Attribute and Structure Preserving Graph Contrastive Learning

Jialu Chen, Gang Kou*

School of Business Administration, Faculty of Business Administration, Southwestern University of Finance and Economics
lukac@smail.swufe.edu.cn, kougang@swufe.edu.cn

Abstract

Graph Contrastive Learning (GCL) has drawn much research interest due to its strong ability to capture both graph structure and node attribute information in a self-supervised manner. Current GCL methods usually adopt Graph Neural Networks (GNNs) as the base encoder, which typically relies on the homophily assumption of networks and overlooks node similarity in the attribute space. There are many scenarios where such assumption cannot be satisfied, or node similarity plays a crucial role. In order to design a more robust mechanism, we develop a novel attribute and structure preserving graph contrastive learning framework, named ASP, which comprehensively and efficiently preserves node attributes while exploiting graph structure. Specifically, we consider three different graph views in our framework, i.e., original view, attribute view, and global structure view. Then, we perform contrastive learning across three views in a joint fashion, mining comprehensive graph information. We validate the effectiveness of the proposed framework on various real-world networks with different levels of homophily. The results demonstrate the superior performance of our model over the representative baselines.

Introduction

Graph representation learning (GRL), which aims to learn low-dimensional representations for graphs, has shown its great potential for various application areas, such as graph matching (Li et al. 2019), physics (Shlomi, Battaglia, and Vlimant 2021), chemistry (Do, Tran, and Venkatesh 2018) and biology (Zitnik, Agrawal, and Leskovec 2018). Due to the scarcity of task-specific labeled graph data, recent research efforts for GRL methods have been devoted to self-supervised learning on graphs, where only limited or no labels are needed (Xie et al. 2021; Liu et al. 2022). Among all these methods, graph contrastive learning (GCL), which successfully applys contrastive learning from the field of vision and language to graph data, has drawn considerable research attention. By maximizing the agreement between jointly sampled positive views and drawing apart the distance between negative views, GCL methods have achieved promising results on many graph-based tasks (Veličković et al. 2019; Zhu et al. 2020b; Hassani and Khasahmadi 2020). Typically, GCL methods first generate multiple views for each instance through various data augmentation strategies, and then adopt graph neural networks as the backbone encoder to obtain the corresponding representations for contrastive objectives. Researchers usually improve the performance and generalizability of GCL models by designing novel data augmentation strategies (Zhu et al. 2021b; You et al. 2021), or adjusting suitable contrastive objectives (Sun et al. 2019; Peng et al. 2020b; Jiao et al. 2020).

While the success of GCLs, they ignore a key weakness that built in the backbone encoder they use, i.e., homophily assumption of graph neural networks, where nodes within the same class tend to connect with each other. Since graphs with different level of homophily are widespread in real world, the homophily assumption may not always be satisfied, which restrict the performance of GNNs on general data (Pei et al. 2020). Thus, exisiting GCLs based on graph neural network encoder could also suffer from this issue and thus lead to impaired performance. Hence, a natural question emerges: When the knowledge of labels and downstream tasks are unavailable, how to design the framework of contrastive learning so that it could capture comprehensive information for graphs with different levels of homophily?

Several recent works have studied dealing with graphs with different homophily level in a supervised manner. For example, Pei et al. (2020) proposes a geometric aggregation scheme for graph neural networks to overcome neighborhood structural information losing and long-range dependencies in non-homophilous graphs lacking. Zhu et al. (2020a) combines a set of key designs that can boost learning from the graph structure in heterophily into a model. Jin et al. (2021a) designs a universal propagation mechanism which learns node embeddings by adaptively fusing information from 1-hop, 2-hop, and kNN neighbors. Jin et al. (2021b) develops a feature preserving aggregation which adaptively integrates graph structure and node attributes by incoporating attribute graph. Lim et al. (2021) collects and introduces non-homophilous datasets from a variety of application areas, and proposes a simple scalable learning method for non-homophilous graphs.

However, designing graph contrastive learning methods for graphs with different level of homophily could be challenging. First, adaptive views which could distinguish the

*Corresponding Author.

homophily level of graphs usually are trained under the supervised signal. Since labels and downstream tasks are unknown while training, it is hard to generate such adaptive views under unsupervised scenario. In addition, directly contrasting two views generated from methods for homophilous graphs and methods for non-homophilous graphs seems to be a good choice. However, if two views reveal significantly diverse information, the embeddings are far away from each other, thus being bad views for contrastive learning (Tian et al. 2020).

To tackle the challenges discussed above, we propose a novel graph contrastive learning framework named ASP, which preserves both attribute and structure information and achieves competitive performance independent of homophily level. We first incorporate attribute graph into our framework, since node similarity relationship built in attribute graph is proved to be helpful for dealing with non-homophilous graphs (Jin et al. 2021a,b). Then, we design a novel data augmentation strategy: add one view to another view and use the summation of two views as a final view for contrasting. Such design solves the problem that two node embeddings generated from very different views tend to be far from each other. Specifically, we use original graph as one of the contrastive view, and treat the summation of attribute graph and original graph as another contrastive view. Through maximizing the agreement between the two contrastive views, our model preserves the attribute information without annotated labels. Next, to preserve the structure information in the meantime, we propose another contrastive learning module which mainly aims to preserve global structure information. Different from existing methods (Hassani and Khasahmadi 2020; Zhu et al. 2020b), we directly adopt higher-order view of the graph as global structure view. We find that contrasting the higher-order view and original graph view is sufficient for preserving global structure information. Besides, in order to align the representations across different modules, a novel cross-module contrastive loss are proposed. By jointly optimizing the attribute loss, structure loss and cross-module loss, our framework could learn expressive node representations without any human annotated labels. Our main contributions can be summarized as follows:

- We propose a novel framework ASP, which could learn effective node representations for graphs with different levels of homophily in a self-supervised fashion.
- We design two contrastive learning modules which could respectively capture attribute and structure information of the input graph, and train them coordinately.
- Extensive experimental results show that our approach achieves superior performance over the representative supervised and unsupervised baselines[1].

