Neurocomputing 570 (2024) 127147

Contents lists available at ScienceDirect

Neurocomputing

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

Deep Contrastive Clustering via Hard positive sample Debiased ☆

Xinyu Zhang [a,1], Huiying Xu [a,1,*], Xinzhong Zhu [a,b], Yuhang Chen [a]

[a] School of Computer Science and Technology, Zhejiang Normal University, Jinhua, 321004, China
[b] AI Research Institute of Beijing Geekplus Technology Co., Ltd., Beijing, 100101, China

ARTICLE INFO

Communicated by A. Iosifidis

Keywords:
Deep Contrastive Clustering
Hard Sample
Debiased

ABSTRACT

Deep graph clustering aims to reveal the underlying information of the graph and provide accurate embedding for the node clustering task, in which contrastive learning plays an important role. However, the commonly used contrastive loss function incorrectly classifies elements outside the diagonal of the cross view as negative samples, which contain a large number of positive sample pairs. In order to overcome the above problems, we propose a new deep graph contrastive clustering method, which combines hard positive sample debiasing and sample pair weighting, and improves the recognition ability of the network by removing the potential positive sample pairs and hard sample pair weighting in the negative sample pair of the loss function. More specifically, we developed a symmetric graph neural network to encode node representations. Using two sets of node representations, the correctness of negative cases is increased by clustering to generate high-confidence pseudo-label pairs for labels and confidence. The similarity distribution differences are weighted by adapting to different dataset samples to improve the sample recognition ability. To verify the efficacy of our DCHD, we compare it to existing state-of-the-art methods for node clustering tasks on six real-world datasets. Overall, the experimental results show that our proposed method outperforms current state-of-the-art graph clustering methods.

1. Introduction

In recent years, deep learning has achieved notable success across various domains, including object detection [1,2], recommender systems [3–5], general-purpose audio representations [6], and medical image analysis [7]. This approach effectively leverages neural networks to maximize their capacity for extracting valuable hidden information [8]. Contrastive learning has gained widespread adoption within the realm of deep learning. Within these domains, deep graph clustering, which is focused on partitioning nodes into distinct and non-overlapping clusters, has garnered considerable interest in recent years. Contrastive learning enables models to learn high-quality representations without the need for labels or symbols.

Given the impressive performance exhibited by contrastive learning, an increasing number of deep graph clustering techniques have embraced the contrastive learning framework. Recent research has demonstrated the effectiveness of strategies such as mining hard samples in this context. To be more precise, MoCHi [9] aims to create hard-negative samples through the process of sample synthesis. In the case of Progcl [10], it initially identifies false negative samples by modeling the distributions of both true and false negative samples. Subsequently, it generates an array of diverse negative samples based on the harder negative samples selected. HomoGCL [11] employs homogeneity to estimate the likelihood that neighboring nodes represent positive samples. On the other hand, HSAN [12] assigns weights to hard positive and negative samples to enhance its recognition capabilities. Although established as effective, we would like to highlight two shortcomings in current methodologies: (1) Prior investigations have primarily concentrated on addressing the challenges posed by hard samples through techniques such as weight modulation. However, they tend to overlook the inherent bias introduced by hard positive samples themselves. (2) Furthermore, the distribution of hard positive and hard negative samples exhibits a single pattern across different datasets, where either positive or negative hard samples account for a higher proportion. This centralized distribution, coupled with the constraints imposed by the weight distribution function, may limit the recognition ability of the model. Previous studies [13] has characterized hard samples as either false positives, positive samples exhibiting low similarity, or negative samples demonstrating high similarity. Why should we

☆ This work was supported by the National Natural Science Foundation of China (62376252, 61976196, U22A20102); Key Project of Natural Science Foundation of Zhejiang Province (LZ22F030003).

* Corresponding author.
E-mail addresses: zsdzxy2021@zjnu.edu.cn (X. Zhang), xhy@zjnu.edu.cn (H. Xu), zxz@zjnu.edu.cn (X. Zhu), chenyuhang@zjnu.edu.cn (Y. Chen).
[1] indicates equal contribution as first author.

https://doi.org/10.1016/j.neucom.2023.127147
Received 14 September 2023; Received in revised form 28 November 2023; Accepted 6 December 2023
Available online 18 December 2023

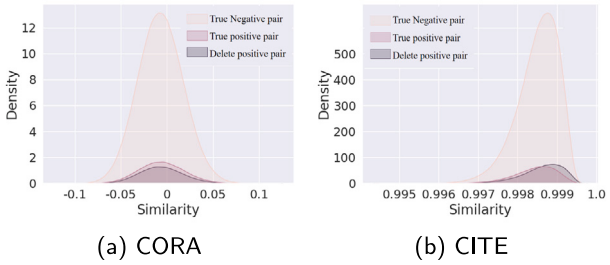

Fig. 1. Density distribution study of the similarity of different nodes of the graph. In order to examine the count of positive sample pairs within the negative example in the infoNCE loss function, we employ a color scheme where yellow signifies the true negative sample pair, red represents the true positive sample pair, and gray denotes the genuine positive sample that has been eliminated as a result of our DCHD methods. (1)It is important to note that since we are classifying all pairs of nodes to estimate the density distribution, the combined gray and red sections represent the entirety of the true positive sample pairs. (2)The figure also illustrates the similarity distribution among initially paired nodes. For instance, in the CITE dataset, a majority of sample pairs exhibit similarity values that are nearly equal to 1 during initialization. This performance analysis emphasizes the fundamental factors contributing to the observed bias within the infoNCE loss function and underscores the disparities in the distribution of difficult samples across diverse datasets.

care about hard samples? Initially, the focus on hard samples was to address the imbalance of training samples, where model parameter updates depend on the gradient of loss. The sample pairs in the loss function are weighted, assigning smaller weights to simple samples. This is done to diminish their impact on the overall loss, effectively reducing the proportion of loss attributed to simple samples. Assign a larger weight to the hard sample, the weight tends to be 1. The primary objective of minimizing the InfoNCE loss is to improve the lower bound of mutual information between the anchor and positive sample, thereby bringing their representations closer together. This is accomplished by employing cosine similarity to assess the distance between their generated vectors, enhancing the similarity between the anchor and corresponding positive samples while diminishing the similarity with negative samples. The above conform to deep learning theory [14–18].

We have generated density distribution plots illustrating the similarity between sample pairs in the initial state of the CORA and CITE datasets in Fig. 1. It is evident that in the CORA dataset, the similarity among node pairs follows a roughly normal distribution centered around a mean of 0. Additionally, a significant proportion of positive sample pairs exhibit low similarity values. In the CITE dataset, the similarity among node pairs roughly conforms to a beta distribution pattern. Notably, the concept of hard positive samples is weakened when a large number of samples exhibit high similarity values. Therefore, we should introduce a larger weight to focus the network's attention on hard negative samples. Therefore, we enhance the discriminative ability of the model solely by weighted hard positive samples. We achieve this through a controlled weighting strategy, allowing us to manage a specific category of hard samples. Additionally, the presence of a substantial proportion of true positive sample pairs is noteworthy, and designating them as negative sample pairs would introduce significant bias. Our approach to debiasing hard positive samples effectively eliminates approximately half of the positive sample pairs, thereby substantially enhancing the model's recognition capabilities.

To address the aforementioned issues, we introduce a novel deep graph clustering method Deep Contrastive Clustering via Hard positive sample Debiased(DCHD). This approach involves the formulation of a hard positive sample debiasing criterion and the implementation of a distinct hard sample weighting strategy. In particular, to enhance the reliability of hard negative sample pairs, we mitigate the model's susceptibility to the influence of hard positive samples through the clustering of pseudo-labels and confidence levels. Furthermore, we introduce a contrastive sample weighting strategy aimed at enhancing the network's recognition capabilities. Initially, we employ a clustering algorithm on the consensus node embeddings to generate clustering pseudo-labels characterized by high confidence. Samples within the same cluster are subsequently identified as potential positive sample pairs. Distinct weighting strategies were employed based on the considered dataset, specifically weighting using hard positive samples or hard negative samples for a particular dataset individually. This article makes the following primary contributions:

- Our DCHD introduces a novel deep graph contrastive clustering approach designed to eliminate the bias exerted by hard positive samples on the model.
- To accommodate variations in the similarity density distribution among sample pairs in different datasets, our DCHD presents a weighting strategy model that exclusively addresses positive or negative samples. This approach helps mitigate adverse effects within the weighting process.
- Furthermore, we conducted thorough experiments on the node clustering task using six benchmark datasets. The results consistently demonstrated that our method outperformed other state-of-the-art deep graph clustering approaches.

