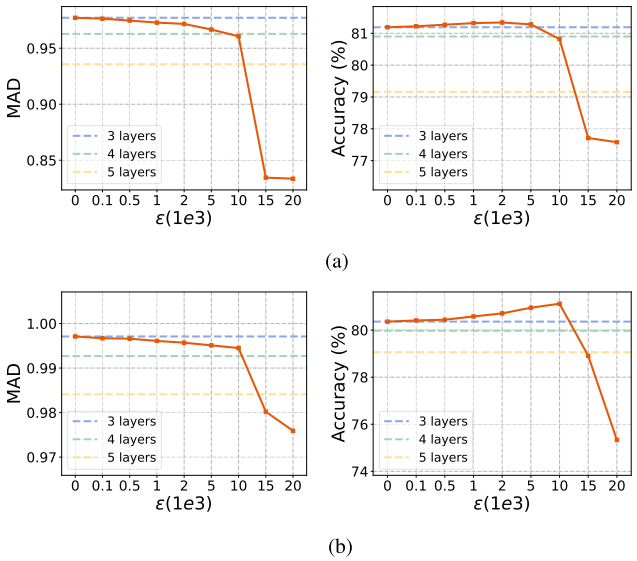
E. Empirical Convergence Analysis (RQ4)

In this section, we analyze the convergence of our proposed method R^{2}FGC, and the curves of the training losses are shown in Fig. 5. It can be observed that our method demonstrates graceful convergence across different datasets AMAP and HHAR. The reason behind this can be attributed to our pretraining learning based on AE and GAE, which provides us with a well-initialized representation. As a result, the initial loss optimization has the correct gradient direction, leading to a rapid decrease in loss. In addition, our method effectively maintains the relational similarity between nodes in the graph while reducing the redundancy of learned representations. This allows the learned representations to possess highly rich semantic information and strong discriminative capabilities. It enables similar nodes closer to each other while better distinguishing unrelated nodes, facilitating the formation of clusters. This also motivates the training objective to converge to lower value, leading to better clustering performance.

F. Analysis of Oversmoothing Issue (RQ5)

To verify the superiority of the proposed PR loss in alleviating oversmoothing issue, we compare the effects of different GCN layers in the GAE encoder and different values of ϵ by mean average distance (MAD) and clustering performance (i.e., ACC). MAD reflects the smoothness of node representations by calculating the mean of the average cosine distance between the nodes and other nodes [64]. A smaller MAD indicates a higher global smoothness. The analysis results are shown in Fig. 6.

It can be observed that as the number of GCN layers in the GAE encoder increases, indicated by the dashed line in



(a)

(b)

Fig. 6. Comparisons of the MAD and clustering ACC with respect to different GCN layers in the GAE encoder and regularization parameter ϵ on (a) AMAP and (b) DBLP datasets.

the figure, both MAD and ACC exhibit a decreasing trend. It implies that larger GCN layers cause nodes to immensely absorb information from farther neighbors and thus can lead to indistinguishable node representations, exacerbating the oversmoothing issue and resulting in performance degradation. On the other hand, our proposed PR loss shows a slight decrease in MAD and a gradual increase in clustering performance as ϵ increases to a certain value (e.g., $2e^3$ for AMAP, $1e^4$ for DBLP). This suggests that our PR loss is equivalent to simulating a GCN of a fractional layer, which possesses the capability to ease the increase in smoothness and meanwhile enhance the expressive power of node representations, thereby promoting the clustering performance. However, when ϵ is particularly large, both MAD and ACC decrease sharply. This is because excessively large ϵ amplifies the risk of oversmoothness. Therefore, selecting an appropriate weight is crucial in balancing node expressiveness and the oversmoothing issue.

G. Visualization of Clustering Results (RQ5)

To visually verify the validity of our proposed R^{2}FGC, we plot 2-D t-distributed stochastic neighbor embedding (t-SNE) visualizations [66] for the learned representations on the ACM, CITE, DBLP, and HHAR datasets. We compare the t-SNE visualizations of the embeddings resulting from R^{2}FGC with those from the raw data and DFCN (the best method among the baselines in Section V-B) to enable a visual comparison. The plots are shown in Fig. 7. The results of t-SNE on the four raw data clearly have poor separability for different clusters. Compared with the raw data, more distinguishing visualizations in R^{2}FGC and DFCN demonstrate that deep graph clustering methods indeed make great performance improvements. Comparing R^{2}FGC with DFCN, the latent representations obtained by our method R^{2}FGC show better separability for different clusters, where the samples from the same cluster have better aggregation and those from different clusters have a bigger gap. Such a phenomenon illustrates

