HeterGCL: Graph Contrastive Learning Framework on Heterophilic Graph

Chenhao Wang[1] , **Yong Liu**[1*] , **Yan Yang**[1*] and **Wei Li**[2]
[1]Heilongjiang University
[2]Harbin Engineering University
2211898@s.hlju.edu.cn, {2010023, yangyan}@hlju.edu.cn, wei.li@hrbeu.edu.cn

Abstract

Graph Contrastive Learning (GCL) has attracted significant research attention due to its self-supervised ability to learn robust node representations. Unfortunately, most methods primarily focus on homophilic graphs, rendering them less effective for heterophilic graphs. In addition, the complexity of node interactions in heterophilic graphs poses considerable challenges to augmentation schemes, coding architectures, and contrastive designs for traditional GCL. In this work, we propose HeterGCL, a novel graph contrastive learning framework with structural and semantic learning to explore the true potential of GCL on heterophilic graphs. Specifically, We abandon the random augmentation scheme that leads to the destruction of the graph structure, instead introduce an adaptive neighbor aggregation strategy (ANA) to extract topology-supervised signals from neighboring nodes at different distances and explore the structural information with an adaptive local-to-global contrastive loss. In the semantic learning module, we jointly consider the original nodes' features and the similarity between nodes in the latent feature space to explore hidden associations between nodes. Experimental results on homophilic and heterophilic graphs demonstrate that HeterGCL outperforms existing self-supervised and semi-supervised baselines across various downstream tasks.

1 Introduction

Graph Neural Networks (GNNs) have demonstrated superior performance in graph-based machine learning tasks like node classification and clustering [Xu *et al.*, 2023; Liu *et al.*, 2024a; Liu *et al.*, 2024b]. Following the homophily assumption [Wang *et al.*, 2023], they iteratively aggregate and transform information from neighbor nodes to learn node representations. Generally, GNNs are designed mainly to fulfill the needs of supervised tasks, which rely heavily on task-specific labeled data. However, obtaining labeled graph datasets in the real world is challenging due to the complexity of the graph structure. This motivates many pioneering works in graph self-supervised learning to alleviate the reliance on manual labels, especially Graph Contrastive Learning (GCL), which transposes contrastive learning from vision and language domains to graph data, attracting extensive research interest.

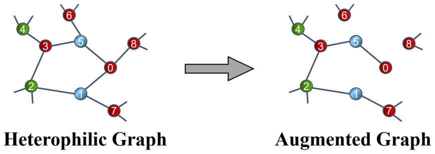


Figure 1: Random graph Augmentation.

Most GCL methods continue the visual contrastive pattern of "augmentation-encoding-contrast" to provide self-supervised learning without labeled data. This involves augmenting the original graph to generate contrasting views, encoding these views using traditional GNNs, and learning more general and robust node representations by maximizing consistency between the augmented views. While the current GCL paradigm has shown promising results on homophilic graphs, where linked nodes share similar features or labels, its effectiveness is greatly limited in heterophilic settings.

First, traditional GCL performs random graph augmentation by removing edges or nodes to obtain different views. However, it may destroy the underlying structural information in graphs. For example, randomly deleting an element node in a molecular graph may lead to a completely different molecular structure with distinct properties. Similarly, removing edges adjacent to hub nodes can impede the rapid propagation of node information and isolate important nodes that should be linked. Figure 1 shows an example where node 7 could have propagated features to node 0 within two hops. However, due to the removal of the edge between nodes 0 and 1, node 7 now requires a longer propagation path ($7 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 0$). Recent theoretical and empirical analyses [Wang *et al.*, 2022a] further reveal that random augmentation maintains low-frequency components in homophilic graphs but suppresses high-frequency information in heterophilic graphs. This explains why GCL tends to achieve favorable results on homophilic graphs but struggles with heterophilic graphs.

Secondly, homophily as a crucial inductive bias on graphs

[*] Corresponding author.

is widely recognised as an important factor for the success of graph neural networks. The encoding structure and contrastive patterns of GCL inherit the implicit homophily in GNNs, which provides appropriate guidance in cases where node labels are unknown. They reconstruct node attributes and interactions through local message passing. Many recent studies [Wang *et al.*, 2022c; Lee *et al.*, 2022] also discovered the effectiveness of using homophily, and they utilized community structure to strengthen local connections. [Li *et al.*, 2023] further exploits homophily directly by treating neighbor nodes as positive nodes. However, it is worth noting that real-world graphs often exhibit heterophily, where nodes with similar semantics may not be geographically close and adhere to the principle of “opposites attract”. In such cases, smoothing features of locally connected nodes will inadvertently merge irrelevant information from different classes, which limits the application of GCL to general data. Overall, the potential of GCL on heterophilic graphs with complex connectivity remains unexplored. It motivates our investigation into improving graph contrastive learning to fill this gap.

To address the above challenges, we analyze the homophilic levels of graphs, and find that nodes within heterophilic graphs exhibit complex interactions, which challenges the traditional “similarity attracts similarity” principle in GCL. Therefore, we propose a novel framework HeterGCL for graph contrastive learning. By incorporating both structural and semantic information, HeterGCL aims to explore the true potential of GCL on heterophilic graphs. In the structural branch, we eliminate random augmentation and design a more comprehensive structural augmentation scheme, ANA, which effectively extracts topology-supervised signals from neighboring nodes at different distances while avoiding feature mixing. In addition, we introduce the adaptive neighbor contrastive loss (ANCLoss) to facilitate the adaptive learning of structural information from local to global. This allows us to overcome the limitations of interactions within the same layer or across layers. Meanwhile, the semantic branch integrates the feature information of nodes in the original graph with the semantic information between similar nodes in the potential feature space. The incorporation of similarity relations into the features has been shown to be beneficial for handling heterophilic graphs. Finally, by jointly optimizing structural and semantic losses, we can learn highly expressive node representations for different downstream tasks without manually annotated labels. Our source code is available at https://github.com/Incendio1/HeterGCL.

Our main contributions can be summarized as follows: (1) We reveal the limitations of the traditional GCL when applied to heterophilic graphs and empirically show the complexity of node interactions, which makes the traditional GCL paradigm inapplicable. (2) To fill the gap of self-supervised learning in heterophilic graphs, we propose a novel GCL framework HeterGCL, which improves the “augmentation-encoding-contrast” pattern by incorporating structure and semantic learning to obtain effective node representations for different homophilic-level graphs. (3) Extensive experiments on different homophily graphs show that our method achieves state-of-the-art performance compared to supervised and unsupervised baselines in various downstream tasks.

