Dual-Kernel Graph Community Contrastive Learning

Xiang Chen[1,2], **Kun Yue**[1,2], **Wenjie Liu**[1,2], **Zhenyu Zhang**[3], **Liang Duan**[1,2*]

[1]School of Information Science and Engineering, Yunnan University, Kunming, China
[2]Yunnan Key Laboratory of Intelligent Systems and Computing, Yunnan University, Kunming, China
[3]College of Artificial Intelligence, Tianjin University of Science and Technology, Tianjin, China
chenx@stu.ynu.edu.cn, duanl@ynu.edu.cn

Abstract

Graph Contrastive Learning (GCL) has emerged as a powerful paradigm for training Graph Neural Networks (GNNs) in the absence of task-specific labels. However, its scalability on large-scale graphs is hindered by the intensive message passing mechanism of GNN and the quadratic computational complexity of contrastive loss over positive and negative node pairs. To address these issues, we propose an efficient GCL framework that transforms the input graph into a compact network of interconnected node sets while preserving structural information across communities. We firstly introduce a kernelized graph community contrastive loss with linear complexity, enabling effective information transfer among node sets to capture hierarchical structural information of the graph. We then incorporate a knowledge distillation technique into the decoupled GNN architecture to accelerate inference while maintaining strong generalization performance. Extensive experiments on sixteen real-world datasets of varying scales demonstrate that our method outperforms state-of-the-art GCL baselines in both effectiveness and scalability.

Code — https://github.com/chenx-hi/DKGCCL
Extended version — https://arxiv.org/abs/2511.08287

Introduction

Graph neural networks (GNNs) learn effective node representations through message passing over graph structures, achieving impressive success across a wide range of graph analysis tasks (Wu et al. 2021). However, most GNN models are trained in a supervised way, and their performance is heavily dependent on the availability of labeled data. To address this limitation, graph contrastive learning (GCL) has emerged as a promising self-supervised approach for graph representation learning (Liu et al. 2023a). The core idea of GCL is to distinguish positive and negative node pairs using a contrastive loss grounded in mutual information maximization. This enables label-free training of GNN, achieving performance comparable to, or even surpassing, that of supervised methods (Chen et al. 2025a).

Despite significant progress in GCL, its application to large-scale graphs remains challenging due to two fundamental bottlenecks. The first is that *training scalability is limited by the quadratic computational complexity of pairwise comparisons in both GNN and contrastive loss*. The second is that *inference inefficiency stems from the intensive message-passing mechanism inherent in GNN architectures*. Most existing methods address these challenges in isolation, lacking a unified framework that improves both scalability and efficiency. For instance, some methods simplify training by eliminating the need for negative sampling (Huang et al. 2025) or reducing the number of augmented views processed by GNN (Mo et al. 2022), but they do not improve inference efficiency. Conversely, other methods achieve fast inference through decoupled GNN-MLP architectures (Xiao et al. 2024), yet still incur high computational training cost due to the contrastive loss.

Recent advancements in graph representation learning have extended beyond the limitations of traditional node-level message passing to reduce the complexity of GNN during training. These methods typically partition the graph into multiple communities, treating each as a single node to yield a coarsened graph. Subsequently, message passing (Chiang et al. 2019) or Graph Transformer attention (Xing et al. 2024) is applied to the coarsened graph, enabling the model to capture long-range dependencies and significantly reduce computational complexity. Notably, this graph coarsening technique has been successfully applied to GCL (Zhang et al. 2024), where it simultaneously addresses scalability issues arising from both GNN and contrastive loss in training phase. However, such oversimplification may lead to excessively uniform node representations within communities, resulting in a loss of fine-grained node information.

In this work, instead of simplifying each community to a single node via coarsening techniques, we envision the input graph as a network of node sets interconnected across communities, which allows us to preserve essential node information during training (Huang et al. 2024). To mitigate the increased complexity arising from this design choice, we integrate Multiple Kernel Learning (MKL) (Celikkanat, Shen, and Malliaros 2022; M. Ghari and Shen 2022) into the graph contrastive loss. By combining node-level and community-level kernels of different granularities, our proposed Dual-Kernel Graph Community Contrastive Learning (GCCL) effectively captures the hierarchical structural information of the graph (Duan et al. 2024). In the training process, we forgo explicit message passing and reduce the computational

*Corresponding author.

complexity of the contrastive loss from quadratic to linear time, thereby directly addressing the scalability challenge.

To improve inference efficiency on large-scale graphs while guaranteeing model performance, we propose a knowledge distillation module based on a decoupled GNN architecture. Specifically, we decouple the feature transformation and message passing steps of the GNN. The linear layer is trained within GCCL, while during inference, a parameter-free message passing operation is employed to propagate structural information across the graph. This decoupled paradigm introduces no additional training overhead on GCCL and improves generalization for downstream tasks. We then use the post-message-passing representations from the decoupled GNN as the distillation target, which enables us to extract a lightweight MLP model that captures graph structural information from the decoupled GNN, making our method suitable for latency-critical applications.

Our main contributions are summarized as follows:

- We propose a dual-kernel graph community contrastive loss by integrating multiple kernel learning, which improves the scalability of GCL training.
- We introduce a knowledge distillation module for decoupled GNN to effectively preserve graph structure information and enable low-latency inference.
- We provide theoretical analyses demonstrating that the kernelized graph community contrastive loss yields high-quality node representations for downstream tasks.
- Extensive experiments show that our method achieves state-of-the-art performance while significantly reducing the computational costs of both training and inference.

Preliminaries

Graph Neural Network. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote an undirected graph, where $\mathcal{V} = \{v_1, \cdots, v_n\}$ is the set of n nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathbf{X} \in \mathbb{R}^{n \times h}$ is the node feature matrix. The i-th row of $\mathbf{X}$ corresponds to the h-dimensional feature vector $\mathbf{x}_i$ of node v_i. The graph structure can be denoted by an adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$, where $\mathbf{A}_{i,j} = 1$ if and only if $(v_i, v_j) \in \mathcal{E}$. For simplicity, the undirected graph $\mathcal{G}$ can also be denoted as $G = (\mathbf{A}, \mathbf{X})$. Given G as input, the GNN encoder $f_\theta(G) : G \to \mathbb{R}^{n \times d}$ can produce effective representations $\mathbf{z}_i = f_\theta(G)[v_i]$ of node v_i:

$$f_\theta(G) = \sigma(\widetilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\mathbf{W}) \tag{1}$$

where $\mathbf{I}$ is an identity matrix, $\widetilde{\mathbf{D}}$ is a diagonal degree matrix of $\mathbf{D} + \mathbf{I}$, $\sigma(\cdot)$ is a non-linear activation function, and $\mathbf{W} \in \mathbb{R}^{h \times d}$ is a learnable parameter matrix corresponding to θ.