The structure of the remainder of this paper is as follows: In Section 2, we offer an overview of related work. Section 3 delves into the methodology, providing comprehensive details and introduce essential notations. Subsequently, Section 4 presents the experimental results. Finally, the last section provides a summary of the paper.

2. Related work

2.1. Contrastive learning on graph

Recently, due to the accomplishments of contrastive learning mechanisms in computer vision and natural language processing, contrastive learning methods are experiencing growing utilization in the fields of deep graph clustering and graph representation learning. In the initial stages, CCA-SSG [19] reduces the correlation of different nodes through affinity constraints. AFGRL [20] applied positive samples that filtered neighbor search and clustering to knowledge distillation. Subsequently, AutoSSL [21] enhances the pre-training effectiveness by conducting a search to fine-tune the weight distribution of the task, leveraging pseudo-homophily as a basis for adjustment. In contrast, NCLA [22] autonomously learns graph enhancement parameters in an end-to-end manner via a multi-head graph attention mechanism and introduces a neighbor contrast loss function. Subsequently, MA-GCL [23] is introduced, which augments the view from the model level through the utilization of three enhancement strategies. Nevertheless, a significant portion of recent research has concentrated on enhancement techniques and train process. Recently, there has been a notable surge in interest regarding the selection of positive and negative samples. In particular, ProGCL [10] calculates the probabilities associated with true and false negative samples to extract hard negative samples and generates synthetic false samples. On the other hand, HSAN [12] dynamically assigns weights to hard positive and negative samples by employing clustering pseudo-labels and confidence levels. However, these approaches tend to overlook the inherent bias introduced by false negative samples on the model. To address this issue, we introduce a novel contrasting deep graph clustering method designed to mitigate the model's susceptibility to the impact of hard true samples. In our proposed approach, hard true samples are eliminated through the utilization of clustering pseudo-labels and confidence levels.

Table 1
Acronym and description.

Acronym	Description
$\mathbf{X} \in \mathbb{R}^{N\times D}$	Attribute matrix
$\mathbf{A} \in \mathbb{R}^{N\times N}$	Adjacency matrix
$\widetilde{\mathbf{X}} \in \mathbb{R}^{N\times D}$	Laplacian filter attribute matrix
$\widetilde{\mathbf{L}_s} \in \mathbb{R}^{N\times N}$	Symmetric Laplacian matrix
$\mathbf{Z}_i \in \mathbb{R}^{N\times d}$	Node embedding of ith view
$S \in \mathbb{R}^{2N\times 2N}$	Cross-view similarity matrix
$\mathbf{Y} \in \mathbb{R}^{N}$	Sample pseudo-labels
$\mathbf{P} \in \mathbb{R}^{N\times N}$	Sample pair pseudo-labels
$\mathbf{H} \in \mathbb{R}^{N\times N}$	High confidence sample pair
$\mathcal{M} \in \mathbb{R}^{2N\times 2N}$	De-biased matrix
$\|\|\cdot\|\|_2$	l-2 regularization

2.2. Hard sample mining

The critical determinants of success in contrastive learning methods lie in the selection of positive and negative samples. Earlier research has demonstrated that negative pairs with high similarity or positive sample pairs with low similarity exhibit significant potential. Given their effectiveness, an increasing number of researchers are now directing their attention towards the mining of hard samples within graph-based scenarios. In particular, GDCL [24] enhances the realism of negative examples by employing a sampling strategy for negative samples. ANML [25] dynamically eliminates indivisible samples through metric learning. Circle loss [26] dynamically adjusts its weight in the backward propagation according to similarity. CuCo [27] uses a scoring function to measure the difficulty of negative sampling and ranks them from easy to difficult. In more recent developments, ProGCL [10] employs a more suitable metric for assessing negative hardness and similarity by estimating the probability that negative samples are genuine. HSAN [12] utilizes a weight modulation function generated through the combination of clustering pseudo-labels and confidence values to dynamically assign weights to both challenging negative and positive samples. Although the prior method demonstrated its efficacy, it overlooked variations in the similarity density distributions among different datasets, leading to suboptimal performance. In our approach, we contend that prioritizing specific categories of hard samples is crucial for different datasets. Building upon this idea, we introduce a controllable sample weighting strategy.

3. Method

In this section, we begin by presenting symbolic notations for graphs and graph contrastive learning methodologies. Subsequently, we introduce our enhanced approach built upon these foundations as shown in Fig. 2.

3.1. Preliminaries and notations

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the node set with C classes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set respectively. Additionally, the attribute matrix and the adjacency matrix are denoted as $\mathbf{X} \in \mathbb{R}^{N\times D}$ and $\mathbf{A} \in \{0,1\}^{N\times N}$, where $\mathbf{A}_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, and D is the dimension of the raw node v. The notations used are summarized in Table 1. Our primary goal is to train a graph encoder capable of generating low-dimensional representations for nodes, which can then be directly utilized for node clustering. GCN [28] is a commonly employed encoder in the field of graphs. However, recent research [10] has demonstrated that the message passing mechanism within this architecture plays a crucial role in shaping the distribution of negative sample different with image contrastive learning. HomoGCL [11] has established that message passing is the focal aspect of contrastive learning. Consequently, researchers have substituted the step involving the aggregation of neighboring nodes in the GCN encoding process with the Laplacian filter. The original message passing mechanism is outlined as follows:

$$\mathbf{X}^{(t+1)} = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}^{(t)}$$

where t represents the times of message passing, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\hat{\mathbf{D}}_i = \sum_i \hat{\mathbf{A}}_{ij}$. As a result, a new message passing approach that designed to exhibit the property of a symmetric Laplacian matrix for non-bipartite and connected graphs, was introduced:

$$\mathbf{X}^{(t+1)} = (\mathbf{I} - \mathbf{L}_s)\mathbf{X}^{(t)}$$

where symmetric Laplacian matrix $\mathbf{L}_s = \hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{L}}\hat{\mathbf{D}}^{-\frac{1}{2}}$, Laplacian matrix $\hat{\mathbf{L}} = \hat{\mathbf{D}} - \hat{\mathbf{A}}$. Here, we illustrate our point using the InfoNCE [29] loss function, one of the most widely adopted loss functions in graph contrastive methods [30], as an example. The embedding v of any node i in one view is considered the anchor. The embedding u of any node i in the alternative view is utilized as the positive sample, while the embeddings of all remaining nodes in both views are designated as negatives.

$$\mathcal{L}_{infoNCE}(u_i, v_j) = -\ log\frac{e^{\theta(v_i,u_i)}}{e^{\theta(v_i,u_i)} + \sum_{j\neq i}(e^{\theta(v_i,v_j)} + e^{\theta(v_i,u_j)})}.$$

Here, $\theta(\cdot)$ represents the cosine similarity between paired samples in the latent space. By minimizing the InfoNCE loss, the objective is to bring similar samples from different views closer together while pushing dissimilar samples apart.