2 Related Work

Graph Neural Networks Meet Heterophily. Most GNNs employ the message passing (MP) to facilitate the feature propagation among nodes and their neighbors, where prominent examples include GCN [Kipf and Welling, 2017] and GAT [Velickovic *et al.*, 2018]. [Wu *et al.*, 2019] further decoupled the MP to explore the global structural information. Unfortunately, the homophily implicit in MP limits the generalization of GNNs to heterophilic graphs, which is outlined in [Pei *et al.*, 2020]. They show that unlike “like attracts like” in homophilic graphs, different nodes in heterophilic graphs tend to be linked. In this case, GNNs may struggle to perform well. Recent works have begun to revisit the heterophily by designing wider messaging ranges to capture distant but similar nodes in the heterophilic graph. JKNet and Mix-Hop [Xu *et al.*, 2018; Abu-El-Haija *et al.*, 2019] transform and connect multilayer neighbor representations, while DAGNN and GPRGNN [Liu *et al.*, 2020; Chien *et al.*, 2021] use graph diffusion to capture higher-order neighbors in heterophilic graphs. Other studies have shown that incorporating feature perspectives, such as node attribute graphs [Jin *et al.*, 2021] and interpretable compatibility matrices [Wang *et al.*, 2022b], can improve GNNs on heterophilic graphs. However, these methods still require a supervised setting, which increases the demand for high-quality datasets.

Graph Self-supervised Learning. With the success of self-supervised learning in improving representation quality when labels are scarce, more and more work has focused on combining self-supervised learning with GNNs. Early works usually adopt traditional network embedding strategies such as random walk or link reconstruction [Qiu *et al.*, 2018; Zhang *et al.*, 2018] , which sacrifice certain topology information and pay more attention to the proximity of nodes. Therefore, recent works have turned to GCL, which maximizes the consistency between enhanced graph views by contrasting positive and negative samples across different views, e.g., DGI, InfoGraoph, and MVGRL [Velickovic *et al.*, 2019; Sun *et al.*, 2020; Hassani and Khasahmadi, 2020]. On the one hand, the core of GCL is how to determine contrast patterns. Grace focuses on node-level contrast. BGRL [Thakoor *et al.*, 2022] uses invariance regularization to perform self-supervised representation learning without negative samples. On the other hand, the GCL framework relies on the design of graph augmentation to facilitate the learning of invariant representations, which includes node dropping, edges removing and adding, etc. However, it may erase the original graph's semantic information. [Lee *et al.*, 2022] also highlighted the need for careful calibration of random augmentation. If edges are over-removed due to “sampling bias”, the graph structure may be damaged, and insufficient removals may not provide sufficient learning signals for the model.

3 Notations and Preliminaries

Notation. An undirected graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of nodes $\{v_1, \ldots, v_N\}$. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of the edges. The adjacency matrix of $\mathcal{G}$ is denoted as $\mathbf{A} \in \mathbb{R}^{N \times N}$ and the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$, where d is the feature dimension size. The diagonal degree matrix of

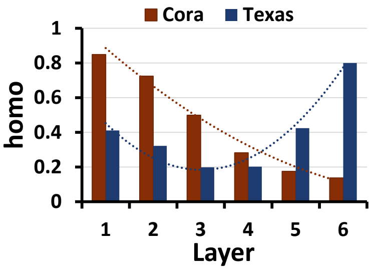


Figure 2: Analysis of homophily level on Cora and Texas.

nodes is denoted as $\mathbf{D} \in \mathbb{R}^{N \times N}$ and $\mathbf{D}_{(i,i)} = \sum_j \mathbf{A}_{(i,j)}$. In addition, $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}} \widetilde{\mathbf{A}} \widetilde{\mathbf{D}}^{-\frac{1}{2}}$ is the re-normalized adjacency matrix, where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ represents the adjacency matrix with self-loop and $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix. $\widetilde{\mathbf{D}}$ is the diagonal matrix corresponding to $\widetilde{\mathbf{A}}$.

Problem Definition. Given a graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, we aim to learn the encoder $\mathcal{F}$ to map the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ to a low-dimensional representation $\mathbf{H} \in \mathbb{R}^{N \times d_f}$, and $d_f << d$. The i-th row $\mathbf{h}_i$ in $\mathbf{H}$ represents the low-dimensional embedding of node v_i. Finally, these representations can be further used for downstream tasks.

4 Method

In this section, we first explore why the existing methods are unsuitable for heterophilic graphs. Then, we present HeterGCL to cope with the current challenges.

4.1 Homophiliy Level Analysis

Existing methods usually assume a strong homophilic relationship between nodes and use adjacency information ($\mathbf{A}$) to improve node predictability. Nevertheless, it is not always the case in practice. The nodes with the same label as the center node are not evenly distributed among each layer. As shown in Figure 2, we calculated the average homophily level **homo** of nodes in different layers using homophilic graph Cora and heterophilic graphs Texas as examples. **homo** is computed based on the ratio of neighboring nodes belonging to the same class as each node in each layer. Our observations reveal that shallow neighbor nodes on Cora exhibit more similar labels with their corresponding center nodes. As the layer depth increases, **homo** shows a decreasing trend, indicating the difficulty in finding neighbor nodes similar to the center node in deeper layers. These observations explain the strong performance of GNN and GCL methods, as they benefit from similar relationships among connected nodes. However, the situation is different for Texas. More than half of the nodes in the shallow layer do not share the same class label as the center node. **homo** gradually decreases from the initial to the third layer and shows an increasing trend after the fourth layer. This empirical evidence suggests that the distribution of neighboring nodes on heterophilic graphs follows a diverse and heterophilic pattern. It is also worth noting that even on the highly homophilic Cora dataset, many nodes in the shallow layers do not belong to the same class as their corresponding center nodes. For example, the first and second layers consist of approximately 15% and 28% neighbor nodes from different classes with the center node. Consequently, simple learning of node connectivity relationships introduces considerable irrelevant noise nodes. We would like to design personalized decision-making schemes for nodes that reasonably utilize neighbor information from different layers.

4.2 HeterGCL

This section formally introduces the HeterGCL framework. The overall details of HeterGCL are displayed in Figure 3.

