Hard Sample Aware Network for Contrastive Deep Graph Clustering

Yue Liu¹, Xihong Yang¹, Sihang Zhou², Xinwang Liu¹ ‡, Zhen Wang³, Ke Liang¹, Wenxuan Tu¹, Liang Li¹, Jingcan Duan¹, Cancan Chen⁴

¹College of Computer, National University of Defense Technology
²College of Intelligence Science and Technology, National University of Defense Technology
³Northwestern Polytechnical University
⁴Beijing Information Science and Technology University

yueliu@nudt.edu.cn

Abstract

Contrastive deep graph clustering, which aims to divide nodes into disjoint groups via contrastive mechanisms, is a challenging research spot. Among the recent works, hard sample mining-based algorithms have achieved great attention for their promising performance. However, we find that the existing hard sample mining methods have two problems as follows. 1) In the hardness measurement, the important structural information is overlooked for similarity calculation, degrading the representativeness of the selected hard negative samples. 2) Previous works merely focus on the hard negative sample pairs while neglecting the hard positive sample pairs. Nevertheless, samples within the same cluster but with low similarity should also be carefully learned. To solve the problems, we propose a novel contrastive deep graph clustering method dubbed Hard Sample Aware Network (HSAN) by introducing a comprehensive similarity measure criterion and a general dynamic sample weighing strategy. Concretely, in our algorithm, the similarities between samples are calculated by considering both the attribute embeddings and the structure embeddings, better revealing sample relationships and assisting hardness measurement. Moreover, under the guidance of the carefully collected high-confidence clustering information, our proposed weighted modulating function will first recognize the positive and negative samples and then dynamically up-weight the hard sample pairs while down-weighting the easy ones. In this way, our method can mine not only the hard negative samples but also the hard positive sample, thus improving the discriminative capability of the samples further. Extensive experiments and analyses demonstrate the superiority and effectiveness of our proposed method. The source code of HSAN is shared at https://github.com/yueliu1999/HSAN and a collection (papers, codes and, datasets) of deep graph clustering is shared at https://github.com/yueliu1999/Awesome-Deep-Graph-Clustering on Github.

Introduction

In recent years, contrastive learning has achieved promising performance in the field of deep graph clustering benefiting from the powerful potential supervision information extraction capability (Liu et al. 2022c; Gong et al. 2022). Among the recent works, researchers demonstrate the effectiveness of hard negative sample mining in contrastive learning. Concretely, GDCL (Zhao et al. 2021) is proposed to correct the bias of the negative sample selection via clustering pseudo labels. ProGCL (Xia et al. 2022a) first filters the false negative samples and then generates a more abundant negative sample set by the sample interpolation.

Although verified to be effective, we point out two drawbacks in the existing methods as follows. 1) While measuring the hardness of samples, the important structural information is neglected for the sample similarity calculation, degrading the representativeness of the selected hard negative samples. 2) Previous works only focus on the hard negative samples while overlooking the hard positive samples, limiting the discriminative capability of samples. We argue that the samples within the same cluster but with low similarity should also be carefully learned.

Figure 1: Sample weighing strategy illustration. In the sample weighting process, according to the high-confidence pseudo labels generated by the integrated clustering procedure, samples from the same clusters are recognized as potential positive sample pairs. Meanwhile, samples from different clusters are combined as potential negative sample pairs. To drive the network to focus more on the hard samples, we assign larger weights to the positive sample pairs with smaller similarities shown in sub-figure (a) and assign smaller weights to the negative sample pairs with small similarities shown in sub-figure (b). In this way, samples in the same cluster with low similarity and those in different clusters with large similarity are mined as hard samples.

To solve the mentioned problems, we propose a novel contrastive deep graph clustering method termed Hard Sample Aware Network (HSAN) by designing a comprehensive similarity measure criterion and a general dynamic sample

