Universal Graph Self-Contrastive Learning

Liang Yang[1], **Yukun Cai**[1], **Hui Ning**[1], **Jiaming Zhuo**[1*], **Di Jin**[2], **Ziyi Ma**[1], **Yuanfang Guo**[3], **Chuan Wang**[4] and **Zhen Wang**[5]
[1]Hebei Province Key Laboratory of Big Data Calculation, School of Artificial Intelligence, Hebei University of Technology, Tianjin, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin, China
[3]School of Computer Science and Engineering, Beihang University, Beijing, China
[4]School of Computer Science and Technology, Beijing JiaoTong University, Beijing, China
[5]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), School of Cybersecurity, Northwestern Polytechnical University, Xi'an, China
yangliang@vip.qq.com, 202222802050@stu.hebut.edu.cn, ninghui048@163.com, jiaming.zhuo@outlook.com, jindi@tju.edu.cn, zyma@hebut.edu.cn, andyguo@buaa.edu.cn, wangchuan@iie.ac.cn, w-zhen@nwpu.edu.cn

Abstract

As a pivotal architecture in Self-Supervised Learning (SSL), Graph Contrastive Learning (GCL) has demonstrated substantial application value in scenarios with limited labeled nodes (samples). However, existing GCLs encounter critical issues in the graph augmentation and positive and negative sampling stemming from the lack of explicit supervision, which collectively restrict their efficiency and universality. On the one hand, the reliance on graph augmentations in existing GCLs can lead to increased training times and memory usage, while potentially compromising the semantic integrity. On the other hand, the difficulty in selecting TRUE positive and negative samples for GCLs limits their universality to both homophilic and heterophilic graphs. To address these drawbacks, this paper introduces a novel GCL framework called GRAph learning via Self-contraSt (GRASS). The core mechanism is node-attribute self-contrast, which specifically involves increasing the feature similarities between nodes and their included attributes while decreasing the similarities between nodes and their non-included attributes. Theoretically, the self-contrast mechanism implicitly ensures accurate node-node contrast by capturing high-hop co-inclusion relationships, thereby enabling GRASS to be universally applicable to graphs with varying degrees of homophily. Evaluations on diverse benchmark datasets demonstrate the universality and efficiency of GRASS.

1 Introduction

Graph Self-Supervised Learning (GSSL) has made significant advancements in learning discriminative representations without relying on expensive hand-labeled nodes (samples) [Hendrycks *et al.*, 2019]. As a representative architecture of GSSL, Graph Contrastive Learning (GCL) aims to maximize the agreement between the representations of two augmented views from the same input while minimizing the agreement between the representations from different augmented views. It has achieved remarkable performance on various node-level downstream tasks, such as node classification [Ma *et al.*, 2023]. Depending on the different information contained in contrastive objects, existing GCLs for node-level tasks can be categorized into: 1) local-global contrast [Velickovic *et al.*, 2019; Hassani and Khasahmadi, 2020] and 2) local-local contrast [Zhu *et al.*, 2020b; Thakoor *et al.*, 2021; Zhuo *et al.*, 2024a; Zhuo *et al.*, 2024c]. Given that the fundamental objective is to contrast nodes, these two categories of GCLs can be collectively termed as *Node-Node Contrast*.

Unfortunately, existing node-node graph contrastive models still face critical issues in the *graph augmentation* and *positive/negative sampling* due to the lack of supervision information. First, graph augmentation not only faces the challenge of maintaining the semantic integrity of the graph during the augmentation process but also inevitably increases computational costs [Thakoor *et al.*, 2022; Trivedi *et al.*, 2022]. Despite some methods successfully working without graph augmentations by selecting positive samples from neighboring nodes [Zhang *et al.*, 2022; Xiao *et al.*, 2022], challenges persist in accurately identifying TRUE positive and negative samples for nodes. On the one hand, positive samples, whose representations will be pulled together, can not be easily identified. Neighborhoods are from the same classes with a high probability on homophilic graphs, and thus can be treated as positive samples. However, this is not feasible on heterophilic graphs. On the other hand, it is more difficult to locally identify negative samples. Negative samples should be the nodes in the different classes. It is commonly accepted that nodes with long-distance may belong to different classes. Therefore, this compromises the model's

*Corresponding Author

universality to graphs with diverse structural characteristics.

To solve the aforementioned drawbacks, this paper seeks to introduce a universal augmentation-free GCL framework. Motivated by the Self-contrastive Learning [Bae *et al.*, 2023], which leverages a multi-exit architecture to generate multiple representations from a single image without relying on data augmentation, this paper aims to explore self-contrastive learning on graphs. To this end, a novel Graph Self-contrastive Learning framework named GRAph learning via Self-contraSt (GRASS) is proposed. It enables self-contrast between nodes and attributes, thereby eliminating the reliance on augmentation and circumventing the challenges associated with sample selection. GRASS first treats attributes as another type of node, *i.e.*, attribute nodes, and generates initial representations for each type of node separately. The representations of nodes and attribute nodes are respectively derived from the adjacency matrix, which describes the link relationships between nodes, and the Positive Pointwise Mutual Information (PPMI) matrix, which captures the co-occurrence relationships of attributes. Then, it identifies positive pairs via inclusion relationships and negative pairs via non-inclusion relationships in the attribute matrix for representation learning. In theory, the framework can implicitly capture the node-node co-inclusion relationship (a high-hop relationship), which guarantees its universality.

The main contributions of this paper are as follows:

- We investigate the challenges faced by existing Graph Contrastive Learning (GCLs) in terms of graph augmentation and positive/negative sample selection.
- We introduce a novel GCL framework named GRAph learning via Self-contraSt (GRASS), with a node-attribute self-contrastive mechanism.
- We provide a theoretical analysis to justify the effectiveness and universality of GRASS.
- We conduct extensive experiments on twelve well-known benchmark datasets with various homophily degrees to demonstrate the performance of GRASS.

2 Related Works

Contrastive Learning. Contrastive Learning (CL) has become highly popular in self-supervised visual representation learning, aiming to learn discriminative representations by contrasting positive and negative sample pairs. Specifically, SimCLR [Chen *et al.*, 2020] maximizes the consistency between representations of different augmented views of the same image; BYOL [Grill *et al.*, 2020] removes negative pairs, using an online network to predict the representations produced by a target network. More recently, CM-SCGC [Guan *et al.*, 2024] has constructed two views by extracting the pixel neighborhood texture information and spatial-spectral information from hyperspectral images.