Graph Community Contrastive Learning. A typical GCL paradigm defines adjacent nodes as positive pairs and all other nodes in G as negative pairs to ensure the adjacent nodes have similar representations (Shen et al. 2023):

$$\mathcal{L}_G = -\frac{1}{n}\sum_{v_i \in \mathcal{V}} \frac{1}{\mathcal{N}(v_i)} \sum_{v_j \in \mathcal{N}(v_i)} \log \frac{\exp\left(\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_j/\tau\right)}{\sum_{v_k \in v_i^-} \exp(\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_k/\tau)} \tag{2}$$

where $\mathcal{N}(v_i)$ is the neighborhoods of node v_i, v_i^- is the negative node set and τ is the temperature hyper-parameter.

Let $\mathcal{P} = \{P_1, \cdots, P_m\}$ be a partition of G with m communities. Each community $P_j \in \mathcal{P}$ is a subset of $\mathcal{V}$, such that $\mathcal{V} = \bigcup_{j=1}^{m} P_j$ and $P_j \cap P_k = \emptyset$ for $j \neq k$. The partition assignment matrix denotes as $\mathbf{P} \in \mathbb{R}^{n \times m}$, where $\mathbf{P}_{i,j}$ is the weight of i-th node in the j-th community. The community-wise (coarsened) graph $\mathbf{A}^{\mathcal{P}}$ can be constructed by $\mathbf{P}^{\mathrm{T}}\mathbf{A}\mathbf{P}$, where $\mathbf{A}^{\mathcal{P}}_{j,k}$ is the connection weight between P_j and P_k.

In this work, we focus on leveraging community structure information to reconstruct the graph contrastive loss in Eq. 2, and refer to the resulting objective as the graph community contrastive loss (GCCL loss).

Multiple Kernel Learning. Kernel-based methods utilize an optimal kernel function $\kappa(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to measure pairwise similarity and have proven powerful for diverse tasks (Celikkanat, Shen, and Malliaros 2022). MKL methods integrate diverse features from different perspectives by combining multiple kernel functions (Liu 2022). The resulting multiple kernel κ_η is defined as:

$$\kappa_\eta(\{\mathbf{z}_i\}_{i=1}^l, \{\mathbf{y}_i\}_{i=1}^l) = g_\eta(\{\kappa_i(\mathbf{z}_i, \mathbf{y}_i)\}_{i=1}^l) \tag{3}$$

where $g_\eta : \mathbb{R}^l \to \mathbb{R}$ is a combinatorial function. In this work, we focus on the pairwise scenario, i.e., $l = 2$. Specially, given two kernels $\kappa_1 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $\kappa_2 : \mathcal{X}' \times \mathcal{X}' \to \mathbb{R}$, we consider two strategies of g_η: the tensor product of kernels and the convex linear of kernels (Gönen and Alpaydın 2011). The tensor product method is defined as:

$$\kappa_\eta((\mathbf{z}, \mathbf{z}'), (\mathbf{y}, \mathbf{y}')) = \kappa_1(\mathbf{z}, \mathbf{y}) \cdot \kappa_2(\mathbf{z}', \mathbf{y}') \tag{4}$$

where $(\mathbf{z}, \mathbf{z}'), (\mathbf{y}, \mathbf{y}')$ are pairs of objects from feature space $\mathcal{X} \times \mathcal{X}'$. The convex linear combination is defined as:

$$\kappa_\eta((\mathbf{z}, \mathbf{z}'), (\mathbf{y}, \mathbf{y}')) = \alpha\kappa_1(\mathbf{z}, \mathbf{y}) + \beta\kappa_2(\mathbf{z}', \mathbf{y}') \tag{5}$$

where $\alpha \in [0, 1]$ is a combination coefficient and $\beta = 1 - \alpha$.

Methodology

In this section, we present the proposed GCCL framework, illustrated in Figure 1. We first reduce the complexity of contrastive loss by leveraging community structure and MKL. We then design a knowledge distillation module for decoupled GNN to speedup inference.

Kernel-Based GCCL

Bi-Level Features Generation. The fundamental concept behind our method is to perceive G as interconnected communities of nodes (Huang et al. 2024). Within this paradigm, the community to which a node set belongs can serve as a bridge for information interaction, enhancing the information flow between its internal nodes and nodes in other communities. This enables our method to effectively capture the hierarchical information of the graph while preserving node-level details. Thus, node v_i in P_j can be characterized by a bi-level pair of features $\{\mathbf{v}_i, \mathbf{c}_j\} \in \mathcal{X}_G \times \mathcal{X}_P$:

$$\mathbf{v}_i = \mathbf{x}_i\mathbf{W}_G,\ \mathbf{c}_j = \sum\nolimits_{v_t \in P_j} \mathbf{P}_{t,j}^{\mathrm{T}} \mathrm{Dropout}\left(\mathbf{x}_t\mathbf{W}_P\right) \tag{6}$$

where $\mathbf{v}_i \in \mathcal{X}_G$ is a node-level feature and $\mathbf{c}_j \in \mathcal{X}_P$ is a community-level feature. $\mathbf{W}_G$ and $\mathbf{W}_P$ are two different

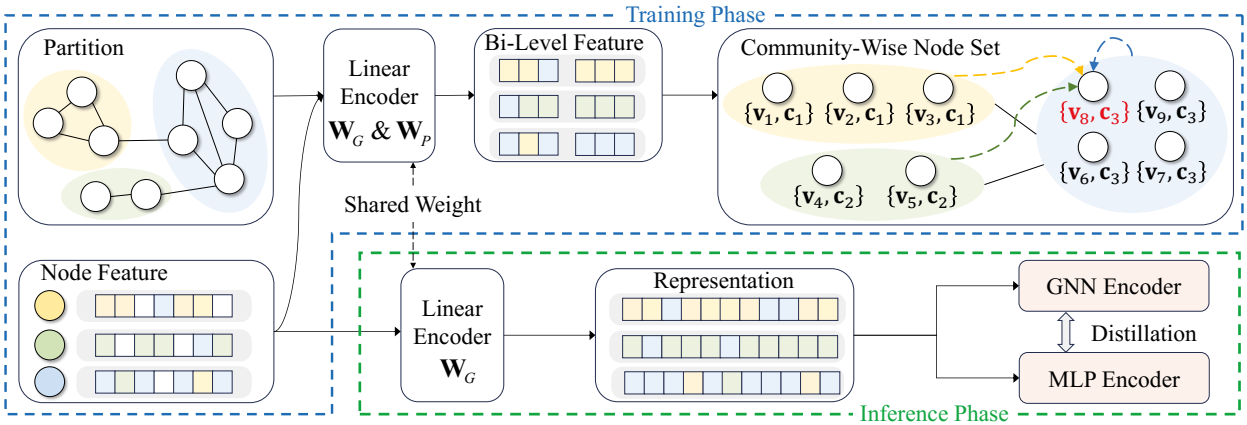

Figure 1: The overall framework of our method.

projection matrices to the node feature space $\mathcal{X}_G$ and the community feature space $\mathcal{X}_P$, respectively.

