HPoolGCL: Augmentation-Free Cross-granularity Graph Contrastive Learning with Hierarchical Pooling

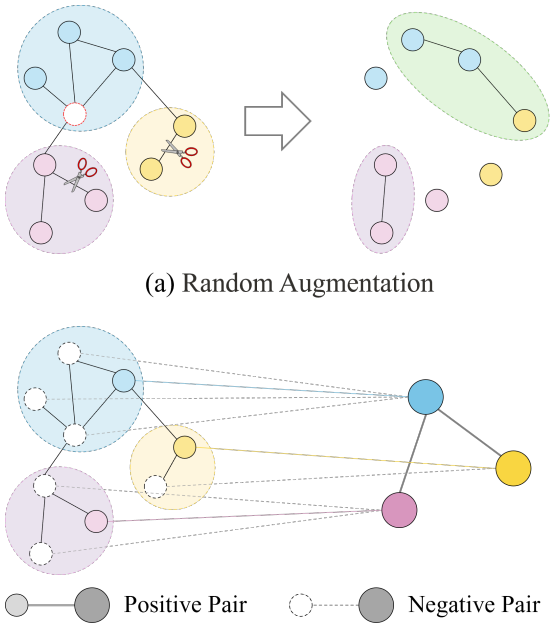
***Abstract*—Graph contrastive learning (GCL) has emerged as a dominant paradigm for self-supervised representation learning for attributed graph data. However, existing GCL methods heavily rely on empirical graph data augmentation, which may distort intrinsic graph semantics and produce poor generalization without carefully chosen or designed augmentation techniques. Furthermore, most GCL approaches focus on same-granularity contrastive learning (e.g., node versus node), neglecting the hierarchical and multi-granular properties inherent in real-world networks, leading to suboptimal performance. To address these limitations, we propose HPoolGCL, a cross-granularity GCL framework compatible with various hierarchical graph pooling methods to capture multi-granularity information. Our framework eliminates the need for handcrafted augmentations, explicit negative sampling, and complex multi-encoder architectures by applying two novel loss functions in hierarchical graph pooling. The theoretical analysis is provided to explain the effectiveness of unified MGC and HiCR losses from three perspectives, namely the information maximization principle, the redundancy reduction principle, and the information bottleneck principle. The experimental results demonstrate that HPoolGCL achieves state-of-the-art performance across multiple downstream tasks on five benchmarks. Our codes are available at https://github.com/Heycen/HPoolGCL.**

***Index Terms*—Graph self-supervised learning, graph contrastive learning, graph pooling, graph neural networks, graph representation learning**

I. Introduction

In recent years, self-supervised learning (SSL) has made significant progress, which enables training deep models on unlabeled data by learning representations from self-generated supervisory signals [1]. Inspired by the remarkable success of SSL in computer vision (CV) [2]–[4] and natural language processing (NLP) [5], [6], researchers have extensively explored SSL techniques for graph-structured data, achieving performance comparable to supervised learning [7]. Graph contrastive learning (GCL), which learns semantically rich representations by pulling positive samples close and pushing negative samples apart, has emerged as a predominant paradigm in graph self-supervised learning. This paradigm has also inspired powerful methods in other domains, such as contrastive clustering [8], [9]. These methods perform joint representation learning and clustering by contrasting at both the instance and cluster levels.

Typically, GCL follows the general framework of visual contrastive learning (VCL) by generating different views through data augmentation. Based on the level of comparison, GCL methods can be categorized into same-granularity [10]–[13] and cross-granularity contrast. However, while same-granularity methods are prevalent, they often struggle to capture high-level structural information. Early cross-granularity attempts [14], [15] were often too simplistic, inadequately accounting for substructures inherently embedded in complex graphs, resulting in suboptimal performance for downstream tasks. Recent research [16] has demonstrated that graph pooling is an effective method for obtaining multi-granularity information from complex graphs, and has highlighted the need to design appropriate constraints to prevent the pooling layer from quickly converging to a trivial solution in unsupervised settings. Despite these advances, most existing methods, whether same-granularity or cross-granularity, rely heavily on graph data augmentation, leaving the potential of pooling underexplored in augmentation-free GCL.



(a) Random Augmentation

(b) Graph Contrastive Pooling

Fig. 1: A comparison between traditional random augmentation and hierarchical graph pooling for creating contrastive views. (a) Random augmentations (e.g., edge dropping or node dropping) can distort the graph's intrinsic community structure and may even create additional isolated communities. (b) In contrast, the purpose of hierarchical graph pooling is to preserve the essential high-level structure, yielding a semantically consistent coarse-grained view.

In VCL, various augmentation strategies are widely adopted to help extract richer semantic information. However, designing task-specific graph augmentations remains empirically driven, due to the irregular and non-Euclidean nature of graph data. This makes it difficult to ensure that the augmentation is particularly useful for graph-specific downstream learning tasks [17], [18]. One possible reason is that random edge/node perturbations may alter graph semantics [19]. As visually

summarized in Fig. 1, random perturbations (a) can inadvertently destroy essential high-level structures like community integrity. In contrast, hierarchical graph pooling (b) offers a superior, deterministic alternative. This naturally raises our first critical question: *can we design a GCL framework that completely avoids empirical data augmentation by leveraging structure-preserving transformations like graph pooling?*

Some recent efforts have pursued augmentation-free GCL paradigms [19]–[21]. However, these methods typically remain confined to same-granularity comparisons, neglecting the inherent hierarchical properties of real-world networks. This reveals a further limitation and leads to our second question: *even if we achieve augmentation-free learning, how can we effectively capture the multi-level, hierarchical information inherent in complex graphs?*

The challenge of integrating multi-granularity learning is also a prominent trend in VCL, with powerful paradigms like Pyramid Contrastive Learning (PCLC) [22] and hybrid-grained methods like DCHL [23] demonstrating its effectiveness. Inspired by this, our work aims to bridge the gap between the distinct research directions of augmentation-free GCL and hierarchical representation learning. A key challenge is that graphs lack the natural pyramidal structure of images. A promising direction, which we pursue in this work, is to explicitly construct a graph hierarchy via pooling, thus successfully adapting this advanced multi-scale contrastive idea to the graph domain.

To address these two fundamental challenges, we propose HPoolGCL, a novel GCL framework that is both augmentation-free and capable of learning multi-granularity representations. First, to preserve rich semantics and capture high-level structural information, we employ graph pooling to coarsen the original graph into multiple coarse-grained graphs, obtaining multi-granularity node representations through shared graph neural network (GNN) encoders. As most pooling methods target supervised scenarios, we design a novel Hierarchical Consistency and Redundancy Reduction (HiCR) loss for self-supervised settings to ensure pooled representations preserve global information while reducing redundancy. After obtaining the coarsened graphs that preserve high-level information, we further focus on enabling the model to learn multi-granularity information across hierarchical levels. To achieve this, we naturally treat nodes discarded during pooling as redundant nodes, which should be pushed apart as negative samples, while the key nodes retained in the coarsened graph serve as positive samples. This strategy allows us to construct a Multi-Granularity Contrastive (MGC) loss to facilitate cross-granularity contrastive learning. Notably, we do not introduce the additional explicit positive/negative sample selection mechanism. Instead, we fully leverage the selection effect of graph pooling, where positive and negative sample pairs naturally emerge during the multi-granularity view construction process. Furthermore, by combining MGC loss with the redundancy reduction term in HiCR loss, our theoretical analysis demonstrates that, under mild assumptions, our framework establishes a formal connection to the information bottleneck (IB) principle. As shown in Table I, the proposed cross-granularity contrast method HPoolGCL does not rely on empirical data augmentation, explicit negative sampling methods, parameterized MI estimators, or multi-encoder frameworks, which are common components in these GCL methods. In summary, our contributions can be outlined as follows:

- We propose a novel cross-granularity GCL framework that deeply integrates hierarchical graph pooling to learn multi-granularity information from complex graph structures. Our architecture eliminates the dependency on empirical data augmentation, explicit negative sampling, and multiple encoders while remaining compatible with diverse graph pooling algorithms.
- By leveraging inherent selection effects in graph pooling, we design two loss functions: (1) MGC loss that aligns hierarchical representations through cross-granularity MI maximization and (2) HiCR loss grounded in redundancy reduction principles. Theoretical analysis shows that the unified loss function simultaneously satisfies MI maximization, redundancy reduction, and the IB principle.
- Extensive experiments demonstrate that HPoolGCL achieves state-of-the-art performance across multiple benchmarks.

The remainder of the paper is structured as follows. Section II reviews the related work. Section III gives a detailed description of the HPoolGCL framework. The experimental study is presented in Section IV, and we conclude the paper in Section V.

TABLE I: Technical comparison of GCL methods. HPoolGCL is a unique framework that supports cross-granularity learning while being free from common complex components. Abbreviations: Cross-Gran. (Cross-Granularity), Aug-Free (Augmentation-Free), Neg-Free (Negative-Free), MI-Free (MI-Estimator-Free), SE (Single-Encoder).

Method	Cross-Gran.	Aug-Free	Neg-Free	MI-Free	SE
DGI	✓	✗	✗	✗	✓
MVGRL	✓	✗	✗	✗	✗
GRACE	✗	✗	✗	✓	✓
GCA	✗	✗	✗	✓	✓
BGRL	✗	✗	✓	✓	✗
AFGRL	✗	✓	✓	✓	✗
iGCL	✗	✓	✓	✓	✗
HPoolGCL	✓	✓	✓	✓	✓

✓: Yes (The method possesses this desirable property, e.g., it is augmentation-free).
✗: No (The method relies on the corresponding component, e.g., it requires data augmentation).

II. Related Work

A. Graph Contrastive Learning

Motivated by the remarkable successes of SSL in CV/NLP, numerous studies have adapted contrastive learning to graphs. Same-granularity contrast focuses on comparing views at the same granularity. Inspired by SimCLR [3], GRACE [10] proposes a dual-view contrastive framework that learns robust representations by maximizing the consistency of node representations at the same granularity. GCA [12] improves

GRACE by introducing an adaptive augmentation strategy that dynamically generates diverse node-level views. To eliminate the dependency on explicit negative sampling, BGRL [24] eliminates negative sampling by adopting the BYOL [4] architecture [4]. SGCL [25] analyzes the components of BGRL and proposes a simplified pipeline framework that utilizes the outputs from two consecutive iterations as positive pairs. CCA-SSG [26] employs canonical correlation analysis to avoid parameterized MI estimators. Representative cross-granularity methods like DGI [14] follow the Deep InfoMax principles [15] by contrasting node representations with global graph summaries to integrate global information. MVGRL [15] employs graph diffusion techniques to create augmented structural views of graphs and uses a discriminator to contrast node-level and graph-level representations across different views. CoLM^{2}S [27] explores multi-scale structure-level and feature-level signals based on a mutual information mechanism and performs contrastive learning both within and across graph relations to improve the expressive power of node representations. To capture more complex structural information, some works leverage graph pooling to construct multi-granularity views. For instance, HCL [28] and CGKS [16] both propose hierarchical contrastive frameworks. However, despite their architectural advancements, they still fundamentally rely on data augmentation to generate contrastive pairs, failing to address its inherent risks.

A parallel line of research focuses on augmentation-free GCL to avoid the potential semantic distortion from random perturbations. Methods like SimGRACE [13] uses the original graph as input, pairing a GNN model with its perturbed variant as dual encoders to obtain two correlated views for contrastive learning without explicit data augmentation. AFGRL [19] proposes an augmentation-free GCL method that selects target nodes sharing local structural and global semantic information as positive samples. iGCL [20] introduces the invariant-discriminative loss (ID loss) to learn invariant and discriminative representations. Another emerging augmentation-free paradigm is based on diffusion models, where implicit augmentation is achieved through a denoising process. For instance, Directional Diffusion Models [29] propose using anisotropic noise to better preserve structural information, while SDMG [30] focuses on reconstructing global, low-frequency information. However, despite their innovations, a common limitation across most of these augmentation-free approaches is their failure to explicitly model and leverage the hierarchical nature of graphs, with diffusion models also often incurring high computational costs.

In contrast to these prior works, our framework presents a fundamentally different and more streamlined approach that is both hierarchical and augmentation-free, uniquely bridging the gap between these distinct research directions.

B. Graph Data Augmentation

In VCL, the strategic combination of data augmentation techniques plays a key role [3]. However, due to the non-Euclidean and irregular nature of graph data, designing effective augmentations is a significant challenge. In GCL, augmentations typically involve perturbing node features or graph topology, such as through node feature masking, edge dropping, or subgraph sampling.

Existing methods predominantly rely on empirical combinations of these strategies. For instance, GRACE [10] and CCA-SSG [26] randomly drop edges and mask features, but such random removals risk discarding critical information. GraphCL [17] explored a wider range of augmentation types, and subsequent works like JOAO [31] have even attempted to automate the selection process. LightGCL [32] utilizes singular value decomposition (SVD) to guide the construction of contrastive views by modifying or generating node attributes to enhance sample diversity. CGSRL [33] employs a discriminator and generator to reconstruct the original features while introducing noise during the process to achieve augmentation effects. Despite these efforts, previous studies confirm that no universal augmentation scheme is effective across all graph structures [17].

Consequently, the heavy reliance on data augmentation in GCL raises two critical issues: (1) the potential for semantic corruption through inappropriate augmentations, leading to degraded performance, and (2) the prohibitive computational cost of searching for optimal augmentation strategies. To circumvent these limitations, we propose leveraging multi-granularity information within graphs, thereby eliminating the reliance on data augmentation.

C. Graph Pooling

Graph pooling methods create coarsened graph structures and are broadly categorized into global and hierarchical approaches. Global pooling methods, such as mean/max pooling and more advanced techniques like Set2Set [34] and SortPool [35], collapse the graph into a single representation at once, making them unsuitable for capturing the intermediate, multi-granularity information central to our work.