Augmentation-based Graph Contrastive Learning. CL has also been successful on graphs. GRACE [Zhu *et al.*, 2020b] generates augmented graphs by randomly dropping edges or nodes and maximizes the consistency of the node representations from the two views; GCA [Zhu *et al.*, 2021] uses an adaptive augmentation method to perturb the unimportant information; MVGRL [Hassani and Khasahmadi, 2020] generates the multiple views via graph diffusion and learns both node-level and graph-level representations. GREET [Liu *et al.*, 2023] introduces a discriminator to assess the homophily of edges and implements random augmentation. AGCL [Yu and Jia, 2024] generates multiple views via graph topology and graph diffusion, capturing local and global information. GOUDA [Zhuo *et al.*, 2024b] makes utilization of augmentation-centric vectors to simulate attribute variations in node neighborhoods, thereby accomplishing a unified augmentation framework.

Augmentation-free Graph Contrastive Learning. Some Graph Contrastive Learning (GCL) models without augmentation are given below. SimGRACE [Xia *et al.*, 2022] takes the original graph as input and Graph Neural Network (GNN) model with its perturbed version as two encoders to obtain two correlated views for contrast. AF-GCL [Wang *et al.*, 2022] leverages the features aggregated by GNN to construct the self-supervision signal instead of augmentations. DSSL [Xiao *et al.*, 2022] imitates a generative process of nodes and links decoupling the potential semantics of different neighborhoods. GCFormer [Chen *et al.*, 2024] develops a new token generator to generate both positive and negative token sequences for each node.

3 Notations and Preliminaries

3.1 Notations

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with node set $\mathcal{V} = \{v_1, ..., v_n\}$ and edge set $\mathcal{E}$, where n denotes the number of nodes. The adjacency matrix $\mathbf{A} = [a_{ij}] \in \{0, 1\}^{n \times n}$ represents the graph topology, where $a_{ij} = 1$ if and only if there exists an edge $e_{ij} = (v_i, v_j)$. The degree matrix $\mathbf{D}$ is a diagonal matrix with diagonal element $d_i = \sum_{i=1}^{n} a_{ij}$ as the degree of node v_i. $\mathbf{X} \in \mathbb{R}^{n \times f}$ denotes the collections of node attributes with the ith rows, *i.e.*, $x_i \in \mathbb{R}^f$ corresponding to node v_i, where f stands for the dimension of the attribute.

3.2 Homophily

Typically, the graph edge homophily ratio [Pei *et al.*, 2020; Zhu *et al.*, 2020a] is defined as the proportion of edges connecting nodes with the same labels. Mathematically, this can be represented by the following equation:

$$h = \frac{\left|\left\{(v_i, v_j) : (v_i, v_j) \in \mathcal{E} \wedge y_{v_i} = y_{v_j}\right\}\right|}{|\mathcal{E}|}. \tag{1}$$

Homophilic graphs have a high edge homophily ratio, *i.e.*, h tends to 1, while heterophilic graphs correspond to a small edge homophily ratio, *i.e.*, h tends to 0. This paper particularly examines graphs with varying degrees of homophily.

3.3 Positive Pointwise Mutual Information

Positive Pointwise Mutual Information (PPMI) is a common tool for measuring the association between two words in computational linguistics [Church and Hanks, 1990]. PPMI between words w_i and w_j is defined by:

$$\mathrm{PPMI}(w_i, w_j) = \left[\log \frac{\mathrm{P}(w_i, w_j)}{\mathrm{P}(w_i)\mathrm{P}(w_j)}\right]_+. \tag{2}$$

Here, $[\mathrm{p}]_+$ denotes $\max(\mathrm{p}, 0)$. $\mathrm{P}(w_i)$ is the occurrence probability of the word w_i, and $\mathrm{P}(w_i, w_j)$ is the co-occurrence probability of two words. If two words are completely independent, $\mathrm{P}(w_i, w_j) = \mathrm{P}(w_i)\mathrm{P}(w_j)$, and the value of PPMI is 0. In contrast, if there is a strong relationship between two words, they will usually co-occur frequently and thus share a high PPMI value, indicating a close association. Given this property, PPMI is employed in our framework to represent the correlations among attributes.

3.4 Contrastive Learning Loss

The key to contrastive learning (CL) is maximizing mutual information (MI) [Tsai *et al.*, 2021]. To this end, an information theory viewpoint is adopted to construct the Graph Contrastive Learning (GCL) objective. Let $I(X;Y)$ be the MI between random variables X and Y, which is commonly represented by the Donsker-Varadhan variational representation [Donsker and Varadhan, 1983; Polyanskiy and Wu, 2014] as follows:

$$I(X;Y) \geq I_\Theta(X,Y), \tag{3}$$

where $I_\Theta(X,Y)$ is the neural information with parameters $\theta \in \Theta$, which is precisely given by the following detailed and comprehensive definition:

$$I_\Theta(X,Y) = \sup_{\theta\in\Theta} \mathbb{E}_{\mathbb{P}_{XY}}[T_\theta] - \log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Y}[e^{T_\theta}]). \tag{4}$$

Based on this variational lower bound, the contrastive loss is formulated by directly optimizing $I_\Theta(X,Y)$:

$$\mathcal{L}_{GCL} = -\mathbb{E}_{\mathbb{P}_{XY}}[T_\theta] + \log(\mathbb{E}_{\mathbb{P}_X \otimes \mathbb{P}_Y}[e^{T_\theta}]). \tag{5}$$

4 Methodology

This section first introduces a novel graph contrastive learning (GCL) framework named GRAph learning via Self-contraSt (GRASS). Subsequently, it theoretically analyzes the effectiveness of this framework.

4.1 GRASS

As shown in Figure 1, the architecture of GRASS consists of three key modules: (1) *View Construction*, which extracts two types of contrastive objects from the input graph. (2) *Sampling Matrix Construction*, which determines positive samples for self-contrast based on the node-attribute inclusion relationship (represented by a bipartite graph). (3) *Graph Self-contrastive Learning Loss*, which guides the feature update process for these two types of objects.

