Hierarchically Contrastive Hard Sample Mining for Graph Self-Supervised Pretraining

Wenxuan Tu, Sihang Zhou, Xinwang Liu, *Senior Member, IEEE*, Chunpeng Ge, Zhiping Cai, *Member, IEEE*, and Yue Liu

***Abstract*— Contrastive learning has recently emerged as a powerful technique for graph self-supervised pretraining (GSP). By maximizing the mutual information (MI) between a positive sample pair, the network is forced to extract discriminative information from graphs to generate high-quality sample representations. However, we observe that, in the process of MI maximization (Infomax), the existing contrastive GSP algorithms suffer from at least one of the following problems: 1) treat all samples equally during optimization and 2) fall into a single contrasting pattern within the graph. Consequently, the vast number of well-categorized samples overwhelms the representation learning process, and limited information is accumulated, thus deteriorating the learning capability of the network. To solve these issues, in this article, by fusing the information from different views and conducting hard sample mining in a hierarchically contrastive manner, we propose a novel GSP algorithm called hierarchically contrastive hard sample mining (HCHSM). The hierarchical property of this algorithm is manifested in two aspects. First, according to the results of multilevel MI estimation in different views, the MI-based hard sample selection (MHSS) module keeps filtering the easy nodes and drives the network to focus more on hard nodes. Second, to collect more comprehensive information for hard sample learning, we introduce a hierarchically contrastive scheme to sequentially force the learned node representations to involve multilevel intrinsic graph features. In this way, as the contrastive granularity goes finer, the complementary information from different levels can be uniformly encoded to boost the discrimination of hard samples and enhance the quality of the learned graph embedding. Extensive experiments on seven benchmark datasets indicate that the HCHSM performs better than other competitors on node classification and node clustering tasks. The source code of HCHSM is available at https://github.com/WxTu/HCHSM.**

***Index Terms*— Contrastive learning, graph self-supervised pretraining (GSP), hard sample mining, multiview learning, mutual information (MI).**

SUMMARY OF MAIN NOTATIONS

Notation	Explanation
$\mathbf{X} \in \mathbb{R}^{N\times D}$	Raw attribute matrix.
$\mathbf{A} \in \mathbb{R}^{N\times N}$	Raw adjacency matrix.
$\mathbf{D} \in \mathbb{R}^{N\times N}$	Degree matrix.
$\mathbf{I} \in \mathbb{R}^{N\times N}$	Identical matrix.
$\mathbf{X}^v \in \mathbb{R}^{N\times D}$	Attribute matrix in the vth view.
$\mathbf{X}_c^v \in \mathbb{R}^{N\times D}$	Corrupted attribute matrix in the vth view.
$\widetilde{\mathbf{A}}^v \in \mathbb{R}^{N\times N}$	Normalized adjacency matrix in the vth view.
$\mathbf{Z}_{\text{pos}}^v \in \mathbb{R}^{N\times d}$	Positive graph embedding in the vth view.
$\mathbf{Z}_{\text{neg}}^v \in \mathbb{R}^{N\times d}$	Negative graph embedding in the vth view.
$\mathbf{Z}_{\text{pos}} \in \mathbb{R}^{N\times d}$	Positive consensus graph embedding.
$\mathbf{Z}_{\text{neg}} \in \mathbb{R}^{N\times d}$	Negative consensus graph embedding.
$\mathbf{g}^v \in \mathbb{R}^{1\times d}$	Global graph embedding in the vth view.
$\mathbf{s}_{\text{pos}}^v \in \mathbb{R}^{N\times 1}$	Positive MI agreement score vector in the vth view.
$\mathbf{s}_{\text{neg}}^v \in \mathbb{R}^{N\times 1}$	Negative MI agreement score vector in the vth view.
$\mathbf{H}^v \in \mathbb{R}^{N\times d}$	Structure-enhanced graph embedding in the vth view.

I. INTRODUCTION

GRAPH self-supervised pretraining (GSP) has gained significant interest among machine learning researchers as an increasingly attractive direction. It aims to learn a graph encoder to preserve the latent structure and attribute information from raw graphs without human-annotated labels for better performance on downstream tasks. Because of the strong representation learning capability of graph neural networks (GNNs), researchers in this field have achieved encouraging performance across various applications, including anomalous citation detection [1], few-shot learning [2], feature selection [3], and knowledge graph [4], [5]. Among all GSP algorithms [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], contrastive GSP has garnered significant attention from researchers and been the dominant paradigm recently since its powerful learning capacity and impressive performance.

With a carefully designed sample contrastive mechanism, contrastive GSP algorithms initially establish positive and

Manuscript received 6 October 2022; revised 20 January 2023 and 26 April 2023; accepted 11 July 2023. Date of publication 17 August 2023; date of current version 30 October 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0107703, in part by the National Natural Science Foundation of China under Grant 61922088 and Grant 62006237, and in part by the Postgraduate Scientific Research Innovation Project in Hunan Province under Grant CX20220076. *(Corresponding authors: Xinwang Liu; Zhiping Cai.)*

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

Wenxuan Tu, Xinwang Liu, Zhiping Cai, and Yue Liu are with the School of Computer, National University of Defense Technology, Changsha 410073, China (e-mail: wenxuantu@163.com; xinwangliu@nudt.edu.cn; zpcai@nudt.edu.cn; yueliu@nudt.edu.cn).

Sihang Zhou is with the School of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China (e-mail: sihangjoe@gmail.com).

Chunpeng Ge is with the School of Software, Shandong University, Shandong 250100, China (e-mail: gechunpeng2022@126.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2023.3297607, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2023.3297607

negative sample pairs and then pretrain one or more encoders to ensure that the representations of a positive sample pair agree with each other, while those of a negative sample pair disagree with each other [20], [21], [22], [23], [24]. Different contrastive GSP algorithms in this field conduct mutual information (MI) estimation at different scales, which can be broadly grouped into two types [25], i.e., **same-scale contrasting** and **cross-scale contrasting**. Concretely, same-scale graph contrastive learning (GraphCL) aims to preserve consistent information by maximizing the MI between two-view node representations (i.e., a node-to-node pair) [26], [27] or graph summaries (i.e., a graph-to-graph pair) [12], [28]. For example, graph contrastive representation learning (GRACE) [29] maximizes the MI agreement of node representations in two augmented views with an improved InfoNCE loss, and GraphCL [12] extracts the information of augmented graphs to learn their global representations for MI estimation, while the goal of cross-scale GraphCL is to preserve the underlying manifold and attribute information of the graph by conducting the MI estimation between different-scale representations of augmented graphs. For instance, deep graph infomax (DGI) [30] and MVGRL [23] estimate the feature similarity between a node embedding and a graph embedding (i.e., a node-to-graph pair) to learn meaningful latent variables via MI maximization. SUGAR [31] and InfoGraph [32] propose a self-supervised MI scheme, which promotes the subgraph embedding to capture global graph properties by maximizing the MI between a subgraph and the entire graph (i.e., a subgraph-to-graph pair).

Although recent efforts have achieved superior performance enhancement by leveraging various GraphCL techniques, we observe that current contrastive GSP algorithms suffer from at least one of the following issues when conducting the MI estimation between a positive (or negative) sample pair. First, the existing MI estimators equally collect and preserve the information of all node representations. Most contrastive GSP assume that all samples contribute equally to the contrastive learning target, e.g., InfoNCE loss [12], [27], [31]. However, the equal treatment optimization strategy would cause the network pretraining to be overwhelmingly guided by easy samples and ignore hard but important boundary samples. Second, most contrastive GSP algorithms fall into a single contrasting pattern within the graph. For a given complete graph, conducting a scale-fixed MI estimation between a sample pair (e.g., a node-to-node pair [21], [29], a node-to-subgraph pair [22], [33], or a node-to-graph pair [23], [30]) would make the network tend to be biased in fitting either the global or very local representations. In this case, it is usually not enough to tell apart similar sample pairs with only one glimpse within a single perspective. Consequently, a multitime comparison between a positive (or negative) sample pair from different perspectives is needed.

To tackle the above issues, we propose a novel GSP algorithm called hierarchically contrastive hard sample mining (HCHSM). The core idea behind our solution is to focus more on hard samples and collect multilevel graph information to boost the quality of their representations for better downstream performance. Specifically, as a core component, the MHSS module is elaborately designed to drive the network to concentrate more on the sample pairs that are hard to tell. Since preserving multilevel information is proved to be crucial for sample representations [34], [35], we propose a hierarchically contrastive scheme to facilitate the implicit training of the MHSS module by utilizing three types of contrasting patterns. This enables the learning of diverse granularity representations for hard samples. In the proposed hierarchically contrastive scheme, as the contrastive granularity goes finer, the representations of hard samples could effectively capture the global, neighbor, and individual information from various perspectives. By this means, multiple observations from different views and levels are constructed to learn higher quality graph embedding for more precise performance. The key contributions of this study are fourfold.

1) A novel contrastive GSP algorithm called HCHSM is proposed for graph data analysis without relying on the labeling information.
2) An MHSS module for contrastive GSP is innovatively designed, which addresses the hard sample mining issue from the MI estimation perspective.
3) A hierarchically contrastive scheme that can collect multilevel graph information for hard sample comparison and representation learning is proposed.
4) Extensive experiments on seven benchmark datasets have demonstrated that the HCHSM is highly competitive and consistently outperforms most state-of-the-art competitors on node classification and clustering downstream tasks.

The rest of this article is structured as follows. Section II reviews and discusses the related work. Section III presents the model design and describes each component of HCHSM. Section IV presents the experiments and analyzes the results. Section V draws a conclusion and discusses future work.

II. Related Work

A. Graph Representation Learning

Early graph representation learning (GRL) algorithms learn node representations by utilizing probability models to handle the random walk paths generated over graphs [36], [37]. However, these random-walk-based algorithms overly emphasize the structure information and overlook the rich attribute information. Because of the rapid development of GNNs, GNN-oriented GRL algorithms, which jointly exploit graph structures and node attributes in a spectral or spatial domain, have been extensively researched in recent years. Based on their learning paradigms, these algorithms can be roughly divided into supervised learning-based algorithms [15], [38], [39], [40], [41] and unsupervised learning-based algorithms [9], [10], [11], [12], [30], [42], [43], [44], [45], [46]. Recently, GSP has gained popularity in unsupervised GRL. Following the GSP paradigm, the proposed HCHSM can learn high-quality representations from supervision signals derived from the data itself, without relying on excessively annotated labels.

B. Graph Self-Supervised Pretraining

GSP aims to learn more generalized representations from massive unlabeled graphs through pretext tasks, so as to

the learned features could be saved and utilized for better performance on downstream tasks. The existing GSP algorithms can be roughly classified into three categories: generative GSP algorithms, predictive GSP algorithms, and contrastive GSP algorithms [25]. For example, as a representative of generative GSP algorithms, SelfTask-GNN [22] combines node-level and graph-level pretraining with a powerful GNN to perform masked node feature reconstruction and edge feature reconstruction on a given graph. Similarly, L2P-GNN [47] introduces a meta-learning-based predictive framework, where both node-level information and graph-level information are extracted as pseudo-labels for GSP. More recently, self-supervised heterogeneous graph pretraining (SHGP) [48] incorporates two attention-based modules that mutually enhance each other, leading to more precise predictions and pseudo-labels for GSP.