3.2. Laplacian filter encoding

From the above, to mitigate the impact of message passing in graph neural networks, we employ the Laplacian filter to filter out high-frequency noise [12], which is represented by the following equation:

$$\widetilde{\mathbf{X}} = (\prod_{i=1}^{t}(\mathbf{I} - \widetilde{\mathbf{L}}))\mathbf{X} = (\mathbf{I} - \widetilde{\mathbf{L}})^t\mathbf{X}, \tag{1}$$

where $\widetilde{\mathbf{L}}$ is the Laplacian filter, $\mathbf{I}$ is denoted Identity matrix and t is the filtering times. Then we denote node embeddings in the feature matrix $\widetilde{\mathbf{X}}$ with Auto-encoder $\mathcal{T}(\cdot)$ as follow:

$$\begin{aligned}\mathbf{Z}_1 = \mathcal{T}_1(\widetilde{\mathbf{X}}); \mathbf{Z}_1^{v_i} = \frac{\mathbf{Z}_1^{v_i}}{\|\mathbf{Z}_1^{v_i}\|_2}, i = 1, 2, \ldots, N;\\ \mathbf{Z}_2 = \mathcal{T}_2(\widetilde{\mathbf{X}}); \mathbf{Z}_2^{v_j} = \frac{\mathbf{Z}_2^{v_j}}{\|\mathbf{Z}_2^{v_j}\|_2}, j = 1, 2, \ldots, N,\end{aligned} \tag{2}$$

where $\mathbf{Z}_1$ and $\mathbf{Z}_2$ represent the feature embedding of the sample. In this context, both $\mathcal{T}_1(\cdot)$ and $\mathcal{T}_2(\cdot)$ refer to simple multi-layer perceptrons (MLPs) that possess identical architectures but do not share parameters. Consequently, $\mathbf{Z}_1$ and $\mathbf{Z}_2$ encapsulate distinct semantic information. In contrast to GCN, we do not integrate neighboring node information into the encoding process. However, we obtain node embeddings that encompass both attribute and topological information for each sample. Subsequently, the node similarity function $\mathcal{S}$ is introduced to compute the similarity between the ith sample and the jth sample using the following formula:

$$\mathcal{S}(v_i^a, v_j^b) = (\mathbf{Z}_a^{v_i})^{\mathrm{T}}\mathbf{Z}_b^{v_j}, \tag{3}$$

In this formula, $a, b \in \{1, 2\}$ and indicate the view in which node v is situated, while T denotes matrix transpose. The equation comprises two cross-view similarity matrices and two self-correlation similarity matrices.

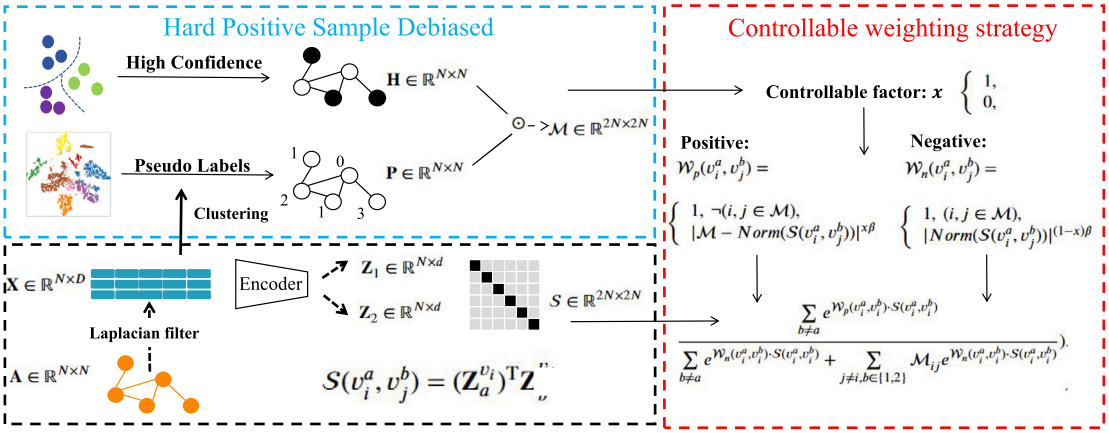

Fig. 2. The pipeline of the DCHD. Firstly generates a Laplace attribute matrix through the Laplacian filter of the original attribute matrix **X** and the adjacency matrix **A**, then it is fed into MLPs encoders that do not share parameters to learn the representations. The representations of $\mathcal{G}$ is utilized to generate sample pseudo-labels **Y** via clustering, and high-confidence samples **H** are screened according to the distance between the sample and the centroids of each cluster. High-confidence positive sample pairs are selected on this basis. In the face of different data sets, we adopt different sample pair weighting strategies: (1) give sample pairs other than $\mathcal{M}$ to the power β of the absolute value of similarity, and (2) calculate the absolute value of their similarity and $\mathcal{M}$ difference to the absolute power of β for diagonal elements of the cross view.

3.3. Hard positive sample de-biased

In this section, we present the hard positive sample debiasing module, which is built upon the enhancement of the InfoNCE loss. Merely designating all nodes except those on the diagonal as negative nodes is not optimal. This approach includes a substantial number of positive sample pairs, and if we treat a pair of nodes that are close to the same cluster as a negative sample pair, it can introduce bias during the backpropagation process. In order to counteract this effect, we eliminate positive sample pairs with a high likelihood of being genuine by utilizing clustering pseudo-labels and confidence measures.

Initially, we derive the pseudo-label $\mathbf{Y} \in \mathbb{R}^N$ for each node by clustering the embeddings **Z**. Reliable positive sample pairs are subsequently generated based on these pseudo-labels. We then choose the top τ samples to form the high-confidence sample set $\mathbf{H} \in \mathbb{R}^M$.

$$\mathbf{P}_{ij} = \begin{cases} 1 & \mathbf{Y}_i = \mathbf{Y}_j, \\ 0 & \mathbf{Y}_i \neq \mathbf{Y}_j. \end{cases} \tag{4}$$

Here, $\mathbf{P}_{ij}$ indicates a pseudo-relationship between the ith sample and the jth sample. Specifically, $\mathbf{P}_{ij} = 1$ implies that the ith and jth samples are more likely to be positive sample pairs, while $\mathbf{P}_{ij} = 0$ signifies that they are more likely to be negative sample pairs.

$$\mathbf{H}_{ij} = \begin{cases} 1 & v_i, v_j \in \mathbf{H}, \\ 0 & \text{else}. \end{cases} \tag{5}$$

where $\mathbf{H}_{ij}$ reveals the credibility relationship between the ith sample and the jth sample. To elaborate, $\mathbf{H}_{ij} = 1$ implies that the relationship between the ith and jth samples is more likely to be true, while $\mathbf{H}_{ij} = 0$ indicates that the relationship between them is more likely to be false.

Based on **H** and **P**, we calculate the high confidence positive sample labels pair $\mathcal{M}$ as follows:

$$\mathcal{M} = \mathbf{P} \odot \mathbf{H}, \tag{6}$$

Leveraging the labels from the high-confidence positive sample pairs $\mathcal{M}_{ij}$, we can then identify challenging positive sample pairs between the ith and jth samples.

3.4. Controllable weighting strategy

Currently, the predominant approach for handling challenging samples primarily relies on using the infoNCE [29] function to adjust the weights of sample pairs. Current methodologies for determining hard sample weights primarily concentrate on hard positive and negative samples. However, based on empirical investigations, we have discovered that adjusting the weights of both positive and negative samples may mitigate the clustering effect. Further, we observed variations in the density distribution of sample pairs across different datasets in Fig. 1. The application of a uniform weighting strategy, whether it concentrates on both hard positive and hard negative samples or exclusively on positive or negative samples, had detrimental effects.

To address this issue, we propose a controllable weight modulation function $\mathcal{W}$ that dynamically fine-tunes the weights of sample pairs throughout the training process. The confidence τ here can be different from the debiased portion of the sample. We use controllable factor x to choose a way to weigh our decisions. We simply select a hard sample to weight in the code. In particular, for datasets where the initial sample pairs exhibit low similarity, we compute their weights using the power β of the similarity among hard sample pairs. In the case of hard positive samples, we determine the weight as the difference between the pseudo-label and the similarity of the sample pair (diagonal element).

Scheme 1:Hard Positive weighting

$$\mathcal{W}_p(v_i^a, v_j^b) = \begin{cases} 1, \ \neg(i, j \in \mathcal{M}), \\ f|\mathcal{M} - Norm(S(v_i^a, v_j^b))|^{x\beta}, \ others. \end{cases} \tag{7}$$

Here, β represents the zoom factor, which dictates the weighting rate for sample pairs. When the samples exhibit high confidence, their weights are adjusted based on the pseudo-information and the similarity of the sample pair. Conversely, for samples without high confidence, i.e., $\neg(i, j \in \mathcal{M})$, we retain the original settings in the loss function. For example, one hard negative sample pair and simple negative sample pair have a similarity of 0.9, β=2, then the weights are 0.81 and 1, respectively. The weight of the hard sample is smaller than that of the simple sample, forcing the network to focus on the simple sample.