View Construction. As discussed in the Introduction, traditional GCL methods rely on perturbation-based augmentations to construct two views, which may result in expensive training costs and compromised semantic integrity. The proposed view construction module regards nodes and attributes as two intrinsic views of the input graph, thus avoiding the need for perturbation-based augmentations. To be specific, for the input graph $\mathcal{G}(\mathbf{A}, \mathbf{X})$, this module generates representations for attributes in addition to nodes, represented as $\mathbf{Z} \in \mathbb{R}^{n\times d}$ and $\mathbf{B} \in \mathbb{R}^{f\times d}$.

The initial node representations are derived through the embedding of the graph topology matrix, while the initial attribute representations are obtained via the embedding of the attribute co-occurrence matrix. The graph topology matrix is selected as the adjacency matrix $\mathbf{A}$. The attribute co-occurrence matrix is constructed through the utilization of Positive Pointwise Mutual Information (PPMI), which is defined as $\mathbf{M}_{i,j} = \mathrm{PPMI}(\mathbf{X}_{:,i}, \mathbf{X}_{:,j})$.

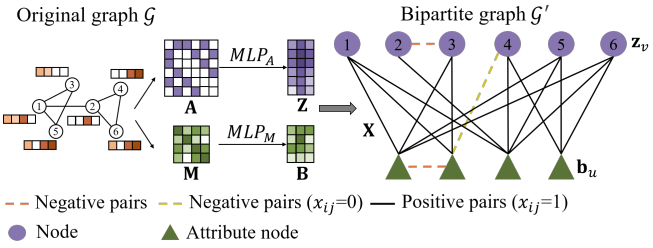

Figure 1: Illustration of the proposed framework GRASS. Firstly, the initial representations are derived through the embedding of the graph topology matrix and the attribute co-occurrence matrix. Secondly, the attributes are regarded as another type of node, and the two types of nodes are connected based on non-zero entries in the attribute matrix $\mathbf{X}$, forming a node-attribute bipartite graph. Finally, positive and negative sample pairs are selected on the bipartite graph to construct the contrastive learning objective function.

After obtaining these matrices, the initial representations are computed by applying Multi-Layer Perceptrons (MLPs) to transform the graph topology matrix and the attribute co-occurrence matrix into feature embeddings, that is:

$$\mathbf{Z} = MLP_A(\mathbf{A}), \quad \mathbf{B} = MLP_M(\mathbf{M}), \tag{6}$$

where MLP_A and MLP_M stand for two distinct Multi-Layer Perceptron (MLP). In this paper, the attributes are regarded as another type of node, referred to as *attribute nodes*.

Sampling Matrix Construction. It is commonly assumed that the attributes associated with nodes can accurately reflect the node categories [Lim *et al.*, 2021]. Based on this assumption, this module aims to identify positive samples for nodes and attribute nodes by constructing a node-attribute bipartite graph $\mathcal{G}'$ using the attribute matrix $\mathbf{X}$.

To ensure symmetry, the adjacency matrix of the bipartite graph, *i.e.*, $\mathbf{A}' \in \mathbb{R}^{(n+f)\times(n+f)}$, is defined as:

$$\mathbf{A}' = \begin{bmatrix} \mathbf{0} & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{0} \end{bmatrix}, \tag{7}$$

where $\mathbf{0}$ denotes an all-zero matrix. The adjacency matrix indicates the node-attribute inclusion relationship, which is regarded as the basis for identifying positive samples. Specifically, for a given type of node, the positive samples are identified as the other type of nodes where the corresponding weight in the adjacency matrix is non-zero.

Graph Self-contrastive Learning Loss. In general, contrastive losses aim to amplify feature similarities between positive sample pairs while reducing them between negative sample pairs [Xiao *et al.*, 2022; Zhu *et al.*, 2020b]. In the proposed framework GRASS, connected node pairs are regarded as positive sample pairs, while all unconnected node pairs are regarded as negative sample pairs. Therefore, it is entirely

feasible and rational to construct the overall objective function in the following comprehensive and detailed manner:

$$\mathcal{L} = -\mathbb{E}_{\mathbf{z}_v,\mathbf{b}_u\sim P_{\mathcal{VU}}}[c_\theta(\mathbf{z}_v,\mathbf{b}_u)] + \log(\mathbb{E}_{\mathbf{h},\mathbf{h}'\sim P_V\otimes P_V}[e^{c_\theta(\mathbf{h},\mathbf{h}')}]), \tag{8}$$

where $c_\theta(\cdot)$ represents the similarity between node pairs calculated using a neural network, specifically a single MLP. $\mathbf{z}_v$ is the vth row of $\mathbf{Z}$, and $\mathbf{b}_u$ is the uth row of $\mathbf{B}$. $P_{\mathcal{VU}}$ denotes the connection probability of positive sample pairs on the bipartite graph, *i.e.*, node pairs $(\mathbf{z}_v,\mathbf{b}_u)$ that are connected by an edge, where $\mathcal{V}$ is the set of original graph nodes and $\mathcal{U}$ is the set of attribute nodes. Meanwhile, $\mathbf{h}$ denotes any node, $\mathbf{h}'$ denotes all nodes except $\mathbf{h}$ on the bipartite graph. $P_V\otimes P_V$ denotes the product of marginal distribution of independently sampling two nodes from the bipartite graph node set $V=\mathcal{V}\cup\mathcal{U}$, which covers all node pairs.

4.2 Theoretical Analysis

This subsection aims to provide a theoretical understanding of the proposed GCL framework. The framework can implicitly capture the node-node co-inclusion relationship (a high-hop relationship), which is important for its universality to both homophilic and heterophilic graphs.