Related Work

Graph Neural Networks. Over the past few years, Graph Neural Networks (GNNs) have emerged as a powerful tool for learning powerful node representations. There are two main categories of GNNs, i.e., spectral-based GNNs and spatial-based GNNs. Spectral-based GNNs (Bruna et al. 2014; Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017) are built on spectral graph theory which defines graph convolutions by introducing filters from the perspective of graph signal processing(Shuman et al. 2012). Spatial-based GNNs (Gilmer et al. 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018) treat graph convolutions as a message passing process in which node information can be passed along edges directly. Moreover, to increase the expressive power, many efforts are taken to explore higher-order GNNs (Li, Han, and Wu 2018; Chen et al. 2020a; Zeng et al. 2022, 2023). Although GNNs have shown impressive power under many scenarios, their performance might be impaired when the homophily assumption fails (Pei et al. 2020). To solve this problem, researchers try to design GNNs with broader information, by considering node attribute graph (Jin et al. 2021b,a), by incorporating an interpretable compatibility matrix in the graph (Zhu et al. 2021a), and many others (Zhu et al. 2020a; Li et al. 2022).

Self-supervised Learning on Graphs. Recently, inspired by the success of self-supervised learning (SSL) in computer vision (He et al. 2020; Chen et al. 2020b) and natural language processing (Devlin et al. 2018), self-supervised learning on graphs has been a promising research direction. There are three main categories of SSL on graphs: contrastive, generative and predictive (Wu et al. 2021). Graph contrastive learning approaches usually design different views and aim at maximizing agreement between representations of views. DGI (Veličković et al. 2019) maximizes the mutual information between patch representations and corresponding high-level summary of graphs. GRACE (Zhu et al. 2020b) focuses on contrasting views at the node level. GraphCL (You et al. 2020a) applies four types of graph augmentations and learns to predict whether two graphs originate from the same graph or not. Some recent works intend to construct graph views without data augmentation, but instead by adding noise to the hidden embedding space (Zhang et al. 2022; Yu et al. 2022) or model parameters (Xia et al. 2022). Compared with contrastive methods, the generative methods (You et al. 2020b; Hu et al. 2020) are based on generative models and treat the original input data as a natural self-supervision. Different from contrastive methods and generative methods, predictive methods (Peng et al. 2020a; Jin et al. 2020) aim to self-generate informative labels from the data as a supervision.

Most of previous graph self-supervised learning methods use graph neural networks as the backbone encoder, and achieve remarkable performance in homophilous datasets. However, few efforts have been devoted to designing self-supervised node representation learning framework for graphs with different levels of homophily.

Notations and Preliminaries

Notations

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an undirected attributed graph, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ is the set of N nodes, $\mathcal{E}$ is the set

[1]Our implementation is available at https://github.com/JialuChenChina/ASP

of edges describing the relations between nodes and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^T \in \mathbb{R}^{n \times f}$ is the node attribute matrix with f attributes per node. In a node classification task, each node v_i is associated with a label $y_i \in \mathcal{Y}$. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix and $\tilde{\mathbf{A}} \in \mathbb{R}^{n \times n}$ denote the adjacency matrix with added self-loops. $\mathbf{D} \in \mathbb{R}^{n \times n}$ and $\tilde{\mathbf{D}} \in \mathbb{R}^{n \times n}$ represent the corresponding degree matrix of $\mathbf{A}$ and $\tilde{\mathbf{A}}$. Besides, we use $(\mathbf{A}, \mathbf{X})$ to represent graph data.

Homophily

Homophily of a graph characterizes the possibility that nodes with the same label are near each other in a graph. There are many measures of homophily that fits this definition. In this paper, we use edge homophily ratio (Zhu et al. 2020a) defined as the fraction of edges in a graph that connect nodes with the same label:

$$\text{homophily ratio} = \frac{|\{(v_i, v_j) : (v_i, v_j) \in \mathcal{E} \wedge y_i = y_j\}|}{|\mathcal{E}|}. \tag{1}$$

Graphs with homophily ratio closer to 1 tend to have more edges connecting nodes within the same class, whereas graphs with homophily ratio closer to 0 have more edges connecting nodes in different classes. A graph is considered to be non-homophilous if the homophily ratio is smaller than 0.5, homophilous otherwise.

Graph Contrastive Learning

Given an input graph, graph contrastive learning aims to learn one or more encoders such that representations of similar instances agree with each other and representations of dissimilar graph instances disagree with each other. We decouple a typical graph contrastive learning into three main modules.

Multiple Views Generation. Given a graph $\mathcal{G}$, M different transformations $\mathcal{T}_1, ..., \mathcal{T}_M$ can be applied to obtain different views $\mathbf{v}_1, ..., \mathbf{v}_M$ of the graph:

$$\mathbf{v}_i = \mathcal{T}_i(\mathbf{A}, \mathbf{X}), i = 1, ..., M. \tag{2}$$

In practice, graph transformation approaches include node attribute masking, edge perturbation, edge diffusion, subgraph sampling, etc.