Scheme 2:Hard negative weighting

Given that we have eliminated positive sample pairs in the loss function, the remaining samples can be regarded as genuine negative sample pairs. In a similar vein, when conducting negative sample weight adjustment independently, it can be configured as the modulation function of its own similarity. For example, since the clustering cannot accurately identify sample pairs, there will be cases where the similarity of hard positive sample pairs and easy positive sample pairs is 0.1, and β=1, the weights are 0.9 and 1, respectively. So we keep the positive sample pair weights as originally set.

$$\mathcal{W}_n(v_i^a, v_j^b) = \begin{cases} 1, \ (i, j \in \mathcal{M}), \\ |Norm(S(v_i^a, v_j^b))|^{(1-x)\beta}, \ others. \end{cases} \tag{8}$$

Algorithm 1 The DCHD training algorithm

Input: Attribute matrix **X**; Adjacency matrix **A**; epoch number **N**; filtering times t; confidence τ ; zoom factor β; controllable factor x.
Output: The clustering results **R**.

```
Obtain the Laplacian filtered attribute matrix X̃ in Eq. (1) .
Utilize the encoder to obtain Z_1 and Z_2 by Eq. (2) .
Calculate similarity matrix S by Eq. (3) .
for n = 1 to N do
    Clustering on node embeddings Z to obtain sample pair pseudo-labels P and high confidence sample pair H in Eq. (4) and Eq. (5) , respectively.
    Conduct the hard positive sample de-biased matrix based on P and H in Eq. (6) .
    Conduct the controllable weighting matrix W based on S and M in Eq. (7) and Eq. (8) .
    Update the parameters of network by minimizing L in Eq. (10) ;
end for
Obtain result R by performing K-means over Z.
return R
```

The weight modulation function $\mathcal{W}$ serves to amplify the importance of hard samples while diminishing the significance of easy ones. To elaborate, when the ith and jth samples are recognized as positive sample pairs, the level of difficulty in pairing them diminishes as their similarity increases. Consequently, $\mathcal{W}$ assigns higher weights to positive pairs with low similarity (representing hard samples) and lower weights to pairs with high similarity (representing straightforward samples).

Based on $\mathcal{S}$ and $\mathcal{W}$, we represent the hard sample contrastive loss of the ith sample in the ath view as follows:

$$\mathcal{L}(v_i^a) = -log\Big(\frac{\sum_{b\neq a} e^{\mathcal{W}_p(v_i^a,v_i^b)\cdot\mathcal{S}(v_i^a,v_i^b)}}{\sum_{b\neq a} e^{\mathcal{W}_n(v_i^a,v_i^b)\cdot\mathcal{S}(v_i^a,v_i^b)} + \sum_{j\neq i,b\in\{1,2\}} \mathcal{M}_{ij} e^{\mathcal{W}_n(v_i^a,v_j^b)\cdot\mathcal{S}(v_i^a,v_j^b)}}\Big). \tag{9}$$

In contrast to the traditional infoNCE loss, our approach initially employs confidence measures and clustering pseudo-labels to rectify the bias stemming from an abundance of positive sample pairs within the negatives. Building upon HSAN [12], we introduce a controllable sample pair weighting strategy $\mathcal{W}$ to accommodate variations in the similarity density distribution among diverse datasets. In summary, the comprehensive loss function for our approach can be expressed as follows:

$$\mathcal{L} = \frac{1}{2N}\sum_{a=1}^{2}\sum_{i=1}^{N}\mathcal{L}(v_i^a). \tag{10}$$

This contrastive loss with differentiated hard sample weighting helps alleviate the influence of hard samples on the network, thus further enhancing the discriminative capability of both positive and negative samples. We can attribute these two key reasons to our approach: (1) The introduced hard positive sample debiasing strategy effectively mitigates the presence of misleading hard samples in the loss function, resulting in a more precise representation of negative samples. (2) The proposed controllable weighting strategy is adept at accommodating variations in the similarity distribution among distinct datasets. In datasets with a high proportion of low similarity sample pairs, it can reduce the weight of negative sample pairs; In datasets with a high proportion of similarity, it can reduce the weight of positive sample pairs.

4. Experiments

In this section, we evaluate the efficacy of our approach using real-world datasets for node clustering. We adhere to consistent experimental procedures across all experiments, including Random seeds in [0-9] and evaluation protocols.

4.1. Experimental setups

Dataset: We conduct experiments on six widely used benchmark datasets, including two citation networks CORA, Citeseer (CITE), three Air-Traffic networks Europe Air-Traffic (EAT), Brazil Air-Traffic (BAT), and USA Air-Traffic (UAT) and one co-purchase networks Amazon-Photo (AMAP).

Baselines: To verify the superiority of our approach. Concretely, we primarily compare our approach to thirteen state of-the-art deep graph clustering methods including based on feature extraction DAEGC [31], ARGVA [32], SDCN [33] and DFCN [34], based on graph contrastive learning AutoSSL [21], SUBLIME [35], AFGRL [20], NCLA [22] and MA-GCL [23], based on hard sample mining GDCL [24], ProGCL [10], Homogcl [11] and HSAN [12].

Implement details and Evaluation protocol: We performed part of the baseline experiment and the rest of the results are from [12]. The experimental results were acquired from a server equipped with an Intel Xeon Platinum 8172M CPU, NVIDIA GeForce RTX 3090 GPU, 128 GB RAM, and the PyTorch deep learning platform. The training is configured for 400 epochs, and we conduct ten runs for all methods. Following the procedures outlined in prior research, each model initially undergoes unsupervised training on the entire graph. Subsequently, results are updated every 10 rounds, and we employ the Adam optimizer to train the parameters. In our model, attribute encoders consist of two parameters unshared one-layer MLPs, with 500 hidden units for UAT/AMAP and 1500 hidden units for other datasets. The BAT/CITE are adapt positive weighting and negative weighting for others. For baseline methods, we utilize their original source code with default settings and replicate the results. The clustering performance is evaluated using four widely employed metrics: ACC (Accuracy), NMI (Normalized Mutual Information), ARI (Adjusted Rand Index), and F1. These metrics are commonly used in deep clustering assessments.

4.2. Complexity analysis

It is worth noting that the estimation of hard samples introduces lighter computational overhead and space overhead on the underlying model. Based on the hard pseudo-label obtained by k-means of t degree, $\mathcal{O}(tkNd)$ is required to obtain the k-clustered centroid. For Eq. (5), we need to calculate the distance between each node and each cluster centroid, which is another $\mathcal{O}(kNd)$ overhead to get the hard-positive sample matrix. Overall, the additional computational overhead for the base model is $\mathcal{O}(kNd)$, which is lightweight compared to the base model because k is usually set to a smaller number. Algorithm 1 summarizes the training algorithm used by the algorithm for the graph clustering task.

We have analyzed the memory usage of six methods during training on the AMAP dataset in Table 3. To ensure fairness, ProGCL accounts for training usage by calculating the probabilities of negative sample true and false occurrences. Regarding time efficiency, we measure the duration from the beginning of an epoch to the completion of the gradient loopback when the training epoch stabilizes. In this context, DCHD and HSAN are categorized into update round time $(\cdot)$ and normal training time. As indicated in the table, our approach exhibits a lower memory footprint compared to HSAN but is higher than other contrastive learning methods. This is attributed to the introduction of a weight matrix, incurring additional overhead. Additionally, the K-means process in DCHD is performed on the GPU, while other methods execute it on the CPU. In terms of time overhead, our method demonstrates a low impact during normal training rounds, with the weight update round consuming a longer duration. However, the update every 10 rounds has a minimal effect on the overall training time.

Table 2
The average clustering performance across ten runs on six benchmark datasets. The performance assessment is carried out using four metrics, with reported mean values and standard deviations. The values highlighted in red and blue indicate the best and second-best results, respectively.