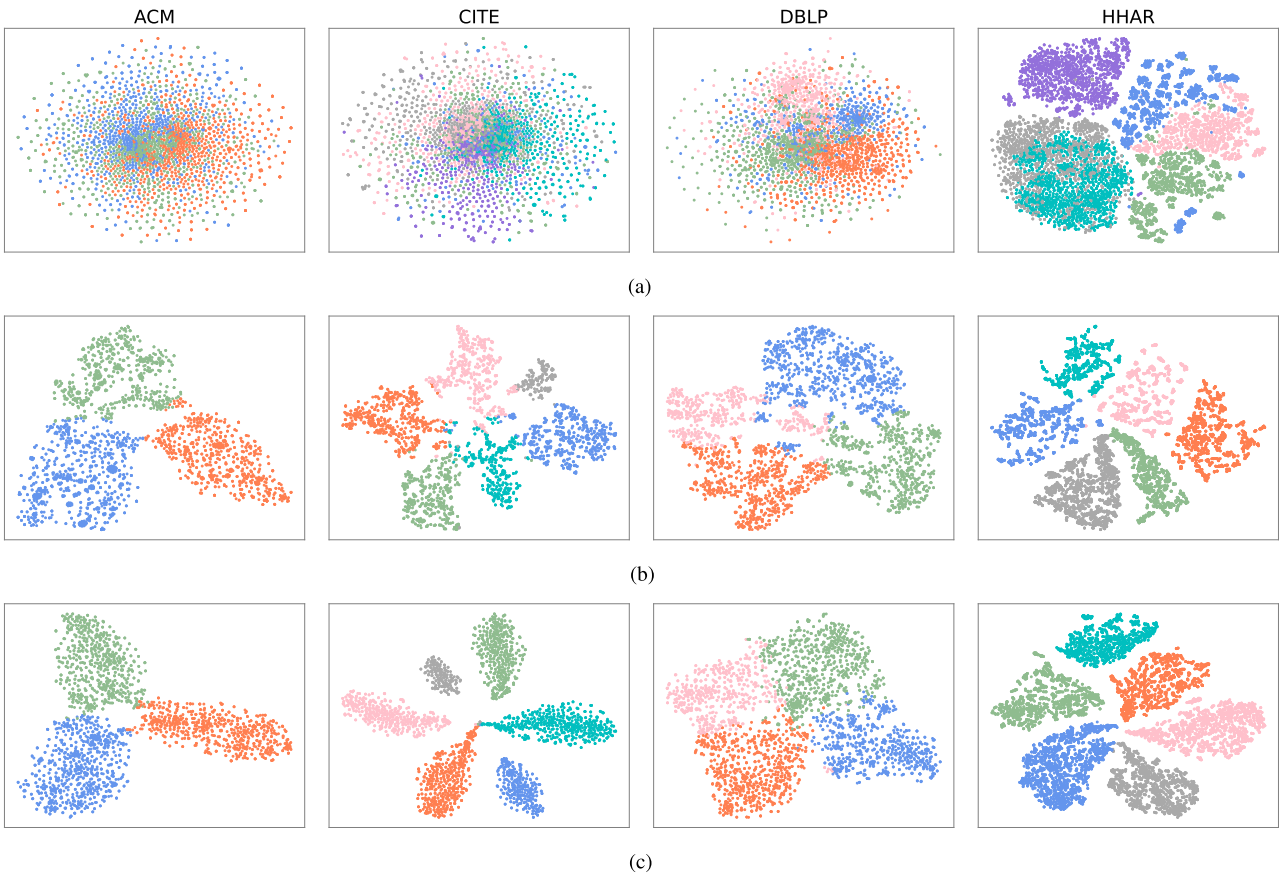


Fig. 7. t-SNE visualizations on the ACM, CITE, DBLP, and HHAR datasets. Distributions of the embeddings from (a) raw data, (b) DFCN, and (c) our proposed R^{2}FGC.

that our proposed method learns more discriminative representations and produces more effective cluster assignments compared with the state-of-the-art methods.

VI. Conclusion

In this article, we study self-supervised deep graph clustering and propose a novel method termed R^{2}FGC. R^{2}FGC introduces the relational learning for the graph-structured data, in which the attribute- and structure-level relationship information among nodes are extracted based on AE and GAE. To achieve effective representations, R^{2}FGC preserves consistent relationships among the nodes under augmentation, whereas the redundancy relationship is filtered for discriminative representations. R^{2}FGC also cooperates a representation fusion mechanism with the relational learning to instruct downstream self-supervised clustering tasks jointly. The experimental results on various benchmark datasets demonstrate the validity and superiority of the proposed method. In the future, we aim to extend relational learning to other scenarios including multiview graph clustering, interpretable clustering, and other promising applications such as face clustering and text clustering.