View Encoding. A set of encoders $f_1, ..., f_M$ take corresponding views as their inputs and output the representations $\mathbf{h}_1, ..., \mathbf{h}_M$ of the graph from each view:

$$\mathbf{h}_i = f_i(\mathbf{v}_i), i = 1, ..., M. \tag{3}$$

Contrastive Objectives. During training, the contrastive objective aims to train encoders to maximize the mutual information $\mathcal{MI}(\mathbf{h}_i, \mathbf{h}_j)$ of two views from the same instance:

$$\max_{\{f_i\}_{i=1}^M} \frac{1}{\sum_{i \neq j} \sigma_{i,j}} \left[\sum_{i \neq j} \sigma_{i,j} \mathcal{MI}(\mathbf{h}_i, \mathbf{h}_j) \right], \tag{4}$$

where $\sigma_{i,j} \in \{0, 1\}$, and $\sigma_{i,j} = 1$ if the mutual information is computed between $\mathbf{h}_i$ and $\mathbf{h}_j$, and $\sigma_{i,j} = 0$ otherwise. In order to efficiently compute the mutual information, three popular lower-bound forms of the mutual information are derived, i.e., Donsker-Varadhan estimator (Belghazi et al. 2018), Jensen-Shannon estimator (Nowozin, Cseke, and Tomioka 2016) and InfoNCE estimator (Gutmann and Hyvärinen 2010).

Proposed Method

In this section, we present our proposed ASP framework in detail. An overview of ASP is shown in Figure 1.

View Generation

View choices control the information node representations capture, and thus are critical for contrastive learning methods. Existing contrastive learning methods mainly construct views from transformations of existing graph data (i.e. $\mathbf{A}$ and $\mathbf{X}$). We argue that such data transformation strategies are not sufficient for learning comprehensive and robust node embeddings. First, comprehensive self-supervised signals such as attribute graph are hardly utilized in current frameworks. Second, data transformation methods such as edge removing or adding may destroy the original graph topology and optimal data transformation strategies are hard to devise. Third, few existing transformation methods consider the heterophily of graphs. Therefore, by jointly considering three different graph views, i.e., original view, attribute view and global structure view, we solve the issues above.

The original graph data without any data augmentation strategy is deemed to **original view** $\mathbf{v}_o$, which acts as an anchor to other views. To capture the rich attribute similarity relationship between nodes, we utilize the attribute matrix $\mathbf{X}$ to construct a k-Nearest-Neighbor (kNN) graph $\mathcal{G}_A$, and treat it as **attribute view** $\mathbf{v}_a$. The kNN graph $\mathcal{G}_A$ could be built based on different distance metrics, such as euclidean distance, cosine distance or Jaccard distance. We use cosine distance and Jaccard distance in our framework. Experimental results in section show that the choice of distance metric could influence the kNN graph we built and thus affect the final model performance. For a given node pair (v_i, v_j), their cosine distance and Jaccard distance are defined as:

$$d_{cos}(v_i, v_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|}, \tag{5}$$

$$d_J(v_i, v_j) = \frac{\text{\# of non-equal dimensions}}{\text{\# of nonzero dimensions}}, \tag{6}$$

where $|\cdot|$ denotes the magnitude of a vector. The adjacency matrix and corresponding degree matrix of the kNN graph are represented by $\mathbf{A}_F$ and $\mathbf{D}_F$.

In order to extract global structure information, we generate a simple yet effective graph view. Unlike the edge perturbation strategies which arbitrarily modifying the graph structure, edge diffusion methods (Hassani and Khasahmadi 2020) are able to keep rich global information of graphs. Current edge diffusion methods mainly employ Personalized PageRank(PPR) (Page et al. 1999) or heat kernel (Kondor and Lafferty 2002) for global view. However, the solutions of such methods involve computation of matrix exponential or matrix inverse, which is computationally inefficient. Although applying approximating could reduce the