On another research line, GraphCL, which aims to learn node representations for downstream tasks by conducting sample discrimination between positive and negative pairs, has emerged as a popular unsupervised technique for GSP over the past three years [14], [21], [22], [23], [26], [27], [29], [30], [31], [49], [50], [51]. According to contrastive granularity, these algorithms can be divided into two types, i.e., same-scale contrasting and cross-scale contrasting. In previous same-scale GraphCL works, GRACE [29] and GRACE with adaptive augmentation (GCA) [26] maximize the node-to-node-level MI agreement between data representations of two augmented views with an improved InfoNCE loss. Graph contrastive coding (GCC) [49] first generates multiple subgraphs over each graph using a random walk algorithm and then contrasts two subgraph-level representations of positive (or negative) sample pairs in the latent space. Moreover, GraphCL [12] first generates two correlated graph views by performing corruption randomly and then learns representations by directly maximizing the MI agreement between two-view graph embeddings. For heterogeneous graphs, contrastive pretraining strategy of GNNs on heterogeneous graph (CPT-HG) [52] conducts semantic-aware pretraining tasks through relation-level and subgraph-level contrastive learning, respectively. Also, cross-scale GraphCL algorithms show encouraging performance for GSP. For instance, DGI [30] estimates the feature similarity between an arbitrary node embedding and a global graph embedding to learn useful information via MI maximization. Sub-graph contrast (SUBG-CON) [33] exploits the intimate correlation between a central node and its neighbors (i.e., a subgraph) to extract contextual structural features. Pretraining GNNs on heterogeneous graph (PT-HGNN) [53] contrastively leverages both node relations and the graph schema to extract heterogeneous structural and semantic information. More recently, SUGAR [31] designs a subgraph framework with reinforcement pooling and MI mechanism, which encourages the subgraph embedding to consider global graph properties by maximizing their MI. Although the existing contrastive GSP algorithms achieve impressive performance, most of these algorithms usually fall into a single contrasting pattern within the graph when conducting a scale-fixed MI estimation for contrastive learning. Consequently, the learned node representations could easily tend to be biased in fitting either the global or very local pattern. Inspired by SelfTask-GNN [22] and L2P-GNN [47], this article explores the collection and preservation of multilevel graph information (i.e., node, subgraph, and graph features) to enhance the learned representations of hard samples, which has not yet been adequately studied in the domain of contrastive GSP.

C. Hard Sample Mining on Graphs

Motivated by the remarkable achievements of hard sample mining in visual deep representation learning [54], [55], [56], there has been an increasing tendency toward proposing and employing techniques to estimate the hardness of samples in the graph domain recently. On the one hand, some efforts leverage graph augmentation techniques to estimate the sample importance at the edge and attribute levels. For instance, MVGRL [23] applies an edge diffusion augmentation to preserve more informative edges and filter noisy connections for cross-view graph contrasting learning. Similarly, GraphCL [12] first samples reliable subgraphs as augmented graphs according to the results of the random-walk algorithm and then employs contrastive learning to predict whether two graphs have the same origin or not. Recently, GCA [26] performs adaptive graph data augmentations by utilizing the network centrality to filter insignificant node attributes and sample connections. Despite their encouraging successes, these methods treat the graph augmentation procedure as a separate step from the model optimization procedure. More recently, attributed graph clustering with dual redundancy reduction (AGC-DRR) [57] integrates data augmentation and representation learning processes into a united optimization framework to refine graph structures for node clustering. However, it assumes that all samples of augmented views have an equal impact on the learning objective, resulting in inferior representations for clustering. On the other hand, to enable the network to concentrate more on hard samples, recent efforts have begun carefully designing specialized hard sample mining techniques for GSP. Heterogeneous graph contrastive learning with structure-aware hard negative mining (HORACE) [58] establishes a structure-aware hard sample mining scheme, which evaluates the sample hardness via structural properties for heterogeneous academic graphs. In ProGCL [59], a two-component beta mixture model (BMM) is fit on the feature similarity to distinguish easy and hard samples, where the BMM estimates the probability of a negative statement being true with respect to a specific anchor. Although proven to be a powerful tool, the technique of hard sample mining on homogenous GSP is relatively less explored. In this work, the proposed HCHSM integrates both hard sample mining and representation learning within a common optimization framework. Moreover, by selecting and preserving hard samples for further exploration in the embedding space, the proposed method offers a more unified and effective solution for GSP.

III. Proposed Approach

A. Notations and Definitions

Consider an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with C classes, where $\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N\}$ and $\mathcal{E}$ are the node set and edge set, respectively, and N denotes the number of samples.

In classical graph machine learning [38], the graph is usually characterized by a raw attribute matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$ and a raw adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Here, D denotes the attribute dimension, and $\mathbf{A}_{ij} = 1$ if there exists an edge between node $\boldsymbol{v}_i$ and node $\boldsymbol{v}_j$ in the graph $\mathcal{G}$; otherwise, $\mathbf{A}_{ij} = 0$. The corresponding degree matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ is defined, such that $\mathbf{D}_{ii} = \sum_{\boldsymbol{v}_j\in\mathcal{V}} \mathbf{A}_{ij}$. With the degree matrix, the raw adjacency matrix can be normalized as $\widetilde{\mathbf{A}} \in \mathbb{R}^{N\times N}$ through calculating $\mathbf{D}^{-(1/2)}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-(1/2)}$, where the identity matrix $\mathbf{I} \in \mathbb{R}^{N\times N}$ represents that each node in $\mathcal{V}$ is linked with a self-loop structure. Nomenclature summarizes the main notations.

Definition 1 (MI Agreement Score): Given a node $\boldsymbol{v}_i$, the agreement score $\mathbf{s}_{\text{pos},i}$ (or $\mathbf{s}_{\text{neg},i}$) between a positive (or negative) sample pair is usually measured by MI [25], which reflects the underlying correlations between two instances within a sample pair.

Definition 2 (Hard Sample): We quantify the sample discriminative capability according to both positive and negative MI agreement scores. Given a graph $\mathcal{G}$, it is feasible to identify hard samples based on the assumption that a node $\boldsymbol{v}_i$ with a high loss (i.e., the result of $\mathbf{s}_{\text{neg},i} - \mathbf{s}_{\text{pos},i}$ maintains a relatively large value in our settings) can be regarded as a hard node.

Definition 3 (Learning Problem): This study mainly focuses on addressing the hard sample mining issue on graphs without label annotation. Our model works for learning a graph encoding function $f_g(\cdot)$ to generate node representations and a hierarchical MI estimating function $f_e(\cdot)$ to predict the MI agreement scores for positive (or negative) sample pairs.

B. Overview of HCHSM

As illustrated in Fig. 1, the proposed HCHSM mainly includes five steps: graph augmentation, sample embedding, MHSS, hierarchically contrastive scheme, and contrastive objective.

1) Graph Augmentation: Similar to previous work (e.g., MVGRL [60] and adaptive graph encoder (AGE) [61]), we apply the personalized page rank (PPR) technique and Laplacian smoothing operation for data augmentations. Concretely, we first conduct the graph structure diffusion over $\widetilde{\mathbf{A}}$ by calculating the following formulation:

$$\widetilde{\mathbf{A}}_{\text{PPR}} = \alpha\left(\mathbf{I} - (1-\alpha)\widetilde{\mathbf{A}}\right)^{-1} \tag{1}$$

where α refers to the teleport probability in a random walk. Next, to filter the high-frequency signals and preserve the low-frequency ones, we apply a Laplacian smoothing filter to smooth $\mathbf{X}$

$$\mathbf{H} = \mathbf{I} - m\left(\widetilde{\mathbf{D}} - \widetilde{\mathbf{A}}\right) \tag{2}$$

$$\mathbf{X}_L = \mathbf{H}^t\mathbf{X} \tag{3}$$

where $\widetilde{\mathbf{D}}$ denotes the degree matrix of $\widetilde{\mathbf{A}}$. t is the frequency of Laplacian smoothing filters and is initialized as 1. m is a real value initialized as (2/3). Thereafter, we take the original graph $\mathcal{G}^1_{\text{pos}} = (\mathbf{X}, \widetilde{\mathbf{A}})$ and the augmented graph $\mathcal{G}^2_{\text{pos}} = (\mathbf{X}_L, \widetilde{\mathbf{A}}_{\text{PPR}})$ as two positive graph views. Based on $\mathcal{G}^1_{\text{pos}}$ and $\mathcal{G}^2_{\text{pos}}$, we then rowwise shuffle all nodes using a corruption function $f_c(\cdot)$ [30] to construct negative versions $\mathcal{G}^1_{\text{neg}} = (\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}})$ and $\mathcal{G}^2_{\text{neg}} = (\widetilde{\mathbf{X}}_L, \widetilde{\mathbf{A}}_{\text{PPR}})$.

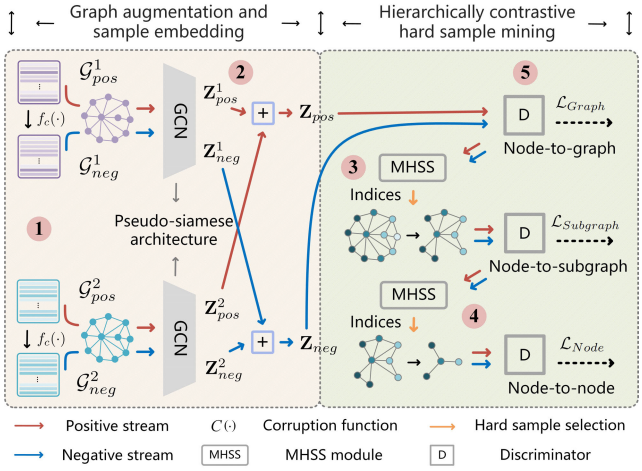

Fig. 1. Overall framework of the proposed HCHSM, which mainly contains five steps. ① *Graph augmentation*—in this step, we generate two views of the original graph and construct contrastive objects. ② *Sample embedding*—in this step, we first adopt two GCN-based encoders to extract features, and then, the resultant multiview graph embeddings are aggregated with a simple elementwise addition layer. ③ *Hard sample selection*—in this step, we utilize the MHSS module to filter easy samples and obtain hard samples for further learning according to node indices. ④ *Hierarchically contrastive learning*—in this step, we conduct a hierarchically contrastive scheme to collect multilevel graph information for hard sample representation learning. ⑤ *Objective calculation*—in this step, we gather multilevel contrastive objectives to serve as the final loss function for network training. The byproduct of the contrastive objective, i.e., the MI agreement scores, is intimately associated with the criterion of hard sample selection. For more detailed information, please refer to the descriptions in the method section.