Structure Augmentation via ANA. Following the GCL pattern, we first need to augment the input graphs to obtain graph views for contrast. However, current random augmentation destroys the structural integrity of the graph. In addition, considering the high-frequency preference of heterophilic graphs, structural perturbations will impact the middle and high-frequency components of the graph. Therefore, we drop the random augmentation to preserve the structural knowledge. Based on our analysis, capturing homophilic nodes spread across different layers is critical in heterophilic graphs. It motivates us to explore the replacement of random augmentation with graph diffusion [Klicpera *et al.*, 2019]. We explicitly set the transfer matrix $\widehat{\mathbf{A}}^{(k)}$ to construct self-supervised signals by incorporating the information from neighboring nodes. However, since neighbor distributions in heterophilic graphs usually show different heterophilic patterns, recursive aggregation tends to include more heterophilic nodes. Figure 4 shows an example where nodes of the same label share the same color. When the center node v_0 aggregates features from multiple layers, information is transmitted to v_0 layer-by-layer via connected edges. As a result, node v_0 indiscriminately collects information from various layers, including noisy nodes with different classes as v_0.

To address this problem, we propose Adaptive Neighbor Aggregation (ANA) instead of random augmentation. ANA is automatically compatible with various graph datasets without requiring a priori domain knowledge. Specially, We do not assume a strong correlation between the center node and its neighbors. Instead, we untangle the interconnections between multiple layers and group neighbors according to their layer orders, which avoids information being forced to propagate along the graph structure layer by layer leading to mutual interference between different layers. It is defined as follows:

$$\begin{gathered} \mathbf{M}^{(l)} = \operatorname{sgn}\left[\widehat{\mathbf{A}}^{(l)}\right] - \operatorname{sgn}\left[\widehat{\mathbf{A}}^{(l-1)}\right], \\ \widetilde{\mathbf{M}}^{(l)} = \mathbf{M}^{(l)} + \mathbf{I}, l = 1, 2, \ldots, K. \end{gathered} \tag{1}$$

where the $\operatorname{sgn}[\cdot]$ function is used to indicate a result of 1 when $\widehat{\mathrm{A}}_{ij}^{(l)} > 0$, and 0 otherwise. When $l = 1$, $\widehat{\mathbf{A}}^{(0)} = \mathbf{I}$. From Theorem 1, it can be inferred that $\mathbf{M}^{(l)}$ effectively captures all pairs of nodes with the shortest path length of l. Thus, for each node, the set $\{\mathbf{M}^{(1)}, ..., \mathbf{M}^{(l)}\}$ contains the neighbor nodes from layer 1 to layer l, where nodes occurring in both the shallow and deep layers are stored only in the shallow group. Finally, by adding self-loops to each node, $\widetilde{\mathbf{M}}^{(l)}$ can be obtained. Grouping allows for a more flexible configuration of neighbor nodes from different layers without interference from shallow nodes. This flexibility in the aggregation

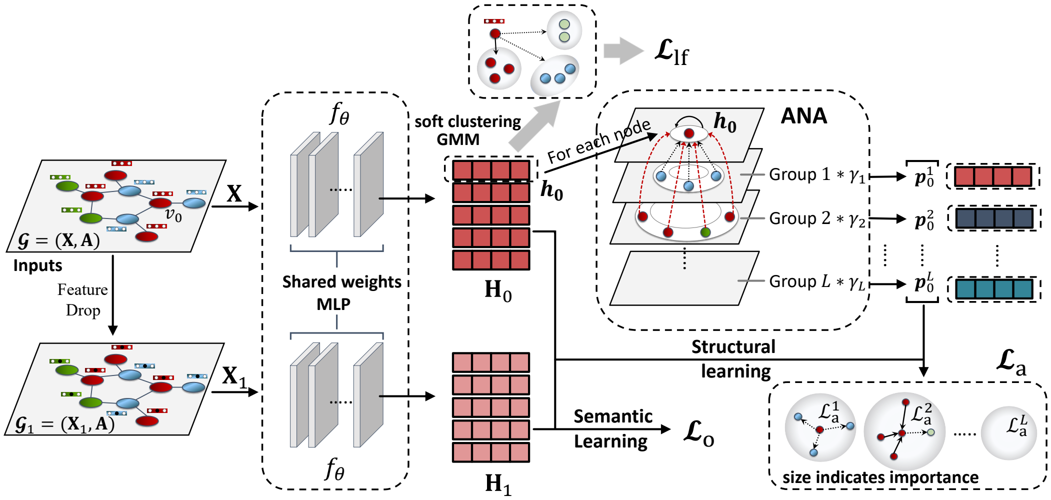


Figure 3: The overall architecture of HeterGCL.The original graph $\mathcal{G}$ and the perturbation graph $\mathcal{G}_1$ into the shared MLP encoder and learn the feature semantic information without negative samples by loss $\mathcal{L}_{\rm o}$. The $\mathcal{L}_{\rm lf}$ is used to learn the potential feature similarity. For structure learning, we generate local-to-global views for contrastive loss $\mathcal{L}_{\rm a}$ by an unperturbed structural augmentation method ANA.

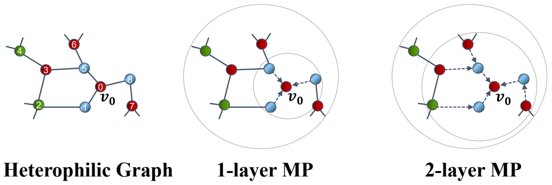


Figure 4: Traditional message passing mechanism (MP).

enables the inclusion of more valuable information from farther away nodes in heterophilic graphs.

Theorem 1. *Assume that $\widehat{\mathbf{A}}$ represents the normalized adjacency matrix of an undirected graph $\mathcal{G}$. Then* $\operatorname{sgn}\left[\widehat{\mathbf{A}}^{(l)}\right] - \operatorname{sgn}\left[\widehat{\mathbf{A}}^{(l-1)}\right]$ *records all pairs of nodes whose shortest path length is equal to l.*

Proof. We consider the unnormalized adjacency matrix as an example. Suppose $\mathbf{A}$ is the adjacency matrix corresponding to graph $\mathcal{G}$, recording all node pairs with path lengths equal to 1. $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ records all node pairs with path length less than or equal to 1 (the path length is 0 from itself to itself). $\widetilde{\mathbf{A}}^{(2)}$ records all pairs of nodes whose path length is less than or equal to 2 (all pairs of nodes whose distance is equal to 0, 1, 2). By the induction hypothesis, $\widetilde{\mathbf{A}}^{(l-1)}$ records all node pairs with path lengths less than or equal to $l-1$, and $\widetilde{\mathbf{A}}^{l}$ records all node pairs with path lengths less than or equal to l. Therefore, $\operatorname{sgn}\left[\widetilde{\mathbf{A}}^{(l)}\right] - \operatorname{sgn}\left[\widetilde{\mathbf{A}}^{(l-1)}\right]$ records all pairs of nodes whose shortest path length is exactly l. □