Acknowledgment

The authors are grateful to the anonymous reviewers for critically reading the article and for giving important suggestions to improve their article.

References

[1] F. Liu et al., "Deep learning for community detection: Progress, challenges and opportunities," 2020, *arXiv:2005.08225*.

[2] X.-R. Sheng, D.-C. Zhan, S. Lu, and Y. Jiang, "Multi-view anomaly detection: Neighborhood in locality matters," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 4894–4901.

[3] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8725–8735.

[4] M. Xu, H. Wang, B. Ni, H. Guo, and J. Tang, "Self-supervised graph-level representation learning with local and global structure," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11548–11558.

[5] W. Ju et al., "Unsupervised graph-level representation learning with hierarchical contrasts," *Neural Netw.*, vol. 158, pp. 359–368, Jan. 2023.

[6] X. Luo et al., "CLEAR: Cluster-enhanced contrast for self-supervised graph representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 8, 2022, doi: 10.1109/TNNLS.2022.3177775.

[7] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001, pp. 1–8.

[8] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[9] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[10] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, "ClusterGAN: Latent space clustering in generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 4610–4617.

[11] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 10, pp. 8547–8555.

[12] H. Zhong et al., "Graph contrastive clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9224–9233.

[13] C. Liu et al., "Self-guided partial graph propagation for incomplete multiview clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 3, 2023, doi: 10.1109/TNNLS.2023.3244021.

[14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[15] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.

[16] W. Ju et al., "A comprehensive survey on deep graph representation learning," 2023, *arXiv:2304.05055*.

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

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

[19] Z. Peng, H. Liu, Y. Jia, and J. Hou, "Attention-driven graph clustering network," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 935–943.

[20] H. Zhao, X. Yang, Z. Wang, E. Yang, and C. Deng, "Graph debiased contrastive learning with joint representation clustering," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 3434–3440.

[21] W. Tu et al., "Deep fusion clustering network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 11, pp. 9978–9987.

[22] P. Zhu, J. Li, Y. Wang, B. Xiao, S. Zhao, and Q. Hu, "Collaborative decision-reinforced self-supervision for attributed graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 18, 2022, doi: 10.1109/TNNLS.2022.3171583.

[23] H. Zhang, P. Li, R. Zhang, and X. Li, "Embedding graph auto-encoder for graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 25, 2022, doi: 10.1109/TNNLS.2022.3158654.

[24] X. He, B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin, "Parallelly adaptive graph convolutional clustering model," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 26, 2022, doi: 10.1109/TNNLS.2022.3176411.

[25] Y. Liu et al., "Deep graph clustering via dual correlation reduction," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7603–7611.

[26] X. Peng, J. Cheng, X. Tang, J. Liu, and J. Wu, "Dual contrastive learning network for graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 7, 2023, doi: 10.1109/TNNLS.2023.3244397.

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

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

[29] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[30] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[33] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[34] J. Yuan, X. Luo, Y. Qin, Y. Zhao, W. Ju, and M. Zhang, "Learning on graphs under label noise," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[35] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[36] W. Ju et al., "TGNN: A joint semi-supervised framework for graph-level classification," 2023, *arXiv:2304.11688*.

[37] W. Ju, J. Yang, M. Qu, W. Song, J. Shen, and M. Zhang, "KGNN: Harnessing kernel-based networks for semi-supervised graph classification," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, Feb. 2022, pp. 421–429.

[38] W. Ju et al., "GLCC: A general framework for graph-level clustering," 2022, *arXiv:2210.11879*.

[39] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[40] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1753–1759.

[41] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.

[42] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, and Y. Yang, "Rank-constrained spectral clustering with flexible embedding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6073–6082, Dec. 2018.

[43] Z. Li, F. Nie, X. Chang, Y. Yang, C. Zhang, and N. Sebe, "Dynamic affinity graph construction for spectral clustering using multiple features," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6323–6332, Dec. 2018.

[44] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang, "MGAE: Marginalized graph autoencoder for graph clustering," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 889–898.

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

[46] S. Pan, R. Hu, S.-F. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020.

[47] S. Fan, X. Wang, C. Shi, E. Lu, K. Lin, and B. Wang, "One2Multi graph autoencoder for multi-view graph clustering," in *Proc. Web Conf.*, Apr. 2020, pp. 3070–3076.

[48] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[49] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[51] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[52] J.-B. Grill et al., "Bootstrap your own latent—A new approach to self-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21271–21284.

[53] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2021.

[54] C. Yan et al., "ZeroNAS: Differentiable generative adversarial networks search for zero-shot learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9733–9740, Dec. 2022.