Note that we apply a random mask via Dropout$(\cdot)$ to all dimensions of $\mathbf{x}_t\mathbf{W}_P$ per training epoch to obtain $\mathbf{c}_j$. This method serves as a special data augmentation strategy, providing more diverse community-level features in $\mathcal{X}_P$ for community structure-based GCL. Specifically, we regard the construction of community-level features as a message passing process from nodes to community centroids. Based on the findings of Dropout$(\cdot)$ in the message passing mechanism of GNN (Luo, Wu, and Zhu 2025), we can derive the following proposition.

Proposition 1 *Let the feature dimension of the community-level feature space $\mathcal{X}_P$ be d^P. Then, the expected number of distinct partitioned substructures generated by the Dropout$(\cdot)$ operation for each partition P_j is:*

$$\mathbb{E}[|P_j^s| s = 1, \cdots, d^P|] = d^P \left(1 - (1-p)^{|P_j|}\right) \quad (7)$$

where P_j^s is a substructure of P_j on the feature dimension s, and p is the dropout probability.

Proposition 1 demonstrates that Dropout$(\cdot)$ generates a set of substructures (Luo, Wu, and Zhu 2025), whose quantity increases with both the dropout probability p and the dimension d^P. In subsequent experiments, we found that the diversity of substructures can reduce the training cycle of GCL.

Dual-Kernel GCCL Loss. After obtaining the bi-level features, we consider how to use a simple kernel trick to accelerate the computational process of GCL. The success of existing GCL methods lies in emphasizing the neighborhood similarity of node representations (Shen et al. 2023), a phenomenon also observed in coarsened graphs (Zhang et al. 2024), which aligns with the graph homophily assumption. This motivates us to treat the target node v_i and its interconnected node communities as positive pairs.

Definition 1 *Given a bi-level kernel $\kappa_B : \mathcal{X}_G \times \mathcal{X}_P \to \mathbb{R}_+$, the graph community contrastive loss can be expressed as:*

$$\mathcal{L}_{\mathcal{P}} = -\frac{1}{n}\sum\nolimits_{v_i \in \mathcal{V}} \log \ell(v_i, P_j), \quad \textit{where } \ell(v_i, P_j) = \frac{\sum_{P_k \in \mathcal{N}(P_j)} \mathbf{A}^{\mathcal{P}}_{j,k} \sum_{v_t \in P_k} \mathbf{P}_{t,k} \cdot \kappa_B(\{\mathbf{v}_i, \mathbf{c}_j\}, \{\mathbf{v}_t, \mathbf{c}_k\})}{\sum_{P_k \in \mathcal{P}} \sum_{v_t \in P_k} \mathbf{P}_{t,k} \cdot \kappa_B(\{\mathbf{v}_i, \mathbf{c}_j\}, \{\mathbf{v}_t, \mathbf{c}_k\})}$$

where P_j is the community to which v_i belongs, and $\mathcal{N}(P_j)$ is the set of communities connected to P_j in $\mathbf{A}^{\mathcal{P}}$.

In this definition, κ_B helps integrate information between nodes and communities, while $\mathbf{A}^{\mathcal{P}}_{j,k}$ can adjust the weight of positive pair based on the connectivity between P_j and P_k.

Let the non-negative kernel functions of feature spaces $\mathcal{X}_G$ and $\mathcal{X}_P$ be κ_G and κ_P, respectively, and we represent κ_G via its feature map as $\kappa_G(\mathbf{v}_i, \mathbf{v}_t) \approx \phi(\mathbf{v}_i)^{\mathrm{T}}\phi(\mathbf{v}_t)$. According to the tensor product of kernels (Eq.4), we have:

$$\kappa_B(\{\mathbf{v}_i, \mathbf{c}_j\}, \{\mathbf{v}_t, \mathbf{c}_k\}) = \kappa_P(\mathbf{c}_j, \mathbf{c}_k) \cdot \phi(\mathbf{v}_i)^{\mathrm{T}}\phi(\mathbf{v}_t)$$

Then, we can derive the variant of $\ell(v_i, P_j)$ as $\ell_{tp}(v_i, P_j)$.

Definition 2 *The dual-kernel GCCL loss with tensor product method $\ell_{tp}(v_i, P_j)$ can be formulated as:*

$$\frac{\phi(\mathbf{v}_i)^{\mathrm{T}}\left[\sum_{P_k \in \mathcal{N}(P_j)} \kappa'_P(\mathbf{c}_j, \mathbf{c}_k) \sum_{v_t \in P_k} \phi'(\mathbf{v}_t)\right]}{\phi(\mathbf{v}_i)^{\mathrm{T}}\left[\sum_{P_k \in \mathcal{P}} \kappa_P(\mathbf{c}_j, \mathbf{c}_k) \sum_{v_t \in P_k} \phi'(\mathbf{v}_t)\right]} \quad (8)$$

where the valid kernel $\kappa'_P(\mathbf{c}_j, \mathbf{c}_k) = \mathbf{A}^{\mathcal{P}}_{j,k} \cdot \kappa_P(\mathbf{c}_j, \mathbf{c}_k)$ and the feature map $\phi'(\mathbf{v}_t) = \mathbf{P}_{t,k} \cdot \phi(\mathbf{v}_t)$.

The tensor product method of MKL enables interactions across all dimensions of features at different granularity levels (Gönen and Alpaydın 2011), which naturally allows us to effectively capture the dependencies between node-level and community-level features in the combined feature space $\mathcal{X}_G \times \mathcal{X}_P$. Another key advantage of $\ell_{tp}(v_i, P_j)$ is that the summation term of negative pairs in the denominator is shared across all nodes, so it only needs to be calculated once and can be re-used for other nodes. The summation term of positive pairs in the numerator is shared among nodes within the same community. These properties avoid the quadratic computational complexity of node pairs in vanilla contrastive loss.

Next, we discuss another variant of $\ell(v_i, P_j)$. According to the linear combination of kernels (Eq.5), we have:

$$\kappa_B(\{\mathbf{v}_i, \mathbf{c}_j\}, \{\mathbf{v}_t, \mathbf{c}_k\}) = \alpha\phi(\mathbf{v}_i)^{\mathrm{T}}\phi(\mathbf{v}_t) + \beta\kappa_P(\mathbf{c}_j, \mathbf{c}_k)$$

Then, we can derive the variant of $\ell(v_i, P_j)$ as $\ell_{lc}(v_i, P_j)$.

Definition 3 *The dual-kernel GCCL loss with linear combination method $\ell_{lc}(v_i, P_j)$ can be formulated as:*

$$\frac{\sum_{P_k \in \mathcal{N}(P_j)} \left(\phi(\mathbf{v}_i)^{\mathrm{T}} \sum_{v_t \in P_k} \alpha\phi''(\mathbf{v}_t) + \beta\kappa'_P(\mathbf{c}_j, \mathbf{c}_k)\right)}{\phi(\mathbf{v}_i)^{\mathrm{T}} \left(\alpha \sum_{P_k \in \mathcal{P}} \sum_{v_t \in P_k} \phi'(\mathbf{v}_t)\right) + \beta \sum_{P_k \in \mathcal{P}} \kappa_P(\mathbf{c}_j, \mathbf{c}_k)} \tag{9}$$

where the feature map $\phi''(\mathbf{v}_t) = \mathbf{A}^{\mathcal{P}}_{j,k} \cdot \phi'(\mathbf{v}_t)$.