2) Sample Embedding: By taking $\mathcal{G}^v_{\text{pos}}$ and $\mathcal{G}^v_{\text{neg}}$ as inputs, we denote a vth-view graph convolutional network (GCN)-based encoder $f^v_g(\cdot)$: $\mathbf{Z}^v = f^v_g(\mathcal{G}^v)$, where $v \in [1, 2]$, to learn the positive graph embedding $\mathbf{Z}^v_{\text{pos}}$ and the negative graph embedding $\mathbf{Z}^v_{\text{neg}}$ in the vth view. Note that two encoders are structure-identical but parameter-decoupled. By performing an elementwise addition operation $\sum_{v=1}^{V} \mathbf{Z}^v_{\text{pos}}$, the extracted multiview graph embeddings are integrated to obtain the positive consensus graph embedding $\mathbf{Z}_{\text{pos}}$. The generation of the negative one $\mathbf{Z}_{\text{neg}}$ is similar to that of $\mathbf{Z}_{\text{pos}}$.

3) MI-Based Hard Sample Selection: A sample selection module is designed to select and preserve those nodes that are hard to tell for further exploration. In this module, hard samples can be determined according to a normalized score vector, which quantifies the MI agreement gap between negative and positive sample pairs.

4) Hierarchically Contrastive Scheme: Three discriminators are employed to contrast positive and negative sample pairs, where the agreement between patch representations is evaluated by applying the techniques of MI estimation. Following this principle, we conduct a hierarchically contrastive scheme, which collects multiview intrinsic information of graphs from different levels to boost the discriminative capability of hard sample representations.

5) Contrastive Objective: For each node $\boldsymbol{v}_i$, we formulate a contrastive objective $\mathcal{L}$ to estimate its loss. Concretely, $\mathcal{L}$ enables the network to learn to discriminate a predefined positive sample pair $p^+_i = (\mathbf{z}_{\text{pos},i}, \mathbf{y}^v_i)$ from a predefined negative sample pair $p^-_i = (\mathbf{z}_{\text{neg},i}, \mathbf{y}^v_i)$. $\mathbf{z}_{\text{pos},i}$ and $\mathbf{z}_{\text{neg},i}$ are the positive and the negative consensus graph embedding of node

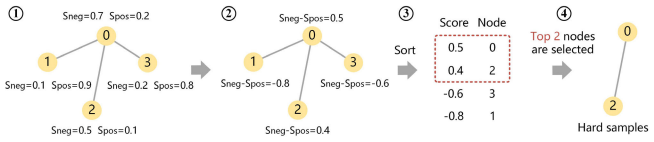

Fig. 2. Toy illustration of the MHSS criterion, which mainly consists of four steps. ① Calculate consensus MI agreement scores of positive (i.e., $\mathbf{s}_{\text{pos}}$) and negative (i.e., $\mathbf{s}_{\text{neg}}$) sample pairs. ② Subtract $\mathbf{s}_{\text{pos}}$ from $\mathbf{s}_{\text{neg}}$. ③ Sort the resultant score vector in descending order. ④ Select top-2 elements (i.e., node 0 and node 2) and obtain their indices for hard sample selection. In our sample-selection criterion, a node $\boldsymbol{v}_i$ with a high loss (i.e., the result of $\mathbf{s}_{\text{neg},i} - \mathbf{s}_{\text{pos},i}$ maintains a relatively large value in our settings) can be regarded as a hard node.

$\boldsymbol{v}_i$, respectively. In addition, $\mathbf{y}_i^v$ denotes patch representations in the vth view, such as a summary of graph features or subgraph features. Note that the consistency of sample pairs is measured through MI in our algorithm.

In the following parts, we will introduce the core components of HCHSM in detail, mainly including the MHSS, the hierarchically contrastive scheme, and the optimization target.

C. Hierarchically Contrastive Hard Sample Mining

Since hard samples usually confuse network learning, treating all samples equally during optimization would limit the discriminative capability of the learned node representations. To solve this issue, the primary consideration is how to design a reliable criterion for selecting hard samples in an unsupervised scenario. Moreover, it is crucial to involve more informative information in the process of hard sample representation learning. To this end, we propose a novel hard sample mining strategy to encourage the network to focus more on the selected hard nodes. In the following, we first introduce the criterion of hard sample selection and next illustrate a hierarchically contrastive scheme.

1) MI-Based Hard Sample Selection: As shown in Fig. 1, we perform two MHSS modules to select hard samples based on the node-to-graph and node-to-subgraph mutual dependencies, respectively. Here, we take the first one as an example for illustration, and the procedure of the second one is similar. Specifically, given a vth-view graph embedding $\mathbf{Z}_{\text{pos}}^v \in \mathbb{R}^{N\times d}$ extracted by $f_g^v(\cdot)$, we first leverage a readout function $f_r(\cdot)$ to learn the vth-view global graph embedding $\mathbf{g}^v \in \mathbb{R}^{1\times d}$ of $\mathcal{G}_{\text{pos}}^v$

$$\mathbf{g}^v = f_r\left(\mathbf{Z}_{\text{pos}}^v\right) \tag{4}$$

where $f_r(\cdot)$ maps all samples into a united vector $\mathbf{g}^v$ that reflects the global information of the graph.

For a node $\boldsymbol{v}_i$, its embedding $\mathbf{z}_{\text{pos},i} \in \mathbb{R}^{1\times d}$ is taken as a positive sample. Naturally, the negative one can be represented as $\mathbf{z}_{\text{neg},i} \in \mathbb{R}^{1\times d}$. Then, we adopt a discriminator $\mathcal{D}(\cdot)$ (i.e., a simple bilinear function) to predict a probability score $\mathbf{s}_{\text{pos},i}^v$ assigned to a vth-view positive sample pair

$$\mathbf{s}_{\text{pos},i}^v = \mathcal{D}\left(\mathbf{z}_{\text{pos},i}, \mathbf{g}^v\right) = f_{\text{sig}}\left(\mathbf{z}_{\text{pos},i}\mathbf{W}(\mathbf{g}^v)^\top\right) \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{d\times 1\times d}$ indicates a learnable weight tensor and $f_{\text{sig}}(\cdot)$ refers to the Sigmoid function, aiming at mapping the predicted score into the probability of $p_i^+ = (\mathbf{z}_{\text{pos},i}, \mathbf{g}^v)$ being a positive sample pair. Similarly, we can obtain $\mathbf{s}_{\text{neg},i}^v$ that quantifies the MI agreement of a vth-view negative sample pair.

After that, we take two-view MI agreement score vectors of positive and negative sample pairs as inputs and transfer them into the MHSS module. As illustrated in Fig. 2, the criterion of hard sample selection within this module mainly includes four steps. First, we integrate the input vectors of all views with a linear combination operation to achieve a consensus MI estimation

$$\mathbf{s}_{\text{pos}} = \beta\mathbf{s}_{\text{pos}}^1 + (1-\beta)\mathbf{s}_{\text{pos}}^2 \tag{6}$$

$$\mathbf{s}_{\text{neg}} = \beta\mathbf{s}_{\text{neg}}^1 + (1-\beta)\mathbf{s}_{\text{neg}}^2 \tag{7}$$

where $\mathbf{s}_{\text{pos}} \in \mathbb{R}^{N\times 1}$ and $\mathbf{s}_{\text{neg}} \in \mathbb{R}^{N\times 1}$ denote the consensus MI agreement score vectors, which reflect the two-source MI agreement with sufficient negotiation between positive sample pairs and negative sample pairs, respectively. The balanced coefficient β is learnable and determines the importance of two-source information selectively. In our settings, we initialize β as 0.5 and then fine-tune it automatically using a gradient descent algorithm.

Second, we define the contrastive labels of positive and negative sample pairs as 1 and 0, respectively. Our goal is to train the network by calculating a contrastive objective (i.e., a cross-entropy-like loss function), which enables the predicted MI agreement score $\mathbf{s}_{\text{pos},i}$ (or $\mathbf{s}_{\text{neg},i}$) of a positive (or negative) sample pair gradually be close to its contrastive label

$$\mathcal{L}_{\text{Graph}} = -\frac{1}{2N}\sum_{v=1}^{V}\left(\sum_{i=1}^{N}\log\mathcal{D}\left(\mathbf{z}_{\text{pos},i}, \mathbf{g}^v\right) + \sum_{i=1}^{N}\log\left(1-\mathcal{D}\left(\mathbf{z}_{\text{neg},i}, \mathbf{g}^v\right)\right)\right) \tag{8}$$

where V and N indicate the number of views and sample pairs, respectively. It is intuitive to observe that contrastive loss serves as a natural metric of the hardness of sample embedding. Therefore, according to (5) and (8), it is feasible to determine hard samples based on the assumption that a node $\boldsymbol{v}_i$ with a high loss (i.e., the result of $\mathbf{s}_{\text{neg},i} - \mathbf{s}_{\text{pos},i}$ maintains a relatively large value in our settings) can be regarded as a hard node. Following this assumption, we subtract $\mathbf{s}_{\text{pos}}$ from $\mathbf{s}_{\text{neg}}$ to obtain a score vector that quantifies the MI agreement gap between positive and negative sample pairs. Third, we sort all nodes in descending order according to the score vector and finally denote the top-K nodes as hard samples in accordance with their indices

$$\text{ind} = \mathcal{T}\left(\mathbf{s}_{\text{neg}} - \mathbf{s}_{\text{pos}}, K\right) \tag{9}$$

where $\mathcal{T}(\cdot)$ denotes an index slicing operation. Specially, K is set to Nr and Nr^2 in the first and second MHSS modules, respectively, where r refers to the hard sample selection ratio. With this module, we could evaluate the discriminative capability of each sample in the latent space and, thus, formulate a novel solution to select hard samples according to the results of MI estimation in different views.

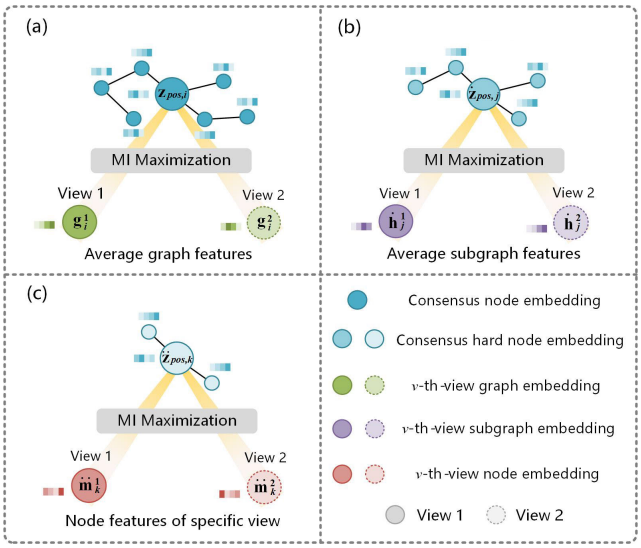

Fig. 3. Illustration of the hierarchically contrastive scheme. (a) Node-to-graph MI maximization for all nodes. (b) Node-to-subgraph and (c) node-to-node MI maximization for hard nodes. For each hard node selected according to (a), we introduce its multiview local structure information to boost its representation by conducting the MI maximization between its embedding and the corresponding average subgraph embedding. The learning procedure of (c) is similar. By doing this, the network encourages the representations of hard samples discriminative enough to tell the true positive (or negative) nodes apart from false positive (or negative) ones.