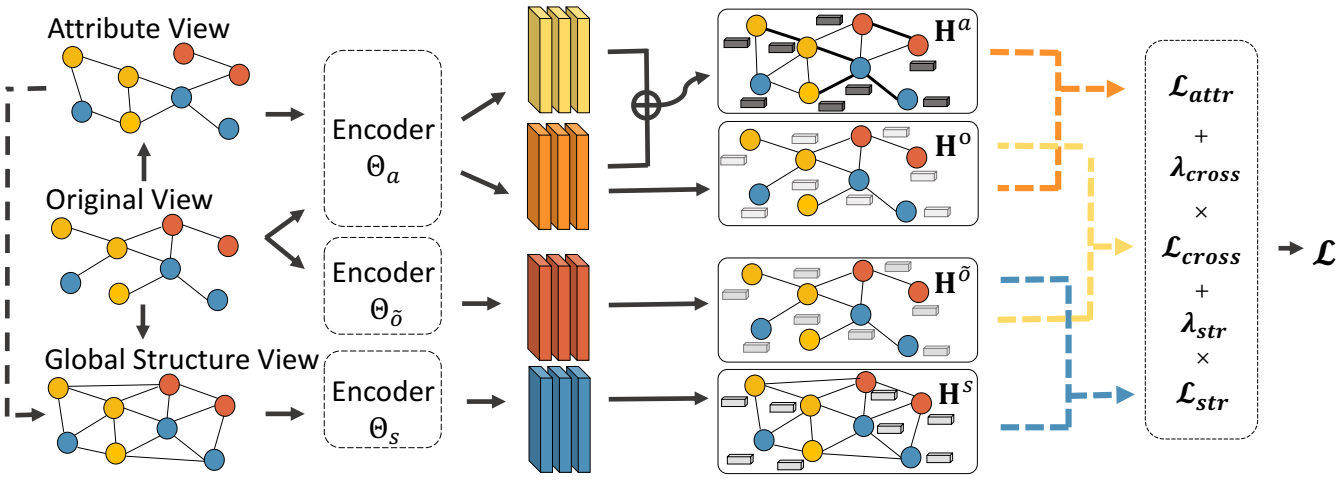

Figure 1: The overview of our proposed framework: ASP. We first generate three graph views from the input graph: original view, attribute view and global structure view. Then the three graph views are fed into the encoders to obtain the corresponding node representations. Instead of directly contrasting these graph views, we use the summation of node representations from attribute view and original view as the new node representations $\mathbf{H}^a$ for subsequent contrastive learning. Through minimizing attribute contrastive loss $\mathcal{L}_{attr}$, structure contrastive loss $\mathcal{L}_{str}$, and cross-module loss $\mathcal{L}_{cross}$ between $\mathbf{H}^{\tilde{o}}$ and $\mathbf{H}^o$, ASP learns expressive node embeddings.

computational complexity, extra hyperparameter tuning is tedious and time-consuming. Different from existing methods, we directly adopt higher-order view of the graph as **global structure view** $\mathbf{v}_s$. Under global structure view, each node captures information from the neighbors which are l hops away. As l increases, more global information is included. Both original graph $\mathcal{G}$ and the kNN graph $\mathcal{G}_A$ could be used for generating higher-order views. We find that simply contrasting original graph views with higher-order views could lead to comparable performance.

Attribute Preserving Contrastive Learning

Existing GCL methods usually adopt Graph Neural Networks (GNNs) as the base encoder, which captures rich structure information of graphs. Despite their success, they commonly overlook the node attribute similarity relationship that rooted in the attribute matrices. In this work, we design a more robust mechanism, which preserves the attribute knowledge based on original graph view and attribute graph view.

Although most GNNs are allowed in our framework, we choose SGC (Wu et al. 2019) as the base encoder, due to its model simplicity and comparable performance. SGC simplifies the structure of GCN (Kipf and Welling 2017) by removing the intermediate nonlinearities, and the learnable parameters collapse into a single matrix:

$$\mathbf{H} = \mathbf{S}^P\mathbf{X}\mathbf{\Theta}, \tag{7}$$

where $\mathbf{S}$ denotes the normalized adjacency matrix with added self-loops $\mathbf{S} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, and $\mathbf{\Theta}$ is a trainable weight matrix. Note that $\mathbf{S}^P\mathbf{X}$ denotes the representations obtained by propagating information from nodes that are P-hop away. Normally, P is a positive integer in SGC, while we allow P to be 0, which turns SGC to one-layer MLP. The reason is that we find MLP is more suitable for non-homophilous graphs. For homophilous graphs, P is set to be a positive integer, which could aggregate useful information from similar neighbors.

Instead of directly comparing original view with attribute view, we harness attribute view as a complement for original view. Specifically, we add the node embeddings from attribute view to original view, and treat the summation of the node embeddings as the final contrastive view. Formally, we have

$$\begin{aligned}\mathbf{H}^o &= \mathbf{S}^p\mathbf{X}\mathbf{\Theta}_a,\\ \mathbf{H}^a &= \mathbf{S}^p\mathbf{X}\mathbf{\Theta}_a + \mathbf{S}_F\mathbf{X}\mathbf{\Theta}_a,\end{aligned} \tag{8}$$

where $\mathbf{S}_F = \mathbf{D}_F^{-1/2}\mathbf{A}_F\mathbf{D}_F^{-1/2}$. The weight matrix $\mathbf{\Theta}_a$ is shared between $\mathbf{H}^o$ and $\mathbf{H}^a$.

After obtaining the node embeddings $\mathbf{H}^o$ and $\mathbf{H}^a$, we adopt InfoNCE (Gutmann and Hyvärinen 2010) to estimate the lower bound of the mutual information between them. For node v_i, the learned node embeddings $\mathbf{h}_i^o$ and $\mathbf{h}_i^a$ form the positive samples, and nodes embeddings other than them are naturally regarded as negative samples. With the defined positive and negative samples, the loss function of attribute preserving contrastive learning module can be defined as:

$$\mathcal{L}_{attr}(v_i) = -\log\frac{e^{\mathcal{D}(\mathbf{h}_i^o,\mathbf{h}_i^a)/\tau}}{\sum_{j=1}^{N} e^{\mathcal{D}(\mathbf{h}_i^o,\mathbf{h}_j^a)/\tau} + \sum_{v\in\{o,a\}}\sum_{j=1}^{N}\mathbb{1}_{[j\neq i]}e^{\mathcal{D}(\mathbf{h}_i^v\cdot\mathbf{h}_j^v)/\tau}}, \tag{9}$$

where τ denotes the temperature parameter, and $\mathcal{D}(\cdot)$ is the discriminator that computes the agreement score between

two vectors. Here, we use cosine similarity as $\mathcal{D}(\cdot)$.

Structure Preserving Contrastive Learning

Although attribute information has been revealed from attribute preserving contrastive learning, the global structural information hasn't been explored. Hence, to increase robustness and expressiveness of the proposed framework, structure preserving contrastive learning module is introduced.