Copyright © 2023. Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

‡Corresponding author.
learning mechanisms, the existing methods can be roughly categorized into three classes: generative methods (Wang et al. 2017; Mrabah et al. 2022; Xia et al. 2022e), adversarial methods (Pan et al. 2019; Gong et al. 2022), and contrastive methods. More information about the fast-growing deep graph clustering can be checked in our survey paper (Liu et al. 2022d). In this work, we focus on the last category, i.e., the contrastive deep graph clustering. Recently, contrastive mechanisms have succeeded in many domains such as images (Xu and Lang 2020) and graphs (Zhu et al. 2020; Wu et al. 2021b), and knowledge graphs (Liang et al. 2022a). Inspired by their success, the contrastive deep graph clustering methods are increasingly proposed. A pioneer AGE (Cui et al. 2020) conducts contrastive learning by a designed adaptive encoder. MVGRL (Hassani and Khasahmadi 2020) adopts the InfoMax loss (Hjelm et al. 2018) to maximize the cross-view mutual information. Subsequently, SCAGC (Xia et al. 2022d) pulls together the positive samples while pushing away negative ones across augmented views. After that, DCRN and its variants (Liu et al. 2022c,f) alleviate the collapsed representation by reducing correlation in a dual manner. Then SCGC (Liu et al. 2022e) is proposed to reduce high time cost of the existing methods by simplifying the data augmentation and the graph convolutional operation. More recently, the selection of positive and negative samples has attracted great attention. Concretely, GDCL (Zhao et al. 2021) develops a debiased sampling strategy to correct the bias for negative samples. However, most of the existing methods treat the easy and hard samples equally, leading to indiscriminative capability. To solve this problem, we propose a novel contrastive deep graph clustering by mining the hard samples. In our proposed method, the weights of the hard positive and negative samples will dynamically increased while the weights of the easy ones will be decreased.

### Hard Sample Mining

In the contrastive learning methods, one key factor of promising performance is the positive and negative sample selection. Previous works (Chuang et al. 2020; Robinson et al. 2020; Kalantidis et al. 2020) on images have demonstrated that the hard negative samples are hard yet useful. Motivated by their success, more researchers take attention to the hard negative sample mining on graphs. Concretely, GDCL (Zhao et al. 2021) utilizes the clustering pseudo labels to correct the bias of the negative sample selection in the attribute graph clustering task. Besides, CuCo (Chu et al. 2021) selects and learns the samples from easy to hard in the graph classification task. In addition, to better mine true and hard negative samples, STENCIL (Zhu et al. 2022) advocates enriching the model with local structure patterns of heterogeneous graphs. More recently, ProGCL (Xia et al. 2022a) built a more suitable measure for the hardness of negative samples together with similarity by a designed probability estimator. Although verified the effectiveness, previous methods neglect the hard positive sample pair, thus leading to sub-optimal performance. In this work, we argue that the samples with the same category but low similarity should also be carefully learned. From this motivation, we propose a general sample weighting strategy to guide the network focus more on both the hard positive and negative samples.

### Method

In this section, we propose a novel Hard Sample Aware Network (HSAN) for contrastive deep graph clustering by guiding our network to focus on both the hard positive and negative sample pairs. The framework is shown in Figure 2.
Figure 2: Illustration of our proposed hard sample aware network. In attribute and structure encoding, we embed the attribute and structure into the latent space with the attribute encoders and structure encoders. Then the sample similarities are calculated by a learnable linear combination of attribute similarity and structure similarity, thus better revealing the sample relations. Moreover, guided by the high-confidence information, a general dynamic sample weighting strategy is proposed to up-weight hard sample pairs while down-weighting the easy ones. Overall, the hard sample aware contrastive loss guides the network to focus more on both hard positive and negative sample pairs, thus further improving the discriminative capability of samples.

Notation and Problem Definition

Let $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ be a set of $N$ nodes with $C$ classes and $\mathcal{E}$ be a set of edges. In the matrix form, $X \in \mathbb{R}^{N \times d}$ and $A \in \mathbb{R}^{N \times N}$ denotes the attribute matrix and the original adjacency matrix, respectively. Then $G = \{X, A\}$ denotes an undirected graph. The degree matrix is formulated as $D = diag(d_1, d_2, \ldots, d_N) \in \mathbb{R}^{N \times N}$ and $d_i = \sum_{(v_i, v_j) \in \mathcal{E}} a_{ij}$. The graph Laplacian matrix is defined as $L = D - A$. With the renormalization trick $\hat{A} = A + I$ in GCN (Kipf and Welling 2017), the symmetric normalized graph Laplacian matrix is denoted as $\hat{L} = I - D^{-\frac{1}{2}}A D^{-\frac{1}{2}}$. The notations are summarized in Table 1.