Although Equation (1) preserves the nodes of different layers, it sacrifices more connection relations between nodes. Specifically, $\widetilde{\mathbf{M}}^{(l)}$ only retains the shortest paths between nodes and disregards other paths, which means that neighbor nodes in the same layer have the same influence on the center node. Figure 4 illustrates this limitation, where there is only one direct path between node 0 and node 7. In contrast, there are multiple connected paths $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ and $0 \rightarrow 5 \rightarrow 3$ between node 3 and node 0. Intuitively, the interaction strength between nodes 3 and 0 should be greater than between nodes 7 and 0. However, after node grouping, $0 \rightarrow 5 \rightarrow 3$ is the only propagation path between nodes 0 and 3, which weakens the interaction strength between nodes 0 and 3. Therefore, we maintain interaction strength by integrating this multipath relationship:

$$\mathbf{R}^{(l,L)} = \sum_{i=l}^{L} \widehat{\mathbf{A}}^{(i)}. \tag{2}$$

where L is a hyperparameter limiting the maximum path length between nodes to be less than or equal to L. $\widehat{\mathbf{A}}^{(i)}_{n,m}$ measures the interaction strength between nodes n and m generated by path of length i. The interaction strength generated by each path depends on the degree of all nodes on the path. Therefore, the interaction strength of all nodes can be generalized to $\mathbf{R}^{(l,L)}$. Finally, the interaction strength of all nodes in each layer can be calculated as follows:

$$\mathbf{A}^{(l)}_{ana} = \widetilde{\mathbf{M}}^{(l)} \odot \mathbf{R}^{(l,L)}, l = 1, 2, \ldots, L. \tag{3}$$

where $\odot$ denotes Hadamard product. With this approach, we avoid the negative effects of random augmentation and can set different l to obtain low-order to high-order structural information about the nodes in the graph.

Structural Learning via Adaptive Neighbor Contrast. Previous GCL methods usually employ GNNs to reconstruct feature information in different views. However, the implied homophily or heterophily in GNNs will affect node encoding. Therefore, we use MLP to encode the original view and separate feature and structural information into different branches to optimize the utilization of both types of information. The coding process is defined as follows:

$$\mathbf{H}_0 = \mathrm{MLP}(\mathbf{X}). \tag{4}$$

where row i of $\mathbf{H}_0$ represents the representation $\mathbf{h}_i$ of node v_i. To improve the utilization of neighbor information, we would like to follow the principle of mutual information maximization, in which it is crucial to define and select positive samples. However, directly considering multi-hop information as positive anchor nodes may ignore the homophily level in different layers (e.g., Figure 2). To ensure the quality of the positive samples, we further combine the l-hop representation with adaptive weights to encode the neighbor's features:

$$\mathbf{P}^{(l)} = \gamma_l(\mathbf{A}_{ana}^{(l)}\mathbf{H}_0). \tag{5}$$

where γ denotes the trainable weight factor and $\boldsymbol{p}_i^{(l)}$ is the i-th row of $\mathbf{P}^{(l)}$. As l expands, we get a set of multi-order views $\{\mathbf{P}^{(1)}, ..., \mathbf{P}^{(l)}\}$. Then, we introduce Adaptive Neighbor Contrastive Loss (ANCLoss) to estimate the lower bound of the mutual information local to global views. It extracts contextual information by aligning node representations with l-th hop neighbor representations, similar to knowledge distillation. The ANCLoss for the l-th hop is defined as follows:

$$\mathcal{L}_{\mathrm{a}}^{(l)} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\left(\boldsymbol{h}_i \cdot \boldsymbol{p}_i^{(l)}/\tau\right)}{\sum_{v_k\in\mathcal{V}}\mathbf{1}_{[k\neq i]}\exp\left(\boldsymbol{h}_i \cdot \boldsymbol{p}_k^{(l)}/\tau\right)}. \tag{6}$$

where $\mathbf{1}_{[k\neq i]}$ denotes 1 when k is not equal to i and 0 when equal. τ is the temperature parameter, and $\exp(\cdot)$ denotes the exponential function. For each node, we use $\mathbf{A}_{ana}^{(l)}$ to adaptive find its l-layer neighbors and treat them as positive samples, while other nodes are negative. In the representation space, the loss encourages each node to extract the contextual information presented by the adaptive neighbor representation. This allows the encoder to learn strong correlations between anchor nodes and different layer nodes without recursive message passing. In addition, the adaptive weights balance the information contained in different hops. As a result, the positive sample $\boldsymbol{p}_i^{(l)}$ retains important information in l-th hop and reduces the effect of noise or redundant information. The overall structural contrastive loss is defined as:

$$\mathcal{L}_{\mathrm{a}} = \sum_{l=1}^{K}\mathcal{L}_{\mathrm{a}}^{(l)}. \tag{7}$$

Semantic Learning via original Graph. After obtaining structural information, attribute knowledge rooted in the graph is also essential for graph learning. [Wang *et al.*, 2022b] pointed out that models using only the original features of heterophilic graph nodes outperform many complex GNN models. Inspired by canonical correlation analysis [Zhang *et al.*, 2021], we propose Original Feature Analysis (OFA). For contrastive schemes, we need to specify positive and negative view samples for anchor points. However, specifying accurate negative samples for heterophilic graphs is challenging if only node features are considered. Therefore, OFA employs a feature-level self-supervised learning approach based on invariant regularization, which maximizes the mutual information between node embeddings and original features to mine more original semantic information.

First, we generate a perturbation attribute view for the original feature view. Then, the new view is fed into the MLP shared with the topology channel to obtain node representations. The detailed definition is as follows:

$$\begin{aligned}\mathbf{X}_1 &= \mathrm{FeatDrop}(\mathbf{X}, p),\\ \mathbf{H}_1 &= \mathrm{MLP}(\mathbf{X}_1).\end{aligned} \tag{8}$$

where FeatDrop denotes the feature drop operation and p is the drop probability. To avoid destroying the structure of heterophilic graphs as much as possible, we only mask some features. The embeddings of the new view and the original view are $\mathbf{H}_1$ and $\mathbf{H}_0$, respectively.

$$\mathcal{L}_o = \underbrace{\|\mathbf{H}_0 - \mathbf{H}_1\|_F^2}_{\text{invariance}} + \underbrace{\lambda\left(\left\|\mathbf{H}_0^\top\mathbf{H}_0 - \mathbf{I}\right\|_F^2 + \left\|\mathbf{H}_1^\top\mathbf{H}_1 - \mathbf{I}\right\|_F^2\right)}_{\text{decorrelation}}. \tag{9}$$

where λ is a non-negative hyperparameter for tuning the invariant and decorrelation terms. By optimizing Equation (9), the mutual information between node embeddings and original features is maximized.