The convex linear combination of MKL provides the flexibility to combine the effects of features at different granularity levels (Gönen and Alpaydın 2011), allowing us to adjust the contribution of node-level and community-level information to the similarity metric of sample pairs via α. Similarly, the summation terms in the numerator and denominator of $\ell_{lc}(v_i, P_j)$ are shared among nodes within the same community and all nodes, respectively. Such a property enables our method to operate on large-scale graphs with fewer computational resources. We will discuss the applicability of variants $\ell_{tp}(v_i, P_j)$ and $\ell_{lc}(v_i, P_j)$ on different datasets in the experimental section.

In practice, we employ the simple graph partition algorithm Metis (Karypis and Kumar 1998) to generate a partition $\mathcal{P}$ and ensure the training efficiency of GCL. For the feature map $\phi(\mathbf{v})$, we use the sigmoid function to ensure that the similarity in the feature space $\mathcal{X}_G$ remains positive. The commonly used exponential dot product $\exp(\mathbf{c}_j^{\mathrm{T}}\mathbf{c}_k/\tau)$ in contrastive loss is adopted as κ_P. It should be noted that since the number of communities is much smaller than that of nodes, our method is of linear complexity.

Efficient Model Inference

Decoupled GNN Architecture. The Over-smoothing problem presents a critical challenge hindering the expressive power of GNN (Xing et al. 2024). Here, we investigate the impact of our community contrastive loss on node smoothness and illustrate the necessity of incorporating prior information about G. Without loss of generality, we take the node classification task as an example. In this task, each node is associated with a label for classification.

Proposition 2 *Let $\bar{\mathbf{A}}$ be the normalized adjacency matrix constructed from positive node pairs in the contrastive loss $\mathcal{L}_{\mathcal{P}}$ and $y(v)$ denote the label of v. Then, the bound of the smoothness between node embeddings is:*

$$\left\|\mathbf{V} - \bar{\mathbf{A}}\mathbf{V}\right\|_F \leq \sqrt{2}L \sum_{v_i \in \mathcal{V}} \frac{1}{1 + \frac{\varepsilon(v_i)\lambda_{v_i}}{(1-\varepsilon(v_i))\gamma_{v_i}}} \tag{10}$$

where $\lambda_{v_i} = \mathbb{E}_{v_t \in \mathcal{N}(v_i), y(v_i)=y(v_t)} \kappa_B(\{\mathbf{v}_i, \mathbf{c}_j\}, \{\mathbf{v}_t, \mathbf{c}_k\})$ and $\gamma_{v_i} = \mathbb{E}_{v_t \in \mathcal{N}(v_i), y(v_i)\neq y(v_t)} \kappa_B(\{\mathbf{v}_i, \mathbf{c}_j\}, \{\mathbf{v}_t, \mathbf{c}_k\})$. L is the Lipschitz constant, and $\varepsilon(v_i)$ is the one-hop homophily score of node v_i in $\bar{\mathbf{A}}$, defined as:

$$\varepsilon(v_i) = \frac{1}{|\mathcal{N}(v_i)|} \sum\nolimits_{v_t \in \mathcal{N}(v_i)} \mathbb{1}[y(v_i) = y(v_t)] \tag{11}$$

where $\mathbb{1}[\cdot]$ is the indicator function.

This proposition establishes a significant relationship between the smoothness of node embeddings and two key factors: the homophyily score $\varepsilon(\cdot)$ of positive pairs and the bi-level kernel κ_B. Notably, the smoothness is negatively correlated with $\varepsilon(\cdot)$. This indicates that the excessively expanded community structures can lead to over-smoothing, making node representations indistinguishable. This issue can be addressed by incorporating graph-level structural information as additional details, which complements community information to enhance the node representations.

$$\mathbf{Z}^* = \sigma(\mathbf{X}\mathbf{W}_G + 1/K \sum\nolimits_{k=1}^{K} \widetilde{\mathbf{A}}^k \mathbf{X}\mathbf{W}_G) \tag{12}$$

where K denotes capturing local information from the K-hops neighborhood of $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\widetilde{\mathbf{D}}^{-\frac{1}{2}}$.

Note that the standard GNN in Eq. 1 can be viewed as a model that tightly couples linear feature transformation with message passing ($\mathbf{Z} = \widetilde{\mathbf{A}}\mathbf{X}\mathbf{W}_G$), while the message passing in Eq. 12 occurs in the post-processing phase of GCL model training. This means that our method adopts a decoupled paradigm for GNN. Specifically, we first use a linear layer and incorporate community information into this linear transformation process ($\mathbf{V} = \mathbf{X}\mathbf{W}_G$) via our dual-kernel contrastive loss. Then, a training-free graph convolution operator is performed ($\mathbf{Z} = \widetilde{\mathbf{A}}^k\mathbf{V}$). This decoupled paradigm reduces the training burden of GCL and retains the powerful graph-level information processing ability of GNN.

Graph Representational Similarity Distillation. We adopt a knowledge distillation technique to avoid the significant computational overhead incurred by GNN during inference. Instead of using soft labels as in most previous works (Huo et al. 2023), we directly use the node representations after message passing as the distillation target to encourage the MLP to learn structural information:

$$\mathcal{L}_D = ||\mathrm{MLP}(\mathbf{X}\mathbf{W}_G) - 1/K \sum\nolimits_{k=1}^{K} \widetilde{\mathbf{A}}^k \mathbf{X}\mathbf{W}_G)||_F^2 \tag{13}$$

Thus, Eq. 12 can be rewritten as:

$$\mathbf{Z}^* = \sigma(\mathbf{X}\mathbf{W}_G + \mathrm{MLP}(\mathbf{X}\mathbf{W}_G)) \tag{14}$$

Notably, we use the node-level features output by the GCCL as input to the distillation model. Thus, the MLP can capture both community structure and positional information of G, which has been shown to be beneficial for graph representational similarity distillation (Tian et al. 2023).

Theoretical Analysis

We provide theoretical evidence to support the effectiveness of our dual-kernel GCCL loss and distillation loss.

Properties of Dual-Kernel GCCL Loss

First, we show that the dual-kernel GCCL loss can approximate the graph contrastive loss on a k-step graph diffusion matrix of Eq. 2.