Hierarchical pooling, in contrast, generates a sequence of progressively coarser graphs, making it highly relevant to our framework. This paradigm follows two main strategies. Node clustering approaches, pioneered by DiffPool [36], learn a soft assignment matrix to map nodes to clusters. This idea has been refined from spectral perspectives by MincutPool [37] and through memory modules by MemPool [38], though these methods often incur high computational costs. Node selection methods offer a more efficient alternative by scoring and selecting a subset of informative nodes. This ranges from early methods using simple projections like gPool [39] to GNN-based attention scoring like SAGPool [40], which we adopt in our work. A notable example in this category is HGP-SL [41], which combines adaptive node selection with a structure learning mechanism. Subsequent efforts have enhanced the pooling process by improving the scoring mechanism through multi-view strategies, as seen in MVPool [42], or by introducing unpooling operators, as in AdamGNN [43]. Another line of research grounds the selection in information-theoretic principles, with methods like VIPool [44] and CGIPool [45] aiming to maximize mutual information. However, these methods are typically designed for supervised settings.

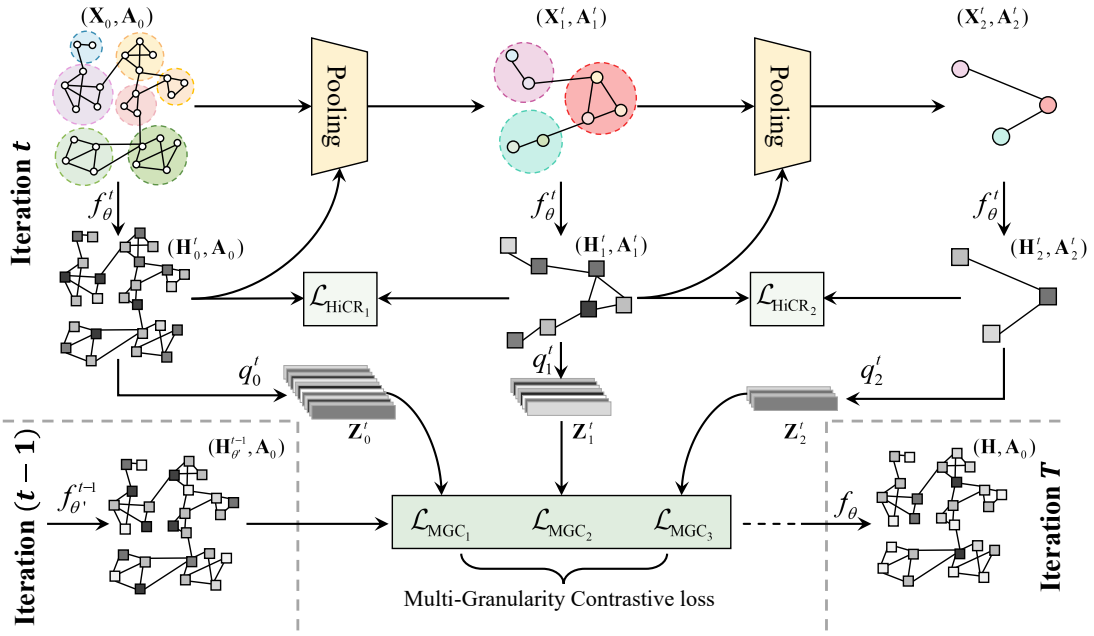


Fig. 2: Illustration of the proposed HPoolGCL. We use two pooling layers as a demonstration. During each training iteration t, we first apply stacked pooling layers to the input graph $(\mathbf{X}_0, \mathbf{A}_0)$ to construct a hierarchy of coarsened (multi-granularity) graphs $(\mathbf{X}_1^t, \mathbf{A}_1^t)$ and $(\mathbf{X}_2^t, \mathbf{A}_2^t)$. Then we feed these multi-granularity graphs into the online encoder f_θ^t to obtain node representations $\{\mathbf{H}_k^t\}_{k=0}^2$. The projectors are then used to generate $\{\mathbf{Z}_k^t\}_{k=0}^2$ from the node representations. In parallel, $\mathbf{H}_{\theta'}^{t-1}$ is produced by the target encoder$f_{\theta'}^{t-1}$, which is essentially the online encoder from iteration $t-1$. Finally, we compute HiCR loss using $\{\mathbf{H}_k^t\}_{k=0}^2$ and perform cross-granularity contrastive between $\{\mathbf{Z}_k^t\}_{k=0}^2$ and $\mathbf{H}_{\theta'}^{t-1}$. After training, the encoder generates node representations $\mathbf{H}$ from the original graph for downstream tasks.

Our work builds upon the node selection paradigm for its conceptual simplicity and efficiency. We distinguish ourselves by proposing a novel framework where the pooling method acts as a pluggable module. This flexibility allows our core contribution, HiCR loss, to provide explicit, unsupervised guidance to various existing pooling techniques (e.g., SAGPool). This is particularly significant as it enables these methods, which are typically designed for supervised settings, to be effectively utilized in a fully unsupervised manner.

III. Methodology

In this section, we propose a cross-granularity and augmentation-free GCL method called HPoolGCL. Unlike AFGRL [19] or iGCL [20], which maximize the similarity between representations from online and target encoders, our method aligns multi-granularity node representations from iteration $t \in \{1, \ldots, T\}$ with finest-grained node representations from iteration $t-1$.

As shown in Fig. 2, HPoolGCL consists of the following main components:

(1) A graph encoder f_θ to learn node representations at different granularities.

(2) A set of stacked pooling layers for extracting multi-granularity information from the original graph.

(3) MGC loss $\mathcal{L}_{\text{MGC}}$ for aligning multi-granularity information and HiCR loss $\mathcal{L}_{\text{HiCR}}$ to ensure the effectiveness of the pooling process.

The pseudo-algorithm of HPoolGCL is described in Algorithm 1.

Next, we first investigate whether the exponential moving average (EMA) mechanism is intrinsically effective for augmentation-free GCL. Then, we introduce the details of HPoolGCL and theoretically analyze the effectiveness of the loss function we designed.

A. Preliminary

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with N nodes, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively. We represent the node feature matrix as $\mathbf{X} \in \mathbb{R}^{N \times F}$, where F is the input dimension, and the adjacency matrix as $\mathbf{A} \in \mathbb{R}^{N \times N}$. The adjacency matrix $\mathbf{A}$ is defined such that $\mathbf{A}_{ij} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$, and $\mathbf{A}_{ij} = 0$ otherwise.

Our goal is to learn an encoder $f(\cdot)$ that generates the low dimensional node representations $\mathbf{H} = f(\mathbf{X}, \mathbf{A}) \in \mathbb{R}^{N \times D}$, where D is the embedding size. Specifically, our goal is to learn node representations that generalize well to various downstream tasks without using any class information.

B. EMA Mechanism in Augmentation-Free GCL

Siamese network architectures are widely adopted in contrastive learning. As a pioneering work on asymmetric Siamese architectures, BYOL [4] introduced the EMA mechanism to eliminate the dependency on explicit negative sampling. Specifically, BYOL employs an online encoder f_θ and a target encoder f_ξ, where θ is updated via gradient descent while ξ is updated through the EMA mechanism: $\xi \leftarrow \tau\xi + (1-\tau)\theta$.

Algorithm 1: HPoolGCL training process.

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, adjacency matrix $\mathbf{A}_0$, feature matrix $\mathbf{X}_0$, graph encoder $f_\theta(\cdot)$, pooling layers $\{pool_k(\cdot)\}_{k=1}^{K}$, projectors $\{q_k^t\}_{k=0}^{K}$, maximum number of pooling layers K, maximum number of iterations T;

Output: The learned encoder $f_\theta(\cdot)$;

Initialize target representation $\mathbf{H}_{\theta'}^0 = f_{\theta'}^0(\mathbf{X}_0, \mathbf{A}_0)$ using randomly initialized encoder parameters;

for *each iteration* $t = 1$ *to* T **do**

  Finest-grained online representation generation: $\mathbf{H}_0^t = f_\theta^t(\mathbf{X}_0, \mathbf{A}_0)$;

  Calculate $\mathbf{Z}_0^t = q_0^t(\mathbf{H}_0^t)$;

  Calculate finest-grained contrastive loss $\mathcal{L}_{\text{MGC}_0}$ according to Eq. (3);

  for *each granularity* $k = 1$ *to* K **do**

    Compute idx_k using pooling layer $pool_k(\mathbf{H}_{k-1}^t, \mathbf{A}_{k-1}^t)$;

    Generate the coarsened graph $(\mathbf{X}_k, \mathbf{A}_k)$ according to Eq. (2);

    k-th granularity online representation generation: $\mathbf{H}_k^t = f_\theta^t(\mathbf{X}_k, \mathbf{A}_k)$;

    Calculate $\mathbf{Z}_k^t = q_k^t(\mathbf{H}_k^t)$;

    Calculate k-th granularity contrastive loss $\mathcal{L}_{\text{MGC}_k}$ according to Eq. (4);

    Calculate k-th granularity HiCR loss $\mathcal{L}_{\text{HiCR}_k}$ according to Eq. (6);

  end

  Calculate total loss $\mathcal{L}$ according to Eq. (8);

  Update θ_t using the optimizer;

  Set θ_t as the target encoder parameters θ_t';

  Target representation generation: $\mathbf{H}_{\theta'}^t = f_{\theta'}^t(\mathbf{X}_0, \mathbf{A}_0)$;

end

return $f_\theta(\cdot)$.

This mechanism was commonly believed indispensable for preventing model collapse. Consequently, subsequent GCL works (e.g., BGRL [24], AFGRL [19], and iGCL [20]) have retained EMA in their frameworks. To investigate whether EMA is fundamentally necessary for augmentation-free GCL, a scenario in which models are theoretically more prone to collapse, we conduct ablation studies on AFGRL specifically targeting the EMA mechanism. As shown in Fig. 3, the model maintains robust performance even when EMA is deactivated ($\tau = 0$). This insight motivates us to design a more streamlined model framework. Notably, a recent study reached similar conclusions regarding the negligible impact of EMA in BGRL. This observation aligns with SimSiam's conclusion [46] that collapse prevention primarily stems from stop-gradient operations rather than from the EMA mechanism.

C. Graph Encoder

HPoolGCL follows the general philosophy of BYOL, BGRL, and AFGRL, which is to learn invariant representations by maximizing the similarity between the outputs of the online encoder and the target encoder. However, inspired by the results of Section III-B and the architectural simplification proposed in SGCL [25], we remove the EMA mechanism and retain only a single encoder f_θ. The encoder in the current iteration t is treated as the online encoder f_θ^t, and the encoder from the previous iteration $t-1$ serves as the target encoder $f_{\theta'}^{t-1}$. Here, θ' refers to the parameters from the previous iteration, which are kept frozen during the current iteration. This design inherently implements stop-gradient by freezing the target encoder during backpropagation.

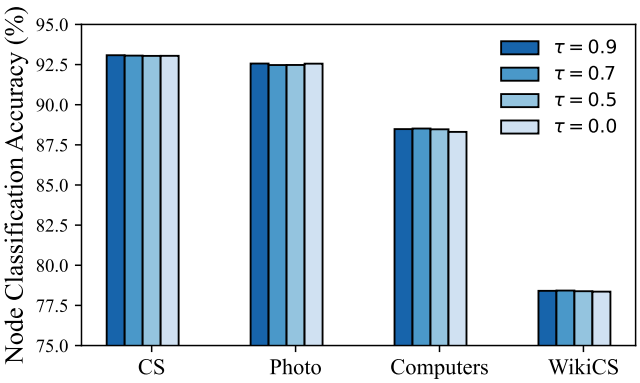


Fig. 3: Impact of different τ values on AFGRL performance across Coauthor-CS, Amazon-Photo, Amazon-Computers, and WikiCS datasets.

Specifically, in each iteration t, we apply K stacked graph pooling layers to the original graph $(\mathbf{X}_0, \mathbf{A}_0)$, generating a hierarchy of coarsened graphs: $(\mathbf{X}_1^t, \mathbf{A}_1^t), \ldots, (\mathbf{X}_k^t, \mathbf{A}_k^t), \ldots, (\mathbf{X}_K^t, \mathbf{A}_K^t)$, where $\mathbf{X}_k^t \in \mathbb{R}^{N_k \times F}$ denotes the node feature matrix and $\mathbf{A}_k^t \in \{0,1\}^{N_k \times N_k}$ represents the adjacency matrix at the granularity level k, with $N_0 > N_1 > \cdots > N_K$ indicating progressively reduced graph sizes. After each pooling step, we interleave encoding operations using the same online encoder f_θ^t, where the representations from the current level simultaneously serve as input features for the next pooling layer. Finally, we obtain multi-granularity online representations of the original graph: $\{\mathbf{H}_0^t, \ldots, \mathbf{H}_K^t\}$, where $\mathbf{H}_k^t = f_\theta^t(\mathbf{X}_k^t, \mathbf{A}_k^t)$, each $\mathbf{H}_k^t \in \mathbb{R}^{N_k \times D}$ contains D-dimensional node representations for the k-th granularity level.

For the target representation, we use the finest-grained node representations generated in iteration $t-1$ by the optimized parameters of the encoder $f_{\theta'}^{t-1}$, that is, $\mathbf{H}_{\theta'}^{t-1} = f_{\theta'}^{t-1}(\mathbf{X}_0, \mathbf{A}_0), \mathbf{H}_{\theta'}^{t-1} \in \mathbb{R}^{N \times D}$. For the first iteration, the target representation is generated using the online encoder with randomly initialized parameters. As for the specific architecture of the graph encoder, we simply adopt the widely studied graph convolutional networks (GCNs) [47] for a fair comparison. Note that the encoder can be flexibly replaced by other architectures, such as GraphSAGE [48], graph attention networks (GATs) [49].

D. Multi-granularity View Construction

We construct multiple coarsened graphs through hierarchical downsampling to preserve intermediate structural hierarchies,

as illustrated in Fig. 1 (b). The pooling process selectively retains critical nodes and their corresponding edges as coarse-grained representations of the original graph view, where node selection is determined by importance metrics. These coarsened graphs retain high-level information from the original data, such as functional groups in molecular systems and communities in social networks.