Semantic Learning via latent feature graphs. Next, we utilize the similarity of nodes in the latent feature space to discover hidden associations between nodes. Nodes belonging to different classes on heterophilic graphs may exhibit similar features in the feature space. For example, in protein networks, despite the tendency of different amino acids to interact and create new proteins, amino acids still share common properties with each other. Considering the success of DeepCluster [Caron *et al.*, 2018] training, which uses feature cluster assignments as pseudo-labels, we are naturally inspired to identify potential homophilic structures by using K-means in the latent feature space to group similar nodes together. However, the non-differentiability of the K-means hard clustering process will make the optimization process tricky. To solve the problem, we transform K-means into the special Gaussian Mixture Models (GMM) [Jin *et al.*, 2022]. Specifically, by utilizing centroids $\{c_1, c_2, ..., c_k\}$ defined by the mean embedding of nodes with different hard labels, we compute a posterior probability to achieve soft clustering assignment to the original graph:

$$p\left(\boldsymbol{h}_i \mid \boldsymbol{c}_j\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\|\boldsymbol{h}_i - \boldsymbol{c}_j\|_2}{2\sigma^2}\right). \tag{10}$$

where σ^2 is the variance of Gaussian distribution. By considering an equal prior $p\left(c_1\right) = p\left(c_2\right) = ... = p\left(c_k\right)$, the probability of node feature $\boldsymbol{h}_i$ belonging to cluster c_j can be calculated by the Bayes rule as:

$$p\left(\boldsymbol{c}_j \mid \boldsymbol{h}_i\right) = \frac{p\left(\boldsymbol{c}_j\right)p\left(\boldsymbol{h}_i \mid \boldsymbol{c}_j\right)}{\sum_{r=1}^{k}p\left(\boldsymbol{c}_r\right)p\left(\boldsymbol{h}_i \mid \boldsymbol{c}_r\right)} = \frac{\exp\left(-\frac{\left(\boldsymbol{h}_i-\boldsymbol{c}_j\right)^2}{2\sigma^2}\right)}{\sum_{r=1}^{k}\exp\left(-\frac{\left(\boldsymbol{h}_i-\boldsymbol{c}_r\right)^2}{2\sigma^2}\right)} \tag{11}$$

In this way, we can get a cluster assignment matrix $\mathbf{R} \in \mathbb{R}^{N\times k}$ where $\mathbf{R}_{ij} = p\left(c_j \mid \boldsymbol{h}_i\right)$ indicates the soft clustering value between node v_i and cluster c_j. Then we can construct the latent feature loss function (LFLoss) as follows:

$$\mathcal{L}_{\mathrm{lf}} = \frac{1}{k|\mathcal{E}|}\sum_{r=1}^{k}\sum_{(v_i,v_j)\in\mathcal{E}}\mathrm{MSE}\left(p\left(\boldsymbol{c}_r \mid \boldsymbol{h}_i\right), p\left(\boldsymbol{c}_r \mid \boldsymbol{h}_j\right)\right). \tag{12}$$

where MSE$(\cdot)$ is the Mean Square Error.

	Heterophily				Homophliy		
Dataset	**Cornell**	**Texas**	**Wisconsin**	**Actor**	**Cora**	**Citeseer**	**Pubmed**
Nodes	183	183	251	7600	2708	3327	19717
Edges	277	279	499	26752	5278	4676	44324
Features	1703	1703	1703	931	1433	3703	500
Classes	5	5	5	5	6	7	3
H.R	0.131	0.108	0.196	0.219	0.810	0.736	0.802
Edge Density	0.0179	0.0194	0.0164	0.0010	0.0014	0.0008	0.0002

Table 1: Benchmark dataset statistics.

4.3 Model Training

We jointly optimize contrastive losses to train HeterGCL end-to-end. The overall objective function is defined as follows:

$$\mathcal{L} = \alpha\mathcal{L}_a + (1-\alpha)(\mathcal{L}_o + \mathcal{L}_{\text{lf}}). \tag{13}$$

where α is a weight factor. Our goal is to optimize node representations for downstream tasks by minimizing $\mathcal{L}$.

5 Experiments

5.1 Datasets and Experimental Settings

Datasets. We evaluated the performance of HeterGCL and existing methods on seven representative homophilic or heterophilic datasets. Specifically, for the heterophilic datasets, we select three webpage datasets, Cornell, Texas, Wisconsin and an actor co-occurrence network, Actor [Pei *et al.*, 2020]. For the homophilic dataset, we select the widely used standard citation network datasets Cora, Citeseer, and Pubmed [Sen *et al.*, 2008], where nodes and edges denote document and citation relationships, respectively. In addition, we calculate the homophilic ratio of graphs by the following metrics:

$$H.R = \frac{|\{(u,v) : (u,v) \in \mathcal{E} \wedge y_u = y_v\}|}{|\mathcal{E}|}. \tag{14}$$

where $H.R \in [0,1]$, is the fraction of edges with connected nodes of the same class. When $H.R$ is closer to 1, the graph is more homophilic. Conversely, the graph is more heterophilic. Table 1 gives detailed statistics for all datasets.

Baselines for Comparison. To demonstrate the effectiveness and scalability of our method, we compare HeterGCL with several groups of representative baselinses on node classification and node clustering, including four self-supervised models Grace, MVGRL [Hassani and Khasahmadi, 2020], BGRL [Thakoor *et al.*, 2022] and SELENE [Zhong *et al.*, 2022] (design for heterophilic graphs), eight supervised baselines, including GCN [Kipf and Welling, 2017], GAT [Velickovic *et al.*, 2018], SGC [Wu *et al.*, 2019], JKNET [Xu *et al.*, 2018], GCNII [Chen *et al.*, 2020], MixHop [Abu-El-Haija *et al.*, 2019], H2GCN [Zhu *et al.*, 2020], and GPRGNN [Chien *et al.*, 2021] . For the node clustering task, we add feature- or structure-based network embedding models AE [Hinton and Salakhutdinov, 2006], Struct2vec [Ribeiro *et al.*, 2017], LINE [Tang *et al.*, 2015], VGAE [Kipf and Welling, 2016], SDCN [Bo *et al.*, 2020],DGI [Velickovic *et al.*, 2019], GMI [Peng *et al.*, 2020] and FAGCN [Bo *et al.*, 2021].