Dataset	Metric	Based on feature extraction				Based on contrastive learning					Based on hard sample mining				
		DAEGC IJCAI 19	ARGVA IJCAI 19	SDCN WWW 20	DFCN AAAI 21	AutoSSL ICLR 22	SUBLIME WWW 22	NCLA AAAI 23	MA-GCL AAAI 23	AFGRL AAAI 22	HomoGCL KDD 23	GDCL IJCAI 21	ProGCL ICML 22	HSAN AAAI 23	DCHD Ours
CORA	ACC	70.43±0.36	65.97±1.15	35.60±2.83	36.33±0.49	63.81±0.57	71.14±0.74	69.68±4.92	51.91±12.1	26.25±1.24	50.28±9.73	70.83±0.47	57.13±1.23	77.74±1.15	78.67±0.87
	NMI	52.89±0.69	49.30±0.54	14.28±1.91	19.36±0.87	47.62±0.45	53.88±1.02	57.56±1.71	41.05±9.59	12.36±1.54	40.64±8.73	56.60±0.36	41.02±1.34	60.02±1.02	60.13±0.99
	ARI	49.63±0.43	41.28±1.93	07.78±3.24	04.67±2.10	38.92±0.77	50.15±0.14	51.19±4.02	28.60±14.8	14.32±1.87	25.72±11.5	48.05±0.72	30.71±2.70	58.61±1.50	59.60±1.62
	F1	68.27±0.57	63.71±2.09	24.37±1.04	26.16±0.50	56.42±0.21	63.11±0.58	66.45±6.02	47.37±13.2	30.20±1.15	45.11±10.9	52.88±0.97	45.68±1.29	75.76±1.37	76.84±0.79
CITE	ACC	64.54±1.39	59.30±1.38	65.96±0.31	69.50±0.20	66.76±0.67	68.25±1.21	68.18±0.81	63.88±5.62	31.45±0.54	45.61±8.04	66.39±0.65	65.92±0.80	71.30±0.87	71.13±1.04
	NMI	36.41±0.86	31.80±0.81	38.71±0.32	43.90±0.20	40.67±0.84	43.15±0.14	44.23±0.32	39.33±4.09	15.17±0.47	27.35±6.58	39.52±0.38	39.59±0.39	44.86±1.01	44.97±1.20
	ARI	37.78±1.24	31.28±1.22	40.17±0.43	45.50±0.30	38.73±0.55	44.21±0.54	45.28±0.34	39.12±5.81	14.32±0.78	19.51±9.24	41.07±0.96	36.16±1.11	46.67±1.12	46.17±1.90
	F1	62.20±1.32	56.05±1.13	63.62±0.24	64.30±0.20	58.22±0.68	63.12±0.42	64.18±0.70	59.45±6.88	30.20±0.71	41.87±8.52	61.12±0.70	57.89±1.98	63.13±1.73	62.71±2.08
AMAP	ACC	75.96±0.23	61.46±2.71	53.44±0.81	76.82±0.23	54.55±0.97	27.22±1.56	74.30±5.00	46.84±8.16	75.51±0.77	61.01±2.16	43.75±0.78	51.53±0.38	77.37±0.35	77.81±0.68
	NMI	65.25±0.45	53.25±1.91	44.85±0.83	66.23±1.21	48.56±0.71	06.37±1.89	68.57±3.30	41.54±10.0	64.05±0.15	55.84±1.17	37.32±0.28	39.56±0.39	67.35±0.26	67.08±0.88
	ARI	58.12±0.24	38.44±4.69	31.21±1.23	58.28±0.74	26.87±0.34	05.36±2.14	57.81±3.92	18.53±12.7	54.45±0.48	37.77±2.40	21.57±0.51	34.18±0.89	58.12±0.48	58.71±1.47
	F1	69.87±0.54	58.50±1.70	50.66±1.49	71.25±0.31	54.47±0.83	15.97±1.53	70.25±5.98	40.09±7.87	69.99±0.34	60.24±3.55	38.37±0.29	31.97±0.44	72.06±0.45	72.37±1.41
BAT	ACC	52.67±0.00	63.66±0.78	53.05±4.63	55.73±0.06	42.43±0.47	45.04±0.19	42.37±1.61	45.65±2.60	50.92±0.44	44.35±4.02	45.42±0.54	55.73±0.79	78.32±1.04	78.85±1.08
	NMI	21.43±0.35	40.79±0.95	25.74±5.71	48.77±0.51	17.84±0.98	22.03±0.48	15.80±1.55	18.36±3.58	27.55±0.62	16.39±4.81	31.70±0.42	28.69±0.92	54.18±1.13	54.30±1.54
	ARI	18.18±0.29	29.89±1.14	21.04±4.97	37.76±0.23	13.11±0.81	14.45±0.87	08.75±2.23	14.46±4.37	21.89±0.74	10.66±5.45	19.33±0.57	21.84±1.34	52.33±1.64	52.75±2.02
	F1	52.23±0.03	63.90±0.66	46.45±5.90	50.90±0.12	34.84±0.15	44.00±0.62	41.93±1.62	43.12±2.59	46.53±0.57	43.41±3.83	39.94±0.57	56.08±0.89	78.19±1.05	78.80±1.03
EAT	ACC	36.89±0.15	50.35±0.41	39.07±1.51	49.37±0.19	31.33±0.52	38.80±0.35	36.62±1.22	36.67±2.50	37.42±1.24	37.79±4.35	33.46±0.18	43.36±0.87	57.84±0.40	57.84±0.49
	NMI	05.57±0.06	33.68±1.71	08.83±2.54	32.90±0.41	07.63±0.85	14.96±0.75	07.49±1.39	06.68±1.75	11.44±1.41	10.26±5.55	13.22±0.33	23.93±0.45	34.55±0.41	34.87±0.54
	ARI	05.03±0.08	23.28±1.55	06.31±1.95	23.25±0.18	02.13±0.67	10.29±0.88	05.39±0.49	05.44±2.13	06.57±1.73	07.97±5.79	04.31±0.29	15.03±0.98	27.76±0.52	28.03±0.49
	F1	34.72±0.16	48.64±0.52	33.42±3.10	42.95±0.04	21.82±0.98	32.31±0.97	33.26±1.03	33.63±3.34	30.53±1.47	35.25±3.71	25.02±0.21	42.54±0.45	58.15±0.34	58.16±0.46
UAT	ACC	52.29±0.49	51.82±0.83	52.25±1.91	33.61±0.09	42.52±0.64	48.74±0.54	47.46±2.98	48.70±5.91	41.50±0.25	54.06±1.56	48.70±0.06	45.38±0.58	55.29±1.21	56.78±1.17
	NMI	21.33±0.44	26.30±1.18	21.61±1.26	26.49±0.41	17.86±0.22	21.85±0.62	16.55±1.16	22.63±5.63	17.33±0.54	27.20±0.72	25.10±0.01	22.04±2.23	26.15±1.08	27.33±1.66
	ARI	20.50±0.51	18.66±1.68	21.63±1.49	11.87±0.23	13.13±0.71	19.51±0.45	14.98±1.61	15.64±4.31	13.62±0.57	18.13±0.93	21.76±0.01	14.74±1.99	23.72±2.40	24.85±1.91
	F1	50.33±0.64	51.32±1.16	45.59±3.54	25.79±0.29	34.94±0.87	46.19±0.87	44.95±1.40	46.42±6.80	36.52±0.89	53.36±2.48	45.69±0.08	39.30±1.82	54.40±1.90	55.93±1.84

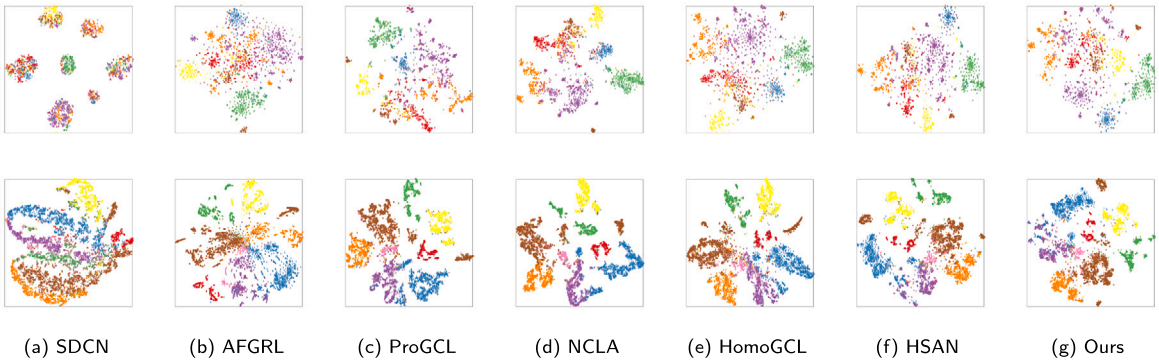

Fig. 3. We present 2D *t*-SNE visualizations of seven methods applied to two benchmark datasets. The first row corresponds to the CORA dataset, and the second row corresponds to the AMAP dataset.