Theorem 1. *Let $N_2(v)$ denote the set of two-hop neighbors of $\mathbf{z}_v$ on the bipartite graph $\mathcal{G}'$. Minimizing the GRASS objective in Equation (8) is approximately minimizing the following alignment loss between two-hop neighbors:*

$$\mathcal{L}_{two-hop} = \frac{1}{2}\frac{1}{|V|}\sum_v\sum_{u\in N(v)}\frac{1}{|N(u)|}\sum_{u_2\in N(u)}\|\mathbf{z}_v-\mathbf{z}_{u_2}\|_2^2.$$

Proof. By removing the second loss term and retaining only the positive sample pair loss, Equation (8) can be formally elaborated as follows:

$$-\frac{1}{|V|}\sum_{v\in V}(\frac{1}{|N(v)|}\sum_{u\in N(v)}(\mathbf{z}_v\mathbf{b}_u^\top)), \tag{9}$$

where $\mathbf{z}_v$ and $\mathbf{b}_u$ are one-hop neighbors of each other on the bipartite graph. $|V|$ denotes the number of nodes on the bipartite graph and $N(v)$ denotes the set of neighbors of node $\mathbf{z}_v$. Expanding Equation (9), the following equation is subsequently and precisely obtained:

$$\begin{aligned}&-\frac{1}{|V|}\sum_{v\in V}\frac{1}{|N(v)|}\sum_{u\in N(v)}\frac{1}{\sqrt{d_v}\sqrt{d_u}}(\sqrt{d_v}\mathbf{z}_v\sqrt{d_u}\mathbf{b}_u^\top)\\ \geq&-\frac{1}{|V|}\sum_{v\in V}\sum_{u\in N(v)}(\frac{1}{\sqrt{d_v}\sqrt{d_u}}(\sqrt{d_v}\mathbf{z}_v\sqrt{d_u}\mathbf{b}_u^\top)).\end{aligned} \tag{10}$$

Let $\mathbf{P}_v=\sqrt{d_v}\mathbf{z}_v$ be the vth row of the matrix $\mathbf{P}\in\mathbb{R}^{(n+f)\times d}$, and $\mathbf{Q}_u=\sqrt{d_u}\mathbf{b}_u$ be the uth row of the matrix $\mathbf{Q}\in\mathbb{R}^{(n+f)\times d}$. $\tilde{\mathbf{A}}'=\tilde{\mathbf{D}}'^{-1/2}\mathbf{A}'\tilde{\mathbf{D}}'^{-1/2}$, where $\mathbf{A}'$ denotes the adjacency matrix of the bipartite graph $\mathcal{G}'$. d_v and d_u denote the degree of $\mathbf{z}_v$ and $\mathbf{b}_u$. Then Equation (10) can be specifically formulated as:

$$-\frac{1}{|V|}tr(\tilde{\mathbf{A}}'\mathbf{P}\mathbf{Q}^\top)\geq-\frac{1}{2}\frac{1}{|V|}(\|\tilde{\mathbf{A}}'\mathbf{P}\|_2^2+\|\mathbf{Q}^\top\|_2^2). \tag{11}$$

For Equation (11), the inequality is utilized: $\mathrm{tr}(\mathbf{PQ})\leq\frac{1}{2}\left(\|\mathbf{P}\|_2^2+\|\mathbf{Q}\|_2^2\right)$ for any two matrices $\mathbf{P}\in\mathbb{R}^{m\times k}$ and $\mathbf{Q}\in\mathbb{R}^{k\times t}$. Then, the property of the trace $\mathrm{tr}(\mathbf{X}^\top\mathbf{X})=\|\mathbf{X}\|_2^2$ is used to obtain the following equation:

$$-\frac{1}{2}\frac{1}{|V|}(tr(\mathbf{P}^\top\tilde{\mathbf{A}}'^\top\tilde{\mathbf{A}}'\mathbf{P})+\|\mathbf{Q}^\top\|_2^2). \tag{12}$$

Equation (12) is extended using the cyclic invariant property of traces to derive the following equation:

$$-\frac{1}{2}\frac{1}{|V|}(tr(\mathbf{P}\mathbf{P}^\top\tilde{\mathbf{A}}'^\top\tilde{\mathbf{A}}')+\|\mathbf{Q}^\top\|_2^2). \tag{13}$$

Removing the $\|\mathbf{Q}^\top\|_2^2$ term and expanding the formula for the trace in Equation (13) can be found:

$$\begin{aligned}&tr(\mathbf{P}\mathbf{P}^\top\tilde{\mathbf{A}}'^\top\tilde{\mathbf{A}}')=\sum_v(\mathbf{P}\mathbf{P}^\top\tilde{\mathbf{A}}'^\top\tilde{\mathbf{A}}')_{vv}\\ =&\sum_{v,u_2\in V}\sum_{u\in N(v)\cap N(u_2)}\tilde{\mathbf{A}}'^\top_{vu}\tilde{\mathbf{A}}'_{uu_2}\sqrt{d_v}\sqrt{d_{u_2}}\mathbf{z}_v^\top\mathbf{z}_{u_2},\end{aligned} \tag{14}$$

where $\mathbf{z}_{u_2}$ is the two-hop neighbor of $\mathbf{z}_v$. Expanding the adjacency matrix in Equation (14) obtains:

$$\begin{aligned}&\sum_v\sum_{u\in N(v)}\sum_{u_2\in N(u)}\frac{1}{d_u}\frac{1}{\sqrt{d_v}\sqrt{d_{u_2}}}\mathbf{z}_v^\top\mathbf{z}_{u_2}\sqrt{d_v}\sqrt{d_{u_2}}\\ \Rightarrow&-\sum_v\sum_{u\in N(v)}\sum_{u_2\in N(u)}\frac{1}{|N(u)|}\|\mathbf{z}_v-\mathbf{z}_{u_2}\|_2^2.\end{aligned} \tag{15}$$

Combining Equation (15) and Equation (13), the final conclusion can be obtained:

$$\frac{1}{2}\frac{1}{|V|}\sum_v\sum_{u\in N(v)}\frac{1}{|N(u)|}\sum_{u_2\in N(u)}\|\mathbf{z}_v-\mathbf{z}_{u_2}\|_2^2. \tag{16}$$

□

Theorem 1 shows that the optimization objective of GRASS is equivalent to minimizing the distance between two-hop neighbors on the constructed bipartite graph. Since these two-hop neighbors correspond to node pairs that share attributes on the original graph, the framework effectively encourages attribute-similar nodes to learn consistent representations. Besides, two-hop neighbors on the bipartite graph are brought close, which effectively captures the high-hop relationship on the original graph. Compared to traditional node-node contrast, the proposed node-attribute contrast can more accurately and comprehensively capture node-node relationships, thereby achieving universality for both homophilic and heterophilic graphs.