Implementation. We adopt the strictly unsupervised scheme and transductive setting to pre-train node representations. We fed the embedding obtained from HeterGCL into a logistic regression classifier to learn the node representations for node classification. Each dataset is randomly split into training/validation/test sets with $10\%/10\%/80\%$. We run each model 10 times and report the average accuracy. For node clustering, we use the K-means algorithm and select 3 evaluation metrics of accuracy (ACC), normalized mutual information (NMI), and average rand index (ARI). The number of clusters is set to the number of ground truth classes. The experiments are conducted on a single NVIDIA GeForce RTX 3090 machine. In addition, we perform a grid search to tune the hyperparameters and use the Adam optimizer to select the learning rate to train the model from $\{5\text{e}-3, 6\text{e}-3, 2\text{e}-2\}$. For HeterGCL, we search for λ and α in steps of 0.01 from 0 to 5. The dropout rate is searched from $\{0, 0.1, 0.5\}$. The weight decay is adjusted from $\{5\text{e}-4, 5\text{e}-3, 3\text{e}-3\}$. L is searched in steps 1 from 1 to 10.

5.2 Node Classification Results

Table 2 summarizes the results of node classification. We can see that HeterGCL outperforms all baseline models on the heterophilic graph and achieves the best performance. We attribute this superiority to the following factors: (1) HeterGCL changes the MP strategy unsuitable for heterophilic graphs to avoid mixing multi-hop information. (2) HeterGCL better mines the semantic information in the original node features. (3) Using MLP as an encoder to process structure and feature information separately avoids the low expressiveness of mixing structure and feature information in GNN encoders in the heterophilic environment. With the help of these extra information, HeterGCL even outperforms supervised GNNs. In addition, although HeterGCL does not achieve the best performance on homophilic graphs, it is still competitive. In fact, most baselines are designed for homophilic graphs and do not apply to heterophilic graphs, which supports our motivation. Similar evidence is that for baselines such as JKNET and MixHop, which also utilize multi-hop neighbor information, the multi-hop aggregation scheme in HeterGCL yields better performance. Finally, compared to the self-supervised baselines, HeterGCL is more general in different graphs.

5.3 Node Clustering Results

Table 3 shows the node clustering results. First, compared to the baseline models, HeterGCL achieves significantly superior clustering performance overall across the three clustering evaluation metrics. Second, only attribute-based AutoEncoder (AE) performs better than structure-based network embedding methods on heterophilic graphs. This empirical validation demonstrates the importance of node initial features learning on heterophilic graphs. Finally, HeterGCL significantly outperforms baselines such as VGAE and SELENE, which stems from the fact that HeterGCL explores the union of initial features and global semantics. It is not available with existing methods. In addition, GCL models generally outperform traditional network embedding methods, again demonstrating the superiority of the GCL pattern.

5.4 Ablation Study

We set up an ablation study to explore the contribution of different components in HeterGCL. Specifically, We construct the following variants: (1) Remove structure learning

Method	**Avaliable data**	**Cornell**	**Texas**	**Wisconsin**	**Actor**	**Cora**	**Citeseer**	**Pubmed**
GCN	X,A,Y	54.8 ± 2.9	58.4 ± 4.2	51.8 ± 6.0	28.2 ± 0.4	83.2 ± 1.2	70.2 ± 1.1	85.2 ± 0.1
GAT	X,A,Y	54.8 ± 3.1	56.7 ± 3.1	51.0 ± 4.3	26.9 ± 0.8	83.7 ± 1.0	70.6 ± 0.8	84.7 ± 0.3
SGC	X,A,Y	48.8 ± 7.1	53.8 ± 8.9	44.3 ± 5.9	25.3 ± 0.5	82.2 ± 1.1	71.4 ± 0.9	81.3 ± 0.5
JKNET	X,A,Y	45.3 ± 3.2	61.4 ± 3.5	56.0 ± 4.3	26.8 ± 1.0	80.5 ± 0.8	65.7 ± 1.8	84.8 ± 0.2
GCNII	X,A,Y	64.3 ± 5.7	61.2 ± 9.8	60.7 ± 7.6	32.6 ± 0.5	84.4 ± 1.6	71.9 ± 1.6	**89.1 ± 0.5**
MixHop	X,A,Y	52.8 ± 6.3	55.5 ± 3.3	51.5 ± 5.4	29.0 ± 1.0	81.0 ± 1.6	66.4 ± 1.8	84.9 ± 0.5
H2GCN	X,A,Y	62.4 ± 5.8	64.1 ± 7.8	63.5 ± 7.2	33.8 ± 0.4	81.4 ± 1.2	71.8 ± 0.9	85.7 ± 0.3
GPRGNN	X,A,Y	63.4 ± 8.4	66.4 ± 7.8	65.2 ± 8.4	33.3 ± 0.7	**85.2 ± 1.1**	72.5 ± 0.8	87.6 ± 0.3
GRACE	X,A	56.4 ± 2.1	63.5 ± 2.6	53.8 ± 3.6	28.1 ± 0.8	83.7 ± 0.7	71.4 ± 1.0	77.6 ± 1.0
MVGRL	X,A	56.2 ± 2.4	61.7 ± 3.9	50.6 ± 5.9	31.4 ± 0.8	83.5 ± 1.1	72.3 ± 0.7	80.1 ± 0.7
BGRL	X,A	56.7 ± 2.1	65.8 ± 2.7	59.8 ± 4.1	29.8 ± 0.3	83.0 ± 0.7	72.3 ± 0.6	84.7 ± 0.4
SELENE	X,A	56.1 ± 2.5	64.0 ± 1.7	55.5 ± 4.8	33.2 ± 0.4	56.2 ± 1.5	54.1 ± 1.1	81.7 ± 0.3
HeterGCL	X,A	**75.5 ± 2.8**	**74.7 ± 3.6**	**75.6 ± 4.5**	**37.2 ± 0.4**	83.0 ± 0.8	**73.0 ± 0.6**	86.2 ± 0.2
w/o ANCLoss	-	67.4 ± 5.5	68.8 ± 3.2	71.5 ± 4.3	35.2 ± 0.5	70.4 ± 1.1	63.4 ± 0.7	78.5 ± 0.4
w/o OFALoss	-	66.1 ± 5.5	69.8 ± 4.2	65.6 ± 6.5	36.4 ± 0.4	82.7 ± 0.9	72.3 ± 0.6	86.2 ± 0.2
w/o ANA	-	72.5 ± 4.6	72.3 ± 2.9	74.0 ± 4.2	35.2 ± 0.5	82.4 ± 1.1	71.7 ± 0.8	85.5 ± 0.2
w/o LFLoss	-	73.9 ± 3.8	73.8 ± 2.5	73.8 ± 2.5	36.8 ± 0.4	82.8 ± 0.8	72.7 ± 0.7	86.0 ± 0.2

Table 2: Overall results for classification accuracy. The best result is in bold, and the second best is underlined.