Table 3
Time and memory (GPU) efficiency analysis.

Method	Time (S)	Memory (GB)
MA-GCL	0.026	2.96
NCLA	0.052	4.14
HomoGCL	0.213	5.37
ProGCL	0.379	8.77
HSAN	0.121 (1.149)	21.95
DCHD	0.054 (2.095)	15.51

4.3. Comparison analysis

We present the node clustering results for various methods across the six datasets in the table. Our implementation of DCHD is based on HSAN. From the outcomes outlined in Table 2, we can draw the following three conclusions: Compared with the traditional deep clustering methods, DCHD demonstrates a significant improvement in accuracy (ACC) of 8.24% and 15.19% on the CORA and BAT datasets, respectively. This suggests that the inclusion of the comparison mechanism enhances the network's ability to capture more valuable information. When compared to contrastive learning methods, DCHD exhibits ACC improvements of 7.53%, 2.3%, and 27.93% on the CORA, AMAP, and UAT datasets, respectively. This indicates that the hard positive sample debiasing module effectively addresses bias within the contrastive learning paradigm, correcting it and leading to improved performance. In contrast to existing hard-sample mining methods, our controllable weighting strategy module contributes to performance enhancement. This underscores the impact of the similarity differences among samples in different datasets on the weighting strategy. In contrast with HSAN, our emphasis on hard positive sample debiasing contributes to refining the contrastive learning paradigm and results in superior performance. To summarize, the experimental results validate the effectiveness of the proposed two modules in improving the performance of the DCHD method.

4.4. Ablation studies

In this section, we conduct an ablation study to confirm the efficacy of the proposed hard positive sample debiasing method denoted as "D" and the Controllable weighting strategy represented by "W". Specifically, in Fig. 4, we use "B" to denote the baseline. Furthermore, "B+D", "B+W", and "Ours" signify the baselines incorporating D, W, and both, respectively. The "B+D" and "B+W" configurations for CORA, AMAP, and BAT datasets have shown improvements in all four metrics. The "Ours" configuration has reached the optimal state. When CITE is used individually, there is a slight performance decrease, but when combined, it is at its best. EAT and UAT perform best when using "D", but performance degrades when using "W", suggesting possible issues with hyperparameter selection. In CORA, the confidence of "D" is set to 0, and the zoom β of UAT is set to 0. The results from Fig. 4 lead to the following three observations. (1) Our proposed method shows significant performance improvement compared to the baseline. This improvement stems from the hard positive sample debiasing technique

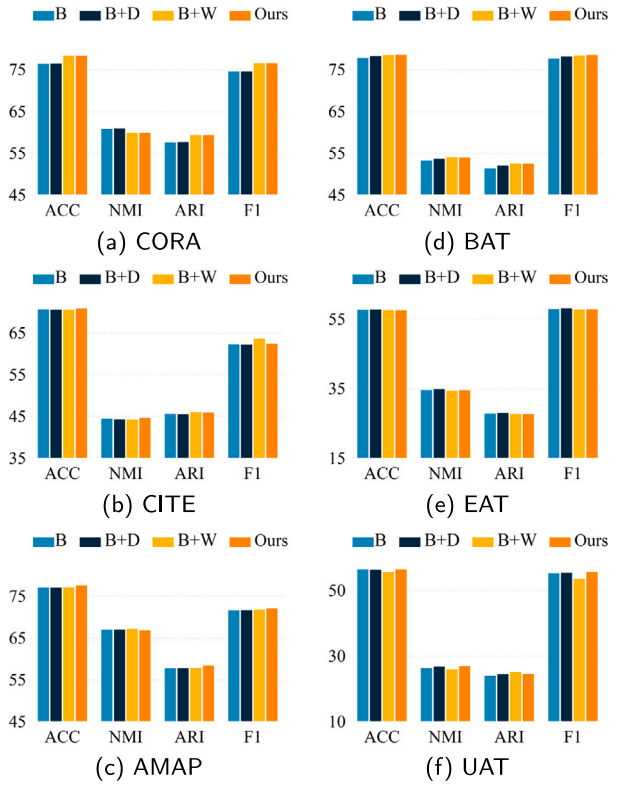

Fig. 4. Ablation studies of the proposed hard positive sample debiased D and weighting strategy function $\mathcal{W}$ on six datasets.

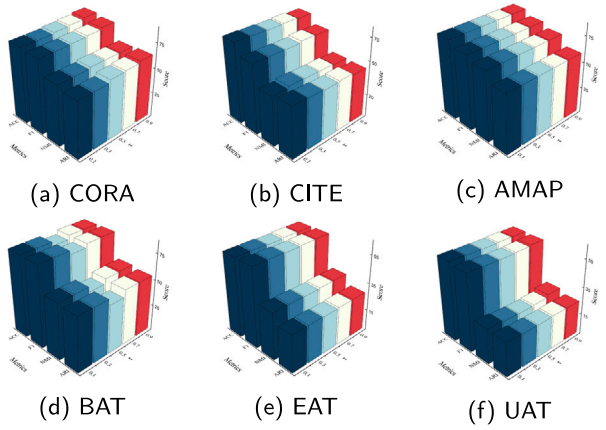

Fig. 5. Analysis of the confidence hyper-parameter τ on six datasets.

D, which effectively reduces the error message in the loss function by eliminating false negative samples (i.e., true positive samples) from the loss function, thereby guiding the network to build the correct model. (2) "B+W" improves the performance of "B". The results show that our proposed controllable weighting strategy, W improves the adaptability of the model to different datasets by guiding our network to weight different hard samples. (3) The combination of S and M yielded favorable clustering performance in all test configurations.

4.5. Sensitivity analysis

We examined the sensitivity of the confidence parameter τ and the number of Laplacian transformations on six datasets.

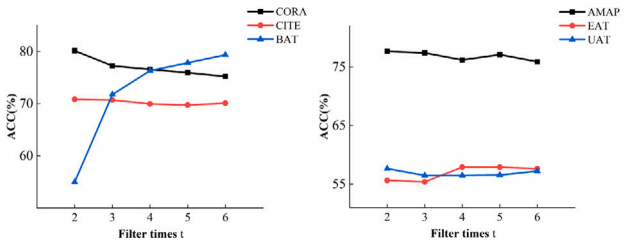

Fig. 6. Analysis of the Laplacian filter times hyper-parameter t.

Confidence parameter τ: Fig. 5 shows the node clustering ACC, NMI, ARI, and F1 of DCHD on different datasets with τ values of [0.1, 0.3, 0.5, 0.7, 0.9]. We observed that DCHD did not have significant fluctuations in probability on the AMAP and BAT|EAT|UAT datasets, with BAT achieving the worst performance at 0.5 and generally showing a trend of first decreasing and then increasing. CORA generally showed an increasing trend with increasing values, while CITE decreased with increasing values of the parameter τ. The results indicate that we suspect that when the confidence parameter τ is low, the number of hard samples identified is relatively small, leads to incorrect weighting of false negative samples during the dynamic weighting process, affecting the model's recognition ability. Fig. 1 shows that the CITE dataset contains a substantial number of challenging negative samples. Due to the ongoing bias in distinguishing between positive and negative samples, the debiased of hard positive samples ends up removing valuable information from true negative samples, making it impossible to learn this useful information. DCHD reduces the bias in the loss function by clustering pseudo labels and confidence screening hard positive samples, and adapts to the differences in the initial state similarity distribution of different datasets through controlled weighting strategies.

Laplacian filter times t: We compared the accuracy (ACC) of our method on six datasets, observing the changes in the Laplacian filter t parameter in [2,6]. As shown in Fig. 6, DCHD is less affected by changes in t except for the BAT dataset. CORA and CITE decrease as t increases, while BAT is the opposite, and the other three datasets have relatively flat changes. Overall, maintaining a low filter times yields better results. In consequence, our model exhibits strong robustness, avoids excessive hyperparameter tuning with augmentation and does not necessitate parameter size adjustments based on the dataset.