2) Hierarchically Contrastive Scheme for Hard Samples: To improve the quality of hard sample representation learning, as illustrated in Fig. 3, we develop a hierarchically contrastive scheme to reveal the intrinsic information of the selected hard samples by calculating node-to-subgraph and node-to-node associations, respectively.

Before introducing the node-to-subgraph MI estimation, we first calculate the structure-enhanced graph embedding $\mathbf{H}^v \in \mathbb{R}^{N\times d}$ in the vth view

$$\mathbf{H}^v = f_{\text{sig}}\left(\widetilde{\mathbf{A}}^v \mathbf{Z}^v_{\text{pos}}\right) \tag{10}$$

where $\mathbf{h}_i^v \in \mathbb{R}^{1\times d}$ indicates the embedding of node $\boldsymbol{v}_i$ that integrates the first-order neighbor information. According to node indices calculated by the first MHSS module, we can obtain the positive, the negative, and the vth-view structure-enhanced graph embeddings of the selected hard samples

$$\dot{\mathbf{Z}}_{\text{pos}} = \mathbf{Z}_{\text{pos,ind}}, \quad \dot{\mathbf{Z}}_{\text{neg}} = \mathbf{Z}_{\text{neg,ind}}, \quad \dot{\mathbf{H}}^v = \mathbf{H}^v_{\text{ind}}. \tag{11}$$

In accordance with the contrastive learning paradigm, we aim to promote agreement between the positive sample embedding $\dot{\mathbf{z}}_{\text{pos},j}$ of node $\boldsymbol{v}_j$ and the vth-view structure-enhanced embedding $\dot{\mathbf{h}}_j^v$, while simultaneously encouraging disagreement between those of a negative sample pair $p_j^- = (\dot{\mathbf{z}}_{\text{neg},j}, \dot{\mathbf{h}}_j^v)$. By doing this, the optimization target of the second discriminator can be formulated as follows:

$$\mathcal{L}_{\text{Subgraph}} = -\frac{1}{2\text{Nr}}\sum_{v=1}^{V}\left(\sum_{j=1}^{\text{Nr}}\log \mathcal{D}\left(\dot{\mathbf{z}}_{\text{pos},j}, \dot{\mathbf{h}}_j^v\right) + \sum_{j=1}^{\text{Nr}}\log\left(1-\mathcal{D}\left(\dot{\mathbf{z}}_{\text{neg},j}, \dot{\mathbf{h}}_j^v\right)\right)\right). \tag{12}$$

By minimizing (12), the network encourages these selected hard samples to maximally express their multiview neighbors in the latent space. By doing this, with the help of underlying localized features within neighbors to be preserved and merged, the updated representations of hard nodes tend to be more distinguishable and informative.

After that, we perform the second MHSS module to further select hard samples according to the computed results of the node-to-subgraph association, as shown in Fig. 1. Specifically, we first adopt the sigmoid function $f_{\text{sig}}(\cdot)$ to normalize the vth-view representations of hard samples $\dot{\mathbf{Z}}^v_{\text{pos}}$ selected by the first MHSS module

$$\dot{\mathbf{M}}^v = f_{\text{sig}}\left(\dot{\mathbf{Z}}^v_{\text{pos}}\right). \tag{13}$$

Similarly, according to node indices ind$'$ calculated by the second MHSS module, we then select and preserve a ratio of hard samples once again

$$\ddot{\mathbf{Z}}_{\text{pos}} = \dot{\mathbf{Z}}_{\text{pos,ind}'}, \quad \ddot{\mathbf{Z}}_{\text{neg}} = \dot{\mathbf{Z}}_{\text{neg,ind}'}, \quad \ddot{\mathbf{M}}^v = \dot{\mathbf{M}}^v_{\text{ind}'}. \tag{14}$$

With these patch representations, we finally transfer them into the last discriminator to calculate the node-to-node association

$$\mathcal{L}_{\text{Node}} = -\frac{1}{2\text{Nr}^2}\sum_{v=1}^{V}\left(\sum_{k=1}^{\text{Nr}^2}\log \mathcal{D}\left(\ddot{\mathbf{z}}_{\text{pos},k}, \ddot{\mathbf{m}}_k^v\right) + \sum_{k=1}^{\text{Nr}^2}\log\left(1-\mathcal{D}\left(\ddot{\mathbf{z}}_{\text{neg},k}, \ddot{\mathbf{m}}_k^v\right)\right)\right). \tag{15}$$

In (15), the network encourages the representations of hard nodes to preserve view-specific information as well as multiview delicate features, further making them discriminative enough to tell the true positive (or negative) samples apart from the false positive (or negative) ones.

D. Optimization Target

The total optimization target of HCHSM consists of three terms

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{Graph}}}_{\text{All samples}} + \underbrace{\lambda\mathcal{L}_{\text{Subgraph}} + \gamma\mathcal{L}_{\text{Node}}}_{\text{Hard samples}} \tag{16}$$

where λ and γ are hyperparameters that balance the importance of three parts. The applied MI technique is similar to the existing contrastive GSP algorithms, such as DGI [30] and MVGRL [23], which learn representations through maximizing the MI between a node and a global summary vector of the graph. However, two major differences exist between this work and our algorithm. First, most advanced contrastive GSP algorithms treat all samples equally during optimization,

Algorithm 1 Training Procedure of HCHSM

Input: Multi-view graphs $\mathcal{G}^1_{\text{pos}} = (\mathbf{X}, \widetilde{\mathbf{A}})$ and $\mathcal{G}^2_{\text{pos}} = (\mathbf{X}_L, \widetilde{\mathbf{A}}_{\text{PPR}})$; multi-view corrupted graphs $\mathcal{G}^1_{\text{neg}} = (\widetilde{\mathbf{X}}, \widetilde{\mathbf{A}})$ and $\mathcal{G}^2_{\text{neg}} = (\widetilde{\mathbf{X}}_L, \widetilde{\mathbf{A}}_{\text{PPR}})$; iteration number $iter\_max$; hyper-parameters r, γ, λ.
Output: Positive consensus graph embedding $\mathbf{Z}_{\text{pos}}$.
1: Initialize graph encoders $f_g^v(\cdot)$ and all discriminators $\mathcal{D}(\cdot)$ with an Xavier initialization;
2: **for** $iter = 1$ to $iter\_max$ **do**
3: Generate multi-view positive (or negative) graph embeddings and integrate them to obtain $\mathbf{Z}_{\text{pos}}$ (or $\mathbf{Z}_{\text{neg}}$);
4: Conduct node-to-graph MI estimation by Eq. (4) and Eq. (8);
5: Obtain node indices of hard samples by Eq. (5) - (7) and Eq. (9);
6: Select hard samples according to node indices;
7: Conduct node-to-subgraph MI estimation within the selected hard samples by Eq. (10) - (12);
8: Repeat step 5 and step 6 based on step 7;
9: Conduct node-to-node MI estimation within the selected hard samples by Eq. (13) - (15);
10: Update the network by minimizing $\mathcal{L}$ in Eq. (16);
11: **end for**
12: **return** $\mathbf{Z}_{\text{pos}}$

while HCHSM focuses more on hard samples that usually limit performance improvement. Second, the existing efforts fall into a single contrasting pattern to optimize the network, while HCHSM conducts a hierarchically contrastive scheme over hard samples to boost their discriminative capability for performance improvement. The whole training procedure of HCHSM is illustrated in Algorithm 1.

Compared with the existing GSP algorithms, the merits of the proposed HCHSM could be summarized as the following factors. First, more naturally handles GRL in an unsupervised scenario. Our optimization target $\mathcal{L}$ accomplishes the contrastive GSP by exploring the multilevel graph information from the data itself without any label guidance. Second, more discriminative that unifies the processes of hard sample mining and representation learning to keep filtering easy nodes and have the network to focus more on hard nodes. Third, more comprehensive that reveals the intrinsic information of hard samples by performing a hierarchically contrastive scheme.

IV. Experiments

To evaluate the superiority and effectiveness of HCHSM against state-of-the-art algorithms, we conduct extensive experiments to answer the following six research questions.

1) *Q1:* How does the performance of HCHSM compare with other baseline algorithms in node classification and node clustering tasks? (See Section IV-B.)
2) *Q2:* How does the proposed hard sample selection strategy influence the performance of HCHSM? (See Section IV-C.)
3) *Q3:* How does the proposed hierarchically contrastive scheme influence the performance of HCHSM? (See Section IV-D.)
4) *Q4:* How about the model sensitivity to different hyper-parameter settings? (See Section IV-E.)
5) *Q5:* How about the running time of HCHSM compared with other competitors? (See Section IV-F.)
6) *Q6:* How does the proposed HCHSM influence the learned latent space? (See Section IV-G.)

TABLE I
Dataset Summary

Dataset	Samples	Edges	Dimension	Classes
Cora	2,708	10,556	1,433	7
Citeseer	3,327	9,228	3,703	6
Pubmed	19,717	88,651	500	3
Amac	13,752	574,418	767	10
Amap	7,650	287,326	745	8
Corafull	19,793	130,622	8,710	70
Ogbn-arxiv	169,343	1,166,243	128	40

In the following, we briefly introduce the experimental setup and then provide detailed experiment results with corresponding analyses.

A. Experimental Setup

1) Benchmark Datasets: The proposed HCHSM is evaluated on seven benchmark datasets. Especially, five datasets come from the citation graphs, and others come from the co-purchase graphs. Table I summarizes a brief statistical overview of these datasets.

1) Cora,[1] Citeseer,[1] Pubmed,[1] Corafull,[2] and Ogbn-arxiv[3] are five popular citation graph datasets. Especially, nodes refer to scientific publications, while edges represent the citation relationships between them. Each node possesses a predefined feature with corresponding dimensions.
2) Amazon Photo[4] (Amap) and Amazon Computers[4] (Amac) are subdivisions within the Amazon co-purchase graph. In this graph, the nodes denote goods, while the edges denote the frequent co-purchases of two goods. The node features are represented by product reviews encoded in the form of bag of words, and the product category determines the class labels.

In the pretext task, we feed all graph samples into the network for unsupervised pretraining. In downstream tasks, the following hold. First, *semisupervised node classification*—we follow the same train/validation/test sample split as [38] on Cora, CiteSeer, and PubMed. Taking the Cora that contains 2708 nodes with seven classes for example, we allow for only 20 nodes per class (140 nodes in total) to be used for the training phase of the downstream task. Thereafter, the predictive power of the learned representations is evaluated on 1000 test nodes. For Amazon Photo, Amazon Computers, Corafull, and Ogbn-arxiv, since these datasets have no available split, we use the random split where 7%, 7%, and 86% samples are randomly sampled to be the train, validation, and test set, respectively. Second, *node clustering*—all learned representations of samples are fed into a classical clustering algorithm, e.g., K-means [62].