[55] W. Ju et al., "Kernel-based substructure exploration for next POI recommendation," 2022, *arXiv:2210.03969*.

[56] N. Lee, D. Hyun, J. Lee, and C. Park, "Relational self-supervised learning on graphs," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 1054–1063.

[57] W. Ju et al., "Few-shot molecular property prediction via hierarchically structured learning on relation graphs," *Neural Netw.*, vol. 163, pp. 122–131, Jun. 2023.

[58] J. Gasteiger, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," 2019, *arXiv:1911.05485*.

[59] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., 1999.

[60] C. Lemieux, *Monte Carlo and Quasi-Monte Carlo Sampling*. New York, NY, USA: Springer, 2009.

[61] S.-Y. Yi, Z. Liu, M.-Q. Liu, and Y.-D. Zhou, "Global likelihood sampler for multimodal distributions," *J. Comput. Graph. Statist.*, vol. 32, no. 3, pp. 927–937, 2023.

[62] S.-Y. Yi and Y.-D. Zhou, "Model-free global likelihood subsampling for massive data," *Statist. Comput.*, vol. 33, no. 1, p. 9, Feb. 2023.

[63] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," 2019, *arXiv:1905.10947*.

[64] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3438–3445.

[65] A. Stisen et al., "Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2015, pp. 127–140.

[66] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

Siyu Yi received the B.S. and M.S. degrees in mathematics from Sichuan University, Chengdu, Sichuan, China, in 2017 and 2020, respectively. She is currently pursuing the Ph.D. degree in statistics with Nankai University, Tianjin, China.

Her research interests include graph representation learning, design of experiments, and subsampling in big data.

Wei Ju (Member, IEEE) received the B.S. degree in mathematics from Sichuan University, Chengdu, Sichuan, China, in 2017, and the Ph.D. degree in computer science from Peking University, Beijing, China, in 2022.

He is currently a Post-Doctoral Research Fellow in computer science with Peking University. His current research interests lie primarily in the area of machine learning on graphs, including graph representation learning and graph neural networks, and interdisciplinary applications, such as knowledge graphs, drug discovery, and recommender systems. He has published more than 30 articles in top-tier venues.

Dr. Ju has won the Best Paper Finalist in IEEE International Conference on Data Mining (ICDM) 2022.

Yifang Qin received the B.S. degree from the School of Electronics Engineering and Computer Science (EECS), Peking University, Beijing, China, in 2023. He is currently pursuing the Ph.D. degree with the School of Computer Science, Peking University.

His research interests include graph representation learning and recommender systems.

Xiao Luo received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 2017, and the Ph.D. degree from the School of Mathematical Sciences, Peking University, Beijing, China, in 2022.

He is currently a Post-Doctoral Researcher with the Department of Computer Science, University of California at Los Angeles, Los Angeles, CA, USA. His research interests include machine learning on graphs, image retrieval, statistical models, and bioinformatics.

Luchen Liu received the Ph.D. degree in computer science from Peking University, Beijing, China, in 2020.

He is currently a Post-Doctoral Research Fellow in computer science with Peking University. His current research interests lie primarily in the area of deep learning for temporal graph data and interdisciplinary applications, such as intelligent healthcare and quantitative investment.

Yongdao Zhou received the B.S. degree in mathematics and the M.S. and Ph.D. degrees in statistics from Sichuan University, Chengdu, China, in 2002, 2005, and 2008, respectively.

After graduation, he joined Sichuan University, where he was a Professor after 2015. In 2017, he joined Nankai University, Tianjin, China, where he is currently a Professor of statistics. He has published more than 60 articles and five monographs. His research interests include design of experiments and big data analysis.

Dr. Zhou's research publications have won Best Paper Awards in World Congress on Engineering (WCE) 2009 and Sci Sin Math in 2023.

Ming Zhang received the B.S., M.S., and Ph.D. degrees in computer science from Peking University, Beijing, China, in 1988, 1991, and 2005, respectively.

She is currently a Full Professor with the School of Computer Science, Peking University. She has published more than 200 research papers on text mining and machine learning in top journals and conferences.

Prof. Zhang is a member of the Advisory Committee of the Ministry of Education (MOE), China, and the Chair of ACM Special Interest Group on Computer Science Education (SIGCSE) China. She is one of the 15 members of ACM/IEEE CC2020 Steering Committee. She won the Best Paper of International Conference on Machine Learning (ICML) 2014 and the Best Paper Nominee of the Web Conference (WWW) 2016. She is the leading author of several textbooks on data structures and algorithms in Chinese, and the corresponding course is awarded as the National Elaborate Course, National Boutique Resource Sharing Course, National Fine-designed Online Course, and National First-Class Undergraduate Course by MOE, China.