Proposition 3 *Assuming the original features $\mathbf{X}$ and the mapped features $\phi(\mathbf{V})$ are bounded by $S_{\mathbf{X}} := \max_i ||\mathbf{X}_i||_2$ and $S_{\phi(\mathbf{V})} := \max_i ||\phi(\mathbf{V}_i)||_2^2$, respectively. Then, the original contrastive loss of the k-step diffusion graph $\mathbf{A}^k$, denoted as $\mathcal{L}_G$, can be approximated by the dual-kernel community contrastive loss, $\mathcal{L}^{lc}_{\mathcal{P}}$, without considering the influence of combination coefficients:*

$$|\mathcal{L}_G - \mathcal{L}^{lc}_{\mathcal{P}}| \leq L||\mathbf{A}^k - \mathbf{P}\mathbf{P}^{\mathrm{T}}||_F S_{\mathbf{X}} ||\mathbf{W}_P||_2 + S_{\phi(\mathbf{V})}$$

Proposition 3 shows that our method can capture the high-order structural information of multi-hop neighborhoods. Minimizing $||\mathbf{P}\mathbf{P}^{\mathrm{T}} - \mathbf{A}||$ is equivalent to minimizing edges between nodes in different communitys, which is a classic minimum cut problem in graph theory (Hofmeyr 2016). This can be achieved by graph partition algorithms, as these algorithms inherently maximize the sum of degrees within communities relative to their external degrees (Zhang et al. 2024). Next, we establish formal guarantees for the learned graph representations on downstream tasks.

Proposition 4 *Let G be a graph with B classes and the classes are balanced. Then, there exists a linear function $g(\cdot) : \mathcal{X}_G \to \mathbb{R}^B$ such that the error upper bound is*

$$\mathbb{E}_v[||y(v) - g(\mathbf{v})||_2^2] \leq 1 + B^2 \sum\nolimits_v (1 + \mathcal{L}_{\mathcal{P}}(v) - \varepsilon(v)) \quad (15)$$

Proposition 4 shows that the classification error on learned representations is bounded by the dual-kernel contrastive loss $\mathcal{L}_{\mathcal{P}}$ and the one-hop homophily score $\varepsilon(v)$ of node v in $\bar{\mathbf{A}}$. Note that $\varepsilon(v)$ is affected by the graph partition. In general, overly expansive community structures tend to result in a low value of $\varepsilon(v)$. Combining with Proposition 2, this requires introducing appropriate graph-level structural information to ensure performance. Conversely, in heterophilic graphs, an expanding receptive field provides additional information that cannot be captured within local neighborhoods (Xing et al. 2024). This means we can adapt to graphs with different homophily levels by adjusting the number of communities and the range of local neighborhoods.

Remark. The contrastive loss on coarsened graph can be seen as a special case of our method, i.e., when $\alpha = 0$ in Eq. 9. Consequently, our method naturally inherits the properties of these method. For instance, our dual-kernel GCCL loss can be seen as introducing an additional regularization term with better generalization, which makes our method more robust to minor perturbation (Zhang et al. 2024).

Properties of Distillation Loss

Based on the graph homophily assumption, nodes of the same semantic class typically share similar neighborhood representations. Thus, the local neighborhood representation $\mathbf{Z}$ can be viewed as sampled from a standard Gaussian distribution centered at $\mathbf{Z}_Y$, i.e., $Z|Y \sim N(\mathbf{Z}_Y, I)$, where Y denotes the latent semantic class of the K-hop patterns and Z is the random variable corresponding to $\mathbf{Z}$ (Xiao et al. 2023). Then, following (Boudiaf et al. 2020), we have:

Proposition 5 *Minimizing the distillation loss $\mathcal{L}_D$ is equivalent to maximizing mutual information between the representation $\mathbf{V}$ and the K-hop pattern Y:*

$$\mathcal{L}_D \geq H(V|Y) - H(V) = -I(Y;V) \quad (16)$$

where V is the random variable corresponding to $\mathbf{V}$.

Proposition 5 shows that minimizing the distillation loss in Eq. 13 can promote the maximizing of mutual information $I(Y;V)$ between node representations containing community information and the latent semantic classes of the K-hop patterns. This allows the distillation model to simultaneously leverage both community and graph-level structural information. Given the above characteristics, the distilled representations exhibit performance comparable to, or even better than, the pre-distillation ones. We will verify this conclusion in the following experiments.

Experimental Study

Experimental Setup

Datasets. We evaluate our method on 16 datasets, including (i) Seven homophilic graphs: Cora, CiteSeer, PubMed, Wiki-CS, Amazon-Photo, Coauthor-CS, and Coauthor-Physics (Sen et al. 2008; Mernyei and Cangea 2020; Shchur et al. 2018). (ii) Seven heterophilic graphs: Cornell, Texas, Wisconsin, Actor, Crocodile, Amazon-Ratings, and Questions (Pei et al. 2020; Rozemberczki, Allen, and Sarkar 2021; Platonov et al. 2023). (iii) Two large-scale graphs: Ogbn-Arxiv and Ogbn-Products (Hu et al. 2020).

Baselines. We compare our model with the following three categories of methods.

- Classic GCL methods: DGI (Veličković et al. 2019), GCA (Zhu et al. 2021), gCooL (Li, Jing, and Tong 2022), CSGCL (Chen et al. 2023), SP-GCL (Wang et al. 2023), GraphECL (Xiao et al. 2024) and SGRL (He et al. 2024).
- Efficiency-oriented GCL methods: BGRL (Thakoor et al. 2022), SUGRL (Mo et al. 2022), GGD (Zheng et al. 2022), SGCL (Sun et al. 2024), StructComp (Zhang et al. 2024) and E2Neg (Huang et al. 2025).
- Heterophily-aware GCL methods: HGRL (Chen et al. 2022), L-GCL (Zhang et al. 2022), DSSL (Xiao et al. 2022), GREET (Liu et al. 2023b), GraphACL (Xiao et al. 2023), HeterGCL (Wang et al. 2024), PolyGCL (Chen, Lei, and Wei 2024) and M3P-GCL (Chen et al. 2025b).

Evaluation Protocols. We evaluate the performance of node classification task by training a linear classifier on frozen graph representations. Each dataset is run across 10 different random splits to ensure robustness, and the average performance and standard deviation are reported here.

Implementation Details. A two-layer MLP is used as the distillation model. All experiments are implemented using PyTorch and run on a server equipped with an NVIDIA 3090 GPU (24GB memory). The detailed hyperparameter settings can be found in the code we provided.

Experimental Results

Exp-1: Effectiveness Evaluation. We conducted comprehensive node classification experiments on both homophilic and heterophilic graphs to evaluate the effectiveness of our method, as shown in Table 1.