Method	**Cornell**			**Texas**			**Actor**			**Citeseer**		
	ACC	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**
AE	52.2 ± 0.0	17.1 ± 0.0	17.4 ± 0.0	50.5 ± 0.0	16.6 ± 0.0	14.6 ± 0.0	24.2 ± 0.1	1.0 ± 0.0	0.5 ± 0.0	58.8 ± 0.2	30.9 ± 0.2	30.3 ± 0.2
Struct2vec	32.7 ± 0.0	1.5 ± 0.0	-2.2 ± 0.0	49.7 ± 0.0	18.6 ± 0.0	21.0 ± 0.0	22.4 ± 0.3	0.1 ± 0.0	-0.1 ± 0.0	21.2 ± 0.5	1.2 ± 0.1	0.2 ± 0.1
LINE	34.1 ± 0.8	2.9 ± 0.2	-1.5 ± 0.3	49.4 ± 2.1	16.9 ± 1.6	18.1 ± 1.1	22.7 ± 0.1	0.1 ± 0.0	0.1 ± 0.0	28.4 ± 0.9	8.5 ± 0.7	3.5 ± 0.6
VGAE	43.4 ± 1.0	5.5 ± 0.5	4.0 ± 0.5	50.3 ± 1.9	11.7 ± 1.0	21.5 ± 1.8	23.3 ± 0.2	0.2 ± 0.0	0.3 ± 0.1	55.7 ± 0.1	32.5 ± 0.1	28.3 ± 0.1
SDCN	36.9 ± 2.0	6.6 ± 0.9	3.4 ± 1.3	44.0 ± 0.6	14.2 ± 1.9	10.7 ± 3.0	23.7 ± 0.3	0.1 ± 0.1	0.0 ± 0.1	59.9 ± 1.2	30.4 ± 0.8	29.7 ± 1.3
DGI	44.1 ± 2.7	5.8 ± 0.8	4.9 ± 2.0	55.7 ± 0.7	8.7 ± 3.6	8.3 ± 6.8	24.3 ± 0.1	1.4 ± 0.0	0.1 ± 0.0	58.9 ± 0.4	32.6 ± 0.4	33.2 ± 0.6
GMI	33.6 ± 2.1	5.3 ± 1.3	3.1 ± 0.9	35.2 ± 1.2	7.7 ± 0.9	3.0± 0.6	26.2 ± 0.0	0.2 ± 0.0	0.4 ± 0.0	59.0 ± 0.0	32.1 ± 0.0	33.1 ± 0.0
FAGCN	56.2 ± 8.3	17.1 ± 4.0	19.9 ± 13.9	57.9 ± 6.5	23.4 ± 9.0	22.5 ± 10.9	25.6 ± 0.1	3.2 ± 0.1	0.3 ± 0.1	47.4 ± 0.3	20.2 ± 0.3	17.9 ± 0.2
GRACE	43.6 ± 4.6	8.2 ± 1.2	6.4 ± 2.0	57.0 ± 2.2	20.7 ± 1.0	29.5 ± 4.2	25.9 ± 0.5	0.6 ± 0.3	0.9 ± 0.4	54.7 ± 5.4	31.7 ± 3.8	27.4 ± 5.6
MVGRL	43.8 ± 3.0	8.4 ± 2.8	7.1 ± 3.0	62.8 ± 2.3	25.7 ± 1.8	33.5 ± 4.6	28.6 ± 1.0	2.4 ± 0.5	2.8 ± 0.6	45.8 ± 9.1	23.4 ± 7.7	19.9 ± 7.9
BGRL	55.1 ± 1.7	8.0 ± 0.5	3.9 ± 1.0	58.7 ± 1.8	22.0 ± 2.4	23.7 ± 2.3	28.2 ± 0.3	1.8 ± 0.2	2.4 ± 0.1	64.3 ± 1.7	36.6 ± 1.7	36.7 ± 1.9
SELENE	57.8 ± 4.7	17.0 ± 3.5	22.9 ± 5.1	**64.5 ± 4.4**	25.2 ± 8.1	**34.2 ± 11.5**	28.2 ± 0.3	4.7 ± 0.7	1.8 ± 0.1	59.2 ± 2.5	29.9 ± 2.2	29.4 ± 3.1
HeterGCL	**62.9 ± 2.7**	**29.2 ± 3.5**	**33.5 ± 5.2**	63.1 ± 1.3	**39.0 ± 2.8**	31.3 ± 3.5	**30.9 ± 1.0**	**4.9 ± 1.2**	**5.0 ± 1.2**	**68.1 ± 0.8**	**43.6 ± 0.9**	**43.1 ± 1.1**

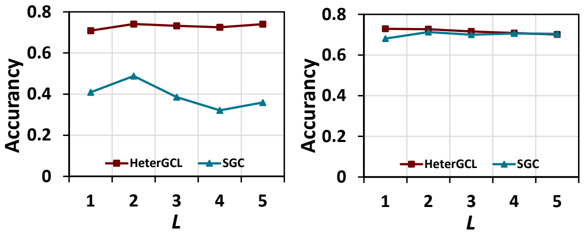
Table 3: Overall results for node clustering. The best result is in bold, and the second-best result is underlined.

(**w/o ANCLoss**); (2) Remove original semantic learning (**w/o OFALoss**); (3) Adaptive Neighbor Aggregation replaced by traditional MP aggregation i.e., $\mathbf{A}_{ana}^{(l)}$ by renormalized neighborhood matrix $\widehat{\mathbf{A}}$ (**w/o ANA**); (4)Remove latent features Semantic Learning (**w/o LFLoss**);

As shown in Table 2, we observe that all components contribute to the performance improvement of HeterGCL. Removing $\mathcal{L}_o$ did not affect the model performance significantly on heterophilic graph Actor. However, on Cornell, Texas, and Wisconsin $\mathcal{L}_o$ plays a significant role, indicating that node features are more helpful on these three small datasets. In contrast, $\mathcal{L}_a$ is more contributive to the larger Actor dataset, which captures more information about nodes with similar features and local structure. In addition, ANA improves performance on all datasets compared to traditional recursive aggregation, demonstrating that our aggregation scheme is more promising for discovering valuable neighbor nodes. Finally, we find that the effect of LFLoss is greater on heterophilic graphs, which supports our conjecture that even though neighbor nodes in heterophilic graph do not belong to the same class, they may still have semantic similarity.