In HPoolGCL, we employ the downsampling component of SAGPool [40] as our pooling layer. This approach leverages GCN to compute node attention scores, thereby incorporating both node features and topological structures during pooling. Specifically, with K stacked pooling layers, the k-th pooling layer ($k \in [1, K]$) processes node representations $\mathbf{H}_{k-1}^t$ and adjacency matrix $\mathbf{A}_{k-1}$ obtained from the $(k-1)$-th layer, where $\mathbf{H}_{k-1}^t = f_\theta^t(\mathbf{X}_{k-1}, \mathbf{A}_{k-1})$. The node importance scores of the previous-layer nodes, which are denoted as $s_{k-1} \in \mathbb{R}^{N\times 1}$, are computed through:

$$s_{k-1} = \sigma\left(\text{GCN}(\mathbf{H}_{k-1}^t, \mathbf{A}_{k-1})\right), \tag{1}$$

where σ denotes the activation function (e.g., $\tanh$). Node selection retains the top-$\lceil pN_{k-1} \rceil$ highest-scoring nodes according to the predetermined pooling ratio $p \in (0, 1]$.

Then the coarsened graph $(\mathbf{X}_k, \mathbf{A}_k)$ is subsequently constructed by:

$$\mathbf{X}_k = \mathbf{X}_{k-1}[idx_k, :], \mathbf{A}_k = \mathbf{A}_{k-1}[idx_k, idx_k], \tag{2}$$

where idx_k denotes the indices of top-ranked nodes in i-th pooling layer according to importance scores s_{k-1}.

Through K iterative pooling steps, we construct multi-granularity views corresponding to the original graph, $(\mathbf{X}_1^t, \mathbf{A}_1^t), \ldots, (\mathbf{X}_k^t, \mathbf{A}_k^t), \ldots, (\mathbf{X}_K^t, \mathbf{A}_K^t)$, where $(\mathbf{X}_0, \mathbf{A}_0)$ corresponds to the original graph view.

It is worth noting that the pooling layers in HPoolGCL can be substituted with other hierarchical graph pooling methods (e.g., downsampling layers in MVPool [42], Graph U-Nets [39]), as demonstrated in the experiments section IV-F where we analyze HPoolGCL's performance across different pooling strategies.

E. Learning Objective

1) Multi-Granularity Contrastive (MGC) loss: To align the multi-granularity representations $\{\mathbf{H}_k^t\}_{k=0}^K$ at iteration t with the target embedding $\mathbf{H}_{\theta'}^{t-1}$ from iteration $t-1$, we propose MGC loss that facilitates multi-granularity information maximization.

First, we employ a set of projectors $q_{\theta_0}^t, \ldots, q_{\theta_k}^t$ to map the representations from the online encoder into distinct latent spaces, obtaining the projected embeddings $\{\mathbf{Z}_{\theta_0}^t, \ldots, \mathbf{Z}_{\theta_K}^t\}$, where $\mathbf{Z}^t = q_k^t(\mathbf{H}_k^t)$. At the finest granularity ($k = 0$), we construct the contrastive loss by maximizing the similarity between each node's projection $z_{0,n}^t$ from $\mathbf{Z}_0^t$ and its corresponding target embedding h_n^{t-1} from $\mathbf{H}_{\theta'}^{t-1}$:

$$\mathcal{L}_{\text{MGC}_0} = -\frac{1}{N}\sum_{n=1}^{N} \log \frac{e^{\text{sim}(z_{0,n}^t, h_n^{t-1})/\tau}}{N}, \tag{3}$$

where τ is the temperature hyperparameter and $\text{sim}(\cdot)$ denotes the cosine similarity, calculated as $\text{sim}(z_{0,n}^t, h_n^{t-1}) = \frac{\langle z_{0,n}^t, h_n^{t-1}\rangle}{\|z_{0,n}^t\|_2 \|h_n^{t-1}\|_2}$. This formulation essentially maximizes the similarity between $z_{0,n}^t$ and h_n^{t-1} for each corresponding node n, and the logarithmic operation serves to normalize $\mathcal{L}_{\text{MGC}_0}$ to the same numerical scale as contrastive losses at other granularity levels.

For coarse-grained views, we can regard the pooling process as a node selection procedure, where each pooling operation essentially preserves critical nodes from the graph and discards unimportant nodes. The coarsened graph formed by these critical nodes preserves essential information from the original graph, while the discarded nodes can be regarded as redundant information or noise. We refer to this process as the *selection effect* of pooling. Therefore, in cross-granularity contrastive learning, our objective is to enable the encoder to learn high-level patterns from coarsened graphs while minimizing the influence of redundant nodes.

To achieve this objective, we implement the cross-granularity contrast in MGC loss using the InfoNCE [50]. As shown in Fig. 1 (b), we construct cross-granularity positive and negative sample pairs between the coarsened graph and the original graph. Specifically, to compute the InfoNCE, in iteration t, we utilize the pooling indices $\{\text{idx}_k\}_{k=1}^K$ to select the representations of positive and negative samples from the target representations $\mathbf{H}_{\theta'}^{t-1}$ as follows:

- Anchors: Projected embeddings from the online encoder: $z_{k,n}^t = q_{\theta_k}^t\left(\mathbf{H}_k^t[n, :]\right) \in \mathbf{Z}_k^t$.
- Positives: Corresponding node representations from the target encoder via pooling indices: $h_{k,n}^{t-1,+} = \mathbf{H}_{\theta'}^{t-1}[\text{idx}_k(n), :]$, where $\text{idx}_k(n)$ denotes the original graph index of the n-th preserved node at pooling layer k.
- Negatives: Target node representations of all discarded nodes, defined as $\{h_{k,m}^{t-1,-}\}_{m=1}^{\tilde{N}} = \mathbf{H}_{\theta'}^{t-1}[\text{idx}_k^{\text{neg}}, :]$, where $\text{idx}_k^{\text{neg}} = \{1, \ldots, N\} \setminus \text{idx}_k(\cdot)$ denotes the indices of non-selected nodes at layer k. Here, N is the total number of nodes, N_k is the number of preserved nodes after pooling at layer k, and $\tilde{N} = N - N_k$.

MGC loss at granularity level k is formulated as:

$$\mathcal{L}_{\text{MGC}_k} = -\frac{1}{N_k}\sum_{n=1}^{N_k} \log \frac{e^{s_{k,n}^+/\tau}}{e^{s_{k,n}^+/\tau} + \sum_{m=1}^{\tilde{N}} e^{s_{k,n,m}^-/\tau}}, \tag{4}$$

where $s_{k,n}^+ = \text{sim}(z_{k,n}^t, h_{k,n}^{t-1,+})$, $s_{k,n,m}^- = \text{sim}(z_{k,n}^t, h_{k,m}^{t-1,-})$.

The total MGC loss integrates all levels through weighted summation:

$$\mathcal{L}_{\text{MGC}} = \alpha \cdot \mathcal{L}_{\text{MGC}_0} + \beta \cdot \frac{1}{K}\sum_{k=1}^{K} \mathcal{L}_{\text{MGC}_k}, \tag{5}$$

where α and β are hyperparameters controlling the importance of contrastive losses at different granularities.

From the above contrastive process, we observe that HPoolGCL does not rely on random data augmentation techniques or introduce additional explicit sampling strategies to construct positive and negative samples. Instead, by leveraging the inherent node selection mechanism in the graph pooling process, our framework simultaneously achieves two objectives during multi-granularity view construction: (1) implicit identification

of critical information to maximize, and (2) automatic suppression of task-irrelevant redundancies.

2) Hierarchical Consistency and Redundancy Reduction (HiCR) loss: Although the positive/negative sample selection strategy in HPoolGCL naturally integrates with the graph pooling process, we face a critical challenge: how to ensure the pooling process properly preserves crucial information from the original graph? Current graph pooling methods are primarily designed for supervised scenarios, where label information can guide model learning. However, unsupervised graph pooling remains underexplored.

To address this, we propose HiCR loss, a novel unsupervised pooling objective based on variance alignment and redundancy penalty. For multi-granularity representations $\{\mathbf{H}_k^t \in \mathbb{R}^{N_k \times D}\}_{k=0}^K$, HiCR loss at layer k (for $1 \leq k \leq K$) is formulated as:

$$\mathcal{L}_{\text{HiCR}_k} = \underbrace{\|\text{diag}(C_{k-1,k-1}) - \text{diag}(C_{k,k})\|_2^2}_{\text{consistency term}} + \lambda \underbrace{\|C_{k,k-1}\|_F^2}_{\text{redundancy reduction term}}, \tag{6}$$

where λ is a trade-off hyperparameter that balances the consistency term and the redundancy reduction term. The covariance matrices are expressed as: $C_{k-1,k-1} = (\mathbf{H}_{k-1}^t)^\top \mathbf{H}_{k-1}^t$, $C_{k,k} = (\mathbf{H}_k^t)^\top \mathbf{H}_k^t$, and $C_{k,k-1} = (\mathbf{H}_k^t)^\top \mathbf{H}_{k-1}^t[\text{idx}_k, :]$, where all covariance matrices are in $\mathbb{R}^{D \times D}$. This formulation is applied across all pooling layers for $k \in [1, K]$, where the pooling indices idx_k naturally handle the index alignment between adjacent layers, including at the boundaries (e.g., between $\mathbf{H}_0^t$ and $\mathbf{H}_1^t$ for $k = 1$).

This loss consists of a consistency term and a redundancy reduction term. The consistency term encourages the preservation of critical information by aligning the per-dimension variances of node representations across adjacent pooling layers. This mechanism extends the Pooling Information Loss proposed by Su et al. [51], which mitigates information loss by minimizing the discrepancy between the inner product matrices before and after pooling. However, we argue that aligning the entire representation distribution may inadvertently retain redundant or noisy information. In contrast, our consistency term focuses solely on aligning the diagonal elements of the covariance matrices, which correspond to the variances of individual representation dimensions. The redundancy reduction term follows the decorrelation strategy proposed in Barlow Twins [52] and extends it to hierarchical pooling scenarios. Specifically, it minimizes the Frobenius norm of the inter-layer covariance matrix, encouraging representations from adjacent pooling layers to be less correlated. This helps reduce redundant information across layers and promotes the preservation of important features after pooling.

For the K pooling operations, the total HiCR loss can be expressed as:

$$\mathcal{L}_{\text{HiCR}} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\text{HiCR}_k}. \tag{7}$$

To harmonize the objectives of multi-granularity representation alignment and pooling information preservation, we unify the contrastive and hierarchical pooling losses into a joint optimization framework:

$$\mathcal{L} = \mathcal{L}_{\text{MGC}} + \gamma_p \mathcal{L}_{\text{HiCR}}, \tag{8}$$

where γ_p governs the trade-off between contrastive learning objectives and pooling regularization. $\mathcal{L}_{\text{MGC}}$ promotes the consistency of the representations across different granularities, ensuring cross-granularity alignment. Meanwhile, $\mathcal{L}_{\text{HiCR}}$ enables the pooling module to preserve key information and suppress redundancy during the pooling process in an unsupervised setting.

F. Theoretical Analysis

1) Connection with the InfoMax principle: Based on the InfoMax principle [15], a primary goal of Graph Contrastive Learning (GCL) is to maximize the mutual information (MI) between representations of related graph instances. Our Multi-Granularity Contrastive (MGC) loss is built upon the InfoNCE objective, which is known to maximize a lower bound on the MI between positive pairs [50].

Specifically, minimizing $\mathcal{L}_{\text{MGC}_k}$ at each granularity level is equivalent to maximizing the MI between the coarsened representations $\mathbf{H}_k^t$ and the target representations $\mathbf{H}_{\theta'}^{t-1}$. A detailed derivation of this connection is provided in Appendix A. Since our total loss $\mathcal{L}_{\text{MGC}}$ is a weighted sum of these individual losses, the overall optimization process aims to maximize the sum of MI across all levels. For conceptual clarity, this objective can be expressed as:

$$\min_{f_\theta^t} \mathcal{L}_{\text{MGC}} \Rightarrow \max_{f_\theta^t} \frac{1}{K+1} \sum_{k=0}^{K} \mathcal{I}(\mathbf{H}_{\theta'}^{t-1}, \mathbf{H}_k^t), \tag{9}$$

where $\mathcal{I}$ denotes the mutual information. This formulation also provides theoretical support for our unsupervised pooling process, as maximizing the MI with the original graph's representation ($\mathbf{H}_{\theta'}^{t-1}$) implicitly encourages the pooling module to preserve essential information.

2) Connection with redundancy reduction principle: Inspired by Barlow's redundancy reduction principle [53], recent SSL methods like Barlow Twins [52] aim to decorrelate the feature dimensions of embeddings. This is typically achieved by minimizing a loss based on the cross-correlation matrix C:

$$\mathcal{L}_{\text{BT}} = \sum_i (1 - C_{ii})^2 + \lambda \sum_i \sum_{i \neq j} C_{ij}^2, \tag{10}$$

where the first term serves as an invariance objective and the second as a redundancy reduction objective. Our HiCR loss adapts and extends this principle to the hierarchical graph pooling setting.

Our HiCR loss consists of two components that correspond to the objectives of this principle. The consistency term serves as an invariance objective analogous to the first term of $\mathcal{L}_{\text{BT}}$. However, instead of enforcing strict invariance by forcing the diagonal of the cross-correlation matrix to be 1, it flexibly aligns only the variance of each feature dimension across adjacent pooling layers. This encourages the preservation of essential statistical properties while allowing for representation transformation during coarsening.

Simultaneously, the redundancy reduction term directly applies the decorrelation objective to a novel, hierarchical context. Unlike Barlow Twins which reduces intra-representation redundancy, our term aims to reduce inter-layer redundancy by penalizing the cross-covariance between adjacent layers. This dual objective guides the pooling process to create coarsened representations that are both informative and compact.