The target of deep graph clustering is to encode nodes with the neural network in an unsupervised manner and then divide them into several disjoint groups. In general, a neural network $\mathcal{F}$ is firstly trained without human annotations and embeds the nodes into the latent space by exploiting the node attributes and the graph structure as follows:

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

where $X$ and $A$ denotes the attribute matrix and the original adjacency matrix. Besides, $E \in \mathbb{R}^{N \times d}$ is the learned node embeddings, where $N$ is the number of samples and $d$ is the number of feature dimensions. After that, a clustering algorithm $C$ such as K-means, spectral clustering, or the clustering neural network layer (Bo et al. 2020) is adopted to divide nodes into $K$ disjoint groups as follows:

$$\Phi = C(E), \quad (2)$$

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X \in \mathbb{R}^{N \times d}$</td>
<td>Attribute matrix</td>
</tr>
<tr>
<td>$\hat{X} \in \mathbb{R}^{N \times d}$</td>
<td>Low-pass filtered attribute matrix</td>
</tr>
<tr>
<td>$A \in \mathbb{R}^{N \times N}$</td>
<td>Original adjacency matrix</td>
</tr>
<tr>
<td>$\hat{A} \in \mathbb{R}^{N \times N}$</td>
<td>Adjacency matrix with self-loop</td>
</tr>
<tr>
<td>$L \in \mathbb{R}^{N \times N}$</td>
<td>Graph Laplacian matrix</td>
</tr>
<tr>
<td>$\hat{L} \in \mathbb{R}^{N \times N}$</td>
<td>Symmetric normalized Laplacian matrix</td>
</tr>
<tr>
<td>$Z_{\Phi} \in \mathbb{R}^{N \times d}$</td>
<td>Attribute embeddings in $k$-th view</td>
</tr>
<tr>
<td>$E_{\Phi} \in \mathbb{R}^{N \times d}$</td>
<td>Structure embeddings in $k$-th view</td>
</tr>
<tr>
<td>$P \in \mathbb{R}^{N \times N}$</td>
<td>Sample clustering pseudo labels</td>
</tr>
<tr>
<td>$Q \in \mathbb{R}^{N \times N}$</td>
<td>Sample pair pseudo labels</td>
</tr>
<tr>
<td>$| \cdot |_2$</td>
<td>L-2 norm</td>
</tr>
</tbody>
</table>

Table 1: Notation summary.

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

Attribute and Structure Encoding

In this section, we design two types of encoders to embed the nodes into the latent space. Concretely, the attribute encoder (AE) and the structure encoder (SE) encode the attribute and structural information of samples, respectively.

In the process of attribute encoding, we follow previous work (Cui et al. 2020) and filter the high-frequency noises in the attribute matrix $X$ as formulated:
\[ \tilde{X} = (I - \tilde{L})X = (I - \tilde{L})tX, \]  
where \( I - \tilde{L} \) is the graph Laplacian filter and \( t \) is the filtering times. Then we encode \( \tilde{X} \) with AE_1 and AE_2 as follows:

\[
Z_i^{v_1} = AE_1(\tilde{X}); Z_i^{v_1} = \frac{Z_i^{v_1}}{||Z_i^{v_1}||_2}, i = 1, 2, ..., N; \]

\[
Z_i^{v_2} = AE_2(\tilde{X}); Z_i^{v_2} = \frac{Z_i^{v_2}}{||Z_i^{v_2}||_2}, j = 1, 2, ..., N, \]  
where \( Z_i^{v_1} \) and \( Z_i^{v_2} \) denote two-view attribute embeddings of the samples. Here, AE_1 and AE_2 are both simple multilayer perceptrons (MLPs), which have the same architecture but un-shared parameters, thus \( Z_i^{v_1} \) and \( Z_i^{v_2} \) contain different semantic information. Different the mixup-based methods (Yang et al. 2022a; Wu et al. 2021a), our proposed method is free from augmentation.

In addition, we further propose the structure encoder to encode the structural information of samples. Concretely, we design the structure encoder as follows:

\[
E_i^{v_1} = SE_1(A); E_i^{v_1} = \frac{E_i^{v_1}}{||E_i^{v_1}||_2}, i = 1, 2, ..., N; \]

\[
E_i^{v_2} = SE_2(A); E_i^{v_2} = \frac{E_i^{v_2}}{||E_i^{v_2}||_2}, j = 1, 2, ..., N, \]  
where \( E_i^{v_1} \) and \( E_i^{v_2} \) denote two-view structure embeddings. Similarly, SE_1 and SE_2 are simple MLPs, which have the same architecture but un-shared parameters, thus \( E_i^{v_1} \) and \( E_i^{v_2} \) contain different semantic information during training.