4.6. t-SNE visualization of embeddings

To understand the node embeddings learned by DCHD more intuitively, we visualize the embedding distribution of the CORA and AMAP datasets generated by DCHD in Fig. 3. It is generated by t-SNE [36] and uses different colors to represent different categories. As a comparison, we mapped the node embedding data generated by these seven methods into a low-dimensional space and then grouped them according to their labels. The main difference between these four methods of hard sample mining is that DCHD maps to a lower-dimensional space and the distribution of the same group is more compact. Compared with HSAN [12], DCHD captures more granular class information. In the CORA dataset, other methods usually have a small amount of noise data in each label group. We compared the set of nodes included in each tag and their embeddings. We can see that the nodes in DCHD are more tightly grouped than those in HSAN. This means that our method can capture more detailed class information.

Fig. 7 shows the distribution of high-confidence samples. The t-SNE visualization of high-confidence samples on the right and the visualization of all samples on the left. It is not difficult to find that almost all high-confidence samples are included in the class cluster to which they belong, which provides strong support for positive sample debiased.

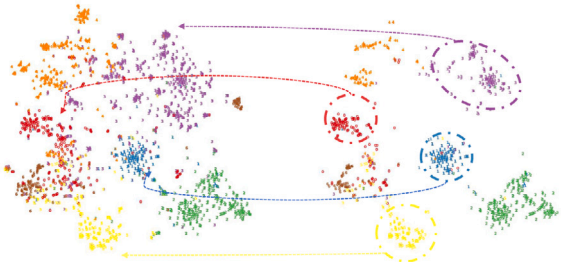

Fig. 7. Visualizations of the high confidence sample.

5. Conclusion

In this paper, we investigate why dynamic weighting of both hard positive and hard negative samples is not effective, and consider that the similarity of initialized samples of different data sets is extreme. In addition, we find that there are a large number of positive pairs in negative sample pairs. Therefore, we designed Deep Contrastive Clustering via Hard positive sample Debiased(DCHD) to screen out potential positive sample pairs by clustering pseudo-labels and confidence criteria. Then, positive sample pairs in the denominator in infoNCE are removed, thus ensuring the purity of negative samples of contrastive learning. In addition, we propose a controllable weighting strategy to accommodate the difference in density distribution of sample pairs of different datasets. In this way, the network pays more accurate attention to positive and negative sample pairs, which further improves the sample identification ability. A large number of experiments show that DCHD has good performance in node clustering tasks on six benchmark datasets. In this work, the accuracy of high-confidence positive sample pairs is not high, and the need to manually adjust the weighting method will be a focal point for future investigations. Moreover, the consideration of constraining the key dimensions of the eigenvector to enhance expressive capability will be a direction for future exploration.

CRediT authorship contribution statement

Xinyu Zhang: Writing – review & editing, Writing – original draft, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Huiying Xu:** Supervision, Resources, Funding acquisition. **Xinzhong Zhu:** Supervision, Funding acquisition. **Yuhang Chen:** Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

We would like to submit the enclosed manuscript entitled "Deep Contrastive Clustering via Hard positive sample Debiased" to Neurocomputing. The work has not been considered to publish elsewhere, in whole or in part. All authors have approved to this submission to your esteemed journal. Its publication is also approved tacitly by the responsible authorities where the work was carried out. We agree that if accepted, the paper will not be published elsewhere in the same form, in English or in any other language, without the written consent of the Publisher.

We believe that this manuscript will make it interesting to general readers of Neurocomputing. This is a part research accomplishment of the National Natural Science Foundation of China (62376252, 61976196); Key Project of Natural Science Foundation of Zhejiang Province (LZ22F030003); Key Projects of National College Student Innovation Training Program (202310345042). The fundamental contributions of our DHCD are primarily highlighted as follows: 1. Our DCHD introduces a novel deep graph contrastive clustering approach designed to eliminate the bias exerted by hard positive samples on the model. 2. To accommodate variations in the similarity density distribution among sample pairs in different datasets, our DCHD presents a weighting strategy model that exclusively addresses positive or negative samples. This approach helps mitigate adverse effects within the weighting process. 3. Furthermore, we conducted thorough experiments on the node clustering task using six benchmark datasets. The results consistently demonstrated that our method outperformed other state-of-the-art deep graph clustering approaches.

Data availability

Data will be made available on request.

Appendix. Proof of method DCHD

In this section, we introduce the proof of our method with Triplet Loss [37].

Proof. Based on the assumptions, we can rearrange the pairwise objective Eq. (9) as:

$$\begin{aligned}\mathcal{L}(v_i^a) &= -log(\frac{\sum_{b\neq a} e^{\mathcal{W}_p(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b)}}{\sum_{b\neq a} e^{\mathcal{W}_n(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b)} + \sum_{j\neq i,b\in\{1,2\}} \mathcal{M}_{ij} e^{\mathcal{W}_n(v_i^a,v_j^b)\cdot S(v_i^a,v_j^b)}})\\ &= log(e^{(\mathcal{W}_n(v_i^a,v_i^b)-\mathcal{W}_p(v_i^a,v_i^b))S(v_i^a,v_i^b)}\\ &+ \sum_{j\neq i,b\in\{1,2\}} \underbrace{\mathbb{0}_{[\mathcal{M}_{ij}=1]}\cdot e^{(\mathcal{W}_n(v_i^a,v_j^b)\cdot S(v_i^a,v_j^b)-\mathcal{W}_p(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b))}}_{Hard\ positive\ Debaised}\\ &+ \underbrace{\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot e^{(\mathcal{W}_n(v_i^a,v_j^b)\cdot S(v_i^a,v_j^b)-\mathcal{W}_p(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b))}}_{True\ negative\ sample\ pair})\end{aligned}$$

The bias caused by the positive sample pairs with similarity tending to 0 is removed by the hard positive sample debiasing module. By Taylor expansion of first order.

$$\begin{aligned}&= log(1+e^{(\mathcal{W}_n(v_i^a,v_i^b)-\mathcal{W}_p(v_i^a,v_i^b))S(v_i^a,v_i^b)} - 1\\ &+ \sum_{j\neq i,b\in\{1,2\}} \underbrace{\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot e^{(\mathcal{W}_n(v_i^a,v_j^b)\cdot S(v_i^a,v_j^b)-\mathcal{W}_p(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b))}}_{True\ negative\ sample\ pair})\\ &\approx e^{(\mathcal{W}_n(v_i^a,v_i^b)-\mathcal{W}_p(v_i^a,v_i^b))S(v_i^a,v_i^b)} - 1\\ &+ \sum_{j\neq i,b\in\{1,2\}} \mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot e^{(\mathcal{W}_n(v_i^a,v_j^b)\cdot S(v_i^a,v_j^b)-\mathcal{W}_p(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b))}\end{aligned}$$

The above equation can be divided into hard positive sample weighting and hard negative sample weighting.

Scheme 1:Hard positive weighting. According to the Eq. (7), it can be divided into two cases.$\mathcal{W}_n(v_i^a,v_i^b)=1$.

$$\begin{aligned}(1)v_i^a\in\mathbf{H}:\quad \mathcal{L}(v_i^a) &\approx 1+(1-\mathcal{W}_p(v_i^a,v_i^b))S(v_i^a,v_i^b)\\ &+\sum_{j\neq i,b\in\{1,2\}}\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot(S(v_i^a,v_j^b)-\mathcal{W}_p(v_i^a,v_i^b)\cdot S(v_i^a,v_i^b))\\ (2)else:\quad \mathcal{L}(v_i^a) &\approx 1+(1-1)S(v_i^a,v_i^b)\\ &+\sum_{j\neq i,b\in\{1,2\}}\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot(S(v_i^a,v_j^b)-S(v_i^a,v_i^b))\\ &=1+\sum_{j\neq i,b\in\{1,2\}}\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot(S(v_i^a,v_j^b)-S(v_i^a,v_i^b))\end{aligned}$$

Scheme 2:Hard negative weighting. According to the Eq. (8), it can be rearranged as:

$$\begin{aligned}&\approx 1+(1-1)S(v_i^a,v_i^b)+\sum_{j\neq i,b\in\{1,2\}}\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot(S(v_i^a,v_j^b)-S(v_i^a,v_i^b))\\ &=1+\sum_{j\neq i,b\in\{1,2\}}\mathbb{1}_{[\mathcal{M}_{ij}=0]}\cdot(\mathcal{W}_n(v_i^a,v_i^b)\cdot S(v_i^a,v_j^b)-S(v_i^a,v_i^b))\end{aligned}$$

which concludes the proof.