3) Connection with the information bottleneck principle: The Information Bottleneck (IB) principle [54] provides a unified theoretical perspective for our framework. In its supervised form, it aims to learn a representation $\mathbf{Z}$ from an input $\mathbf{X}$ that is maximally informative about a target $\mathbf{Y}$:

$$\mathcal{IB}_{\text{sup}} = \mathcal{I}(\mathbf{Y};\mathbf{Z}) - \eta\mathcal{I}(\mathbf{X};\mathbf{Z}), \eta > 0. \tag{11}$$

In self-supervised learning (SSL), this is adapted by replacing the target $\mathbf{Y}$ with a self-supervised signal $\mathbf{S}$ (e.g., an augmented view) [52]:

$$\mathcal{IB}_{\text{ssl}} = \mathcal{I}(\mathbf{S};\mathbf{Z}) - \eta\mathcal{I}(\mathbf{X};\mathbf{Z}), \eta > 0. \tag{12}$$

We extend this principle to our hierarchical setting. For each pooling step k, the goal is to learn a representation $\mathbf{H}_k^t$ that is maximally informative about the global target $\mathbf{H}_{\theta'}^{t-1}$ (our $\mathbf{S}$), while compressing information from its input $\mathbf{H}_{k-1}^t$ (our $\mathbf{X}$). Summing over all levels leads to the hierarchical IB objective for HPoolGCL:

$$\mathcal{IB}_{\text{HPoolGCL}} = \sum_{k=0}^{K} \mathcal{I}(\mathbf{H}_{\theta'}^{t-1};\mathbf{H}_k^t) - \eta \sum_{k=1}^{K} \mathcal{I}(\mathbf{H}_{k-1}^t;\mathbf{H}_k^t). \tag{13}$$

Our unified loss $\mathcal{L} = \mathcal{L}_{\text{MGC}} + \gamma_p\mathcal{L}_{\text{HiCR}}$ naturally optimizes this objective. We analyze how its components at each layer k align with the IB principle:

- MGC loss ($\mathcal{L}_{\text{MGC}_k}$), as established in Section III-F1, maximizes the MI with the target, $\mathcal{I}(\mathbf{H}_{\theta'}^{t-1};\mathbf{H}_k^t)$. This directly serves as the prediction term of the IB objective, encouraging the retention of relevant information.
- The redundancy reduction term within HiCR loss ($\gamma_p\lambda \left\|C_{k,k-1}\right\|_F^2$) acts as the compression term. By penalizing the inter-layer cross-correlation, it serves as an effective proxy for minimizing the compression cost $\mathcal{I}(\mathbf{H}_{k-1}^t;\mathbf{H}_k^t)$, a connection established under Gaussian assumptions in [52].
- The consistency term in HiCR loss ($\gamma_p \left\|\text{diag}(C_{k-1,k-1}) - \text{diag}(C_{k,k})\right\|_2^2$) can be seen as a regularizer on the compression process. It ensures that the compression achieved by pooling does not lead to catastrophic information loss by encouraging the variance of the representations to be preserved.

For a more direct formulation of our loss in the IB framework, please refer to Appendix B. Thus, the joint optimization of MGC and HiCR effectively instantiates a hierarchical IB principle in a self-supervised manner. Unlike supervised IB methods like PSGT [55] that rely on task labels to guide compression, our framework learns what to preserve and compress by enforcing cross-granularity consistency. This provides a strong theoretical rationale for our framework's empirical success.

IV. Experiments and Analysis

A. Datasets

TABLE II: Statistical details of node classification benchmark datasets

Dataset	Nodes	Edges	Features	Classes
CS	18,333	81,894	6,805	15
Physics	34,493	247,962	8,415	5
Computers	13,752	245,861	767	10
Photo	7,650	119,081	745	8
WikiCS	11,701	216,123	300	10
Arxiv_O	169,343	2,315,598	128	40

To evaluate the performance of HPoolGCL, we conduct extensive experiments on six widely used benchmark datasets, including the large-scale Ogbn-Arxiv. As in [19], for datasets without a standard public split, we randomly divide the nodes into training (10%), validation (10%), and testing (80%) sets. For WikiCS and Ogbn-Arxiv, which provide official splits, we directly utilize their given divisions to ensure fair and reproducible comparisons. The statistical information of the datasets is listed in Table II. We also present a detailed description of these datasets in the following.

- WikiCS. The dataset is a web network formed by computer science-themed pages on Wikipedia, where nodes denote the Wikipedia pages, and edges represent cross-references between these pages. Each page is classified into one of ten classes [56].
- Amazon-Computers, Amazon-Photo. Both of them are subgraphs in the Amazon copurchase relationship graph, where nodes denote the products, and edges represent how frequently they are purchased together [57] (abbreviated as Computers and Photo).
- Coauthor-Physics, Coauthor-CS. Both of the datasets are academic networks extracted from the Microsoft Academic Graph, which contains coauthorship relationships, where node features represent author information, and the edges reflect the coauthor relationships [58] (abbreviated as Physics and CS).
- Ogbn-Arxiv. This dataset is a directed citation network of all Computer Science arXiv papers indexed by the Microsoft Academic Graph. Each node represents a paper, with a 128-dimensional feature vector derived from the averaged word embeddings of its title and abstract. The task is to predict the subject area of each paper [59] (abbreviated as Arxiv_O).

B. Baselines

To comprehensively and fairly evaluate the performance of HPoolGCL, we selected three categories of methods as baselines: Unsupervised learning, supervised learning, and contrastive learning.

Unsupervised learning baselines:

- DeepWalk [60]. A truncated random walk is adopted to learn neighborhood information for enhancing node representations.

- node2vec [61]. A random walk with bias is adopted to learn neighborhood information for enhancing node representations.

Supervised learning baselines:

- GCN [47]. It attempts to learn node representations with the classic graph convolutional networks.
- GAT [49]. It introduces the self-attention mechanism to GNNs to adaptively assign weights to neighboring nodes during aggregation in a data-driven manner.
- GraphSAGE [48]. It proposes a sampling approach that allows GNNs to adapt to large-scale graphs.

Contrastive learning baselines:

- DGI [14]. It aims to learn node representations by maximizing the MI between subgraph summary representations of central nodes and the graph-level representation.
- MVGRL [15]. It constructs views of a graph with diffusion kernel and subgraph sampling. Then, it learns to contrast node representations with global summary vector across the two views.
- GRACE [10]. It creates two augmented views of a graph by randomly perturbing nodes/edges and their features and learns node representations by pulling the representations of the same nodes in two augmented graphs closer.
- GCA [12]. An advanced version of GRACE, which proposes multiple augmentation schemes regarding the importance of nodes, edges and their features.
- HCL [28]. A hierarchical contrastive learning framework that performs multi-scale contrastive learning on views generated by a learnable pooling module and data augmentation.
- BGRL [24]. A negative-sampling-free GCL method that employs augmentation-based siamese networks while avoiding trivial solutions via asymmetric architecture and gradient decoupling.
- CCA-SSG [26]. A negative-sampling-free GCL method that proposes a loss function based on canonical correlation analysis and it does not rely on negative samples, and can naturally remove the complicated components.
- SSGE [62]. A recent negative-sampling-free GCL framework that models each node as a Gaussian distribution and minimizes the KL divergence between similar views.
- AFGRL [19]. An augmentation-free GCL method, leveraging the nodes that shared local structural and global semantic relationships as positive samples.
- iGCL [20]. An augmentation-free GCL method that proposes a invariant-discriminative loss to learn invariant and discriminative representations.
- T-GCSA [63]. It proposes a self-evolving adaptive augmented view generating scheme, and focuses on the inherent link between augmented views to learn effective node representations.
- ConGCL [64]. It proposes addressing augmentation inconsistency through semantic-structural context entailment and a consistency agreement loss under stochastic augmentations.

We further categorize the contrastive learning baselines into four groups: (1) early methods, including DGI, MVGRL, GRACE, GCA and HCL; (2) negative-sampling-free methods, including BGRL, CCA-SSG and SSGE; (3) augmentation-free methods, including AFGRL and iGCL; and (4) advanced augmentation methods, including T-GCSA and ConGCL. For all baselines, we report their official results if available, otherwise, we report the results obtained from their official codes when consistent with our evaluation protocol.

C. Experimental Protocol

TABLE III: Hyperparameter settings of HPoolGCL on the datasets.

	D	D^q	K	p	α	β	γ_p	τ
CS	2048	4096	2	0.7	2.0	1.0	400	0.5
Physics	2048	2048	2	0.5	2.0	1.0	100	0.3
Computers	2048	2048	2	0.5	1.0	2.0	8	0.3
Photo	2048	2048	2	0.8	2.2	2.5	10	0.1
WikiCS	1024	2048	2	0.7	1.2	2.0	0.0005	0.9
Arxiv_O	1024	1024	2	0.7	0.5	3.0	0.001	0.3

We evaluate HPoolGCL on three node-level tasks, i.e., node classification, node clustering and node similarity search. Following AFGRL, we train all models in an unsupervised manner. Specifically, we first train the graph encoder in an unsupervised manner. Then, the produced node representations are trained with a ℓ_2-regularized linear classifier without flowing any gradients back to the graph encoder. For node classification, we report the averaged performance over twenty random dataset divisions and model initializations for all datasets apart. For node clustering and similarity search, we perform evaluations on the learned representations at every epoch and report the best performance.

The encoder of the HPoolGCL can be an arbitrary GNN. To compare fairly with the baselines, we use the GCN as the encoder f_θ. Formally, the encoder architecture is defined as:

$$\mathbf{H}^{(l)} = \mathrm{GCN}^{(l)}(\mathbf{X}, \mathbf{A}) = \sigma(\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{W}^{(l)}), \quad (14)$$

where $\mathbf{H}^{(l)}$ is the node embedding matrix of the l-th layer for $l \in [1, ..., L]$, $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops, $\hat{\mathbf{D}} = \sum_i \hat{\mathbf{A}}_i$ is the degree matrix, $\sigma(\cdot)$ is a nonlinear activation function such as ReLU, and $\mathbf{W}^{(l)}$ is the trainable weight matrix for the l-th layer.

All experiments are conducted on an NVIDIA RTX 3090 GPU with 24GB GPU memory and using the PyTorch framework. We perform grid-search on several hyperparameters, such as node embedding dimension size D, hidden size of projectors D^q, number of pooling layers K, pooling ratio p, and temperature τ. The trade-off hyperparameter λ in HiCR loss was fixed to 1 for all experiments. This choice was based on our preliminary sensitivity study, which showed that performance was robust around $\lambda = 1$, in contrast to the high sensitivity of γ_p. Fixing λ allowed us to simplify the complex hyperparameter search while maintaining strong performance. We acknowledge that extensive, dataset-specific tuning of λ remains a potential direction for future work. The optimal hyperparameter configuration of HPoolGCL is shown in Table III.

TABLE IV: Summary of statistical results in terms of mean test set classification accuracies (in percent) and standard deviation on 5 node classification benchmark datasets. **Bold** numbers indicate best results.

		CS	Physics	Computers	Photo	WikiCS
Unsupervised	Raw Feats.	92.01 ± 0.16	93.62 ± 0.13	78.76 ± 0.75	86.23 ± 0.54	72.41 ± 0.58
	node2vec [61]	88.55 ± 0.26	91.85 ± 0.14	84.63 ± 0.41	89.68 ± 0.41	71.89 ± 0.65
	DeepWalk [60]	84.61 ± 0.22	91.77 ± 0.15	85.66 ± 0.08	89.44 ± 0.11	73.42 ± 0.48
Supervised	GCN [47]	92.55 ± 0.17	95.51 ± 0.11	88.38 ± 0.44	92.85 ± 0.38	76.78 ± 0.46
	GAT [49]	92.65 ± 0.51	94.35 ± 0.40	88.31 ± 0.64	92.36 ± 0.59	76.64 ± 0.51
	GraphSAGE [48]	92.90 ± 0.18	95.67 ± 0.13	88.56 ± 0.49	92.80 ± 0.36	76.74 ± 0.45
Contrastive	DGI [14]	92.28 ± 0.16	94.51 ± 0.52	87.45 ± 0.46	91.65 ± 0.32	74.12 ± 0.40
	MVGRL [15]	92.11 ± 0.12	95.33 ± 0.03	87.52 ± 0.11	91.74 ± 0.07	77.52 ± 0.14
	GRACE [10]	92.53 ± 0.11	95.26 ± 0.02	86.65 ± 0.25	92.45 ± 0.24	77.97 ± 0.63
	GCA [12]	92.84 ± 0.14	95.38 ± 0.05	87.85 ± 0.31	92.49 ± 0.33	77.84 ± 0.67
	HCL [28]†	91.70 ± 0.30	93.50 ± 0.40	83.40 ± 0.50	87.30 ± 0.40	-
	BGRL [24]	92.59 ± 0.14	95.48 ± 0.08	89.69 ± 0.37	92.82 ± 0.38	76.86 ± 0.74
	CCA-SSG [26]	93.01 ± 0.20	95.42 ± 0.09	88.76 ± 0.36	92.89 ± 0.28	76.53 ± 0.68
	SSGE [62]	93.46 ± 0.45	95.68 ± 0.15	89.05 ± 0.58	93.15 ± 0.34	79.18 ± 0.57
	AFGRL [19]	93.27 ± 0.17	95.69 ± 0.10	89.88 ± 0.40	93.22 ± 0.28	78.12 ± 0.52
	iGCL [20]	93.27 ± 0.21	95.72 ± 0.11	89.72 ± 0.35	92.86 ± 0.23	78.31 ± 0.57
	T-GCSA [63]	93.51 ± 0.15	95.81 ± 0.24	88.41 ± 0.20	93.37 ± 0.14	78.76 ± 0.13
	ConGCL [64]	93.28 ± 0.09	95.86 ± 0.11	88.98 ± 0.10	93.19 ± 0.13	79.34 ± 0.14
	HPoolGCL	**93.61 ± 0.18**	**95.89 ± 0.08**	**90.23 ± 0.26**	**93.52 ± 0.29**	**80.74 ± 0.39**

†: Results are reported from the original paper.