In this manner, we obtain the attribute embedding and structure embedding of each sample. Subsequently, the attribute-structure similarity function \( S \) is proposed to calculate the similarity between \( i \)-th sample in the \( j \)-th view and \( k \)-th sample in the \( l \)-th view as formulated:

\[
S(i^{v_j}, k^{v_l}) = \alpha \cdot (Z_i^{v_j})^T Z_k^{v_l} + (1 - \alpha) \cdot (E_i^{v_j})^T E_k^{v_l}, \]  
where \( i, k \in \{1, 2, ..., N\} \) and \( j, l \in \{1, 2\} \). Besides, \( \alpha \) denotes a learnable trade-off parameter. The first and second term in Eq. (6) both denotes the cosine similarity. \( S \) can better reveal sample relations by considering both attribute and structure information, thus assisting the hard sample mining.

**Clustering and Pseudo Label Generation**

After encoding, K-means is performed on the learned node embeddings to obtain the clustering results. Then we extract the more reliable clustering information as follows. To be specific, we first generate the clustering pseudo labels \( P \in \mathbb{R}^N \) and then select the top \( \tau \) samples as the high-confidence sample set \( H \in \mathbb{R}^M \). Here, \( \tau \) is the confidence hyper-parameter and \( M \) is the number of high-confidence samples. The confidence is measured by the distance to the cluster center (Liu et al. 2022f). Based on \( P \), we calculate the sample pair pseudo labels \( Q \in \mathbb{R}^N \times N \) as follows:

\[
Q_{ij} = \begin{cases} 
1 & P_i = P_j, \\
0 & P_i \neq P_j. 
\end{cases}
\]  
Here, \( Q_{ij} \) reveals the pseudo relation between \( i \)-th and \( j \)-th samples. Precisely, \( Q_{ij} = 1 \) means \( i \)-th and \( j \)-th samples are more likely to be the positive sample pair while \( Q_{ij} = 0 \) implies they are more likely to be the negative ones.

**Hard Sample Aware Contrastive Learning**

In this section, we firstly introduce the drawback of classical infoNCE loss (Van den Oord et al. 2018) in the graph contrastive methods (Zhu et al. 2020; Xia et al. 2022a) and then propose a novel hard sample aware contrastive loss to guide our network to focus more on hard positive and negative samples.

The classical infoNCE loss for \( i \)-th sample in the \( j \)-th view is formulated as follows:

\[
L_{infoNCE}(i^{v_j}) = -\log e^{\theta(i^{v_j}, j^{v_j})} + \sum_{k \neq i} (e^{\theta(i^{v_j}, k^{v_l})} + e^{\theta(k^{v_l}, i^{v_j})}).
\]  
where \( j \neq l \). Besides, \( \theta(\cdot) \) denotes the cosine similarity between the paired samples in the latent space. By minimizing infoNCE loss, they pull together the same samples in different views while pushing away other samples.

However, we find the drawback of classical infoNCE is that the hard sample pairs are treated equally to the easy ones, limiting the discriminative capability of the network. To solve this problem, we propose a weight modulating function \( M \) to dynamically adjust the weights of sample pairs during training. Concretely, based on the proposed attribute-structure similarity function \( S \) and the generated sample pair pseudo labels \( Q \), \( M \) is formulated as follows:

\[
M(i^{v_j}, k^{v_l}) = \begin{cases} 
1, & -(i, k \in H), \\
Q_{ik} - \text{Norm}(S(i^{v_j}, k^{v_l}))^\beta, & \text{others}. 
\end{cases}
\]  
where \( i^{v_j} \) denotes the \( i \)-th samples in the \( j \)-th view and \( \text{Norm} \) denotes the min-max normalization. In Eq. (9), when \( i \)-th or \( j \)-th sample is not with high-confidence, i.e., \(-\infty, \infty \in H \), we keep the original setting in the infoNCE loss. Differently, the sample weights are modulated with the pseudo information and the similarity of samples, when the samples are with high-confidence. Here, the hyper-parameter \( \beta \in [1, 5] \) is the focusing factor, which determines the down-weighting rate of the easy sample pairs. In the following, we analyze the properties of the proposed weight modulating function \( M \).