These results demonstrate that: (i) Our method exhibits consistent and superior generalization performance across graphs with varying levels of homophily. (ii) Community-based methods, such as gCooL, CSGCL and E2Neg, show significant competitiveness in node classification, confirming the effectiveness of leveraging community structure in GCL. (iii) StructComp performs contrastive learning on

Methods	Cora	CiteSeer	PubMed	Wiki-CS	Amz.Photo	Co.CS	Co.Physics
DGI	82.12±1.28	71.58±1.21	78.87±2.64	75.73±0.13	91.49±0.25	91.95±0.40	94.57±0.39
GCA	79.04±1.39	65.62±2.46	81.55±2.47	79.35±0.42	92.78±0.17	93.32±0.12	95.87±0.15
gCooL	81.63±1.39	71.32±1.64	82.16±1.31	78.87±0.22	93.18±0.12	93.27±0.15	95.13±0.11
CSGCL	79.39±1.57	70.03±1.49	80.37±2.06	78.57±0.14	93.24±0.37	93.59±0.09	95.32±0.24
SP-GCL	82.78±1.35	71.81±1.06	81.14±1.82	80.21±0.37	92.49±0.31	93.05±0.10	95.12±0.15
GraphECL	82.88±0.95	72.26±0.89	82.14±1.63	80.17±0.15	93.39±0.46	94.12±0.16	96.03±0.07
SGRL	82.64±1.92	71.73±1.58	80.91±1.84	80.67±0.26	93.29±0.42	93.61±0.26	95.99±0.10
BGRL	82.33±1.35	71.59±1.42	79.23±1.74	78.74±0.22	93.24±0.29	93.26±0.36	95.76±0.38
SUGRL	81.34±1.23	71.02±1.77	80.53±1.62	79.12±0.67	93.07±0.15	92.83±0.23	95.38±0.11
GGD	82.34±1.57	71.04±1.47	81.28±1.31	78.72±0.61	92.53±0.63	92.44±0.19	95.03±0.21
SGCL	82.57±1.43	71.65±1.31	81.93±1.66	79.85±0.53	93.46±0.31	93.29±0.17	95.78±0.11
St.Comp	81.28±1.29	71.46±1.54	80.47±1.63	80.57±0.11	92.62±0.14	92.56±0.12	95.44±0.10
E2Neg	81.47±1.67	71.69±1.92	80.93±1.49	81.12±0.57	93.36±0.76	93.48±0.59	95.86±0.29
Ours	**83.77±1.37**	**72.68±1.19**	**82.56±1.85**	**81.75±0.36**	**93.86±0.15**	**94.68±0.14**	**96.12±0.17**

Methods	Cornell	Texas	Wisconsin	Actor	Crocodile	Amz.Ratings	Questions
HGRL	51.78±1.03	61.83±0.71	63.90±0.58	27.95±0.30	61.87±0.45	38.37±0.36	-
L-GCL	52.11±2.37	60.68±1.18	65.28±0.52	32.55±1.18	60.18±0.43	-	-
DSSL	53.15±1.28	62.11±1.53	62.25±0.55	28.15±0.31	62.98±0.51	-	-
SP-GCL	52.29±1.21	59.81±1.33	60.12±0.39	28.94±0.69	61.72±0.21	43.11±0.32	75.08±0.49
GREET	72.91±1.13	84.59±4.20	80.98±5.62	36.14±1.38	66.75±0.56	41.19±0.25	-
GraphACL	59.33±1.48	71.08±0.34	69.22±0.40	30.03±0.13	66.17±0.24	41.49±0.45	74.85±0.98
HeterGCL	75.48±2.83	74.71±3.59	75.58±4.47	37.20±0.44	65.42±0.57	-	-
PolyGCL	73.78±3.51	72.16±3.51	76.08±3.33	34.37±0.69	65.95±0.59	44.29±0.43	75.33±0.67
M3P-GCL	75.59±3.81	80.84±1.62	81.67±2.23	35.12±0.97	65.67±0.31	42.91±0.17	-
Ours	**76.49±2.43**	**85.41±3.01**	**85.17±3.02**	**37.74±0.78**	**67.05±0.72**	**47.51±0.68**	**76.35±1.05**

Table 1: Node classification results on homophilic (top) and heterophilic (bottom) graphs (%). AUC is used for Questions and accuracy for all other datasets. Best and second-best results are shown in **bold** and underline, respectively. '-' indicates either unavailability of the official implementation or exceeding 24 GB GPU memory during evaluation.

coarsened graphs, ignoring node-level information, which may result in suboptimal performance on node-level tasks.

Exp-2: Scalability Evaluation. We evaluate the scalability of our method by comparing it with efficiency-oriented GCL methods on large-scale graphs, as shown in Table 2 and Figure 2. For fairness, we excluded memory footprint reports for methods trained with mini-batch processing.

These results demonstrate that: (i) Our method consistently achieves the best performance on large-scale graphs. (ii) Although StructComp achieves the lowest training overhead by ignoring node-level information, our method effectively improves performance with only a slight increase in computational complexity, demonstrating a well-balanced trade-off between scalability and performance. Notably, the accuracy gain over SOTA baselines is 1.7% on Ogbn-Products, a substantial improvement as baselines are carefully fine-tuned on the dataset. (iii) Our method consistently outperforms other methods in inference efficiency, and the inference time scales linearly with graph size. On Ogbn-Products, our method is about 180× faster than the best baseline, which highlights the superiority of knowledge distillation technique for the decoupled GNN.

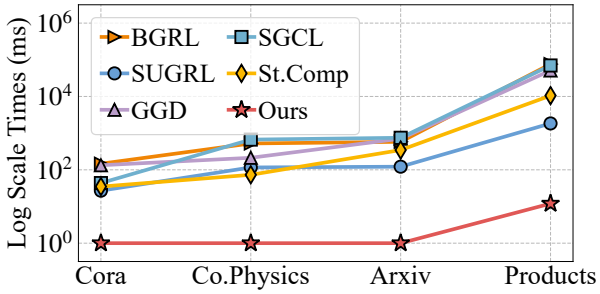

Figure 2: Inference efficiency comparison.

Exp-3: Necessity of Dual-Kernel. We validate the necessity of using dual-kernel contrastive loss to integrate node-level and community-level information by comparing different variants of our method with SOTA baselines. Specifically, we analyze two schemes, the tensor product and linear combination schemes, as well as two additional variants that focus solely on node-level kernel ($\alpha = 0$) and community-level kernel ($\alpha = 1$), as shown in Figure 3.

These results demonstrate that: (i) The dual-kernel

Methods	Ogbn-Arxiv					Ogbn-Products				
	Acc	Time.T(**s**)	Mem.T	Time.I(**s**)	Mem.I	Acc	Time.T(**m**)	Mem.T	Time.I(**s**)	Mem.I
BGRL	71.6	1.43	10.7	0.58	6.1	64.0	53.3	-	76.33	22.8
SUGRL	67.8	0.11	**2.6**	0.12	1.5	72.9	1.5	23.5	1.84	21.3
GGD	71.6	0.95	14.3	0.71	1.9	75.7	12.7	-	143.36	22.8
SGCL	71.0	0.09	5.1	0.75	4.2	76.0	1.9	-	69.94	22.9
St.Comp	71.8	**0.05**	3.4	0.35	1.6	75.5	**0.001**	**5.3**	10.54	12.0
Ours	**72.2**	0.08	4.2	**0.001**	**1.1**	**77.7**	0.003	8.8	**0.01**	**5.3**

Table 2: Scalability evaluation on large-scale datasets. Time.T / I: training / inference time per epoch; Mem.T / I: peak GPU memory usage during training / inference (GB).