We still utilize SGC as the base encoder in structure preserving contrastive learning. In order to capture global structure information, we directly contrast original graph view with global structure view:

$$\begin{aligned} \mathbf{H}^{\tilde{o}} &= \mathbf{S}^{p}\mathbf{X}\mathbf{\Theta}_{\tilde{o}}, \\ \mathbf{H}^{s} &= (\mathbf{S}_G)^{l}\mathbf{X}\mathbf{\Theta}_{s}, \end{aligned} \tag{10}$$

where l is a positive integer which is significantly larger than p, and $\mathbf{S}_G \in \{\mathbf{S}, \mathbf{S}_F\}$. We set $\mathbf{S}_G$ to be $\mathbf{S}_F$ for non-homophilous datasets, since the homophily assumption in GNNs doesn't hold in those datasets. Notice that we use distinct weight matrices for $\mathbf{H}^{o}$ and $\mathbf{H}^{\tilde{o}}$, so that they do not interfere with each other.

Given node embeddings $\mathbf{h}_i^{\tilde{o}}$ and $\mathbf{h}_i^{s}$ of node v_i, we generate the contrastive loss for structure preserving learning:

$$\mathcal{L}_{str}(v_i) = -log\frac{e^{\mathcal{D}(\mathbf{h}_i^{\tilde{o}},\mathbf{h}_i^{s})/\tau}}{\sum_{j=1}^{N} e^{\mathcal{D}(\mathbf{h}_i^{\tilde{o}},\mathbf{h}_j^{s})/\tau} + \sum_{v\in\{\tilde{o},s\}}\sum_{j=1}^{N}\mathbb{1}_{[j\neq i]}e^{\mathcal{D}(\mathbf{h}_i^{v}\cdot\mathbf{h}_j^{v})/\tau}}. \tag{11}$$

Model Training

With the two major components of ASP, next we introduce the cross-module loss which we utilize to align representations of original view across different modules. Thereafter, we present the overall objective loss for ASP.

Cross-Module Loss. The cross-module loss aims to align representations of $\mathbf{H}^{o}$ and $\mathbf{H}^{\tilde{o}}$. We find that by contrasting representations of the same view but different modules can improve the quality of the proposed framework.

$$\mathcal{L}_{cross}(v_i) = -log\frac{e^{\mathcal{D}(\mathbf{h}_i^{o},\mathbf{h}_i^{\tilde{o}})/\tau}}{\sum_{j=1}^{N} e^{\mathcal{D}(\mathbf{h}_i^{o},\mathbf{h}_j^{\tilde{o}})/\tau} + \sum_{v\in\{o,\tilde{o}\}}\sum_{j=1}^{N}\mathbb{1}_{[j\neq i]}e^{\mathcal{D}(\mathbf{h}_i^{v}\cdot\mathbf{h}_j^{v})/\tau}}, \tag{12}$$

Overall Objective Loss. The overall objective loss of ASP is defined to be a sum of attribute preserving loss, structure preserving loss and cross-module loss:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{attr} + \lambda_1\mathcal{L}_{str} + \lambda_2\mathcal{L}_{cross} \\ &= \frac{1}{N}\sum_{i=1}^{N}\Big[\mathcal{L}_{attr}(v_i) + \lambda_1\mathcal{L}_{str}(v_i) + \lambda_2\mathcal{L}_{cross}(v_i)\Big], \end{aligned} \tag{13}$$

where λ_1 and λ_2 are tuning parameters to weight the importance of $\mathcal{L}_{str}$ and $\mathcal{L}_{cross}$, respectively. The detailed description of our framework is provided in Algorithm 1.

Algorithm 1: ASP algorithm

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$

for epoch $\leftarrow$ 1, 2,... **do**
  Obtain node representations for attribute preserving contrastive learning module $\mathbf{H}^{o}$ and $\mathbf{H}^{a}$;
  Obtain node representations for structure preserving contrastive learning module $\mathbf{H}^{\tilde{o}}$ and $\mathbf{H}^{s}$;
  Calculate attribute loss $\mathcal{L}_{attr}$ based on Eq.(9);
  Calculate structure loss $\mathcal{L}_{str}$ based on Eq.(11);
  Calculate cross-module loss $\mathcal{L}_{cross}$ with Eq.(12);
  Update parameters according to the overall objective loss $\mathcal{L}$ in Eq.(13);
end for
return The optimal encoder weights $\mathbf{\Theta}_a$,$\mathbf{\Theta}_{\tilde{o}}$,$\mathbf{\Theta}_s$.

Experimental Results

In this section, we conduct extensive experiments to validate the proposed ASP. In particular, we aim to answer the following research questions:

- How does ASP perform on homophilous and non-homophilous datasets?
- How do different components affect the performance of ASP?
- How do the hyper-parameters in our model affect the final performance?

We first introduce the experimental settings. Then we report and analyze the experimental results.

Experimental Setup

Datasets To evaluate the performance of different methods, we adopt seven public real-world datasets with different levels of homophily. These datasets can be categorized into two types: homophilous datasets and non-homophilous datasets.

For homophilous datasets, we choose three popular public datasets: Cora, Citeseer and Pubmed (Yang, Cohen, and Salakhutdinov 2016), where nodes represent documents and edges represent citation links. Node attributes of these datasets are bag-of-words representation of documents.