1) \( M \) can up-weight the hard samples while down-weighting the easy samples. Concretely, when \( i \)-th, \( j \)-th samples are recognized as positive sample pair \( Q_{ij} = 1 \), the hardness of pulling them together decreases as the similarity increases. Thus, \( M \) up-weights the positive sample pairs with small similarity (the hard ones) while down-weighting that with large similarity (the easy ones). For example, when \( Q_{ij} = 1 \) and \( \beta = 2 \), the easy positive sample pair with 0.90 similarity is weighted with 0.01. Differently, the hard positive sample pair with 0.10 similarity is
weighted with 0.81, which is significantly larger than 0.01. The similar conclusion can be deduced on the negative sample pairs. Experimental evidence can be found in Figure 1.

2) The focusing factor $\beta$ controls the down-weighting rate of easy sample pairs. Concretely, when $\beta$ increases, the down-weighting rate of easy sample pairs increases and vice versa. Take the positive sample pair as an example ($Q_{ij} = 1$), the easy ones with 0.9 similarity be down-weighted to $0.1^\beta$. Here, when $\beta = 1$, the weight is set to 0.1. Differently, when $\beta = 3$, the weight is degraded to 0.001, which is obviously smaller than 0.1. This property is verified by visualization experiments in Figure 2-3 of Appendix.

Based on $S$ and $M$, we formulate the hard sample aware contrastive loss for $i$-th sample in $j$-th view as follows:

$$L(i^v_i) = -\log\left(\sum_{l \neq j} e^{M(i^v_j, i^v_i)} \cdot S(i^v_j, i^v_i) + \sum_{l \in \{1,2\}} e^{M(i^v_j, k^v_i)} \cdot S(i^v_j, k^v_i)\right)$$

(10)

where $k \neq i$. Compared with the classical infoNCE loss, we first adopt a more comprehensive similarity measure criterion $S$ to assist sample hardness measurement. Then we propose a weight modulating function $M$ to up-weight hard sample pairs and down-weight the easy ones. In summary, the overall loss of our method is formulated as follows:

$$L = \frac{1}{2N} \sum_{j=1}^{2} \sum_{i=1}^{N} L(i^v_i).$$

(11)

This hard sample aware contrastive loss can guide our network to focus on not only the hard negative samples but also the hard positive ones, thus improving the discriminative capability of samples further. We summarize two reasons as follows. 1) The proposed attribute-structure similarity function $S$ considers attribute and structure information, thus better revealing the sample relations. 2) The proposed weight modulating function $M$ is a general dynamic sample weighting strategy for positive and negative sample pairs. It can up-weight the hard sample pairs while down-weighting the easy ones during training.

**Complexity Analysis of Loss Function**

In this section, we analyze the time and space complexity of the proposed hard sample aware contrastive loss $L$. Here, we denote the batch size is $B$, the number of the high confident samples in this batch is $M$, and the dimension of embeddings is $d$. The time complexity of $S$ and $M$ is $O(B^2d)$ and $O(M^2d)$, respectively. Since $M < B$, the time complexity of the whole hard sample aware contrastive loss is $O(B^2d)$. Besides, the space complexity of our proposed loss is $O(B^3)$. Thus, the proposed loss will not bring the high time or space costs compared with the classical infoNCE loss. The detailed process of our proposed HSAN is summarized in Algorithm 1. Besides, the PyTorch-style pseudo code can be found in Appendix.

**Experiments**

**Dataset**

To evaluate the effectiveness of our proposed HSAN, we conduct experiments on six benchmark datasets, including CORA, CITE, Amazon Photo (AMAP), Brazil Air Traffic (BAT), Europe Air Traffic (EAT), and USA Air Traffic (UAT). The brief information of datasets is summarized in Table 1 of the Appendix.

**Experimental Setup**

All experimental results are obtained from the desktop computer with the Intel Core i7-7820X CPU, one NVIDIA GeForce RTX 2080Ti GPU, 64GB RAM, and the PyTorch deep learning platform. The training epoch number is set to 400 and we conduct ten runs for all methods. For the baselines, we adopt their source with original settings and reproduce the results. To our model, both the attribute encoders and structure encoders are two parameters un-shared one-layer MLPs with 500 hidden units for UAT/AMAP and 1500 hidden units for other datasets. The learnable trade-off $\alpha$ is set to 0.99999 as initialization and reduces to around 0.4 in our experiments as shown in Figure 4 of Appendix. The hyper-parameter settings are summarized in Table 1 of Appendix. The clustering performance is evaluated by four metrics, i.e., ACC, NMI, ARI, and F1, which are widely used in both deep clustering (Liu et al. 2022c; Xia et al. 2022c; Bo et al. 2020; Tu et al. 2020; Liu et al. 2022f) and traditional clustering (Zhou et al. 2019; Zhang et al. 2022b, 2021; Liu et al. 2022b; Chen et al. 2022a; Li et al. 2022; Zhang et al. 2022a, 2020; Sun et al. 2021; Wan et al. 2022; Wang et al. 2022b, 2021a).