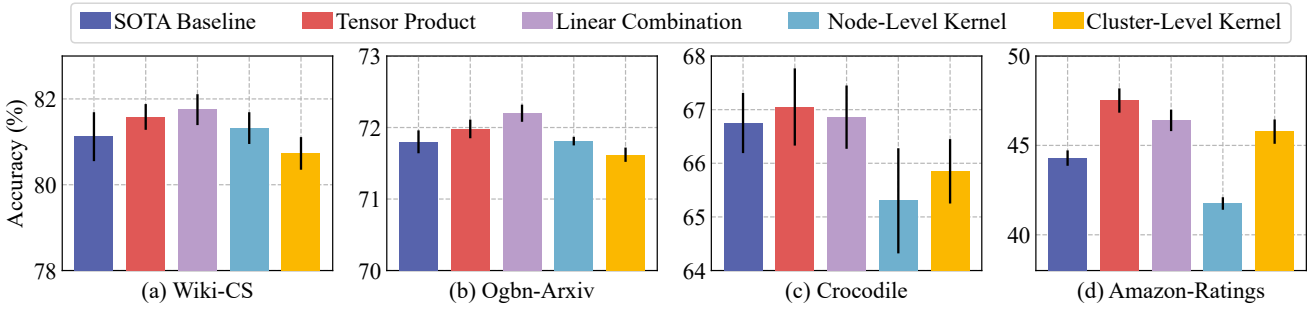

Figure 3: Comparison of different dual-kernel graph community contrastive loss variants.

method outperforms baselines and variants focusing on a single kernel, which highlights the effectiveness of MKL in integrating diverse levels of information. (ii) On homophilic graphs, the linear combination scheme performs better, with the node-level kernel outperforming the community-level kernel. Conversely, on heterophilic graphs, the tensor product scheme and the community-level kernel achieve superior performance. This suggests that homophilic graphs should pay more attention to node-level information.

Related Work

Graph Contrastive Learning with Community. Recent studies have demonstrated the effectiveness of exploiting community structure in GCL (Chen et al. 2025a), which can be categorized into view-optimized and loss-optimized methods. (i) View-optimized methods focus on preserving community information in augmented views. For instance, SEGA (Wu et al. 2023) uses an encoding tree containing community properties as the anchor view. CI-GCL (Tan et al. 2024) constrains view augmentation via based on community invariance. StructComp (Zhang et al. 2024) performs GCL on community-wise graph. (ii) Loss-optimized methods can effectively avoid mislabeling closely connected nodes as negative samples. For example, gCooL (Li, Jing, and Tong 2022) considers nodes and the centroid of their respective communities as positive sample pairs. CS-GCL (Chen et al. 2023) adjusts the weight of contrastive samples based on community strength. E2Neg (Huang et al. 2025) selects representative negative samples from communities. Despite these significant advancements, they are still limited by the message passing mechanism of GNN.

Kernel-Based Representation Learning. Kernel methods have been used to address the scalability issues of Graph Transformers in supervised learning scenarios, as they can bypass the cumbersome explicit computation of all-pairs attentions (Deng, Yue, and Zhang 2024). MKL further enhances the expressiveness of kernel methods by combining multiple kernel functions to integrate features from different perspectives (Gönen and Alpaydın 2011; Celikkanat, Shen, and Malliaros 2022). Examples include recent studies that leverage MKL for federated learning (M. Ghari and Shen 2022), clustering (Liu 2022) and graph classification (Huang et al. 2024). Despite the widespread application of kernel and MKL methods, their utilization for unsupervised graph representation learning remains an underexplored area.

Conclusion

In this work, we propose a scalable and efficient dual-kernel graph community contrastive learning method, underpinned by a straightforward graph partition algorithm and MKL techniques. This design enables us to capture community-level structural features in linear time while preserving essential node-level information. Furthermore, the proposed knowledge distillation technique of the decoupled GNN is particularly suitable for latency-constrained applications. Both theoretical analysis and experimental evaluations verify the effectiveness of our method. We also envision future directions, such as learning adaptive graph partition, integrating edge-based attribute features, and extending to dynamic graphs or more complex graph applications.

Acknowledgments

This work was supported by Yunnan Fundamental Research Project (202501AS070102), Program of Yunnan Key Laboratory of Intelligent Systems and Computing (202405AV340009), Future Industry Science and Technology Special Project of Yunnan University (YDWLCY202505), and Scientific Research Fund Project of Yunnan Education Department (2025Y0061).

References

Boudiaf, M.; Rony, J.; Ziko, I. M.; Granger, E.; Pedersoli, M.; Piantanida, P.; and Ayed, I. B. 2020. A Unifying Mutual Information View of Metric Learning: Cross-Entropy vs. Pairwise Losses. In *Proceedings of the 16th Europran Conference on Computer Vision* (ECCV), 548–564. Springer.

Celikkanat, A.; Shen, Y.; and Malliaros, F. D. 2022. Multiple Kernel Representation Learning on Networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 6113–6125.

Chen, H.; Zhao, Z.; Li, Y.; Zou, Y.; Li, R.; and Zhang, R. 2023. CSGCL: Community-Strength-Enhanced Graph Contrastive Learning. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence* (IJCAI), 2059–2067.

Chen, J.; Lei, R.; and Wei, Z. 2024. PolyGCL: Graph Cntrastive Learning via Learnable Spectral Polynomial Filters. In *Proceedings of the 12th International Conference on Learning Representations* (ICLR), 48613–48642.

Chen, J.; Zhu, G.; Qi, Y.; Yuan, C.; and Huang, Y. 2022. Towards Self-Supervised Learning on Graphs with Heterophily. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* (CIKM), 201–211.

Chen, X.; Yue, K.; Duan, L.; and Yu, L. 2025a. Improving Graph Contrastive Learning with Community Structure. In *Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence* (UAI), 568–578.

Chen, Y.; Guan, D. o.; Yuan, W.; and Zang, T. 2025b. Beyond Homophily: Graph Contrastive Learning with Macro-Micro Message Passing. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence* (AAAI), 15948–15956.

Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-Gcn: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *Proceedings of the 25th ACM Conference on Knowledge Discovery and Data Mining* (SIGKDD), 257–266.

Deng, C.; Yue, Z.; and Zhang, Z. 2024. Polynormer: Polynomial-Expressive Graph Transformer in Linear Time. In *Proceedings of the 12th International Conference on Learning Representations* (ICLR), 18954–18979.

Duan, L.; Chen, X.; Liu, W.; Liu, D.; Yue, K.; and Li, A. 2024. Structuarl Entropy Based Graph Structure Learning for Node Classification. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence* (AAAI), 8372–8379.

Gönen, M.; and Alpaydın, E. 2011. Multiple Kernel Learning Algorithms. *The Journal of Machine Learning Research*, 12: 2211–2268.

He, D.; Shan, L.; Zhao, J.; Zhang, H.; Wang, Z.; and Zhang, W. 2024. Exploitation of a Latent Mechanism in Graph Contrastive Learning: Representation Ccattering. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems* (NeurIPS), 115351–115376.

Hofmeyr, D. P. 2016. Clustering by Minimum Cut Hyperplanes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8): 1547–1560.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems* (NeurIPS), 22118–22133.