For non-homophilous datasets, we choose four public datasets: Actor, Cornell, Texas, and Wisconsin (Pei et al. 2020). Actor is an actor-only induced subgraph of the film-director-actor-writer network, where each node corresponds to an actor, and edges denote co-occurrence on the same Wikipedia page. Node attributes of Actor correspond to some keywords in the Wikipedia pages. As to Cornell, Texas, and Wisconsin, nodes represent web pages, edges represent hyperlinks between them and node attributes are bag-of-words representation of web pages.

Statistics of all datasets are summarized in Table 1.

Baselines We compare our ASP with representative contrastive self-supervised baselines, including DGI (Veličković et al. 2019), GRACE (Zhu et al. 2020b), BGRL (Thakoor et al. 2021), RoSA (Zhu et al. 2022), as well as supervised baselines GCN (Kipf and Welling 2017), GAT

Dataset	Type	#Nodes	#Edges	#Features	#Classes	Edge Density	Homophily Ratio
Cora	Citation	2,708	5,278	1,433	7	0.0014	0.81
Citeseer	Citation	3,327	4,552	3,703	6	0.0008	0.74
Pubmed	Citation	19,717	44,324	500	3	0.0002	0.80
Cornell	Webpage	183	298	1,703	5	0.0179	0.31
Texas	Webpage	183	325	1,703	5	0.0194	0.11
Wisconsin	Webpage	251	515	1,703	5	0.0164	0.20
Actor	Actor Co-occurence	7,600	30,019	932	5	0.0010	0.22

Table 1: Dataset Statistics.

Method	Cora	Citeseer	Pubmed
Supervised Approaches			
GCN	81.5	70.3	79
GAT	83.0±0.7	72.5±0.7	79.0±0.3
SGC	81.0±0.0	71.9±0.1	78.9±0.0
APPNP	83.4±0.5	71.7±0.5	80.1±0.3
Unsupervised Approaches			
DGI	82.3±0.6	71.8±0.7	76.8±0.6
GRACE	80.9±0.7	67.0±0.5	80.5±0.8
BGRL	82.7±0.6	71.1±0.8	79.6±0.5
RoSA	81.8±0.8	70.9±0.4	**82.1±0.5**
ASP-cos	84.2±0.4	72.8±0.7	79.5±0.7
ASP-J	**84.7±0.6**	**73.0±1.0**	80.7±0.6

Table 2: Node classification accuracy(%) results on homophilous graphs. The best performance is highlighted in bold.

(Veličković et al. 2018), APPNP (Gasteiger, Bojchevski, and Günnemann 2019), and Geom-GCN (Pei et al. 2020). GRACE* and RoSA* are variants of GRACE and RoSA, which use MLP as the base encoder instead of GNNs. Note that Geom-GCN is mainly designed for non-homophilous graphs, thus we only report its performance on four non-homophilous graphs. We take the results of baselines from their original paper if available.

Evaluation Protocol We evaluate the representations learned by our proposed ASP on node classification task. First, we train the model with graph data $(\mathbf{A}, \mathbf{X})$ without any annotated labels. Then, the resulting embeddings are used to train and test a simple l_2 regularized logistic regression classifier. We train the model for 5 runs and report the mean classification accuracy with standard deviation. For Cora, Citeseer and Pubmed, we use the public fixed split introduced by Yang, Cohen, and Salakhutdinov (2016). For Actor, Cornell, Texas, and Wisconsin, we randomly split nodes of each class into 60%, 20%, and 20% for training, validation and testing as introduced by Pei et al. (2020).

Note that we include different distance metrics for constructing attribute graph, which leads to two versions of our framework, ASP-cos and ASP-J, where ASP-cos utilizes cosine distance and ASP-J utilizes Jaccard distance. Experimental results of both versions are reported.

Method	Cornell	Texas	Wisconsin	Actor
Supervised Approaches				
GCN	53.4	63.2	49.1	28.7
GAT	55.4	59.9	60.7	26.9
SGC	52.9	59.5	60.7	24.8
APPNP	55.0	59.5	52.8	28.9
Geom-GCN	60.8	67.6	64.1	31.6
Unsupervised Approaches				
DGI	52.7	57.4	47.4	29.9
GRACE	54.7	55.4	43.4	29.3
GRACE*	66.0	74.8	77.5	31.4
BGRL	48.2	53.6	47.4	27.5
RoSA	59.3	55.1	60.3	26.0
RoSA*	74.3	71.1	77.1	31.3
ASP-cos	**78.2**	**81.6**	**79.4**	32.6
ASP-J	73.4	75.5	75.5	**33.6**

Table 3: Node classification accuracy(%) results on non-homophilous graphs. The best performance is highlighted in bold.

Implementation Details We implement our proposed framework and some baselines using Pytorch (Paszke et al. 2017) and Pytorch Geometric (Fey and Lenssen 2019). The hyperparameters we tune include: (1) the initial learning rate $\in \{1e-1, 1e-2, 1e-3, 1e-4\}$, (2) $k \in \{20, 30, 40, 50, 60, 70\}$ for k nearest neighbors, (3) the aggregation hops $l \in \{10, 20, 30\}$. For homophilous datasets, the output dimension of our encoder network is fixed to 256, while tuned within $\{24, 32, 64, 128\}$ for non-homophilous datasets. We set a patience of 20 and a maximum of 500 epochs for early stopping.