2) Training Procedure: HCHSM is implemented using the PyTorch platform and NVIDIA 3090 GPU. The whole training phase of HCHSM consists of two steps. First, we train a contrastive GSP framework to learn the graph embedding for at least 200 iterations by minimizing (16). Second, we evaluate

[1]https://docs.dgl.ai/api/python/dgl.data.html#citation-network-dataset
[2]https://docs.dgl.ai/api/python/dgl.data.html#corafull-dataset
[3]https://ogb.stanford.edu/docs/nodeprop/
[4]https://docs.dgl.ai/api/python/dgl.data.html#amazon-co-purchase-dataset

the quality of the learned graph embedding through downstream tasks, such as node classification and node clustering. Specifically, the following hold: 1) for node classification, we adopt the learned graph embedding as input and train a simple semisupervised classifier for at least 50 iterations until convergence and 2) for node clustering, we directly perform the K-means algorithm over the learned graph embedding. To mitigate the potential impact of randomness, we conduct each experiment ten times for all compared algorithms. The reported results include the average values along with standard deviations.

3) Implementation Details: For all compared algorithms, we implement their public source codes by following the algorithm settings in the corresponding literature and report the reproduced performance. For the proposed algorithm, we adopt two single-layer GCNs with 512 hidden dimensions as graph encoders. To avoid overfitting, we adopt an early stop strategy that the optimization stops when the validation loss reaches a plateau. To make all datasets fit into the GPU memory, we perform a subsampling operation introduced in MVGRL [23] and train the network with Adam optimizer. The learning rates of the proposed HCHSM and the logistic regression classifier are set to $1e-3$ and $1e-2$ for all datasets, respectively. According to the results of parameter sensitivity testing, we fix two balanced hyperparameters γ and λ to 1. We dynamically adjust r by calculating $1-(\text{iter})/(\text{iter\_max})$. iter and iter_max denote the current iteration and the maximum iteration, respectively.

4) Evaluation Metrics: To ensure fairness and comparability, we strictly adhere to the node classification metric used in previous works [21], [23], [30]. Specifically, we evaluate the node classification performance of all compared algorithms using the accuracy (ACC) metric. To further verify the effectiveness and generalization of HCHSM, we conduct the node clustering experiments, and four prevalent clustering metrics are adopted, including clustering accuracy (C-ACC), normalized MI (NMI), adjusted rand index (ARI), and $F1$ score ($F1$) [63], [64], [65].

B. Performance Comparison (Q1)

In this section, we empirically compare HCHSM with the following algorithms from two aspects. Specifically, GCN [38], graph attention network (GAT) [39], and SS-GCNs [66] are from the *supervised learning* aspect, while Deepwalk [36], Node2Vec [37], variational graph auto-encoder (VGAE) [42], DGI [30], graphical mutual information (GMI) [21], AGE [61], MVGRL [23], multi-scale contrastive siamese network (MERIT) [27], GCA [26], and ProGCL [59] are from the *unsupervised learning* aspect.

1) Node Classification: Table II presents an accuracy performance comparison of HCHSM and the aforementioned algorithms. As seen from this table, the proposed algorithm consistently exceeds all baselines by 1.0%–10.9% on average on six datasets. This promising achievement benefits from the novel idea of guiding GSP with an HCHSM mechanism. Specifically, we have the following observations.

1) we first compare HCHSM with five supervised GRL algorithms, including GCN, GAT, SS-GCN-Clu, SS-GCN-Par, and SS-GCN-Comp. Without label information for training, our algorithm exhibits comparable and even better performance than these supervised algorithms. This is because of the following.
   a) HCHSM explores multiview rich information from data itself to learn the graph embedding instead of relying on extremely sparse supervised signals (i.e., label information).
   b) HCHSM focuses more on hard samples and enforces them to inherit multilevel semantic properties from multiview intrinsic information of graphs.
2) HCHSM shows superior performance against the random-walk-based algorithms by a large margin. Specifically, Deepwalk and Node2Vec exploit the graph structure information based on the depth-first sampling (DFS) strategy. However, the sampling process only focuses on a limited number of nodes close to the source node, which may result in neglecting important local structure information. Besides, these algorithms seldom consider attribute information, leading to less discriminative representations and unsatisfied performance. In contrast, HCHSM effectively utilizes both the graph structures and node attributes, which are incorporated through a sufficient negotiation process, thereby enhancing the quality of the learned graph embedding.
3) VGAE and AGE are not comparable to the proposed algorithm, since these algorithms overemphasize the quality of reconstructed information and suffer from unstructured predictions. Instead of reconstructing all sample features, HCHSM learns the graph embedding by estimating the multilevel MI agreement between patch representations, where the sample pair with similar semantic information is encouraged to be close, while the sample pair with unrelated semantic information is pushed away.
4) Compared with contrastive GSP algorithms, such as DGI, GMI, MVGRL, MERIT, GCA, and ProGCL, the proposed HCHSM consistently outperforms them by achieving the best results on all datasets. For example, HCHSM gains at least 1.8%, 1.7%, 3.0%, 1.1%, 0.9%, and 5.6% accuracy increment over DGI on six datasets. Moreover, HCHSM exceeds MVGRL by 0.6%, 1.2%, 2.0%, 0.7%, 0.8%, and 2.3% in terms of accuracy on six datasets. The observations of GMI, GCA, MERIT, and ProGCL are similar. The comparison results have solidly demonstrated the effectiveness of HCHSM in handling the contrastive GSP task. These benefits can be attributed to the following merits.
   a) Different from all compared contrastive GSP algorithms, we elaborately design an MHSS module to enable the network to focus more on hard sample pairs rather than treat all sample pairs equally during optimization.
   b) HCHSM hierarchically reveals the intrinsic information of hard samples from different views, which can make the learned graph embedding more discriminative and accurate.

TABLE II

NODE CLASSIFICATION PERFORMANCE COMPARISON ON SIX DATASETS (MEAN ± STD). THE THIRD COLUMN REPRESENTS THE ALGORITHM SETTINGS, WHERE **X**, **A**, **D**, AND **Y** DENOTE THE ATTRIBUTE INFORMATION, STRUCTURE INFORMATION, DATA AUGMENTATION, AND GROUND-TRUTH LABELS, RESPECTIVELY. ESPECIALLY, THE ALGORITHMS WITH **Y** INDICATE SUPERVISED ALGORITHMS. $\mathbf{s}_{\text{pos}}$ (OR $\mathbf{s}_{\text{neg}}$) INDICATES THE FASHION OF NODE-SELECTION CRITERION. AVG. ↓ REFERS TO THE AVERAGE PERFORMANCE DEGRADATION OVER ALL DATASETS COMPARED WITH HCHSM. "-" REFERS TO OUT-OF-MEMORY FAILURE. BOLDFACE AND UNDERLINE MEAN THE BEST AND THE RUNNER-UP RESULTS, RESPECTIVELY

Type	Algorithm	Setting				Dataset						Avg. ↓
		X	A	D	Y	Cora	Citeseer	Pubmed	Amap	Amac	Corafull	
Supervised	GCN (ICLR' 17)	✓	✓		✓	80.4±0.0	70.4±0.0	78.8±0.0	92.2±0.0	87.7±0.0	54.9 ±0.0	1.7 ↓
	GAT (ICLR' 18)	✓	✓		✓	82.7±0.4	72.3±0.8	79.1±0.5	92.6±0.3	86.9±0.4	-	1.0 ↓
	SS-GCN-Clu (ICML' 20)	✓	✓		✓	81.3±0.0	70.6±0.0	76.9±0.0	90.5±0.0	84.7±0.1	52.3±0.0	3.0 ↓
	SS-GCN-Par (ICML' 20)	✓	✓		✓	79.8±0.0	71.3±0.0	80.0±0.0	90.5±0.0	85.2±0.0	51.4±0.0	1.9 ↓
	SS-GCN-Com (ICML' 20)	✓	✓		✓	80.7±0.0	71.3±0.0	78.7±0.0	91.4±0.0	86.8±0.0	55.3±0.0	1.7 ↓
Unsupervised	Deepwalk (KDD' 14)		✓			66.8±1.1	47.1±0.8	65.5±0.8	91.4±0.5	87.0±0.2	51.5±0.1	10.9 ↓
	Node2Vec (KDD' 16)		✓			68.8±1.1	48.1±1.4	70.8±1.2	90.3±0.2	86.2±0.2	51.3±0.2	9.8 ↓
	VGAE (NeurIPS' 16)	✓	✓			71.5±1.7	62.0±1.6	73.4±1.5	89.6±1.0	80.6±1.1	50.4±1.0	7.8 ↓
	DGI (ICLR' 19)	✓	✓			82.2±0.6	71.8±0.7	77.4±0.8	91.5±0.2	86.9±0.5	50.4±2.4	2.4 ↓
	GMI (WWW' 20)	✓	✓			82.4±0.8	72.9±0.3	79.7±0.5	87.2±0.0	82.2±0.7	-	2.8 ↓
	AGE (KDD' 20)	✓	✓	✓		72.8±0.6	69.5±0.7	66.7±0.8	88.9±0.9	82.6±1.2	51.5±0.7	7.1 ↓
	MVGRL (ICML' 20)	✓	✓	✓		83.4±0.5	72.3±0.5	78.4±0.6	91.9±0.2	87.0±0.2	53.7±1.2	1.3 ↓
	MERIT (IJCAI' 21)	✓	✓	✓		83.3±0.3	72.2±0.1	79.0±0.1	87.4±0.2	81.3±0.3	-	3.1 ↓
	GCA (WWW' 21)	✓	✓	✓		81.8±0.2	71.9±0.4	-	91.8±0.3	87.7±0.2	-	1.2 ↓
	ProGCL (ICML' 22)	✓	✓	✓		82.8±0.2	70.1±0.1	-	87.9±0.1	87.6±0.2	-	2.4 ↓
	HCHSM ($\mathbf{s}_{pos}$)	✓	✓	✓		83.5±0.7	73.0±0.7	79.2±0.9	92.3±0.2	87.5±0.3	55.7±0.8	0.5 ↓
	HCHSM ($\mathbf{s}_{neg}$)	✓	✓	✓		83.7±0.9	73.2±0.7	79.3±1.1	92.4±0.2	87.3±0.3	55.7±1.0	0.6 ↓
	HCHSM	✓	✓	✓		**84.0±0.3**	**73.5±0.4**	**80.4±0.5**	**92.6±0.2**	**87.8±0.1**	**56.0±0.5**	-

TABLE III

NODE CLUSTERING PERFORMANCE COMPARISON ON FOUR DATASETS (MEAN ± STD). HERE WE ADOPT FOUR WIDELY USED CLUSTERING METRICS, I.E., C-ACC, NMI, ARI, AND $F1$, FOR ALGORITHM EVALUATION. AVG. ↓ REFERS TO THE AVERAGE PERFORMANCE DEGRADATION OVER ALL DATASETS COMPARED WITH HCHSM. BOLDFACE AND UNDERLINE MEAN THE BEST AND THE RUNNER-UP RESULTS, RESPECTIVELY