5 Experiments

In this section, to begin with, the proposed framework GRASS is validated by empirically evaluating its performances on the node classification task. Next, an in-depth understanding of the efficacy of this framework is provided through several experiment analyses.

Dataset	Cora	CiteSeer	PubMed	Wiki-CS	Computers	Photo	Chameleon	Squirrel	Actor	Cornell	Texas	Wisconsin
Nodes	2,708	3,327	19,717	11,701	13,752	7,650	2,277	5,201	7,600	183	183	251
Edges	5,429	4,732	44,338	216,123	245,861	119,081	36,101	217,073	33,544	295	309	499
Features	1,433	3,703	500	300	767	745	2,325	2,089	932	1,703	1,703	1,703
Classes	7	6	3	10	10	8	5	5	5	5	5	5
h	0.81	0.74	0.80	0.65	0.78	0.83	0.23	0.22	0.22	0.13	0.11	0.20

Table 1: Statistics of datasets.

Methods	Cora	CiteSeer	PubMed	Wiki-CS	Computers	Photo
GCN	82.43±1.54	71.98±0.89	84.53±0.31	76.89±0.37	86.34±0.48	92.35±0.25
GAT	83.27±1.36	72.36±1.13	84.93±0.44	77.42±0.19	87.06±0.35	92.64±0.42
DeepWalk	77.81±0.35	59.06±0.23	79.62±0.84	74.35±0.06	85.68±0.06	89.44±0.11
Node2Vec	79.16±0.74	59.89±0.54	80.26±0.91	71.79±0.05	84.39±0.08	89.67±0.12
DGI	82.57±0.36	71.52±0.16	85.89±0.15	75.73±0.13	84.09±0.39	91.49±0.25
GMI	82.41±1.34	71.64±0.52	84.57±0.88	75.06±0.13	81.76±0.52	90.72±0.33
GRACE	83.32±0.37	71.48±0.38	**86.30±0.43**	79.16±0.36	87.21±0.44	92.65±0.32
MVGRL	83.02±0.27	72.74±0.36	85.42±0.38	77.97±0.18	87.09±0.27	92.01±0.13
GCA	82.80±0.46	71.14±0.35	85.74±0.75	79.35±0.12	87.84±0.27	92.78±0.17
BGRL	82.66±0.76	71.53±0.56	84.21±0.17	78.74±0.22	88.14±0.33	92.45±0.29
AF-GCL	83.16±0.13	71.96±0.42	83.95±0.75	79.01±0.51	89.68±0.19	92.49±0.31
HLCL	82.34±0.87	72.34±0.84	84.69±0.89	79.26±0.31	86.97±0.35	91.98±0.31
MaskGAE	83.21±0.51	72.46±0.73	82.69±0.31	75.97±0.55	88.46±0.20	92.82±0.04
GraphACL	83.70±0.41	72.84±0.35	84.11±0.21	76.49±0.74	88.92±0.32	92.98±0.26
SGCL	83.56±0.28	72.98±0.69	85.64±0.37	79.85±0.53	88.57±0.43	92.79±0.35
GCIL	83.80±0.73	**73.76±0.52**	84.96±0.63	78.61±0.19	88.32±0.44	93.15±0.68
GRASS	**84.93±1.63**	73.47±0.94	85.18±0.76	**81.35±1.06**	**89.76±0.61**	**93.26±0.57**

Table 2: Node classification accuracy (mean ± std) on homophilic graphs. The best and second best results are highlighted in **bold** and underline, respectively.

5.1 Experimental Settings

Datasets. Experiments are conducted on twelve widely used benchmark datasets with various homophily. The homophilic graph datasets include Cora, CiteSeer, PubMed, Wiki-CS, Amazon-Computers (abbreviated as Computers), and Amazon-Photo (abbreviated as Photo). The heterophilic graph datasets include Chameleon, Squirrel, Actor, Cornell, Texas, and Wisconsin. The statistics of these datasets are summarised in Table 1. *Cora*, *CiteSeer*, and *PubMed* [Sen *et al.*, 2008] are three citation network datasets where nodes represent papers and edges indicate citation relationships between papers. *Wiki-CS* [Mernyei and Cangea, 2020] is a hyperlink network constructed based on Wikipedia. Nodes represent articles in computer science and edges are hyperlinks between articles. *Computers* and *Photo* [Shchur *et al.*, 2018] are co-purchase networks from Amazon. In these networks, nodes represent goods and edges represent two goods being frequently bought together. *Chameleon* and *Squirrel* [Pei *et al.*, 2020] are two Wikipedia networks where nodes denote pages and edges denote links between pages. *Actor* [Pei *et al.*, 2020] is an actor co-occurrence network where nodes denote actors and edges denote two actors co-occurring in the same film. *Cornell*, *Texas*, and *Wisconsin* [Pei *et al.*, 2020] are networks of web pages from computer science departments of diverse universities, where nodes are web pages and edges are hyperlinks between web pages.

Baselines. To verify the superiority and effectiveness of the proposed GRASS, we compare it with three categories of graph learning methods: (1) semi-supervised GNN models for node classification tasks, including vanilla GCN [Pei *et al.*, 2020] and GAT [Veličković *et al.*, 2018]; (2) unsupervised graph learning methods, including DeepWalk [Perozzi *et al.*, 2014] and Node2Vec [Grover and Leskovec, 2016]; (3) self-supervised graph learning methods, including DGI [Velickovic *et al.*, 2019], GMI [Peng *et al.*, 2020], GRACE [Zhu *et al.*, 2020b], MVGRL [Hassani and Khasahmadi, 2020], GCA [Zhu *et al.*, 2021], BGRL [Thakoor *et al.*, 2022], AF-GCL [Wang *et al.*, 2022], HLCL [Yang and Mirzasoleiman, 2023], MaskGAE [Li *et al.*, 2023], GraphACL [Xiao *et al.*, 2023], SGCL [Sun *et al.*, 2024], and GCIL [Mo *et al.*, 2024].