Huang, S.; Song, Y.; Zhou, J.; and Lin, Z. 2024. Cluster-wise Graph Transformer with Dual-granularity Kernelized Attention. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems* (NeurIPS), 33376–33401.

Huang, Y.; Zhao, J.; He, D.; Jin, D.; Huang, Y.; and Wang, Z. 2025. Does GCL Need a Large Number of Negative Samples? Enhancing Graph Contrastive Learning with Effective and Efficient Negative Sampling. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence* (AAAI), 17511–17518.

Huo, C.; Jin, D.; Li, Y.; He, D.; Yang, Y.-B.; and Wu, L. 2023. T2-GNN: Graph Neural Networks for Graphs with Incomplete Features and Structure via Teacher-Student Distillation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence* (AAAI), 4339–4346.

Karypis, G.; and Kumar, V. 1998. A fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on scientific Computing*, 20(1): 359–392.

Li, B.; Jing, B.; and Tong, H. 2022. Graph Communal Contrastive Learning. In *Proceedings of the ACM Web Conference* (WWW), 1203–1213.

Liu, X. 2022. Simplemkkm: Simple Multiple Kernel K-Means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 5174–5186.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. S. 2023a. Graph Self-Supervised Learning: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5879–5900.

Liu, Y.; Zheng, Y.; Zhang, D.; Lee, V. C.; and Pan, S. 2023b. Beyond Smoothing: Unsupervised Graph Representation Learning with Edge Heterophily Discriminating. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence* (AAAI), 4516–4524.

Luo, Y.; Wu, X.; and Zhu, H. 2025. Beyond Random Masking: When Dropout meets Graph Convolutional Networks. In *Proceedings of the 13th International Conference on Learning Representations* (ICLR), 69326–69345.

M. Ghari, P.; and Shen, Y. 2022. Personalized Online Federated Learning with Multiple Kernels. In *Proceedings of the*

36th Annual Conference on Neural Information Processing Systems (NeurIPS), 33316–33329.

Mernyei, P.; and Cangea, C. 2020. Wiki-cs: A Wikipedia-based Benchmark for Graph Neural Networks. *arXiv preprint arXiv:2007.02901*.

Mo, Y.; Peng, L.; Xu, J.; Shi, X.; and Zhu, X. 2022. Simple Unsupervised Graph Representation Learning. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence* (AAAI), 7797–7805.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *Proceedings of the 9th International Conference on Learning Representations* (ICLR), 10247–10258.

Platonov, O.; Kuznedelev, D.; Diskin, M.; Babenko, A.; and Prokhorenkova, L. 2023. A Critical Look at the Evaluation of GNNs under Heterophily: Are We Really Making Progress? In *Proceedings of the 12th International Conference on Learning Representations* (ICLR), 18738–18752.

Rozemberczki, B.; Allen, C.; and Sarkar, R. 2021. Multi-scale Attributed Node Embedding. *Journal of Complex Networks*, 9(2): cnab014.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI magazine*, 29(3): 93–93.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of Graph Neural Network Evaluation. *arXiv preprint arXiv:1811.05868*.

Shen, X.; Sun, D.; Pan, S.; Zhou, X.; and Yang, L. T. 2023. Neighbor Contrastive Learning on Learnable Graph Augmentation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence* (AAAI), 9782–9791.

Sun, W.; Li, J.; Chen, L.; Wu, B.; Bian, Y.; and Zheng, Z. 2024. Rethinking and Simplifying Bootstrapped Graph Latents. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* (WSDM), 665–673.

Tan, S.; Li, D.; Jiang, R.; Zhang, Y.; and Okumura, M. 2024. Community-invariant Graph Contrastive Learning. In *Proceedings of the 41st International Conference on Machine Learning*(ICML), 47579–47606.

Thakoor, S.; Tallec, C.; Azar, M. G.; Azabou, M.; Dyer, E. L.; Munos, R.; Veličković, P.; and Valko, M. 2022. Large-Scale Representation Learning on Graphs via Bootstrapping. In *Proceedings of the 10th International Conference on Learning Representations* (ICLR), 19932–19952.

Tian, Y.; Zhang, C.; Guo, Z.; Zhang, X.; and Chawla, N. 2023. Learning Mlps on Graphs: A Unified View of Effectiveness, Robustness, and Efficiency. In *Proceedings of the 11th International Conference on Learning Representations* (ICLR), 31352–31363.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *Proceedings of the 7th International Conference on Learning Representations* (ICLR), 6633–6649.

Wang, C.; Liu, Y.; Yang, Y.; and Li, W. 2024. HeterGCL: Graph Contrastive Learning Framework on Heterophilic Graph. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence* (IJCAI), 2397–2405.

Wang, H.; Zhang, J.; Zhu, Q.; Huang, W.; Kawaguchi, K.; and Xiao, X. 2023. Single-Pass Contrastive Learning Can Work for Both Homophilic and Heterophilic Graph. *Transactions on Machine Learning Research*.

Wu, J.; Chen, X.; Shi, B.; Li, S.; and Xu, K. 2023. SEGA: Structural Entropy Guided Anchor View for Graph Contrastive Learning. In *Proceedings of the 40th International Conference on Machine Learning* (ICML), 37293–37312.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24.

Xiao, T.; Chen, Z.; Guo, Z.; Zhuang, Z.; and Wang, S. 2022. Decoupled Self-Supervised Learning for Graphs. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems* (NeurIPS), 620–634.

Xiao, T.; Zhu, H.; Chen, Z.; and Wang, S. 2023. Simple and Asymmetric Graph Contrastive Learning without Augmentations. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems* (NeurIPS), 16129–16152.

Xiao, T.; Zhu, H.; Zhang, Z.; Guo, Z.; Aggarwal, C. C.; Wang, S.; and Honavar, V. G. 2024. Efficient Contrastive Learning for Fast and Accurate Inference on Graphs. In *Proceedings of the 41st International Conference on Machine Learning* (ICML), 54363–54381.

Xing, Y.; Wang, X.; Li, Y.; Huang, H.; and Shi, C. 2024. Less is More: on the Over-Globalizing Problem in Graph Transformers. In *Proceedings of the 41st International Conference on Machine Learning* (ICML), 54656–54672.

Zhang, H.; Wu, Q.; Wang, Y.; Zhang, S.; Yan, J.; and Yu, P. S. 2022. Localized Contrastive Learning on Graphs. *arXiv preprint arXiv:2212.04604*.

Zhang, S.; Yang, W.; Cao, X.; Zhang, H.; and Huang, Z. 2024. StructComp: Substituting Propagation with Structural Compression in Training Graph Contrastive Learning. In *Proceedings of the 12th International Conference on Learning Representations* (ICLR), 36245–36264.

Zheng, Y.; Pan, S.; Lee, V.; Zheng, Y.; and Yu, P. S. 2022. Rethinking and Scaling up Graph Contrastive Learning: An Extremely Efficient Approach with Group Discrimination. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems* (NeurIPS), 10809–10820.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *Proceedings of the ACM Web Conference* (WWW), 2069–2080.