TABLE V: Performance on node clustering in terms of NMI and homogeneity.

		Raw Feats.	BGRL	AFGRL	iGCL	ConGCL	HPoolGCL
WikiCS	NMI	0.2948	0.3969	0.4132	0.4155	0.4089	**0.5077**
	Hom.	0.3077	0.4156	0.4307	0.4328	0.4255	**0.5198**
Computers	NMI	0.2443	0.5364	0.5520	0.5531	0.5473	**0.5729**
	Hom.	0.2649	0.5869	0.6040	0.6052	0.5981	**0.6232**
Photo	NMI	0.3174	0.6841	0.6563	0.6580	0.6501	**0.7061**
	Hom.	0.3251	0.7004	0.6743	0.6765	0.6693	**0.7149**
CS	NMI	0.6980	0.7732	0.7859	0.7845	0.7805	**0.7862**
	Hom.	0.7318	0.8041	**0.8161**	0.8143	0.8110	0.8085
Physics	NMI	0.5220	0.5568	**0.7289**	0.7270	0.7242	0.6991
	Hom.	0.5583	0.6018	0.7354	**0.7358**	0.7308	0.7143

TABLE VI: Performance on similarity search. (Sim@n: Average ratio among n nearest neighbors sharing the same label as the query node.)

		Raw Feats.	BGRL	AFGRL	iGCL	ConGCL	HPoolGCL
WikiCS	Sim@5	0.7383	0.7739	0.7811	0.7825	0.7753	**0.8043**
	Sim@10	0.7090	0.7617	0.7660	0.7671	0.7610	**0.7909**
Computers	Sim@5	0.6891	0.8947	0.8966	**0.8981**	0.8908	0.8976
	Sim@10	0.6550	0.8855	0.8890	0.8895	0.8831	**0.8897**
Photo	Sim@5	0.7719	0.9245	0.9236	0.9240	0.9211	**0.9279**
	Sim@10	0.7339	0.9195	0.9173	0.9181	0.9155	**0.9202**
CS	Sim@5	0.8913	0.9112	0.9180	**0.9183**	0.9133	0.9143
	Sim@10	0.8755	0.9086	0.9142	**0.9145**	0.9100	0.9066
Physics	Sim@5	0.9401	0.9504	0.9525	0.9520	0.9508	**0.9533**
	Sim@10	0.9319	0.9464	**0.9486**	0.9481	0.9467	0.9475

D. Comparison with Peer Methods

To evaluate the performance of HPoolGCL across various downstream tasks, we conduct comprehensive assessments on node classification, node clustering, and similarity search tasks. Table IV presents the node classification performance of various models on five benchmark datasets. The results demonstrate that HPoolGCL achieves the highest classification accuracy across all datasets, even surpassing supervised baselines in most cases. Table V and Table VI further illustrate HPoolGCL's superior performance in node clustering and similarity search tasks. Our detailed observations and analyses are summarized as follows.

From the reported experimental results, we summarize our overall analysis of different benchmarks as follows:

- Contrastive learning baselines, even early methods such as GRACE and GCA, consistently outperform traditional unsupervised baselines like DeepWalk and node2vec. Highlighting the superiority of contrastive learning.
- Negative-sampling-free methods such as CCA-SSG and BGRL, achieve comparable performance to previous approaches on most datasets. However, they exhibit significant degradation on the WikiCS dataset, with even the recent approache SSGE failing to achieve optimal performance. This degradation may stem from two factors: the absence of negative sampling weakens the model's ability to learn discriminative representations, and the reliance on random data augmentations introduces semantic distortions.
- Augmentation-free methods such as AFGRL and iGCL, mitigate model collapse by carefully selecting positive samples. The augmentation-free property prevents potential disruptions to the original semantic information, leading to improved performance compared to CCA-SSG and BGRL. However, like previous methods, AFGRL and iGCL are also negative-sampling-free, which results in suboptimal performance on the WikiCS dataset. This suggests that insufficient redundancy reduction may contribute to performance degradation.
- Recent advanced augmentation methods, such as T-GCSA and ConGCL, are designed with carefully crafted

data augmentation strategies tailored to graph structures, demonstrating significant advantages over other baselines on most benchmark datasets. Notably, these methods emphasize the relationships between augmented views, ensuring that enhanced views retain as much original graph information as possible by improving inter-view consistency.

For our proposed method, HPoolGCL, it surpasses all GCL methods in the node classification task while also demonstrating strong performance in node clustering and similarity search tasks. We present the following analysis:

- Compared to state-of-the-art methods that employ carefully designed augmentation strategies, HPoolGCL still exhibits performance improvements (On average, improves by 1.82% and 1.98% than T-GCSA respectively, and improves by 1.25% and 1.40% than ConGCL respectively on Computers and WikiCS datasets). These methods, such as T-GCSA and ConGCL, incorporate augmentation strategies tailored to graph data, leveraging structural properties to mitigate the semantic loss introduced by data augmentation. In contrast, graph pooling is inherently designed for graph data. When constructing coarse-grained views, HPoolGCL naturally incorporates structural properties. Furthermore, by introducing HiCR loss, HPoolGCL is able to construct the finest-grained view that preserve the original graph's semantics and multiple coarse-grained views that capture high-level semantic information.
- Compared to methods without negative sampling, HPoolGCL shows a significant performance improvement, especially on the WikiCS dataset (On average, outperforms CCA-SSG, BGRL, AFGRL, iGCL and SSGE by 4.21%, 3.88%, 2.62%, 2.43%, and 1.56%). This highlights the importance of learning distinguishable representations in GCL. In contrast, HPoolGCL leverages the selection effect of pooling, naturally treating discarded nodes during coarse-grained view construction as redundant. During the construction of coarse-grained views, discarded nodes are naturally treated as redundant. This allows HPoolGCL to learn discriminative representations without explicitly designing negative sampling strategies.
- For augmentation-free methods like AFGRL and iGCL, while they completely avoid the impact of data augmentation, their carefully designed positive sample sampling strategy may not applicable to all datasets (e.g., poor performance on the WikiCS dataset). Moreover, unlike HPoolGCL, which learns multi-granularity information, their contrastive learning is restricted within the same level, limiting their representational capacity.
- In node clustering (Table V) and similarity search (Table VI), HPoolGCL demonstrates superior performance against most baselines, including recent methods. A particularly insightful comparison is with the augmentation-free methods, AFGRL and iGCL. Both methods discard random augmentations and instead construct positive samples by searching for semantically similar nodes in the representation space. This design principle aligns exceptionally well with the nature of node clustering and similarity search, resulting in their strong and highly competitive performance, occasionally surpassing other methods on specific metrics. Despite the inherent advantage of these methods on such tasks, HPoolGCL, which learns semantic invariance through cross-layer and multi-granularity representations, still achieves comparable or superior performance in the majority of cases. This is a strong testament to the effectiveness of learning multi-granularity information, as it allows our model to capture a richer semantic understanding that translates to robust overall performance.

TABLE VII: Node classification accuracy (%) on the Arxiv_O dataset. Best result is in **bold**.

Method	Accuracy (%)
Unsupervised Baselines	
Raw Feats.	55.60 ± 0.18
Node2Vec [61]	69.89 ± 0.16
Supervised Baselines	
GCN [47]	71.88 ± 0.28
GAT [49]	72.08 ± 0.16
Contrastive Baselines	
DGI [14]	70.34 ± 0.16
BGRL [24]	71.44 ± 0.12
CCA-SSG [26]	71.21 ± 0.20
SSGE [62]	71.62 ± 0.19
HPoolGCL	**71.87 ± 0.24**

Therefore, our proposed approach, HPoolGCL, which extracts multi-granularity information through graph pooling and naturally selects important and redundant nodes for cross-granularity contrastive learning, is both promising and effective.

E. Scalability on Large-Scale Graphs

To evaluate the scalability and robustness of HPoolGCL on larger and sparser graphs, as suggested by the reviewer, we conducted additional experiments on the Arxiv_O dataset [59].

As shown in Table VII, HPoolGCL achieves highly competitive performance. It outperforms all other self-supervised baselines and achieves results comparable to strong supervised methods. This success on a large and sparse graph validates that our augmentation-free, pooling-based framework is not only effective on smaller benchmarks but also a robust and scalable solution for real-world, large-scale graph representation learning.

F. Compatibility with Pooling Methods

As introduced in Section III-D, our framework treats the pooling process as a pluggable module for multi-granularity view construction. To validate the compatibility of HPoolGCL with different graph pooling strategies, in addition to SAGPool [40] (used in our main experiments), we implement the following pooling variants while keeping other components fixed:

- Random Pooling: Retains nodes randomly with a fixed pooling ratio, serving as a structure-agnostic baseline that ignores both topological relationships and node features.
- TopKPool: Selects nodes by ranking scores computed through linear transformation of raw node features.
- gPool [39]: The hierarchical pooling component from Graph U-Nets. It first applies graph convolution layers to learn node embeddings, then selects nodes with strongest projections onto a trainable direction vector in the embedding space. This allows retention of semantically meaningful substructures.
- MVPool [42]: The downsampling module of the MVPool architecture. It computes node rankings through coordinated assessments from multiple structural perspectives (local connectivity, feature similarity, etc.), then harmonizes these views via attention-based fusion to determine optimal node subsets.

TABLE VIII: Performance Comparison with Different Pooling Strategies in HPoolGCL Framework. "+ HiCR" denotes adding the proposed HiCR loss.

	CS	Physics	Computers	Photo	WikiCS
Random	93.23	95.22	88.74	92.55	78.67
TopKPool	93.37	95.60	89.77	93.17	79.81
+ HiCR	93.53	95.72	89.92	93.35	79.84
gPool	93.35	95.62	89.86	93.01	80.30
+ HiCR	93.43	95.77	89.93	93.19	80.54
MVPool	93.52	OOM	89.91	93.27	80.34
+ HiCR	**93.63**	OOM	90.02	93.34	**80.78**
HPoolGCL (SAGPool)	93.61	**95.89**	**90.23**	**93.52**	80.74

The results are presented in Table VIII. Our findings can be summarized as follows. First, our framework demonstrates strong compatibility with diverse pooling strategies. Second, we observe that adding HiCR loss consistently improves performance. While these marginal gains may appear modest, it is crucial to understand the synergistic roles of our two losses. MGC loss itself provides an implicit guidance for pooling by requiring coarse-grained views to be predictive of the original graph. HiCR loss then acts as a complementary, explicit regularizer, providing a direct, information-theoretic objective to ensure the quality of the pooled graphs. The essential contribution of this explicit guidance is validated in our Ablation Study (Section IV-G), where the *w/o HiCR* variant (relying only on implicit guidance) suffers a performance drop. Therefore, it is the synergy between MGC's implicit task-driven guidance and HiCR's explicit structural regularization that enables the state-of-the-art performance of our framework. Finally, while sophisticated methods like MVPool show strong results, their prohibitive computational costs motivate our pragmatic choice of SAGPool, which strikes a compelling balance between performance and efficiency.

G. Ablation Study

To validate the benefits of each component in HPoolGCL, we implement the following variants.

- *w/o MGC*: A variant without multi-granularity contrast. This setting is equivalent to a single-granularity contrast baseline, as it only uses Eq. (3) for contrastive learning at the finest granularity.
- *w/o RN*: A variant that exclusively treats preserved nodes after pooling as positive samples while ignoring discarded redundant nodes. Specifically, we replace the cross-granularity contrastive loss in Eq. (4) with Eq. (3).
- *w/o SE*: A variant that disregards the node selection effect during pooling. Instead of distinguishing between preserved and discarded nodes, it adopts a traditional instance discrimination strategy by treating all nodes other than the positive sample as negative samples [65].
- *w/o HiCR*: A variant that removes HiCR loss. This allows us to directly assess the contribution of our proposed unsupervised pooling objective.

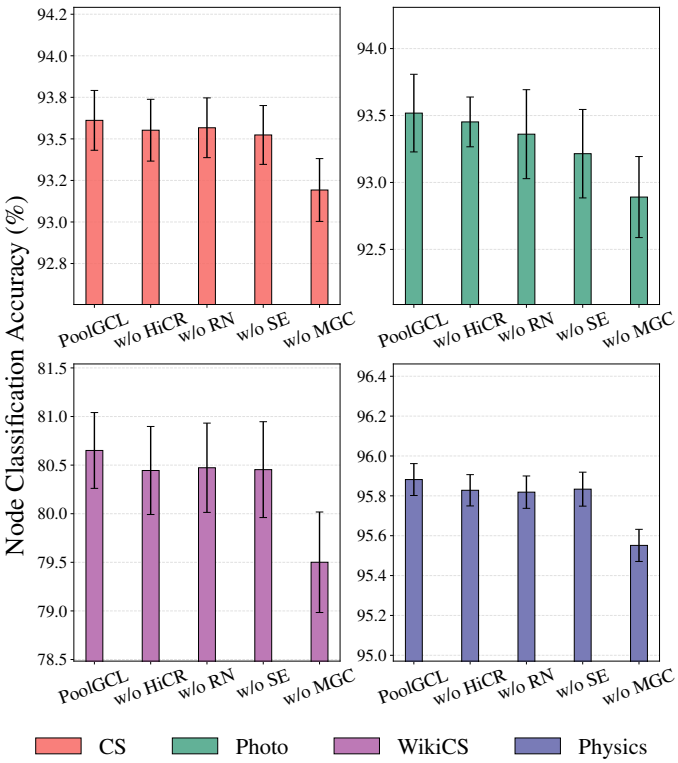


Fig. 4: Ablation study on HPoolGCL.