5.5 Effect of ANA Scope

As shown in Figure.5, we explored the role of ANA on structure learning in a larger neighbor range. We take the SGC with traditional recursive aggregation as a baseline and show the performance of the model under different L. First, the performance curve of HeterGCL on the heterophilic graph Cornell is stable, suggesting that the contrastive pattern of HeterGCL can adapt to heterophilic graphs. In contrast, SGC with recursive MP does not work in slightly deeper layers. It demonstrates the effectiveness of ANA in making rational use of neighbor information. Second, as L increases, the performance of HeterGCL also decreases because the number of homophilic nodes on Citeseer is decreasing. On the contrary, the performance of the SGC improves slightly, benefiting from the additional information brought by the labels.



(a) Cornell. (b) Citeseer.

Figure 5: Impact analysis of L in structural augmentation.

6 Conclusion

In this paper, we focus on self-supervised representation learning for heterophilic graphs and propose a novel network embedding framework HeterGCL. We effectively fuse node features and graph topology information through three contrastive losses. In addition, we propose ANA to guide information propagation to construct self-supervised signals, avoiding information mixing on heterophilic graphs. Extensive experiments show that HeterGCL has better downstream performance on homophilic and heterophilic graphs.

Acknowledgements

This work was supported by the Natural Science Foundation of Heilongjiang Province in China (No. LH2020F043).

References

[Abu-El-Haija *et al.*, 2019] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pages 21–29. PMLR, 2019.

[Bo *et al.*, 2020] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1400–1410. ACM / IW3C2, 2020.

[Bo *et al.*, 2021] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3950–3957, 2021.

[Caron *et al.*, 2018] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV*, volume 11218 of *Lecture Notes in Computer Science*, pages 139–156. Springer, 2018.

[Chen *et al.*, 2020] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International conference on machine learning*, pages 1725–1735. PMLR, 2020.

[Chien *et al.*, 2021] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *9th International Conference on Learning Representations, ICLR*, 2021.

[Hassani and Khasahmadi, 2020] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pages 4116–4126. PMLR, 2020.

[Hinton and Salakhutdinov, 2006] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. 313:504–507, 2006.

[Jin *et al.*, 2021] Wei Jin, Tyler Derr, Yiqi Wang, Yao Ma, Zitao Liu, and Jiliang Tang. Node similarity preserving graph convolutional networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 148–156, 2021.

[Jin *et al.*, 2022] Wei Jin, Xiaorui Liu, Xiangyu Zhao, Yao Ma, Neil Shah, and Jiliang Tang. Automated self-supervised learning for graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations, ICLR*, 2017.

[Klicpera *et al.*, 2019] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13333–13345, 2019.

[Lee *et al.*, 2022] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 7372–7380. AAAI Press, 2022.

[Li *et al.*, 2023] Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. Homogcl: Rethinking homophily in graph contrastive learning. In Ambuj Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye, editors, *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 1341–1352. ACM, 2023.

[Liu *et al.*, 2020] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 338–348, 2020.

[Liu *et al.*, 2024a] Meng Liu, Ke Liang, Yawei Zhao, Wenxuan Tu, Sihang Zhou, Xinbiao Gan, Xinwang Liu, and He Kunlun. Self-supervised temporal graph learning with temporal and structural intensity alignment. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Liu *et al.*, 2024b] Meng Liu, Yue Liu, Ke Liang, Wenxuan Tu, Siwei Wang, Sihang Zhou, and Xinwang Liu. Deep temporal graph clustering. In *The 12th International Conference on Learning Representations*, 2024.

[Pei *et al.*, 2020] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR*, 2020.

[Peng *et al.*, 2020] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 259–270. ACM / IW3C2, 2020.

[Qiu *et al.*, 2018] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 459–467, 2018.

[Ribeiro *et al.*, 2017] Leonardo Filipe Rodrigues Ribeiro, Pedro H. P. Saverese, and Daniel R. Figueiredo. *struc2vec*: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 385–394. ACM, 2017.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008.

[Sun *et al.*, 2020] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *8th International Conference on Learning Representations, ICLR*, 2020.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1067–1077. ACM, 2015.

[Thakoor *et al.*, 2022] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. 2022.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR*, 2018.

[Velickovic *et al.*, 2019] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *7th International Conference on Learning Representations, ICLR*, 2019.

[Wang *et al.*, 2022a] Haonan Wang, Jieyu Zhang, Qi Zhu, and Wei Huang. Augmentation-free graph contrastive learning with performance guarantee. *arXiv preprint arXiv:2204.04874*, 2022.

[Wang *et al.*, 2022b] Tao Wang, Di Jin, Rui Wang, Dongxiao He, and Yuxiao Huang. Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 4210–4218, 2022.

[Wang *et al.*, 2022c] Yanling Wang, Jing Zhang, Haoyang Li, Yuxiao Dong, Hongzhi Yin, Cuiping Li, and Hong Chen. Clusterscl: Cluster-aware supervised contrastive learning on graphs. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1611–1621. ACM, 2022.

[Wang *et al.*, 2023] Chenhao Wang, Yong Liu, and Yan Yang. Network embedding with adaptive multi-hop contrast. In Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos, editors, *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 4330–4334. ACM, 2023.

[Wu *et al.*, 2019] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019.

[Xu *et al.*, 2018] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International conference on machine learning*, pages 5453–5462. PMLR, 2018.

[Xu *et al.*, 2023] Zhe Xu, Yuzhong Chen, Qinghai Zhou, Yuhang Wu, Menghai Pan, Hao Yang, and Hanghang Tong. Node classification beyond homophily: Towards a general solution. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2862–2873, 2023.

[Zhang *et al.*, 2018] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2778–2786. ACM, 2018.

[Zhang *et al.*, 2021] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *Advances in Neural Information Processing Systems*, 34:76–89, 2021.

[Zhong *et al.*, 2022] Zhiqiang Zhong, Guadalupe Gonzalez, Daniele Grattarola, and Jun Pang. Unsupervised heterophilous network embedding via r-ego network discrimination. *CoRR*, abs/2203.10866, 2022.

[Zhu *et al.*, 2020] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.