Dataset	Metric	Algorithm						
		DGI	GMI	MVGRL	GCA	MERIT	ProGCL	HCHSM
Cora	C-ACC	70.8±0.8	69.5±0.5	74.0±0.3	63.6±0.5	63.6±0.7	67.1±0.4	**75.6±0.5**
	NMI	55.7±0.2	54.7±0.6	59.2±0.7	56.8±0.5	48.8±0.8	51.0±0.3	**60.9±0.4**
	ARI	49.8±0.4	46.8±0.3	52.9±0.6	40.3±0.2	41.7±0.7	40.7±0.6	**54.4±0.3**
	F1	67.8±0.8	68.7±0.9	70.1±0.6	55.7±0.4	56.1±0.4	55.6±0.6	**71.1±0.5**
Citeseer	C-ACC	68.0±0.6	66.8±0.5	68.7±0.6	60.4±0.4	68.5±0.8	65.9±0.7	**69.8±0.6**
	NMI	43.7±0.6	41.8±0.4	44.7±0.4	36.1±0.7	41.8±1.0	39.5±0.8	**45.8±0.8**
	ARI	44.0±0.7	42.5±0.6	45.7±0.7	35.2±0.6	41.9±0.5	36.1±0.4	**46.9±0.5**
	F1	63.7±0.7	62.8±0.6	63.6±0.7	56.4±1.2	53.8±0.8	57.8±1.0	**65.2±0.8**
Amap	C-ACC	52.4±0.8	54.5±1.0	54.7±0.7	55.0±1.3	52.8±0.9	51.5±0.8	**56.3±0.8**
	NMI	48.3±0.6	50.8±0.7	51.0±0.6	48.3±0.8	47.9±0.9	39.5±1.0	**53.4±0.7**
	ARI	27.3±0.5	32.9±0.7	32.5±0.8	26.8±0.7	33.4±0.6	33.6±0.4	**34.3±0.7**
	F1	51.5±0.8	52.3±0.8	51.7±1.3	52.5±1.2	49.5±1.1	48.9±0.9	**52.9±1.0**
Amac	C-ACC	47.2±0.8	39.5±0.8	48.1±0.8	44.0±0.8	46.7±0.8	49.6±0.8	**50.3±0.8**
	NMI	46.7±0.7	25.5±0.9	46.1±0.7	35.4±1.1	46.7±1.3	47.0±1.0	**48.2±0.8**
	ARI	27.5±0.5	19.5±0.5	28.9±0.8	27.9±0.6	28.1±1.2	29.9±1.0	**32.3±0.7**
	F1	40.1±0.8	27.4±0.6	41.1±0.6	34.9±0.4	34.4±0.8	41.2±1.2	**43.5±0.6**
Avg. ↓	C-ACC	3.4 ↓	5.4 ↓	1.6 ↓	7.2 ↓	5.1 ↓	4.5 ↓	-
	NMI	3.5 ↓	8.9 ↓	1.8 ↓	7.9 ↓	5.8 ↓	7.8 ↓	-
	ARI	4.8 ↓	6.6 ↓	2.0 ↓	9.4 ↓	5.7 ↓	6.9 ↓	-
	F1	2.4 ↓	5.4 ↓	1.6 ↓	8.3 ↓	9.7 ↓	7.3 ↓	-

5) It is worth noting that performance differences exist among the aforementioned datasets, mainly caused by their different statistical properties. Taking the citation graph datasets for example, the overall performance of all compared algorithms on Cora is much better than that on Corafull. This is because Corafull exhibits a tenfold increase in the number of classes compared with Cora, presenting a significant challenge in enabling the network to learn more discriminative features in an unsupervised scenario.

2) Node Clustering: In the node clustering task, as reported in Table III, we compare the proposed HCHSM with six contrastive GSP algorithms to further illustrate its superiority. From these results, we can see that HCHSM achieves better average performance than all compared algorithms in terms of C-ACC, NMI, ARI, and $F1$ on four datasets. Taking the clustering accuracy results for instance, HSHSM outperforms DGI, GMI, MVGRL, GCA, MERIT, and ProGCL by 3.4%, 5.4%, 1.6%, 7.2%, 5.1%, and 4.5% on average across all datasets. The observations on other metrics are similar. The reasons for performance increment can be summarized below. First, these results illustrate that performing HCHSM is indeed helpful to the clustering task. Second, addressing the hard sample mining issues from the MI estimation perspective can facilitate capturing the most informative graph patterns in the

TABLE IV

ABLATION STUDY FOR THE HIERARCHICALLY CONTRASTIVE SCHEME (MEAN ± STD). $\mathcal{L}_{\text{Graph}}$, $\mathcal{L}_{\text{Subgraph}}$, AND $\mathcal{L}_{\text{Node}}$ CORRESPOND TO MI ESTIMATIONS AMONG A NODE-TO-GRAPH PAIR, A NODE-TO-SUBGRAPH PAIR, AND A NODE-TO-NODE PAIR, RESPECTIVELY. BOLDFACE MEANS THE BEST RESULT

Algorithm	Cora	Citeseer	Pubmed	Amap	Amac	Corafull
Algorithm$_1$ ($\mathcal{L}_{Graph}$)	83.3±0.5	72.8±0.8	78.9±0.7	91.2±0.3	87.1±0.1	54.6±0.8
Algorithm$_2$ ($\mathcal{L}_{Graph}$+$\mathcal{L}_{Subgraph}$)	83.8±0.8	73.0±0.6	80.0±0.9	92.3±0.3	87.4±0.3	55.6±0.5
Algorithm$_3$ ($\mathcal{L}_{Graph}$ +$\mathcal{L}_{Subgraph}$+$\mathcal{L}_{Node}$)	**84.0±0.3**	**73.5±0.4**	**80.4±0.5**	**92.6±0.2**	**87.8±0.1**	**56.0±0.5**

latent space, thereby improving the learning capacity of the contrastive GSP paradigm for node clustering. Furthermore, the effectiveness of HCHSM for node clustering suggests that the learned graph embedding associated with the proposed hard sample mining strategy can be applied to not only the node classification task but also other downstream tasks, verifying its good generalization.

All the above observations on node classification and node clustering tasks have proved the effectiveness of the proposed algorithm for GSP.

C. Ablation on the Fashion of Node-Selection Criterion (Q2)

To illustrate the effect of different fashions of node-selection criterion in the MHSS module, we investigate the proposed HCHSM and its two variants. The major difference between HCHSM ($\mathbf{s}_{\text{pos}}$) and HCHSM ($\mathbf{s}_{\text{neg}}$) lies in the fashion of the hard sample-selection criterion. Specifically, HCHSM ($\mathbf{s}_{\text{pos}}$) indicates the algorithm that quantifies the sample discriminative capability according to the positive MI agreement score vector $\mathbf{s}_{\text{pos}}$ and selects hard samples by calculating $\mathcal{T}(\mathbf{s}_{\text{pos}}, K)$, while the hard sample-selection criterion of HCHSM ($\mathbf{s}_{\text{neg}}$) depends on $\mathbf{s}_{\text{neg}}$. From the results in Table II, some observations can be summarized. First, compared with the existing algorithms that treat all samples equally during optimization, our three proposed algorithms have achieved competitive or even better accuracy performance on all datasets. These results have consistently demonstrated the effectiveness of MI-based hard sample mining strategies. Second, the proposed node-selection criterion of HCHSM contributes more to performance improvement than other solutions. Taking the results on Pubmed for example, HCHSM can boost HCHSM ($\mathbf{s}_{\text{pos}}$) and HCHSM ($\mathbf{s}_{\text{neg}}$) with 1.2% and 1.1% accuracy enhancement, respectively. As observed, our final solution for hard sample selection considers both MI agreements of positive and negative sample pairs, making the quantification of sample discrimination more comprehensive and accurate.

D. Ablation on the Hierarchically Contrastive Scheme (Q3)

In this section, we conduct a performance comparison among three algorithms to demonstrate the effectiveness of the hierarchically contrastive scheme. The results are presented in Table IV. Algorithm$_1$ means that the algorithm merely explores the node-to-graph association without considering hard sample mining. Algorithm$_2$ and Algorithm$_3$ mean that the algorithms implement hard sample selection and conduct multilevel contrastive granularities over the selected hard samples. From this table, we can see that the following hold. First, taking the results on Pubmed and Corafull for example, by adding the MHSS and the node-to-subgraph MI estimation, the accuracy of Algorithm$_2$ has been improved over Algorithm$_1$ by 1.1% and 1.0%, respectively. This illustrates that merely adopting the node-to-graph contrast granularity is not sufficient, and the proposed hard sample mining strategy is indeed conducive to improving the quality of the learned graph embedding for better performance. Second, by adding $\mathcal{L}_{\text{Node}}$, Algorithm$_3$ gains 0.7%–1.5% and 0.2%–0.5% accuracy increment over Algorithm$_1$ and Algorithm$_2$ on six datasets, respectively. These results show that the hierarchically contrastive scheme can make the latent representations more informative and accurate. Moreover, this experiment implies the significance of boosting the discriminative capability of hard samples for performance improvement.

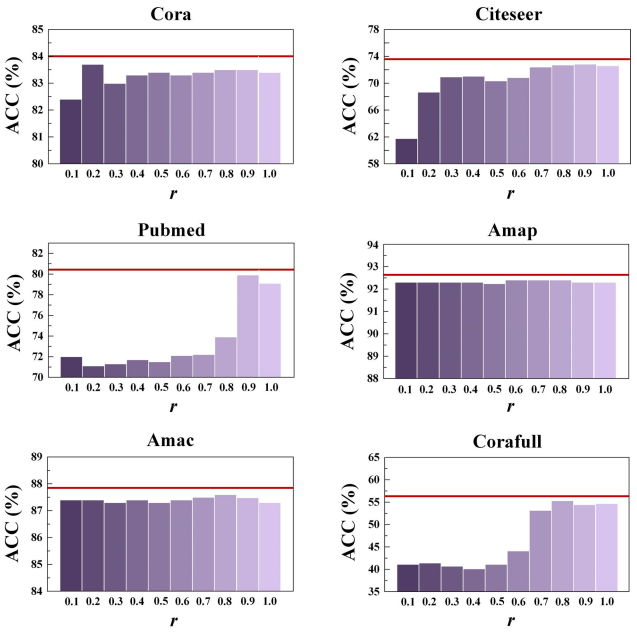

Fig. 4. Hyperparameter analysis on six datasets. ACC performance variation of HCHSM is presented when r varies from 0.1 to 1.0 with 0.1 step size. The X-axis and Y-axis refer to the hard sample selection ratio r and the node classification performance, respectively. In this figure, the result of the proposed automatic r setting mechanism is reported as the red line in subfigures to illustrate its effectiveness.