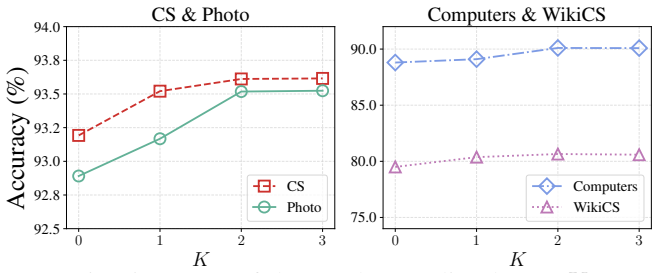


Fig. 5: Impact of the number pooling layers K.

From Fig. 4, we observe that all four simplified variants exhibit lower node classification accuracy than the full HPoolGCL, indicating that the absence of any critical component degrades performance.

The significant performance drop in *w/o MGC* validates the necessity of cross-hierarchy semantic alignment. Notably, the

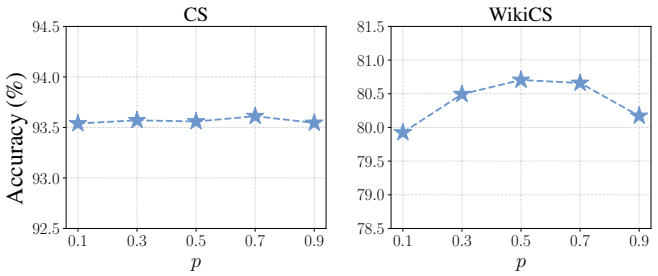


Fig. 6: Impact of the pooling ratio p.

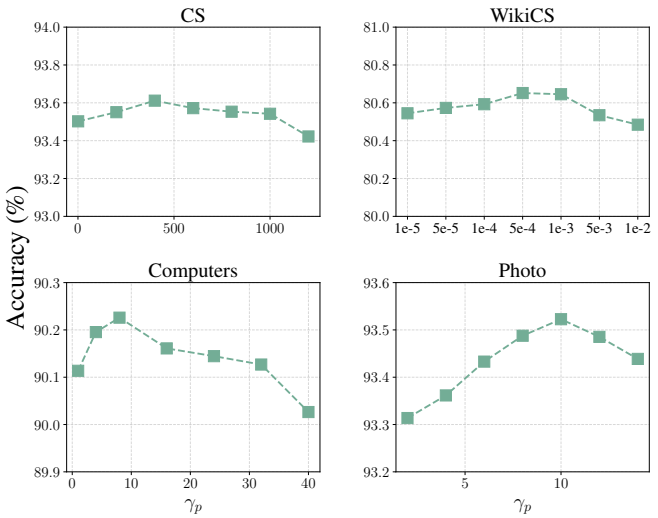


Fig. 7: Impact of HiCR loss weight γ_p.

model does not rapidly collapse to trivial solutions in this case, which aligns with the conclusion in SimSiam [46] that the stop-gradient operation alone suffices to prevent collapse without relying on the EMA mechanism. In HPoolGCL, this is implemented by simply using node representations from the previous iteration as target embeddings, requiring only a single GCN encoder.

When the influence of redundant nodes is ignored (*w/o RN*), the model still outperforms the *w/o MGC* variant, indicating that acquiring information from multi-granularity views is beneficial for GCL. However, it does not fully consider the selection effect introduced by the pooling process, focusing only on the retained nodes during contrastive learning, which leads to lower performance compared to HPoolGCL.

The *w/o SE* variant further validates the importance of modeling the selection effect. Although it introduces negative samples via instance discrimination and slightly outperforms *w/o RN* on some datasets, it fails to utilize explicit retain/discard signals from pooling. As a result, its performance remains suboptimal. This result highlights that the selection effect in our framework provides clearer and more effective contrastive supervision than naively adding negatives.

The performance decline in *w/o HiCR* demonstrates the critical role of HiCR loss in ensuring semantically meaningful graph coarsening under unsupervised settings. Nevertheless, the *w/o HiCR* variant retains competitive performance, consistent with our analysis in Sec. III-F1, where cross-granularity contrastive learning implicitly guides the pooling process to preserve hierarchical information.

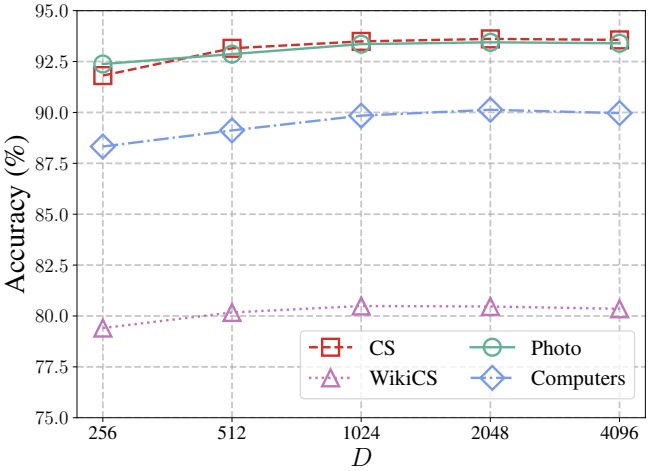


Fig. 8: Impact of the embedding dimension D.

H. Hyperparameter Analysis

In this section, we analyze the impacts of key hyperparameters in HPoolGCL: the number of pooling layers K, the pooling ratio p, HiCR loss weight γ_p, the node embedding dimension D and the granularity balance parameters α and β in MGC loss.

1) Impact of the number of pooling layers: As shown in Fig. 5, increasing the number of pooling layers generally improves performance, indicating that additional coarse-grained views enable richer cross-hierarchy knowledge learning. However, this also introduces higher computational overhead. A practical trade-off between accuracy and efficiency is achieved with $K = 2$, where most datasets attain optimal performance.

2) Impact of the pooling ratio: Different pooling ratio p determines the number of positive/negative samples. Larger p retains more critical nodes, while smaller p aggressively removes redundancies. As shown in Fig. 6 and Table III, the optimal value is highly dependent on the dataset's characteristics. For instance, on Computers, a dense co-purchase network with likely high clustering, performance peaks at a moderate $p = 0.5$ before declining sharply for higher values. This suggests that a relatively small core of nodes is sufficient to represent its well-defined communities, and retaining more nodes introduces noise that weakens the contrastive signal. In contrast to this sharp decline, the performance on the sparser CS network remains relatively stable across all ratios, achieving a slight peak at $p = 0.7$. This indicates that its structure is less sensitive to the pooling ratio. Similarly, WikiCS, a web network with overlapping topics, and Photo, a potentially more interconnected co-purchase graph, also benefit from a higher pooling ratio ($p = 0.7$ and $p = 0.8$, respectively) to capture their more diffuse structural information. These results confirm that while HPoolGCL is robust across a range of p, the optimal ratio is dependent on dataset-specific properties like density and community structure.

3) Impact of HiCR loss weight: HiCR loss weight γ_p controls the strength of the unsupervised pooling constraint, and its impact is highly sensitive and dataset-dependent, as

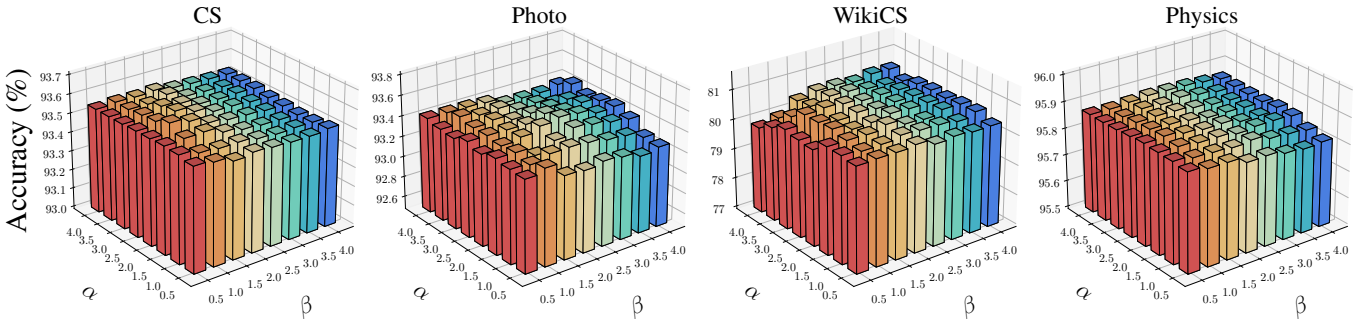
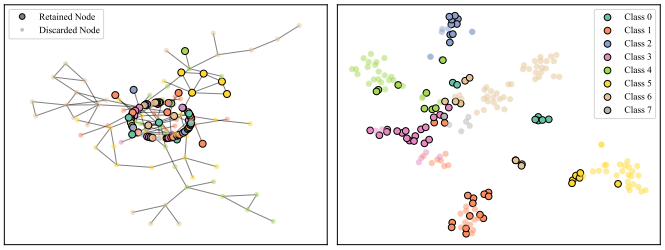


Fig. 9: Impact of the granularity balance parameters α and β.



(a) Pooling on Graph Topology (b) t-SNE of Node Embeddings

Fig. 10: Qualitative analysis of the pooling mechanism on a subgraph from the Photo dataset. (a) On the graph topology, retained nodes (large, solid) are structurally central within their communities (colors), while discarded nodes (small, transparent) are peripheral. (b) In the t-SNE embedding space, retained nodes form the core of their class clusters, whereas discarded nodes occupy the cluster boundaries.

shown in Fig. 7. As detailed in Table III, the optimal value of γ_p varies by several orders of magnitude across datasets, ranging from a large value for CS ($\gamma_p = 400$) to a very small one for WikiCS ($\gamma_p = 0.0005$). This significant variation suggests that the absolute scale of MGC and HiCR loss terms differs substantially depending on the graph's structure and feature distribution. For instance, a graph with clear community structures like CS may benefit from a strong regularization signal to guide the pooling process. In contrast, for a complex network with overlapping communities like WikiCS, the contrastive loss might be inherently harder to optimize, thus requiring only a minimal regularization weight to avoid overwhelming the primary learning objective. While a well-tuned γ_p consistently enhances performance by providing crucial guidance for unsupervised pooling, its high sensitivity is a notable characteristic of our method. Therefore, a careful grid search is essential to find the optimal balance for each specific dataset. The reference values in Table III can serve as a valuable starting point for this process.

4) Impact of the node embedding dimension: We investigate the impact of the node embedding dimension D. As shown in Fig. 8, HPoolGCL benefits from high-dimensional embeddings. When D further increases, the performance of HPoolGCL does not degrade significantly. This observation is consistent with BYOL, AFGRL, and iGCL, which also exhibit robustness to high-dimensional representations. A possible reason is that our method, similar to these approaches, does not adopt an instance discrimination-based contrastive strategy, which has been shown to be more susceptible to the curse of dimensionality [19].

5) Impact of the granularity balance parameters: As illustrated in Fig. 9, we observe distinct dataset-specific preferences for granularity balance: the CS dataset generally achieves better performance when emphasizing fine-grained features ($\alpha > \beta$), while the WikiCS dataset attains optimal performance when coarse-grained views are emphasized ($\alpha < \beta$). Overall, by learning cross-granularity knowledge, HPoolGCL consistently exhibits robustness and excellent performance within reasonable ranges of α and β.

I. Analysis of the Pooling Selection Mechanism

A core component of our framework is the learnable graph pooling mechanism, which implicitly defines positive (retained) and negative (discarded) samples for contrastive learning. To validate its effectiveness, we conducted both qualitative and quantitative analyses to understand which nodes are retained versus dropped.

Our qualitative analysis provides strong evidence that the unsupervised pooling mechanism learns to identify semantically meaningful nodes. Fig. 10 (a) shows that on the graph topology, retained nodes are typically structurally central within their communities. This pattern is mirrored in the feature space, as shown in Fig. 10 (b), where retained nodes form the dense core of their respective class clusters, while discarded nodes are scattered at the periphery. This confirms that the pooling layer effectively preserves nodes that are central in both structure and semantics.

This analysis also highlights the issue of "false negatives," which are the discarded peripheral nodes that may share the same class as retained core nodes. To investigate their impact, we conducted a quantitative analysis by measuring the average false negative rate (FNR) against model performance while varying the pooling ratio p, with results shown in Fig. 11. The analysis yields two key insights. First, the relationship between p and FNR is non-monotonic, often forming a U-shape, which reflects the dynamic nature of the learnable pooling. Second, and more importantly, we observe no direct correlation between a lower FNR and higher accuracy. For instance, on the CS dataset, peak performance occurs at $p = 0.7$, far from the FNR minimum at $p = 0.3$. This demonstrates

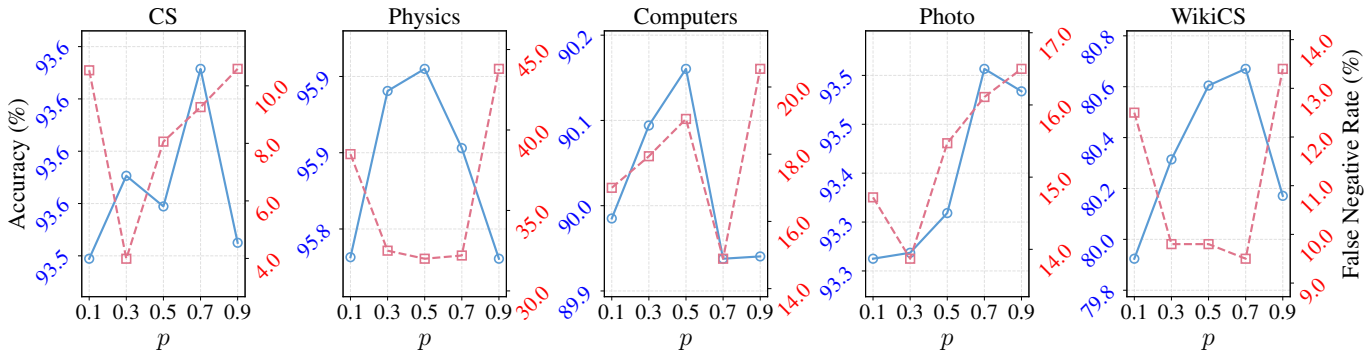


Fig. 11: Quantitative analysis of the average False Negative Rate (FNR) versus model accuracy across five datasets as a function of the pooling ratio p. The results show no direct correlation between lower FNR and higher accuracy, indicating a complex trade-off.

a complex trade-off between negative sample purity (low FNR), information preservation (high p), and negative set diversity (low p). Our model's strong performance suggests it effectively navigates this trade-off, benefiting from the known robustness of InfoNCE loss and potentially leveraging the contrast between core and peripheral same-class nodes as a form of beneficial regularization.