Performance Comparison

We use classification accuracy as the evaluation metric, the relevant results are shown in Table 2 and Table 3. Table 2 shows the results for three homophilous datasets. It is observed that our approach achieves the state-of-the-art results and competes the best one with respect to the existing unsupervised approaches. Compared with supervised baselines, our approach outperforms all the baselines.

We report the results of our method with the strong baselines for four non-homophilous datasets in Table 3. In particular, for GRACE and RoSA, we use both GNN and MLP as

the base encoder. From the results we can see that GRACE and RoSA with MLP as the base encoder performs significantly better than GNN as the base encoder, which shows that structure information extracted by GNN may degrade the expressivity and overall quality of contrastive methods at low level of homophily. ASP outperforms all baselines on all four datasets including methods with MLP encoder GRACE* and RoSA*, which validates the effectiveness of our method for non-homophilous graphs. In particular, on Cornell and Texas, our model ASP outperforms the best baselines RoSA* and GRACE* by a very large margin, i.e., 3.9% and 6.8%, which proves the superiority of our method.

Besides, ASP-J performs better than ASP-cos on all three homophilous datasets (Cora, Citeseer, and Pubmed) and one non-homophilous dataset (Actor), while ASP-cos performs better on other non-homophilous datasets (Cornell, Texas, and Wisconsin). This phenomenon shows that different distance metrics affect the kNN graphs we construct, thus influencing the performance of the whole ASP framework.

Ablation Study

Our proposed ASP employs two contrastive learning modules to learn rich self-supervised information of the input graph. To shed light on the contributions of the main components of our framework, we conduct ablation experiments masking different components under the same hyperparameters and training scheme. Specifically, we build the following ablations:

- Removing the attribute preserving contrastive learning module (w/o AP).
- Removing the structure preserving contrastive learning module (w/o SP).
- Removing the cross-module loss $\mathcal{L}_{cross}$.

As shown in Table 2 and Table 3, ASP achieves more improvement on non-homophilous datasets. Hence, we mainly report the ablation results on non-homophilous datasets. Besides, the choice of distance metric influences the performance of ASP, so we choose the version of ASP which performs better for each dataset, which means for Actor, ASP-J is utilized for ablation study, and for Cornell, Texas and Wisconsin, we use ASP-cos. The results are summarized in Table 4. We can observe that the performance degrades when any of the components is dropped, which indicates the effectiveness of the corresponding components. Especially, when attribute preserving contrastive learning module (AP) is masked, the performance of all four datasets drops significantly, which shows that attribute preserving contrastive learning module essentially boost the performance while structure preserving contrastive learning module and cross-module loss $\mathcal{L}_{cross}$ can further improve the performance.

Parameter Sensitivity

We perform sensitivity analysis on several critical parameters of ASP. Firstly, we measure how the performance is affected by varying the number of nearest neighbors k for kNN graph in the range of $\{20, 30, 40, 50, 60, 70\}$. The results on Cora and Texas are depicted in Figure 2. We get the best results as k reaches 40 for Cora and 70 for Texas. Therefore, choosing a relative large number of nearest neighbors is a reasonable way. Moreover, for both homophilous dataset Cora and non-homophilous dataset Texas, our model is robust to the changes of parameter k. Even in the worst settings of k, ASP still achieves the best performance above all other baselines in Table 2 and Table 3.

Secondly, we test the impact of hop number l and show the results in Figure 2. For homophilous dataset Cora, when l equals a relatively smaller value 10, our model achieves the best result. While for non-homophilous dataset Texas, when l reaches 30, our model achieves the best result. In addition, our model is also robust to the parameter l, as under all settings, ASP achieves comparable performance.

Ablation	Cornell	Texas	Wisconsin	Actor
ASP	78.2	81.6	79.4	33.6
-w/o AP	65.4	64.9	53.3	32.1
-w/o SP	68.6	76.8	52.5	33.5
-w/o $\mathcal{L}_{cross}$	72.0	74.6	53.3	32.6

Table 4: Ablation study on ASP.

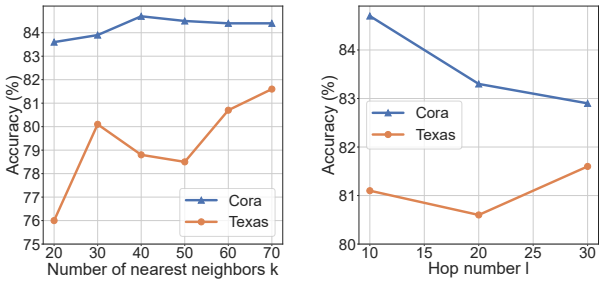

Figure 2: Analysis on critical hyperparameters. The left figure shows the impact of the number of nearest neighbors k in kNN graph, the right figure embodies the influence of hop number l.

Conclusion

In this work, we propose a novel contrastive learning framework, ASP, which effectively preserves both attribute and structure information from the input graph. By performing contrastive learning across three different graph views in a joint fashion, the encoder network learns expressive node representations. Extensive experiment results demonstrate that our model outperforms representative baselines on various real-world datasets with different levels of homophily.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants U1811462, 71725001, and 71910107002, and the National Social Science Foundation of China under Grant 22XJY013. We thank the anonymous reviewers for their insightful and helpful comments.

References

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual Information Neural Estimation. In *ICML*, 531–540.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2014. Spectral Networks and Locally Connected Networks on Graphs. In Bengio, Y.; and LeCun, Y., eds., *ICLR*.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020a. Simple and deep graph convolutional networks. In *International conference on machine learning*, 1725–1735.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Do, K.; Tran, T.; and Venkatesh, S. 2018. Graph Transformation Policy Network for Chemical Reaction Prediction. arXiv:1812.09441.