E. Hyperparameter Analysis (Q4)

1) Analysis of Hyperparameter r: In (12), the hyperparameter r refers to the hard sample selection ratio, which determines the number of hard samples selected for further exploration in the next round. The larger r value means that more samples will be selected as hard samples. To show its influence in depth, we conduct experiments to investigate the effectiveness

of setting the hyperparameter r and report the results (i.e., the red line in each subfigure) of our proposed automatic r tuning mechanism on all datasets. Fig. 4 illustrates the ACC performance of HCHSM when we vary r from 0.1 to 1.0 with a step size of 0.1. As can be seen in Fig. 4, some major observations can be obtained. First, a small ratio of hard samples has the risk of not covering the large variety of the whole samples, leading to relatively poor performance. This phenomenon is quite obvious on Pubmed and Corafull. Second, a too-large ratio of hard samples (i.e., $r = 1.0$) may also suffer from the risk of considering too many samples and pays not enough attention to the real hard samples. Third, the best r is various across different datasets. The performance of the algorithm is relatively stable when r is set in the scope from 0.9 to 1.0. Fourth, searching r within an inappropriate range poses a great impact on model performance in some cases (e.g., acc $= 73.6$ for $r = 0.8$ and acc $= 79.7$ for $r = 0.9$ on Pubmed). This is because dropping the r value may make a certain proportion of real hard samples unselected and jumbled together with well-categorized samples, which increases the risk of representation degeneration [67] and disturbs the model optimization, resulting in worse results. Fifth, according to the observations, the model with our automatic parameter-tuning strategy performs better across all datasets. This observation aligns with our intuition that dynamically changing r with iterations will provide more diverse samples for hard sample mining according to the model learning capability, because the split of all samples becomes more diverse, as the model pretraining process incorporates a wider range of different r values. Besides, HCHSM can be trained with fewer resources without careful manual hyperparameter tuning.

2) Analysis of Hyperparameter λ and γ: We also investigate the effect of hyperparameters λ and γ, which adjust the trade-off among different contrastive objectives. Fig. 5 illustrates the node classification performance variation of HCHSM on six datasets when λ and γ range from 0.01 to 100. From these figures, we can observe that the following hold. First, λ and γ are effective in improving the quality of the learned graph embedding, validating the effectiveness of exploiting an HCHSM scheme. Second, taking the results on Cora for example, increasing the values of λ and γ first improves the performance, and continually increasing them to a higher value obtains relatively stable performance. This indicates that HCHSM needs proper trade-off coefficients to learn discriminative graph features for hard samples. Third, the proposed HCHSM tends to perform well by setting λ and γ to 1 according to the results of all datasets. Therefore, these observations validate our assumption that a trade-off exists among three types of contrastive granularities.

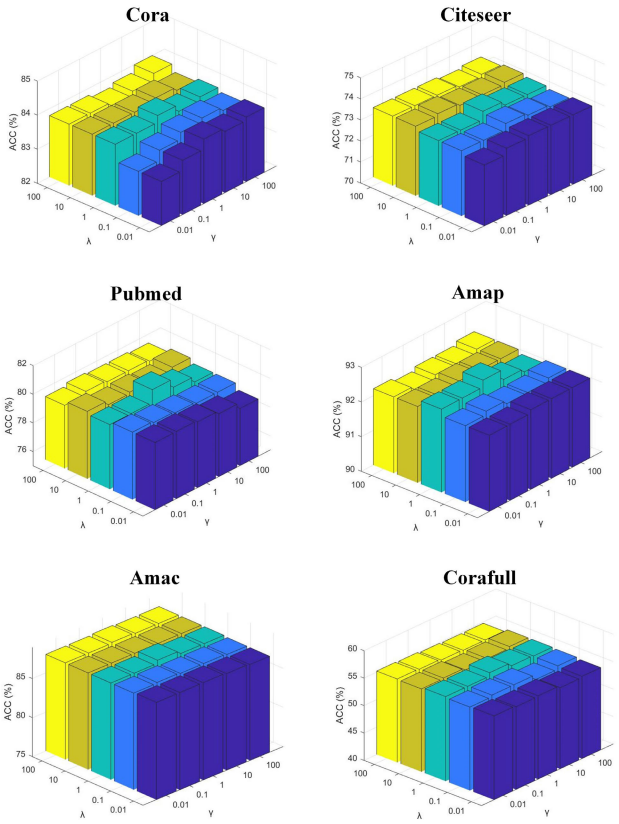

Fig. 5. Sensitivity of HCHSM with the variation of hyperparameter λ and hyperparameter γ on Cora, Citeseer, Pubmed, Amap, Amac, and Corafull.

F. Running Time Comparison (Q5)

To evaluate the computational efficiency of the proposed algorithm, Table V presents the running time of seven contrastive GSP algorithms on three large-scale datasets (i.e., Pubmed, Corafull, and Ogbn-arxiv). Moreover, we report the classification accuracy of all compared algorithms for a clear overall performance illustration. Note that all algorithms are evaluated on the same device with one NVIDIA-3090 GPU card. Here, the running time refers to the average time for the pretraining phase of each epoch. To make the data preprocessing fit into the CPU memory, both MVGRL and HCHSM perform the edge dropping to generate the augmented graph on Ogbn-arxiv. Note that we follow the subsampling strategy of MVGRL to pretrain the model. From the results reported in Table V, we can see that the following hold. First, although HCHSM has a slightly longer running time on Pubmed and Corafull when compared with DGI and MVGRL, HCHSM significantly exceeds these algorithms by 3.0%/2.0% and 5.6%/2.3% accuracy increments, respectively. These experimental results have demonstrated that HCHSM can achieve promising performance without introducing much computation cost. Second, HCHSM consistently outperforms GMI, AGE, MERIT, GCA, and ProGCL algorithms in terms of classification accuracy with much less running time. Third, HCHSM outperforms MVGRL by 2.0% ACC on Ogbn-arxiv, once again demonstrating that our method can efficiently handle larger-scale graph data as well as obtain better performance. Fourth, most of the compared algorithms suffer from out-of-memory failure on Ogbn-arxiv. This indicates that contrastive GSP algorithms require much more time cost and memory cost to process overlarge graph datasets due to the resource-consuming nature of contrastive learning techniques.

G. T-SNE of the Graph Embedding (Q6)

To intuitively verify the superiority of HCHSM, we compare the visual performance among several algorithms, including

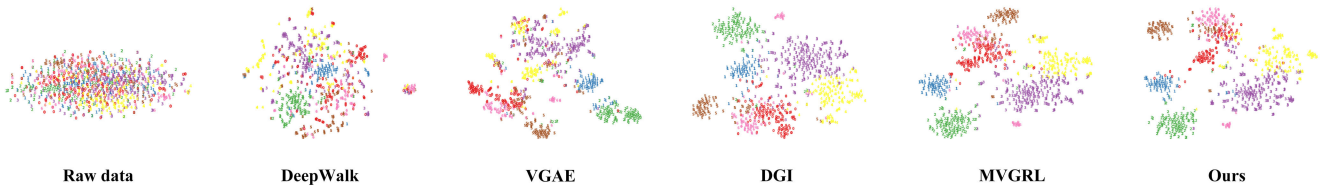

Fig. 6. T-SNE visualization of raw data and five unsupervised GSP algorithms on Cora. The visual results show that the proposed HCHSM presents cleaner partitions among categories than other competitors.

TABLE V

ACCURACY AND RUNNING TIME COMPARISON BETWEEN BASELINES AND HCHSM ON THREE LARGE-SCALE BENCHMARK DATASETS (IN SECONDS). "-" REFERS TO OUT-OF-MEMORY FAILURE THAT MAINLY HAPPENED IN MODEL OPTIMIZATION OR EMBEDDING EXTRACTION

Algorithm	Pubmed		Corafull		Ogbn-arxiv	
	Time	Accuracy	Time	Accuracy	Time	Accuracy
DGI (ICLR' 19)	0.0956	77.4±0.8	0.2123	50.4±2.4	-	-
GMI (WWW' 20)	0.1698	79.7±0.5	-	-	-	-
AGE (KDD' 20)	1.3985	66.7±0.8	2.4137	51.5±0.7	-	-
MVGRL (ICML' 20)	0.0784	78.4±0.6	0.2363	53.7±1.2	0.0512	60.1±1.1
MERIT (IJCAI' 21)	16.6227	79.0±0.1	-	-	-	-
GCA (WWW' 21)	-	-	-	-	-	-
ProGCL (ICML' 22)	-	-	-	-	-	-
HCHSM	0.1249	80.4±0.5	0.2830	56.0±0.5	0.0924	62.1±1.1

DeepWalk, VGAE, DGI, MVGRL, and the proposed HCHSM. We employ the t-distributed stochastic neighbor embedding (T-SNE) algorithm [68] to visualize the distribution of the learned graph embedding in 2-D latent space on Cora, where samples with different colors indicate different categories predicted by algorithms. As illustrated in Fig. 6, we clearly observe that the following hold. First, DeepWalk, VGAE, and DGI do not perform well on Cora; more mixed samples of different categories exist in the latent space, and the boundary of each category is blurry. This is because the baseline algorithms treat all samples equally during optimization and fail to accurately discriminate the false and true positive (or negative) samples, leading to poorer visual performance. Second, the proposed HCHSM presents clearer partitions and denser cluster structures than DeepWalk, VGAE, and DGI, demonstrating that the graph embedding learned by HCHSM is more compact and discriminative. Third, MVGRL has great potential to reveal hidden patterns within the graph data and achieves the most competitive T-SNE visual performance compared with ours. This is because the consensus representations sufficiently negotiated by different information sources could be more discriminative. Hence, the samples belonging to different categories tend to be more easily partitioned in multiview learning [63], [69].

In summary, under the guidance of the HCHSM scheme, the proposed HCHSM focuses more on the samples that are hard to tell rather than treating all of them equally during optimization. Hence, it can accurately distinguish samples belonging to different categories. However, HCHSM makes minor visual modifications to the embedded representations compared with MVGRL. This implies that the study of hard sample mining for GSP remains an open problem, and we believe that accurately selecting hard samples plays a crucial role in successfully implementing the model. In future work, we plan to investigate leveraging other advanced techniques, e.g., reinforcement learning, to better help the model interact in a closed loop with its environment for hard sample selection and exploration, and, thereby, further eliminate the mixed nodes in the latent space.

V. CONCLUSION AND FUTURE WORK

This article proposes a novel framework termed HCHSM for contrastive GSP. In the proposed algorithm, by leveraging the techniques of MI estimation, we design a new MI-based sample-selection criterion within a hard sample selection module to enable the network to focus more on hard nodes and pick them up for further exploration. Meanwhile, we introduce a hierarchically contrastive scheme, which can exploit and unify multilevel semantic information to improve the discriminative capability of hard samples. It shows that both components seamlessly integrated into a unified framework can improve the quality of the graph embedding for downstream tasks. Experiments on seven benchmark datasets also show that HCHSM consistently outperforms state-of-the-art baseline algorithms on node classification and node clustering tasks.

The investigation of hard sample selection techniques for GSP remains an open issue. The hard sample-selection criterion of the proposed MHSS module can be successfully applied to graphs without annotations, but a large number of negative samples are required for MI estimation, leading to impractical memory costs in real-world applications. Future work may extend the hard sample selection procedure to a negative sample-free version. For instance, it is worth studying how to leverage the reinforcement learning technique to help the model interact in a closed loop with its environment for hard sample selection. Also, how to enable HCHSM to process and analyze incomplete unlabelled graphs is another interesting direction.