**Performance Comparison**

To demonstrate the superiority of our proposed HSAN, we conduct extensive comparison experiments on six benchmark datasets. Concretely, we category these thirteen state-of-the-art deep graph clustering methods into three types, i.e., classical deep graph clustering methods (Wang et al. 2017, 2019; Bo et al. 2020; Tu et al. 2020), contrastive...
Table 2: The average clustering performance of ten runs on six datasets. The performance is evaluated by four metrics with mean value and standard deviation. The bold and underlined values indicate the best and runner-up results, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>Classical Deep Graph Clustering</th>
<th>Contrastive Deep Graph Clustering</th>
<th>Hard Sample Mining</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DAEGC</td>
<td>ARGA</td>
<td>DFCN</td>
</tr>
<tr>
<td>CORA</td>
<td>ACC</td>
<td>70.43±0.36</td>
<td>71.04±0.25</td>
<td>63.34±0.49</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>52.89±0.69</td>
<td>51.06±0.52</td>
<td>39.36±0.87</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>49.63±0.43</td>
<td>47.71±0.33</td>
<td>43.67±2.10</td>
</tr>
<tr>
<td></td>
<td>FI</td>
<td>68.27±0.57</td>
<td>69.27±0.39</td>
<td>26.16±5.00</td>
</tr>
<tr>
<td>CITE</td>
<td>ACC</td>
<td>64.54±1.39</td>
<td>61.07±0.49</td>
<td>69.30±0.20</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>36.41±0.86</td>
<td>34.40±0.71</td>
<td>43.90±0.20</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>37.78±1.24</td>
<td>40.17±0.43</td>
<td>45.31±0.41</td>
</tr>
<tr>
<td></td>
<td>FI</td>
<td>62.20±1.32</td>
<td>58.23±0.31</td>
<td>64.30±0.20</td>
</tr>
<tr>
<td>AMAP</td>
<td>ACC</td>
<td>75.96±0.23</td>
<td>69.28±2.30</td>
<td>76.82±0.23</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>65.25±0.45</td>
<td>58.36±2.76</td>
<td>66.23±1.21</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>58.12±0.24</td>
<td>44.18±4.41</td>
<td>58.20±0.74</td>
</tr>
<tr>
<td></td>
<td>FI</td>
<td>69.87±0.54</td>
<td>64.30±1.95</td>
<td>71.25±0.31</td>
</tr>
<tr>
<td>BAT</td>
<td>ACC</td>
<td>52.67±0.00</td>
<td>67.86±0.80</td>
<td>55.73±0.06</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>21.43±0.35</td>
<td>49.09±0.54</td>
<td>48.77±0.51</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>18.18±0.29</td>
<td>42.02±1.21</td>
<td>37.76±0.23</td>
</tr>
<tr>
<td></td>
<td>FI</td>
<td>52.23±0.03</td>
<td>67.02±1.15</td>
<td>50.90±0.12</td>
</tr>
<tr>
<td>EAT</td>
<td>ACC</td>
<td>36.89±0.15</td>
<td>52.13±0.00</td>
<td>49.37±0.19</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>05.57±0.06</td>
<td>22.48±1.21</td>
<td>32.90±0.41</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>03.03±0.08</td>
<td>17.29±0.50</td>
<td>32.05±0.18</td>
</tr>
<tr>
<td></td>
<td>FI</td>
<td>34.72±0.36</td>
<td>52.75±0.97</td>
<td>32.95±0.04</td>
</tr>
<tr>
<td>UAT</td>
<td>ACC</td>
<td>52.29±0.49</td>
<td>49.31±0.15</td>
<td>33.64±0.09</td>
</tr>
<tr>
<td></td>
<td>NMI</td>
<td>21.33±0.44</td>
<td>25.44±0.31</td>
<td>26.49±0.41</td>
</tr>
<tr>
<td></td>
<td>ARI</td>
<td>20.50±0.51</td>
<td>16.57±0.31</td>
<td>22.87±1.03</td>
</tr>
<tr>
<td></td>
<td>FI</td>
<td>50.33±0.64</td>
<td>50.26±0.16</td>
<td>25.79±0.29</td>
</tr>
</tbody>
</table>

Figure 3: 2D t-SNE visualization of seven methods on two benchmark datasets. The first row and second row corresponds to CORA and AMAP dataset, respectively.
3) The combination of $S$ and $M$ achieves the most superior clustering performance. For example, “Ours” exceeds “B” 8.48% with the NMI metric on the CORA dataset. Overall, the effectiveness of our proposed attribute-structure similarity function $S$ and weight modulating function $M$ is verified by extensive experimental results.