Experimental Details. For reproducibility, the detailed settings of the experiments are described below. The experiments are performed on Nvidia GeForce RTX 3090 (24GB) GPU cards. Furthermore, the representations are obtained using a 1-3 layer MLP, with the hidden dimension of each layer being subject to hyperparameter tuning. Based on the representations, we train a Logistic classifier to perform downstream tasks. In all the experiments, we use the Adam optimizer. The training epoch is 200 with full batch training. For hyperparameter settings, the learning rates are tuned in the range $\{0.1, 0.05, 0.01, 0.005, 0.001\}$. Besides, the

Methods	Chameleon	Squirrel	Actor	Cornell	Texas	Wisconsin
GCN	59.63±2.23	36.28±1.52	30.83±0.77	57.03±3.30	60.00±4.80	56.47±6.55
GAT	56.38±2.19	32.09±3.27	28.06±1.48	59.46±3.63	61.62±3.78	54.71±6.87
DeepWalk	47.74±2.05	32.93±1.58	22.78±0.64	63.35±4.61	60.59±7.56	55.41±5.96
Node2Vec	41.93±3.29	22.84±0.72	28.28±1.27	42.94±7.46	41.92±7.76	37.45±7.09
DGI	39.95±1.75	31.80±0.77	29.82±0.69	63.35±4.61	60.59±7.56	55.41±5.96
GMI	46.97±3.43	30.11±1.92	27.82±0.90	54.76±5.06	50.49±2.21	45.98±2.76
GRACE	48.05±1.81	31.33±1.22	29.01±0.78	54.86±6.95	57.57±5.68	50.00±5.83
MVGRL	51.07±2.68	35.47±1.29	30.02±0.70	64.30±5.43	62.38±5.61	62.37±4.32
GCA	49.80±1.81	35.50±0.91	29.65±1.47	55.41±4.56	59.46±6.16	50.78±4.06
BGRL	47.46±2.74	32.64±0.78	29.86±0.75	57.30±5.51	59.19±5.85	52.35±4.12
AF-GCL	59.56±1.69	40.64±0.67	32.26±0.69	64.76±3.58	68.89±2.16	67.75±3.26
HLCL	58.80±1.68	34.11±0.88	30.98±0.94	58.45±3.24	65.26±1.78	66.96±3.76
MaskGAE	54.12±7.65	37.93±1.15	31.12±0.89	55.41±4.56	65.95±6.53	56.86±4.47
GraphACL	60.21±0.35	**41.36±4.57**	30.12±0.21	59.67±1.39	70.24±1.56	68.34±2.03
SGCL	58.96±1.43	38.49±0.86	31.23±0.64	62.76±2.94	68.31±4.33	63.26±3.06
GCIL	60.14±2.26	40.68±1.35	31.67±0.94	63.57±3.19	67.54±6.27	65.24±1.85
GRASS	**60.25±2.48**	40.38±2.64	**34.94±0.94**	**70.25±5.66**	**77.41±2.17**	**77.05±5.98**

Table 3: Node classification accuracy (mean ± std) on heterophilic graphs. The best and second best results are highlighted in **bold** and underline, respectively.

weight decay is tuned from {0.0, 0.001, 0.005, 0.01, 0.1}. Finally, the representation dimension is tuned in the range {256, 512, 1024, 2048, 4096}. For homophilic graphs, all nodes are randomly divided into three parts, 10% nodes for training, 10% nodes for validation and the remaining 80% nodes for testing. The performance on heterophilic graph datasets is evaluated on the commonly used 48%/32%/20% training/validation/testing.

5.2 Experiment Results

Results on Homophilic Graphs. The comparison of accuracy between GRASS and the baselines on six homophilic graphs is shown in Table 2. First, it can be observed that GRASS achieves the optimal performance on four of the six datasets, which illustrates the superiority of GRASS for processing homophilic graphs. GRASS even surpasses all supervised comparison models on all datasets. To be specific, on the Cora dataset, GRASS outperforms the second-best self-supervised model (*i.e.*, GCIL) by 1.13%, and on the WikiCS dataset, GRASS outperforms the second-best model (*i.e.*, SGCL) by 1.50% in classification accuracy. This highlights the effectiveness of GRASS in learning consistency between node representations and attribute representations.

Results on Heterophilic Graphs. Table 3 shows the experimental results on heterophilic graphs. It can be observed that GRASS outperforms the baselines on five of the six datasets. Specifically, contrastive strategies with augmentations (*e.g.*, GRACE, GCA, and BGRL) can not work well on heterophilic graphs compared to homophilic graphs. In contrast, augmentation-free models such as GRASS, AF-GCL, and GraphACL achieve consistent performance advantages across five heterophilic benchmark datasets. Existing studies have shown that graph augmentations preserve the low-frequency components of the graph while potentially perturbing the high-frequency components in heterophilic graphs [Lee *et al.*, 2022]. Therefore, achieving better performance without graph augmentations may benefit from the preservation of high-frequency components. Additionally, GRASS outperforms the second-best model (*i.e.*, AF-GCL and GraphACL) by 5.49%, 7.17%, and 8.71% on the Cornell, Texas, and Wisconsin datasets, respectively. Compared to GraphACL, the performance advantage of GRASS stems from its ability to identify more accurate and numerous node neighbors than GraphACL, which relies on two-hop neighbors on the graph. As shown in Figure 3, even in heterophilic graphs, the identified two-hop neighbors have over 90% similarity, which is significantly higher than the two-hop monophily reported in GraphACL.

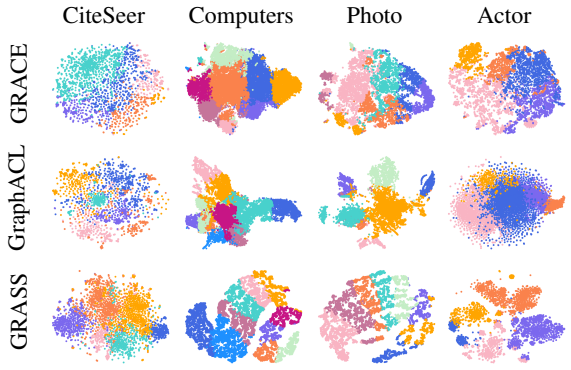

Figure 2: t-SNE visualization of node embeddings from GRACE, GraphACL, and GRASS on CiteSeer, Computers, Photo, and Actor.