V. Conclusion

In this paper, we propose HPoolGCL, a cross-granularity GCL framework that leverages hierarchical graph pooling to learn multi-granularity representations. By integrating graph pooling into contrastive learning, HPoolGCL can capture hierarchical structural in complex graphs without relying on empirical data augmentation strategies that risk semantic distortion. Within the HPoolGCL framework, we introduce MGC loss, which naturally constructs contrastive pairs through the selection effect of pooling, and HiCR loss, which ensures unsupervised hierarchical pooling retains task-relevant information while reducing redundancies. Theoretical analysis demonstrates that our unified loss function aligns with the IB principle, balancing MI maximization and redundancy suppression. Through experiments on multiple graphs and various downstream tasks, the effectiveness of HPoolGCL is empirically verified. Furthermore, HPoolGCL's compatibility with diverse pooling strategies further enhance its practicality.

A. Limitations and Future Work

Our empirical studies were focused on static, homogeneous graphs. A primary direction for future work is extending HPoolGCL to more complex graph types. For heterogeneous graphs, this would involve developing relation-aware GNN encoders and pooling mechanisms, potentially extending HiCR loss to enforce consistency along specific meta-paths. For dynamic graphs, adapting the framework could involve exploring temporal pooling strategies and integrating our single-encoder design into a continual learning paradigm to handle evolving graph structures.

A second limitation pertains to the pooling mechanism itself. While our framework is compatible with various pooling methods, their effectiveness is inherently tied to the graph's topology. Our hierarchical approach is most beneficial for graphs with clear community structures and may offer diminished returns on highly random or regular "flat" graphs. This observation, coupled with the fact that most existing pooling methods are not optimized for unsupervised contrastive objectives, highlights a promising research avenue: the design of novel pooling operators specifically tailored for graph contrastive learning. Such methods could, for instance, be explicitly co-designed with the contrastive objective to maximize information gain while preserving essential structural invariants.

References

[1] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 1597–1607.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2019, pp. 4171–4186.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[7] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 12, pp. 9052–9071, Dec. 2024.

[8] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 10, 2021, pp. 8547–8555.

[9] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng, "Twin contrastive learning for online clustering," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2205–2221, 2022.

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

[11] C. Mavromatis and G. Karypis, "Graph infoclust: Maximizing coarse-grain mutual information in graphs," in *Advances in Knowledge Discovery and Data Mining: 25th Pacific-Asia Conference, PAKDD 2021, Virtual Event, May 11–14, 2021, Proceedings, Part I*, May 2021, pp. 541–553.

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

[13] J. Xia, L. Wu, J. Chen, B. Hu, and S. Z. Li, "Simgrace: A simple framework for graph contrastive learning without data augmentation," in *Proceedings of the ACM Web Conference 2022*, Apr. 2022, pp. 1070–1079.

[14] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *International Conference on Learning Representations*, Sep. 2018.

[15] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 4116–4126.

[16] Y. Zhang, Y. Chen, Z. Song, and I. King, "Contrastive cross-scale graph knowledge synergy," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2023, pp. 3422–3433.

[17] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 5812–5823.

[18] T. Zhao, W. Jin, Y. Liu, Y. Wang, G. Liu, S. Günnemann, N. Shah, and M. Jiang, "Graph data augmentation for graph machine learning: A survey," *arXiv preprint arXiv:2202.08871*, 2022.

[19] N. Lee, J. Lee, and C. Park, "Augmentation-free self-supervised learning on graphs," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 7372–7380, Jun. 2022.

[20] H. Li, J. Cao, J. Zhu, Q. Luo, S. He, and X. Wang, "Augmentation-free graph contrastive learning of invariant-discriminative representations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 8, pp. 11 157–11 167, Aug. 2024.

[21] H. Zhao, X. Yang, K. Wei, C. Deng, and D. Tao, "Unsupervised graph transformer with augmentation-free contrastive learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 7296–7307, Nov. 2024.

[22] Z.-F. Zhou, D. Huang, and C.-D. Wang, "Pyramid contrastive learning for clustering," *Neural Networks*, vol. 185, p. 107217, 2025.

[23] D. Huang, X. Deng, D.-H. Chen, Z. Wen, W. Sun, C.-D. Wang, and J.-H. Lai, "Deep clustering with hybrid-grained contrastive and discriminative learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 10, pp. 9472–9483, 2024.

[24] S. Thakoor, C. Tallec, M. G. Azar, R. Munos, P. Veličković, and M. Valko, "Bootstrapped representation learning on graphs," in *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, Mar. 2021.

[25] W. Sun, J. Li, L. Chen, B. Wu, Y. Bian, and Z. Zheng, "Rethinking and simplifying bootstrapped graph latents," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, Mar. 2024, pp. 665–673.

[26] H. Zhang, Q. Wu, J. Yan, D. Wipf, and P. S. Yu, "From canonical correlation analysis to self-supervised graph neural networks," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 76–89.

[27] B. Han, Y. Wei, Q. Wang, and S. Wan, "Colm2s: Contrastive self-supervised learning on attributed multiplex graph network with multi-scale information," *CAAI Transactions on Intelligence Technology*, vol. 8, no. 4, pp. 1464–1479, 2023.

[28] J. Wang, W. Li, C. Hou, X. Tang, Y. Qiao, R. Fang, P. Li, P. Gao, and G. Xie, "Hcl: Improving graph representation with hierarchical contrastive learning," in *International Semantic Web Conference*. Springer, 2022, pp. 108–124.

[29] R. Yang, Y. Yang, F. Zhou, and Q. Sun, "Directional diffusion models for graph representation learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 32 720–32 731, 2023.

[30] J. Zhu, L. He, C. Gao, D. Hou, Z. Su, P. S. Yu, J. Kurths, and F. Hellmann, "SDMG: Smoothing your diffusion models for powerful graph representation learning," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=lNyaQIJ5Z7

[31] Y. You, T. Chen, Y. Shen, and Z. Wang, "Graph contrastive learning automated," in *Proceedings of the 38th International Conference on Machine Learning*, Jul. 2021, pp. 12 121–12 132.

[32] X. Cai, C. Huang, L. Xia, and X. Ren, "Lightgcl: Simple yet effective graph contrastive learning for recommendation," in *The Eleventh International Conference on Learning Representations*, Sep. 2022.

[33] C. Xu, T. Wang, M. Chen, J. Chen, and Z. Pan, "Class-aware graph siamese representation learning," *Neurocomputing*, vol. 620, p. 129209, Mar. 2025.

[34] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.06391

[35] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[36] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," *Advances in neural information processing systems*, vol. 31, 2018.

[37] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International conference on machine learning*. PMLR, 2020, pp. 874–883.

[38] A. H. Khasahmadi, K. Hassani, P. Moradi, L. Lee, and Q. Morris, "Memory-based graph networks," in *International Conference on Learning Representations*.

[39] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning*. PMLR, 2019, pp. 2083–2092.

[40] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *International conference on machine learning*. pmlr, 2019, pp. 3734–3743.

[41] Z. Zhang, J. Bu, M. Ester, J. Zhang, C. Yao, Z. Yu, and C. Wang, "Hierarchical graph pooling with structure learning," *arXiv preprint arXiv:1911.05954*, 2019.

[42] Z. Zhang, J. Bu, M. Ester, J. Zhang, Z. Li, C. Yao, H. Dai, Z. Yu, and C. Wang, "Hierarchical multi-view graph pooling with structure learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 545–559, 2021.

[43] Z. Zhong, C.-T. Li, and J. Pang, "Multi-grained semantics-aware graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 7251–7262, 2022.

[44] M. Li, S. Chen, Y. Zhang, and I. Tsang, "Graph cross networks with vertex infomax pooling," *Advances in neural information processing systems*, vol. 33, pp. 14 093–14 105, 2020.

[45] Y. Pang, Y. Zhao, and D. Li, "Graph pooling via coarsened graph infomax," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 2177–2181.

[46] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[47] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[48] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[49] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

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

[51] Z. Su, Z. Hu, and Y. Li, "Hierarchical graph representation learning with local capsule pooling," in *Proceedings of the 3rd ACM International Conference on Multimedia in Asia*, 2021, pp. 1–7.

[52] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," Jun. 2021.

[53] H. B. Barlow *et al.*, "Possible principles underlying the transformation of sensory messages," *Sensory communication*, vol. 1, no. 01, pp. 217–233, 1961.

[54] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," Apr. 2000.

[55] J. Zhu, C. Gao, Z. Yin, X. Li, and J. Kurths, "Propagation structure-aware graph transformer for robust and interpretable fake news detection," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 4652–4663.

[56] P. Mernyei and C. Cangea, "Wiki-cs: A wikipedia-based benchmark for graph neural networks," *arXiv preprint arXiv:2007.02901*, 2020.

[57] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 43–52.

[58] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. Hsu, and K. Wang, "An overview of microsoft academic service (mas) and applications," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 243–246.

[59] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *arXiv preprint arXiv:1811.05868*, 2018.

[60] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[61] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[62] Y. Liu, T. He, T. Zheng, and J. Zhao, "Negative-free self-supervised gaussian embedding of graphs," *Neural Networks*, vol. 181, p. 106846, 2025.

[63] J. Miao, F. Cao, M. Li, B. Yang, and H. Ye, "Triplet teaching graph contrastive networks with self-evolving adaptive augmentation," *Pattern Recognition*, vol. 142, p. 109687, 2023.

[64] W. Bu, X. Cao, Y. Zheng, and S. Pan, "Improving augmentation consistency for graph contrastive learning," *Pattern Recognition*, vol. 148, p. 110182, 2024.

[65] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.

APPENDIX

A. Derivation of the Connection to the InfoMax Principle

Here, we provide a more detailed derivation of how minimizing our MGC loss is equivalent to maximizing MI, following the original proof for the InfoNCE loss in [50].

Let us first specify the sampling process in our MGC loss at a single coarse-grained level k, $\mathcal{L}_{\text{MGC}_k}$ (Eq. (4)). For each of the N_k anchor representations $\{z^t_{k,n}\}$, there is one deterministic positive sample $h^{t-1,+}_{k,n}$ representing the corresponding node from the target view. The negative samples comprise a shared set of $\tilde{N}$ representations $\{h^{t-1,-}_{k,m}\}$ derived from all nodes discarded during the pooling process.

For the purpose of theoretical analysis, we model this process in a probabilistic framework, following [50]. We consider a given anchor z drawn from a distribution $p(\mathbf{Z}_k)$. Its positive sample h^+ is considered to be drawn from the conditional distribution $p(h|z)$, while the $\tilde{N}$ negative samples $\{h^-\}$ are considered to be drawn from the marginal, or proposal, distribution $p(h)$.

The InfoNCE loss for a single anchor $z^t_{k,n}$ is given by:

$$\mathcal{L}_n = -\log \frac{e^{s^+_{k,n}/\tau}}{e^{s^+_{k,n}/\tau} + \sum_{m=1}^{\tilde{N}} e^{s^-_{k,n,m}/\tau}} \tag{15}$$

where $s^+_{k,n} = \text{sim}(z^t_{k,n}, h^{t-1,+}_{k,n})$ and $s^-_{k,n,m} = \text{sim}(z^t_{k,n}, h^{t-1,-}_{k,m})$ are the similarity scores, consistent with Eq. (4). This loss is minimized by training the encoder f_θ and the projector q.

As shown in [50], assuming the model has sufficient capacity, the optimal scoring function that our model's s/τ aims to approximate, let's denote it as $s^*(z,h)/\tau$, is proportional to the log-density ratio:

$$s^*(z,h)/\tau \propto \log \frac{p(h|z)}{p(h)}. \tag{16}$$

This connects our loss to the mutual information $\mathcal{I}(\mathbf{Z}_k; \mathbf{H}^{t-1}_{\theta'})$, which is defined as the expected value of this log-density ratio:

$$\mathcal{I}(\mathbf{Z}_k; \mathbf{H}^{t-1}_{\theta'}) = \mathbb{E}_{(z,h)\sim p(z,h)} \left[\log \frac{p(h|z)}{p(h)} \right]. \tag{17}$$

Following the derivation in the appendix of [50], by substituting this optimal scoring function back into the loss, the expected optimal loss $\mathbb{E}[\mathcal{L}^{\text{opt}}_n]$ can be bounded. Under the assumption that the $\tilde{N}$ negative samples are drawn independently from $p(h)$, the expectation of the loss is:

$$\mathbb{E}[\mathcal{L}^{\text{opt}}_n] \approx \mathbb{E}\left[-\log \frac{\frac{p(h^+|z)}{p(h^+)}}{\frac{p(h^+|z)}{p(h^+)} + \sum_{j=1}^{\tilde{N}} \frac{p(h^-_j|z)}{p(h^-_j)}} \right] \tag{18}$$

$$\leq -\mathbb{E}\left[\log \frac{\frac{p(h^+|z)}{p(h^+)}}{\frac{p(h^+|z)}{p(h^+)} + \tilde{N}} \right] \quad \text{(by Jensen's inequality)} \tag{19}$$

$$= \mathbb{E}\left[\log \left(1 + \tilde{N} \frac{p(h^+)}{p(h^+|z)} \right) \right] \tag{20}$$

$$\approx -\mathcal{I}(\mathbf{Z}_k; \mathbf{H}^{t-1}_{\theta'}) + \log(\tilde{N}). \tag{21}$$

Rearranging this gives us the well-known lower bound on mutual information:

$$\mathcal{I}(\mathbf{Z}_k; \mathbf{H}_{\theta'}^{t-1}) \gtrsim \log(\tilde{N}) - \mathbb{E}[\mathcal{L}_n]. \tag{22}$$

Since our trained model aims to minimize the loss, and $\mathcal{L}_{\text{MGC}_k}$ is the average of $\mathcal{L}_n$ over all anchors, minimizing $\mathcal{L}_{\text{MGC}_k}$ is equivalent to maximizing this lower bound on the mutual information between the projected representations $\mathbf{Z}_k$ and the target representations $\mathbf{H}_{\theta'}^{t-1}$.