REFERENCES

[1] J. Liu, F. Xia, X. Feng, J. Ren, and H. Liu, "Deep graph learning for anomalous citation detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2543–2557, Jun. 2022.

[2] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 240–252, Jan. 2022.

[3] R. Zhang, Y. Zhang, and X. Li, "Unsupervised feature selection via adaptive graph learning and constraint," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1355–1362, Mar. 2022.

[4] K. Liang et al., "Knowledge graph contrastive learning based on relation-symmetrical structure," *IEEE Trans. Knowl. Data Eng.*, early access, Jun. 12, 2023, doi: 10.1109/TKDE.2023.3282989.

[5] K. Liang et al., "Learn from relational correlations and periodic events for temporal knowledge graph reasoning," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2023, pp. 1559–1568.

[6] J. Park, M. Lee, H. J. Chang, K. Lee, and J. Y. Choi, "Symmetric graph convolutional autoencoder for unsupervised graph representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6518–6527.

[7] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang, "Attributed graph clustering: A deep attentional embedding approach," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3670–3676.

[8] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, "GPT-GNN: Generative pre-training of graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1857–1867.

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

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

[11] W. Tu et al., "Deep fusion clustering network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9978–9987.

[12] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.

[13] W. Tu et al., "Initializing then refining: A simple graph attribute imputation network," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3494–3500.

[14] Y. Liu et al., "Deep graph clustering via dual correlation reduction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 7, 2022, pp. 7603–7611.

[15] H. Wang et al., "Learning graph representation with generative adversarial nets," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3090–3103, Aug. 2021.

[16] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Hierarchical representation learning in graph neural networks with node decimation pooling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 5, pp. 2195–2207, May 2022.

[17] J. Chen et al., "Adversarial caching training: Unsupervised inductive network representation learning on large-scale graphs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7079–7090, Dec. 2022.

[18] H. Huang, Y. Song, Y. Wu, J. Shi, X. Xie, and H. Jin, "Multitask representation learning with multiview graph convolutional networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 983–995, Mar. 2022.

[19] J. Wang, Z. Ma, F. Nie, and X. Li, "Fast self-supervised clustering with anchor graph," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4199–4212, Sep. 2022.

[20] Y. Liu et al., "Graph self-supervised learning: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5879–5900, Jun. 2023.

[21] Z. Peng et al., "Graph representation learning via graphical mutual information maximization," in *Proc. Web Conf.*, Apr. 2020, pp. 259–270.

[22] W. Hu et al., "Strategies for pre-training graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[23] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4116–4126.

[24] Y. Wang, Y. Min, X. Chen, and J. Wu, "Multi-view graph contrastive representation learning for drug-drug interaction prediction," in *Proc. Web Conf.*, Apr. 2021, pp. 2921–2933.

[25] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, "Self-supervised learning of graph neural networks: A unified review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2412–2429, Feb. 2023.

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

[27] M. Jin, Y. Zheng, Y.-F. Li, C. Gong, C. Zhou, and S. Pan, "Multi-scale contrastive Siamese networks for self-supervised graph representation learning," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1477–1483.

[28] V. Verma, T. Luong, K. Kawaguchi, H. Pham, and Q. V. Le, "Towards domain-agnostic contrastive learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10530–10541.

[29] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," 2020, *arXiv:2006.04131*.

[30] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[31] Q. Sun et al., "SUGAR: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *Proc. Web Conf.*, Apr. 2021, pp. 2081–2091.

[32] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, "InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[33] Y. Jiao, Y. Xiong, J. Zhang, Y. Zhang, T. Zhang, and Y. Zhu, "Sub-graph contrast for scalable self-supervised graph representation learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 222–231.

[34] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 15509–15519.

[35] Y. Wang, J. Peng, H. Wang, and M. Wang, "Progressive learning with multi-scale attention network for cross-domain vehicle re-identification," *Sci. China Inf. Sci.*, vol. 65, no. 6, pp. 6–65, Jun. 2022.

[36] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[37] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[38] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[39] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[40] Y. Wu, O. E. F. Bourahla, X. Li, F. Wu, Q. Tian, and X. Zhou, "Adaptive graph representation learning for video person re-identification," *IEEE Trans. Image Process.*, vol. 29, pp. 8821–8830, 2020.

[41] G. Te, W. Hu, Y. Liu, H. Shi, and T. Mei, "AGRNet: Adaptive graph representation learning and reasoning for face parsing," *IEEE Trans. Image Process.*, vol. 30, pp. 8236–8250, 2021.

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

[43] Z. Tao, H. Liu, J. Li, Z. Wang, and Y. Fu, "Adversarial graph embedding for ensemble clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 3562–3568.

[44] L. Gong, W. Tu, S. Zhou, L. Zhao, Z. Liu, and X. Liu, "Deep fusion clustering network with reliable structure preservation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 17, 2022, doi: 10.1109/TNNLS.2022.3220914.

[45] S. Wang et al., "Highly-efficient incomplete largescale multiview clustering with consensus bipartite graph," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9766–9775.

[46] D. Hu, K. Liang, S. Zhou, W. Tu, M. Liu, and X. Liu, "ScDFC: A deep fusion clustering method for single-cell RNA-seq data," *Briefings Bioinf.*, vol. 24, no. 4, Jul. 2023, Art. no. bbad216.

[47] Y. Lu, X. Jiang, Y. Fang, and C. Shi, "Learning to pre-train graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4276–4284.

[48] Y. Yang et al., "Self-supervised heterogeneous graph pre-training based on structural clustering," in *Proc. Neural Inf. Process. Syst.*, 2022.

[49] J. Qiu et al., "GCC: Graph contrastive coding for graph neural network pre-training," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1150–1160.

[50] Y. Liu et al., "Simple contrastive graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 27, 2023, doi: 10.1109/TNNLS.2023.3271871.

[51] Y. Liu et al., "Hard sample aware network for contrastive deep graph clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 8914–8922.

[52] X. Jiang, Y. Lu, Y. Fang, and C. Shi, "Contrastive pre-training of GNNs on heterogeneous graphs," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 803–812.

[53] X. Jiang, T. Jia, Y. Fang, C. Shi, Z. Lin, and H. Wang, "Pre-training on large-scale heterogeneous graph," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 756–766.

[54] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," 2017, *arXiv:1703.07737*.

[55] W. Ge, W. Huang, D. Dong, and M. R. Scott, "Deep metric learning with hierarchical triplet loss," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 272–288.
[56] Y. Suh, B. Han, W. Kim, and K. M. Lee, "Stochastic class-based hard example mining for deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7244–7252.
[57] L. Gong, S. Zhou, W. Tu, and X. Liu, "Attributed graph clustering with dual redundancy reduction," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3015–3021.
[58] Y. Zhu, Y. Xu, H. Cui, C. Yang, Q. Liu, and S. Wu, "Structure-aware hard negative mining for heterogeneous graph contrastive learning," 2021, *arXiv:2108.13886*.
[59] J. Xia, L. Wu, G. Wang, J. Chen, and S. Z. Li, "ProGCL: Rethinking hard negative mining in graph contrastive learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 24332–24346.
[60] G. Jeh and J. Widom, "Scaling personalized web search," in *Proc. 12th Int. Conf. World Wide Web (WWW)*, 2003, pp. 271–279.
[61] G. Cui, J. Zhou, C. Yang, and Z. Liu, "Adaptive graph encoder for attributed graph embedding," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 976–985.
[62] J. A. Hartigan and M. A. Wong, "A *k*-means clustering algorithm," *J. Roy. Stat. Soc.*, vol. 28, no. 1, pp. 100–108, Oct. 1979.
[63] X. Liu et al., "Late fusion incomplete multi-view clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2410–2423, Oct. 2019.
[64] X. Liu et al., "Absent multiple kernel learning algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1303–1316, Jun. 2020.
[65] X. Liu et al., "Multiple kernel *k*k-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, May 2020.
[66] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?" in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10871–10880.
[67] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, Feb. 2022, pp. 813–823.
[68] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[69] L. Li et al., "Local sample-weighted multiple kernel clustering with consensus discriminative graph," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 15, 2022, doi: 10.1109/TNNLS.2022.3184970.

Wenxuan Tu is currently pursuing the Ph.D. degree with the School of Computer, National University of Defense Technology (NUDT), Changsha, China.

He has published several papers in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), *Information Science*, AAAI Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), Computer Vision and Pattern Recognition Conference (CVPR), Conference on Neural Information Processing Systems (NeurIPS), International Conference on Machine Learning (ICML), and ACM International Conference on Multimedia (ACM MM). His research interests include clustering analysis, graph machine learning, and image semantic segmentation.

Sihang Zhou received the bachelor's degree in information and computing science from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2012, and the M.S. degree in computer science and the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2014 and 2019, respectively.

He is currently an Associate Professor with the School of Intelligence Science and Technology, NUDT. His current research interests include machine learning, pattern recognition, and medical image analysis.

Xinwang Liu (Senior Member, IEEE) received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2013.

He is currently a Full Professor with the College of Computer, NUDT. He has published more than 100 peer-reviewed papers, including those in highly regarded journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE TRANSACTIONS ON MULTIMEDIA (TMM), the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS), International Conference on Machine Learning (ICML), Conference on Neural Information Processing Systems (NeurIPS), International Conference on Computer Vision (ICCV), Computer Vision and Pattern Recognition Conference (CVPR), AAAI Conference on Artificial Intelligence (AAAI), and International Joint Conference on Artificial Intelligence (IJCAI). His current research interests include kernel learning and unsupervised feature learning.

Dr. Liu serves as an Associated Editor for the *Information Fusion Journal*, the IEEE TRANSACTIONS ON CYBERNETICS (TCYB), and the IEEE TNNLS. More information can be found at https://xinwangliu.github.io.

Chunpeng Ge received the Ph.D. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016.

He was a Research Fellow with the Singapore University of Technology and Design, Singapore, and the University of Wollongong, Wollongong, NSW, Australia. He is currently a Professor with Shandong University, Shandong, China. He has published more than 60 papers in journals and conferences, including the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING (TDSC) and the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS). His current research interests include cybersecurity, especially in the area of artificial intelligence (AI) security information security, and privacy-preserving for blockchain.

Zhiping Cai (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 1996, 2002, and 2005, respectively, all in computer science and technology.

He is currently a Full Professor with the College of Computer, NUDT. His current research interests include artificial intelligence, network security, and big data.

Prof. Cai is a Senior Member of China Computer Federation (CCF).

Yue Liu received the bachelor's degree in computer science from Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2021. He is currently pursuing the master's degree in computer science with the National University of Dense Technology (NUDT), Changsha, China.

He has published several peer-reviewed papers, including International Conference on Learning Representations (ICLR), International Conference on Machine Learning (ICML), AAAI Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), ACM International Conference on Multimedia (ACM MM), International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), and the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). His current research interests include self-supervised learning, knowledge graph, and deep graph clustering. More information can be found at https://yueliu1999.github.io/.