Fey, M.; and Lenssen, J. E. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR*.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *ICML*, 1263–1272.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, 297–304.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive Multi-View Representation Learning on Graphs. In *ICML*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2020. Strategies for Pre-training Graph Neural Networks. In *International Conference on Learning Representations*.

Jiao, Y.; Xiong, Y.; Zhang, J.; Zhang, Y.; Zhang, T.; and Zhu, Y. 2020. Sub-graph Contrast for Scalable Self-Supervised Graph Representation Learning. arXiv:2009.10273.

Jin, D.; Yu, Z.; Huo, C.; Wang, R.; Wang, X.; He, D.; and Han, J. 2021a. Universal Graph Convolutional Networks. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Jin, W.; Derr, T.; Liu, H.; Wang, Y.; Wang, S.; Liu, Z.; and Tang, J. 2020. Self-supervised Learning on Graphs: Deep Insights and New Direction. arXiv:2006.10141.

Jin, W.; Derr, T.; Wang, Y.; Ma, Y.; Liu, Z.; and Tang, J. 2021b. Node Similarity Preserving Graph Convolutional Networks. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Kondor, R. I.; and Lafferty, J. 2002. Diffusion kernels on graphs and other discrete structures. In *ICML*, 315–322.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.

Li, X.; Zhu, R.; Cheng, Y.; Shan, C.; Luo, S.; Li, D.; and Qian, W. 2022. Finding Global Homophily in Graph Neural Networks When Meeting Heterophily. arXiv:2205.07308.

Li, Y.; Gu, C.; Dullien, T.; Vinyals, O.; and Kohli, P. 2019. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. In *ICML*, 3835–3845.

Lim, D.; Hohne, F. M.; Li, X.; Huang, S. L.; Gupta, V.; Bhalerao, O. P.; and Lim, S.-N. 2021. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. In *Advances in Neural Information Processing Systems*.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. 2022. Graph Self-Supervised Learning: A Survey. *IEEE Transactions on Knowledge and Data Engineering*.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization. In *International Conference on Neural Information Processing Systems*, 271–279.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic Differentiation in PyTorch. In *NIPS 2017 Workshop on Autodiff*.

Pei, H.; Wei, B.; Chang, K. C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*.

Peng, Z.; Dong, Y.; Luo, M.; Wu, X.-M.; and Zheng, Q. 2020a. Self-Supervised Graph Representation Learning via Global Context Prediction. arXiv:2003.01604.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020b. Graph Representation Learning via Graphical Mutual Information Maximization. In *Proceedings of The Web Conference*.

Shlomi, J.; Battaglia, P.; and Vlimant, J.-R. 2021. Graph neural networks in particle physics. *Machine Learning: Science and Technology*.

Shuman, D.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2012. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Processing Magazine*, 30.

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2019. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

Thakoor, S.; Tallec, C.; Azar, M. G.; Azabou, M.; Dyer, E. L.; Munos, R.; Veličković, P.; and Valko, M. 2021. Large-Scale Representation Learning on Graphs via Bootstrapping. arXiv:2102.06514.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What Makes for Good Views for Contrastive Learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. *ICLR*. Accepted as poster.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *ICLR*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying Graph Convolutional Networks. In *ICML*, 6861–6871.

Wu, L.; Lin, H.; Tan, C.; Gao, Z.; and Li, S. Z. 2021. Self-supervised Learning on Graphs: Contrastive, Generative, or Predictive. *IEEE Transactions on Knowledge and Data Engineering*.

Xia, J.; Wu, L.; Chen, J.; Hu, B.; and Li, S. Z. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *Proceedings of the ACM Web Conference 2022*.

Xie, Y.; Xu, Z.; Zhang, J.; Wang, Z.; and Ji, S. 2021. Self-Supervised Learning of Graph Neural Networks: A Unified Review. arXiv:2102.10757.

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *ICML*, 40–48.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph Contrastive Learning Automated. arXiv:2106.07594.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020a. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems*, 5812–5823.

You, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020b. When Does Self-Supervision Help Graph Convolutional Networks? In *ICML*.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zeng, D.; Liu, W.; Chen, W.; Zhou, L.; Zhang, M.; and Qu, H. 2023. Substructure Aware Graph Neural Networks. In *Proc. of AAAI*.

Zeng, D.; Zhou, L.; Liu, W.; Qu, H.; and Chen, W. 2022. A Simple Graph Neural Network via Layer Sniffer. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5687–5691. IEEE.

Zhang, Y.; Zhu, H.; Song, Z.; Koniusz, P.; and King, I. 2022. COSTA: Covariance-Preserving Feature Augmentation for Graph Contrastive Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Zhu, J.; Rossi, R. A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021a. Graph Neural Networks with Heterophily. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12).

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020a. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *Advances in Neural Information Processing Systems*, 7793–7804.

Zhu, Y.; Guo, J.; Wu, F.; and Tang, S. 2022. RoSA: A Robust Self-Aligned Framework for Node-Node Graph Contrastive Learning. arXiv:2204.13846.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020b. Deep Graph Contrastive Representation Learning. In *ICML*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021b. Graph Contrastive Learning with Adaptive Augmentation. In *Proceedings of the Web Conference 2021*, WWW '21, 2069–2080. Association for Computing Machinery.

Zitnik, M.; Agrawal, M.; and Leskovec, J. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*.