References

[1] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, P. Sun, Z. Li, P. Luo, Detco: Unsupervised contrastive learning for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8392–8401.

[2] C. Yang, Z. Wu, B. Zhou, S. Lin, Instance localization for self-supervised detection pretraining, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3987–3996.

[3] X. Li, A. Sun, M. Zhao, J. Yu, K. Zhu, D. Jin, M. Yu, R. Yu, Multi-intention oriented contrastive learning for sequential recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 411–419.

[4] X. Cai, C. Huang, L. Xia, X. Ren, Lightgcl: Simple yet effective graph contrastive learning for recommendation, 2023, arXiv preprint arXiv:2302.08191.

[5] B. Li, T. Guo, X. Zhu, Q. Li, Y. Wang, F. Chen, Sgccl: Siamese graph contrastive consensus learning for personalized recommendation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 589–597.

[6] A. Saeed, D. Grangier, N. Zeghidour, Contrastive learning of general-purpose audio representations, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 3875–3879.

[7] H. Wu, F. Xiao, C. Liang, Dual contrastive learning with anatomical auxiliary supervision for few-shot medical image segmentation, in: European Conference on Computer Vision, Springer, 2022, pp. 417–434.

[8] X. Chen, S. Xie, K. He, An empirical study of training self-supervised visual transformers. arxiv e-prints, 2021, arXiv preprint arXiv:2104.02057.

[9] Y. Kalantidis, M.B. Sariyildiz, N. Pion, P. Weinzaepfel, D. Larlus, Hard negative mixing for contrastive learning, Adv. Neural Inf. Process. Syst. 33 (21) (2020) 721–798, 809.

[10] J. Xia, L. Wu, G. Wang, J. Chen, S.Z. Li, Progcl: Rethinking hard negative mining in graph contrastive learning, 2021, arXiv preprint arXiv:2110.02027.

[11] W.-Z. Li, C.-D. Wang, H. Xiong, J.-H. Lai, Homogcl: Rethinking homophily in graph contrastive learning, 2023, arXiv preprint arXiv:2306.09614.

[12] Y. Liu, X. Yang, S. Zhou, X. Liu, Z. Wang, K. Liang, W. Tu, L. Li, J. Duan, C. Chen, Hard sample aware network for contrastive deep graph clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, no. 7, 2023, pp. 8914–8922.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.

[14] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley others, Unsupervised and transfer learning challenge: A deep learning approach, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning. JMLR Workshop and Conference Proceedings, 2012, pp. 97–110.

[15] Y. Bengio, I. Goodfellow, A. Courville, Deep learning, MIT press Cambridge, MA, USA, 2017, p. 1.

[16] A. Goyal, Y. Bengio, Inductive biases for deep learning of higher-level cognition, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 478 (2266) (2022) 20210068.

[17] I. Goodfellow, Deep learning of representations and its application to computer vision, 2015.

[18] I.J. Goodfellow, M. Mirza, D. Xiao, A. Courville, Y. Bengio, An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2013, arXiv preprint arXiv:1312.6211.

[19] H. Zhang, Q. Wu, J. Yan, D. Wipf, P.S. Yu, From canonical correlation analysis to self-supervised graph neural networks, Adv. Neural Inf. Process. Syst. 34 (2021) 76–89.

[20] N. Lee, J. Lee, C. Park, Augmentation-free self-supervised learning on graphs, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, no. 7, 2022, pp. 7372–7380.

[21] W. Jin, X. Liu, X. Zhao, Y. Ma, N. Shah, J. Tang, Automated self-supervised learning for graphs, 2021, arXiv preprint arXiv:2106.05470.

[22] X. Shen, D. Sun, S. Pan, X. Zhou, L.T. Yang, Neighbor contrastive learning on learnable graph augmentation, 2023, arXiv preprint arXiv:2301.01404.

[23] X. Gong, C. Yang, C. Shi, Ma-gcl: Model augmentation tricks for graph contrastive learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, no. 4, 2023, pp. 4284–4292.

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

[25] K. Song, J. Han, G. Cheng, J. Lu, F. Nie, Adaptive neighborhood metric learning, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2021) 4591–4604.

[26] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, Y. Wei, Circle loss: A unified perspective of pair similarity optimization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6398–6407.

[27] G. Chu, X. Wang, C. Shi, X. Jiang, Cuco: Graph representation with curriculum contrastive learning, in: Proc. of IJCAI, 2021.

[28] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[29] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv preprint arXiv:1807.03748.

[30] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in: Proceedings of the Web Conference, Vol. 2021, 2021, pp. 2069–2080.

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

[32] S. Pan, R. Hu, S. f. Fung, G. Long, J. Jiang, C. Zhang, Learning graph embedding with adversarial training methods, IEEE Trans. Cybern. 50 (6) (2019) 2475–2487.

[33] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, P. Cui, Structural deep clustering network, in: Proceedings of the Web Conference, Vol. 2020, 2020, pp. 1400–1410.

[34] W. Tu, S. Zhou, X. Liu, X. Guo, Z. Cai, E. Zhu, J. Cheng, Deep fusion clustering network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, no. 11, 2021, pp. 9978–9987.

[35] Y. Liu, Y. Zheng, D. Zhang, H. Chen, H. Peng, S. Pan, Towards unsupervised deep graph structure learning, in: Proceedings of the ACM Web Conference, Vol. 2022, 2022, pp. 1392–1403.

[36] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (11) (2008).

[37] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.

Xinyu Zhang received B.E. degrees from Zhejiang Normal University, China. He is currently pursuing M.E. degree in computer science and technology from Zhejiang Normal University, China. His current research interests include graph representation learning, deep clustering, and neural networks.

Huiying Xu received the M.S. degree from National University of Defense Technology(NUDT), China. She is an associate professor with the School of Computer Science and Technology, Zhejiang Normal University, and also the researcher of Research Institute of Ningbo Cixing Co. Ltd, PR, China. Her research interests include Kernel learning and feature selection, Object Detection, Vision SLAM, Computer vision, Image processing, Pattern recognition, Computer simulation, Deep clustering, Generative Adversarial Network, Diffusion Model, Clustering Ensemble, Multiple Kernel Learning, Learning with incomplete data and their applications. She is a member of the China Computer Federation. She has published papers, including those in highly regarded journals such International Journal of Intelligent Systems, IEEE Transactions on Cybernetics, IEEE Transactions on Multimedia, etc.

Xinzhong Zhu received the Ph.D. degree from Xidian University and M.S. degree from National University of Defense Technology (NUDT), China. He is a professor with the School of Computer Science and Technology, Zhejiang Normal University, and also the chief scientist of Beijing Geekplus Technology Co., Ltd. and president of Research Institute of Ningbo Cixing Co., Ltd., China. His research interests include Machine learning, Deep clustering, Computer vision, Manufacturing informatization, Robotics and System integration, Laser SLAM, Vision SLAM, Diffusion Model, Low Quality Data Learning, Multiple Kernel Learning, and Intelligent manufacturing. He is a member of the ACM and certified as CCF distinguished member. Dr. Zhu has published more than 30 peer-reviewed papers, including those in highly regarded journals and conferences such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Multimedia, the IEEE Transactions on Knowledge and Data Engineering, CVPR, NeurIPS, AAAI, IJCAI, etc. He served on the Technical Program Committees of IJCAI 2020 and AAAI 2020.

Yuhang Chen received his B.E. degrees in chemistry from Huzhou University, Huzhou, China, in 2020. He is currently pursuing the M.E. degree in computer science and technology from Zhejiang Normal University, Jinhua, China. His current research interests include deep clustering, graph neural networks and self-supervised learning.