Furthermore, the data processing inequality states that information cannot be created by post-processing. Since the projected representations $\mathbf{Z}_k$ are a function of the encoder's output $\mathbf{H}_k^t$ (i.e., $\mathbf{Z}_k = q_k(\mathbf{H}_k^t)$), we have $\mathcal{I}(\mathbf{H}_k^t, \mathbf{H}_{\theta'}^{t-1}) \geq \mathcal{I}(\mathbf{Z}_k, \mathbf{H}_{\theta'}^{t-1})$. This means that by maximizing the MI for the projected features, we are optimizing a tractable lower bound for the MI of the source features themselves.

Therefore, we can conclude that the optimization of our MGC loss at each layer drives the maximization of MI between the learned hierarchical representations and the target:

$$\min_{f_\theta^t, q_k^t} \mathcal{L}_{\text{MGC}_k} \Rightarrow \max_{f_\theta^t} \mathcal{I}(\mathbf{H}_k^t, \mathbf{H}_{\theta'}^{t-1}). \tag{23}$$

A similar, though less direct, argument applies to $\mathcal{L}_{\text{MGC}_0}$. As our total loss $\mathcal{L}_{\text{MGC}}$ is a weighted sum of these individual losses, the overall optimization process aims to maximize the sum of mutual information across all levels, as conceptually expressed in Eq. (9).

B. Alternative Formulation of the IB Objective

In the main text (Section III-F3), we established the connection between our unified loss and the Information Bottleneck (IB) principle by mapping its components to the prediction and compression terms. Here, we provide an alternative perspective by directly reparameterizing the loss function in terms of mutual information (MI) under certain assumptions.

Our analysis begins with the loss function for the k-th coarse-grained graph:

$$\begin{aligned}\mathcal{L}_k &= \mathcal{L}_{\text{MGC}_k} + \gamma_p \mathcal{L}_{\text{HiCR}_k} \\ &= \underbrace{-\frac{1}{N_k}\sum_{n=1}^{N_k} \log \frac{e^{s_{k,n}^+/\tau}}{e^{s_{k,n}^+/\tau} + \sum_{m=1}^{\tilde{N}} e^{s_{k,n,m}^-/\tau}}}_{\text{contrastive loss}} \\ &+ \gamma_p \underbrace{\|\text{diag}(C_{k-1,k-1}) - \text{diag}(C_{k,k})\|_2^2}_{\text{consistency term}} \\ &+ \quad \gamma_p \lambda \quad \underbrace{\|C_{k,k-1}\|_F^2}_{\text{redundancy reduction term}}\end{aligned} \tag{24}$$

We can analyze the three components of $\mathcal{L}_k$ from an information-theoretic viewpoint:

- Contrastive loss ($\mathcal{L}_{\text{MGC}_k}$)**:** As established in Appendix A, this term maximizes the MI between the coarse-grained representations $\mathbf{H}_k^t$ and the target representations $\mathbf{H}_{\theta'}^{t-1}$, aligning with the prediction term $\mathcal{I}(\mathbf{Y}; \mathbf{Z})$ of the IB objective.
- Redundancy reduction term in HiCR loss: Following the insight from Barlow Twins [52], penalizing the cross-correlation between representations serves as a proxy for minimizing their mutual information under Gaussian assumptions. Thus, this term aligns with the compression term $\mathcal{I}(\mathbf{X}; \mathbf{Z})$ of the IB objective, encouraging the compression of irrelevant information from the input layer $\mathbf{H}_{k-1}^t$.
- Consistency term in HiCR loss: This term acts as a regularizer. By encouraging the variance of representations to be preserved during pooling, it ensures that the compression process does not lead to catastrophic information loss.

Based on this analysis, and assuming the pooling layer effectively preserves critical information (causing the consistency term to approach zero), the optimization of the combined loss can be conceptually linked to the IB objective. MGC loss drives the maximization of MI with the target, while the redundancy reduction term drives the minimization of MI with the input. This allows us to express the optimization of $\mathcal{L}_k$ in the language of MI:

$$\min_{f_\theta^t} \mathcal{L}_k \Rightarrow \max_{f_\theta^t} \left(\mathcal{I}(\mathbf{H}_{\theta'}^{t-1}; \mathbf{H}_k^t) - \gamma_p' \mathcal{I}(\mathbf{H}_{k-1}^t; \mathbf{H}_k^t) \right), \tag{25}$$

where γ_p' is an effective trade-off parameter related to γ_p and λ.

Extending this to the total loss across all layers (Eq. (8)), we can reparameterize the entire optimization process in the IB framework as:

$$\begin{aligned}\min_{f_\theta^t} \mathcal{L} \Rightarrow \max_{f_\theta^t} \Bigg(&\sum_{k=0}^{K} \mathcal{I}(\mathbf{H}_{\theta'}^{t-1}; \mathbf{H}_k^t) \\ &- \eta \sum_{k=1}^{K} \mathcal{I}(\mathbf{H}_{k-1}^t; \mathbf{H}_k^t) \Bigg),\end{aligned} \tag{26}$$

where the hyperparameter η (related to $\gamma_p \lambda$) directly corresponds to the trade-off parameter in our hierarchical IB objective (Eq. (13)). This reformulation demonstrates that our objective can be viewed as an instantiation of the IB principle for hierarchical self-supervised learning.

C. Analysis of Loss Dynamics

To provide a deeper insight into the optimization process and the sensitivity of HiCR loss weight γ_p, we visualize the training dynamics of $\mathcal{L}_{\text{MGC}}$ and $\mathcal{L}_{\text{HiCR}}$ across four representative datasets in Fig. 12. The $\mathcal{L}_{\text{HiCR}}$ term is plotted on a logarithmic scale to reveal its fine-grained behavior.

The comparison reveals two distinct optimization patterns corresponding to different graph types. On the academic networks (CS and Physics), the optimization is remarkably stable, indicating a synergistic convergence on these well-structured graphs. In stark contrast, the dynamics on the co-purchase networks (Computers and Photo) are significantly more volatile, illustrating a dynamic "tug-of-war" on a more rugged loss landscape. The fact that HPoolGCL achieves strong performance in both scenarios is evidence of its robustness and adaptability to diverse graph topologies.

These curves also explain the significant variation in the optimal HiCR loss weight, γ_p. The role of γ_p is to balance the contributions of $\mathcal{L}_{\text{MGC}}$ and $\mathcal{L}_{\text{HiCR}}$ in the total loss. As the figure

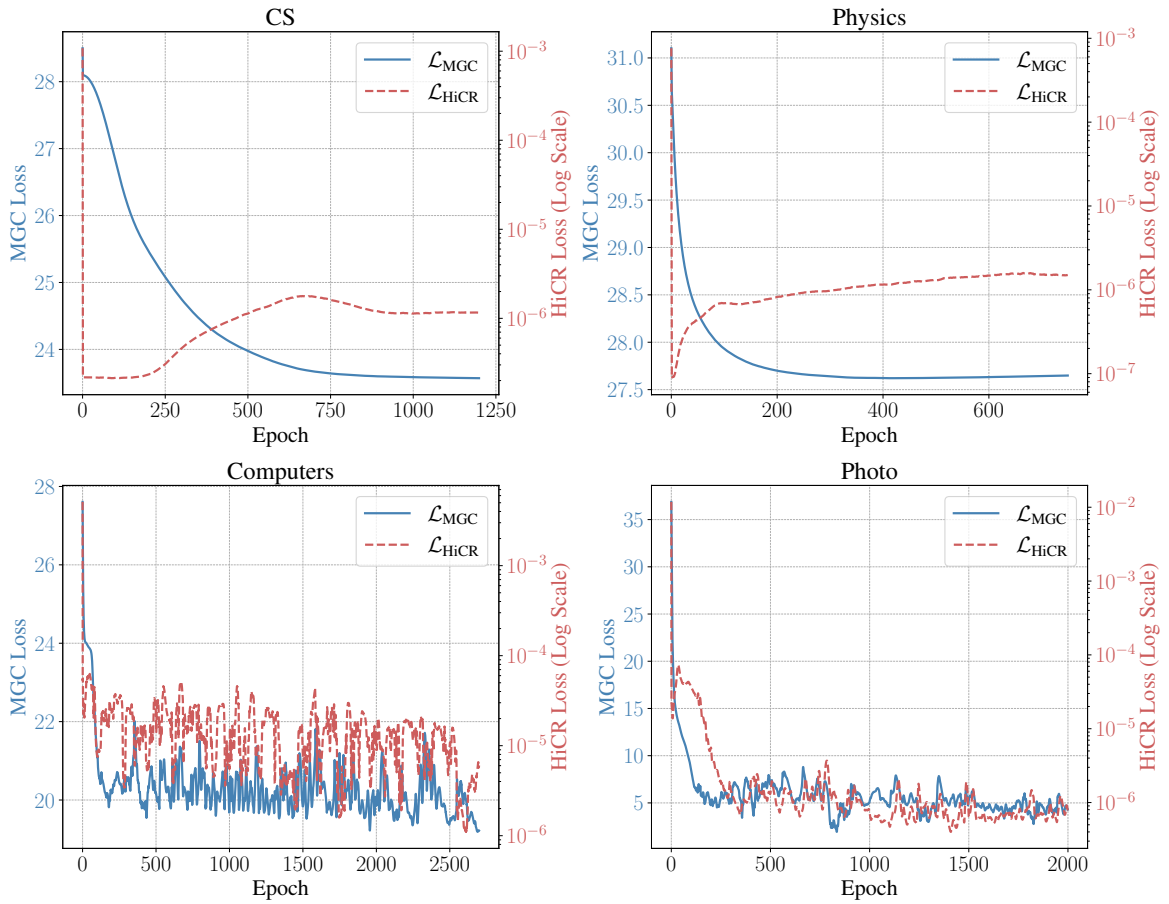


Fig. 12: Training loss dynamics of $\mathcal{L}_{\text{MGC}}$ (blue, left axis) and $\mathcal{L}_{\text{HiCR}}$ (red, right log-scale axis) on four datasets. The dynamics reveal two distinct optimization patterns: a stable, synergistic convergence on academic networks (CS, Physics) and a more volatile, adaptive interplay on co-purchase networks (Computers, Photo).

shows, the absolute magnitudes of these two losses differ substantially across datasets. For instance, on CS, the very small $\mathcal{L}_{\text{HiCR}}$ (around 10^{-6}) requires a very large γ_p (400) to make its contribution comparable to the large $\mathcal{L}_{\text{MGC}}$, which is in the interval (24, 28). Conversely, on Computers and Photo, where the loss magnitudes are closer, a moderate γ_p (8-10) is sufficient. This confirms that the optimal γ_p is not arbitrary but a necessary scaling factor dependent on the graph's structure. Therefore, a careful grid search for γ_p is essential for new datasets.

D. Efficiency Analysis

To evaluate the practical utility and computational cost of our method, we conducted an efficiency analysis comparing HPoolGCL with key baselines. We report the average training time per epoch and the peak GPU memory usage on five benchmark datasets. All metrics were benchmarked on a single NVIDIA RTX 3090 GPU, and the results are summarized in Table IX.

The analysis reveals a clear trade-off between computational overhead and the representational power gained from hierarchical learning. Regarding training time, HPoolGCL demonstrates remarkable efficiency. It is significantly faster per epoch than other advanced methods like AFGRL and ConGCL. The substantial speedup over AFGRL can be attributed to our avoidance of its computationally expensive k-NN search for positive pair mining. Similarly, HPoolGCL is orders of magnitude faster than ConGCL, which relies on time-consuming subgraph sampling and complex distributional alignment losses designed to mitigate the very augmentation inconsistencies that our method avoids by design. This highlights that our pooling-based approach to generating views is computationally much more efficient than both alternative information capture strategies (like AFGRL's k-NN) and augmentation-repairing strategies (like ConGCL).

In terms of memory usage, HPoolGCL occupies a moderate footprint. While it consumes more memory than simpler same-granularity methods like GRACE and BGRL due to storing multi-level representations, it is considerably more memory-efficient than ConGCL. In summary, HPoolGCL embodies a compelling design trade-off, achieving state-of-the-art performance with a highly efficient computational throughput and a moderate and acceptable increase in memory resources.

TABLE IX: Efficiency comparison on five benchmark datasets. We report the average training time per epoch (s) and peak GPU memory usage (GB). All metrics were measured on an NVIDIA RTX 3090 GPU. The symbol '-' indicates an Out-of-Memory (OOM) error on the 24GB GPU.

	CS		Physics		Computers		Photo		WikiCS	
	Time	Mem	Time	Mem	Time	Mem	Time	Mem	Time	Mem
GRACE	0.49	11.63	-	-	0.35	6.07	0.30	2.14	0.50	4.46
BGRL	0.18	2.64	0.20	4.77	0.13	0.96	0.09	0.61	0.14	1.88
CCA-SSG	1.83	2.06	3.91	7.59	0.64	1.69	0.54	1.29	1.15	2.66
AFGRL	9.44	6.02	-	-	3.98	2.70	2.22	1.45	6.59	6.82
ConGCL	-	-	50.54	21.67	47.53	18.37	20.92	7.34	52.22	16.91
HPoolGCL	0.68	8.34	1.05	12.80	0.34	8.34	0.26	4.83	0.43	8.26