<table>
<thead>
<tr>
<th>(a) CORA</th>
<th>(b) AMAP</th>
<th>(c) EAT</th>
<th>(d) CITE</th>
<th>(e) BAT</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Ablation plots" /></td>
<td><img src="image2.png" alt="Ablation plots" /></td>
<td><img src="image3.png" alt="Ablation plots" /></td>
<td><img src="image4.png" alt="Ablation plots" /></td>
<td><img src="image5.png" alt="Ablation plots" /></td>
</tr>
</tbody>
</table>

Figure 4: Ablation studies of the proposed similarity function $S$ and weight modulating function $M$ on six datasets.

**Analysis**

**Hyper-parameter Analysis** In this section, we analyze the hyper-parameters $\tau$ and $\beta$ in our method. For the confidence $\tau$, we select it in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. As shown in Figure 5, we observe that our method achieves promising performance when $\tau \in [0.1, 0.3]$ on BAT / CITE datasets and when $\tau \in [0.7, 0.9]$ on other datasets. In this paper, the confidence is set to a fixed value, thus a possible future work is to design a learnable or dynamical confidence. Besides, we analyze $\beta$ in Appendix. Two conclusions are deduced as follows. 1) The focusing factor $\beta$ controls the down-weighing rate of easy sample pairs. When $\beta$ increases, the down-weighing rate of easy sample pairs also increases and vice versa. 2) HSAN is not sensitive to $\beta$. Experimental evidence can be found in Figure 1-3 in Appendix.

**Visualization Analysis** To further demonstrate the superiority of HSAN intuitively, we conduct 2D t-SNE (Van der Maaten and Hinton 2008) on the learned node embeddings in Figure 3. It is observed that our HSAN can better reveal the cluster structure compared with other baselines.

**Convergence Analysis** In addition, we analyze the convergence of our proposed loss. Specifically, we plot the trend of loss and the clustering ACC of our method in Figure 6. Here, “Ours” and “infoNCE” denotes our method with our hard sample aware contrastive loss, and the infoNCE loss, respectively. We observe 1) the loss and ACC gradually converge after 350 epochs. 2) after 50 epochs, “Ours” calculates the larger loss value while achieving better performance compared with “infoNCE”. The reason is that our loss down-weights the easy sample pairs while up-weighting the hard sample pairs, thus increasing the loss value. Meanwhile, the proposed loss guides the network to focus on the hard sample pairs, leading to better performance.

**Conclusion**

This paper proposes a Hard Sample Aware Network (HSAN) to mine the hard samples in contrastive deep graph clustering. The core idea is to up-weight the hard sample pairs while down-weighting the negative ones. In this manner, the proposed hard sample aware contrastive loss forces the network to focus on positive and negative sample pairs, thus further improving the discriminative capability. The time and space analysis of the proposed loss demonstrates that it will not bring large time or space costs compared with the classical infoNCE loss. Experiments demonstrate the effectiveness and superiority of our proposed method. The weakness of HSAN is that the confidence parameter is set to a fixed value. Thus, one worth future work is to design a learnable or adaptive confidence parameter.
Acknowledgments
This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61976196, 62006237, and 61872371).

References
Mrabah, N.; Bouguessa, M.; Touati, M. F.; and Ksantini, R. 2022. Rethinking graph auto-encoder models for attributed graph clustering. IEEE Transactions on Knowledge and Data Engineering.


Xia, W.; Gao, Q.; Wang, Q.; Gao, X.; Ding, C.; and Tao, D. 2022c. Tensorized Bipartite Graph Learning for Multi-View Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence.


Zhao, H.; Yang, X.; Wang, Z.; Yang, E.; and Deng, C. 2021. Graph debiased contrastive learning with joint representation clustering. In Proc. of IJCAI.