Visualization. This experiment aims to intuitively demonstrate the representation ability of the proposed GRASS. For this purpose, the t-SNE [Van der Maaten and Hinton, 2008] method is exploited to perform feature reduction and visualization of the trained representations. Figure 2 exhibits the experiment results (scatter plots) on four benchmark datasets (*i.e.*, CiteSeer, Computers, Photo, and Actor), where col-

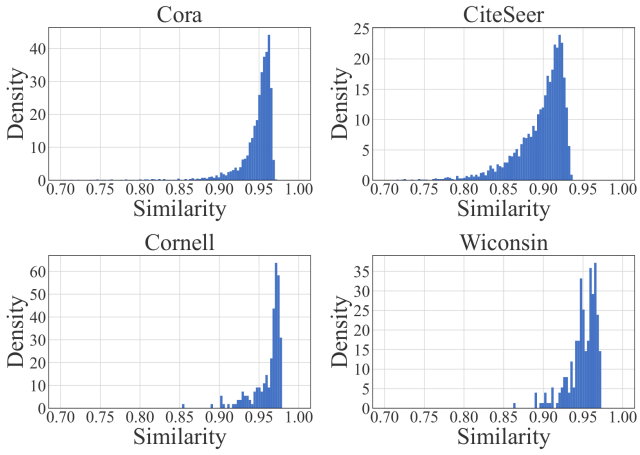

Figure 3: The effect of two-hop neighbors on the bipartite graph.

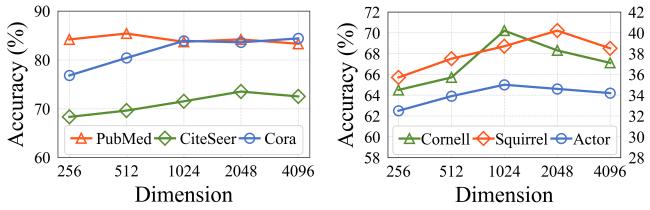

Figure 4: The effect of the representation dimension.

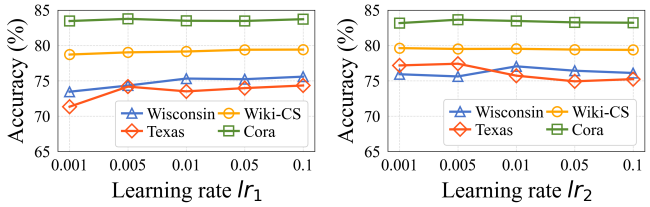

Figure 5: Node classification performance of two MLPs with unshared parameters at different learning rates.

ors stand for the classes of nodes. It can be observed that compared to all the comparison models (*i.e.*, GRACE and GraphACL), the proposed GRASS framework is enabled to produce the most informative representations. As exemplified by the CiteSeer and Actor datasets, compared to the self-supervised models, GRASS achieves more compact clusters in node embeddings. Specifically, the embeddings of the same class are closer, while the embeddings of different classes exhibit more significant differences. The experiment results emphasize the effectiveness of GRASS in enhancing representation ability. Furthermore, GRASS is able to address the uneven distribution in the dataset due to class imbalance. As exemplified by the CiteSeer dataset, GRASS can achieve a more balanced distribution of nodes across different classes, thereby mitigating the distribution disparity among classes compared to GRACE.

5.3 Similarity Analysis

Figure 3 plots the pairwise cosine similarity distribution of two-hop neighbor pairs on the bipartite graph. It can be observed that in both homophilic and heterophilic graphs, the similarity is mostly concentrated in the range of 0.75 to 1, indicating that GRASS is able to efficiently identify pairs of nodes with similar attributes in the original graph. This phenomenon coincides with Theorem 1. In homophilic graphs, connected nodes tend to belong to the same class, *i.e.*, they share similar attributes. The two-hop neighbors in the bipartite graph are connected by the same attribute, thus they are inclined to be similar. Besides, nodes shared attributes (*i.e.*, two-hop neighbors on the bipartite graph) often correspond to the high-hop relationship on the original graph. In heterophilic graphs, directly connected nodes are usually from different classes. Therefore, the high-hop relationship is more likely to uncover nodes of the same class.

5.4 Hyperparameter Study

To analyze the sensitivity of the proposed model, experiments have been conducted with various hyperparameters.

Representation Dimension. In Figure 4, the effect of the representation dimension on performance is analyzed. The results show that higher representation dimensions generally lead to better performance, especially on datasets with large initial attribute dimensions such as CiteSeer and Squirrel. This indicates that preserving more feature information contributes to improved performance on both homophilic and heterophilic graphs.

Learning Rate. In Figure 5, the impact of the hyperparameters lr_1 and lr_2 on node classification accuracy across different graphs is analyzed. lr_1 and lr_2 control the learning rates for the topology part (MLP_A) and the attribute part (MLP_M) of the model, respectively. In the experiments, when lr_1 is adjusted, lr_2 is kept fixed, and vice versa. The results show that in homophilic graphs, the performance is relatively stable with respect to both learning rates, which reflects the insensitivity to learning rates on homophilic graphs. However, in heterophilic graphs, results show that adjusting lr_2 leads to more significant accuracy improvements, while tuning lr_1 has a relatively smaller effect. This indicates that the attribute representation plays a more important role in handling heterophilic structures, with the model relying more on attributes to distinguish node categories.

6 Conclusions

This paper proposes a novel Graph Contrastive Learning (GCL) framework named GRAph learning via Self-contraSt (GRASS) to overcome the limitations of traditional GCL methods, including the reliance on complex graph augmentations and the difficulty of selecting TRUE positive and negative samples. GRASS adopts a node-attribute self-contrast mechanism that enhances the similarity between nodes and their included attributes while suppressing the similarity with non-included attributes. This framework implicitly ensures accurate node-node contrast by capturing high-hop co-inclusion relationships. The theoretical analysis and empirical results on multiple benchmark datasets validate the efficiency and universality of GRASS across both homophilic and heterophilic graphs.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. U22B2036, 62261136549, 62376088, 62272340, 92370111, 62272020), in part by the Hebei Natural Science Foundation (No. F2024202047, F2024202068), in part by the Science Research Project of Hebei Education Department (BJK2024172), in part by the Guangxi Key Laboratory of Machine Vision and Intelligent Control (2023B03), in part by the Hebei Yanzhao Golden Platform Talent Gathering Programme Core Talent Project (Education Platform) (HJZD202509), in part by the Post-graduate's Innovation Fund Project of Hebei Province (CXZZBS2025036), and in part by the Tencent Foundation and XPLORER PRIZE.

References

[Bae *et al.*, 2023] Sangmin Bae, Sungnyun Kim, Jongwoo Ko, Gihun Lee, Seungjong Noh, and Se-Young Yun. Self-contrastive learning: single-viewed supervised contrastive framework using sub-network. In *AAAI*, volume 37, pages 197–205, 2023.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2024] Jinsong Chen, Hanpeng Liu, John E. Hopcroft, and Kun He. Leveraging contrastive learning for enhanced node representations in tokenized graph transformers. In *NeurIPS*, 2024.

[Church and Hanks, 1990] Kenneth Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[Donsker and Varadhan, 1983] Monroe D Donsker and SRS386024 Varadhan. Asymptotic evaluation of certain markov process expectations for large time, iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

[Grover and Leskovec, 2016] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, pages 855–864, 2016.

[Guan *et al.*, 2024] Renxiang Guan, Zihao Li, Wenxuan Tu, Jun Wang, Yue Liu, Xianju Li, Chang Tang, and Ruyi Feng. Contrastive multiview subspace clustering of hyperspectral images based on graph convolutional networks. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2024.

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

[Hendrycks *et al.*, 2019] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.

[Lee *et al.*, 2022] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. In *AAAI*, volume 36, pages 7372–7380, 2022.

[Li *et al.*, 2023] Jintang Li, Ruofan Wu, Wangbin Sun, Liang Chen, Sheng Tian, Liang Zhu, Changhua Meng, Zibin Zheng, and Weiqiang Wang. What's behind the mask: Understanding masked graph modeling for graph autoencoders. In *SIGKDD*, pages 1268–1279, 2023.

[Lim *et al.*, 2021] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *NeurIPS*, 2021.

[Liu *et al.*, 2023] Yixin Liu, Yizhen Zheng, Daokun Zhang, Vincent CS Lee, and Shirui Pan. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *AAAI*, volume 37, pages 4516–4524, 2023.

[Ma *et al.*, 2023] Yixuan Ma, Xiaolin Zhang, Peng Zhang, and Kun Zhan. Entropy neural estimation for graph contrastive learning. In *MM*, pages 435–443, 2023.

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

[Mo *et al.*, 2024] Yanhu Mo, Xiao Wang, Shaohua Fan, and Chuan Shi. Graph contrastive invariant learning from the causal perspective. In *AAAI*, volume 38, pages 8904–8912, 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 *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 *WWW*, pages 259–270, 2020.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, pages 701–710, 2014.

[Polyanskiy and Wu, 2014] Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. *Lecture Notes for ECE563 (UIUC) and*, 6(2012-2016):7, 2014.

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

[Shchur *et al.*, 2018] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan

Günnemann. Pitfalls of graph neural network evaluation. arxiv 2018. *arXiv preprint arXiv:1811.05868*, 2018.

[Sun *et al.*, 2024] Wangbin Sun, Jintang Li, Liang Chen, Bingzhe Wu, Yatao Bian, and Zibin Zheng. Rethinking and simplifying bootstrapped graph latents. In *WSDM*, pages 665–673, 2024.

[Thakoor *et al.*, 2021] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR*, 2021.

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

[Trivedi *et al.*, 2022] Puja Trivedi, Ekdeep Singh Lubana, Mark Heimann, Danai Koutra, and Jayaraman J Thiagarajan. Analyzing data-centric properties for graph contrastive learning. In *NeurIPS*, 2022.

[Tsai *et al.*, 2021] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective. In *ICLR*, 2021.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *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 *ICLR*, 2019.

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

[Xia *et al.*, 2022] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *WWW*, pages 1070–1079, 2022.

[Xiao *et al.*, 2022] Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. Decoupled self-supervised learning for graphs. In *NeurIPS*, 2022.

[Xiao *et al.*, 2023] Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. Simple and asymmetric graph contrastive learning without augmentations. In *NeurIPS*, 2023.

[Yang and Mirzasoleiman, 2023] Wenhan Yang and Baharan Mirzasoleiman. Contrastive learning under heterophily. *arXiv preprint arXiv:2303.06344*, 2023.

[Yu and Jia, 2024] Jiajun Yu and Adele Lu Jia. Agcl: Adaptive graph contrastive learning for graph representation learning. *Neurocomputing*, 566:127019, 2024.

[Zhang *et al.*, 2022] Hengrui Zhang, Qitian Wu, Yu Wang, Shaofeng Zhang, Junchi Yan, and Philip S Yu. Localized contrastive learning on graphs. *arXiv preprint arXiv:2212.04604*, 2022.

[Zhu *et al.*, 2020a] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: current limitations and effective designs. In *NeurIPS*, 2020.

[Zhu *et al.*, 2020b] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

[Zhu *et al.*, 2021] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, pages 2069–2080, 2021.

[Zhuo *et al.*, 2024a] Jiaming Zhuo, Can Cui, Kun Fu, Bingxin Niu, Dongxiao He, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, and Liang Yang. Graph contrastive learning reimagined: Exploring universality. In *WWW*, pages 641–651, 2024.

[Zhuo *et al.*, 2024b] Jiaming Zhuo, Yintong Lu, Hui Ning, Kun Fu, Bingxin Niu, Dongxiao He, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, et al. Unified graph augmentations for generalized contrastive learning on graphs. In *NeurIPS*, 2024.

[Zhuo *et al.*, 2024c] Jiaming Zhuo, Feiyang Qin, Can Cui, Kun Fu, Bingxin Niu, Mengzhu Wang, Yuanfang Guo, Chuan Wang, Zhen Wang, Xiaochun Cao, and Liang Yang. Improving graph contrastive learning via adaptive positive sampling. In *CVPR*, pages 23179–23187, 2024.