Multi-View Robust Graph Representation Learning for Graph Classification

Guanghui Ma[1] , **Chunming Hu**[1,2,3,*] , **Ling Ge**[1] and **Hong Zhang**[4]
[1]School of Computer Science and Engineering, Beihang University, Beijing, China
[2]College of Software, Beihang University, Beijing, China
[3]Zhongguancun Laboratory, Beijing, China
[4]National Computer Network Emergency Response Technical Team / Coordination Center of China
{maguanghui, hucm, geling}@buaa.edu.cn, zhangh@isc.org.cn

Abstract

The robustness of graph classification models plays an essential role in providing highly reliable applications. Previous studies along this line primarily focus on seeking the stability of the model in terms of overall data metrics (e.g., accuracy) when facing data perturbations, such as removing edges. Empirically, we find that these graph classification models also suffer from semantic bias and confidence collapse issues, which substantially hinder their applicability in real-world scenarios. To address these issues, we present MGRL, a multi-view representation learning model for graph classification tasks that achieves robust results. Firstly, we proposes an instance-view consistency representation learning method, which utilizes multi-granularity contrastive learning technique to perform semantic constraints on instance representations at both the node and graph levels, thus alleviating the semantic bias issue. Secondly, we proposes a class-view discriminative representation learning method, which employs the prototype-driven class distance optimization technique to adjust intra- and inter-class distances, thereby mitigating the confidence collapse issue. Finally, extensive experiments and visualizations on eight benchmark dataset demonstrate the effectiveness of MGRL.

1 Introduction

Graph classification models represented by graph neural networks (GNN), which aggregate graph nodes information using the graph topology for representation learning, have demonstrated remarkable performance in various graph classification tasks [Wang *et al.*, 2022; Ma *et al.*, 2022a]. However, this learning mechanism is susceptible to cascade effects, making graph classification models highly sensitive to the graph-structure input [Zhu *et al.*, 2021]. This results in extremely poor robustness of graph classification models against noisy data, severely hindering their implementation in real-world scenarios.

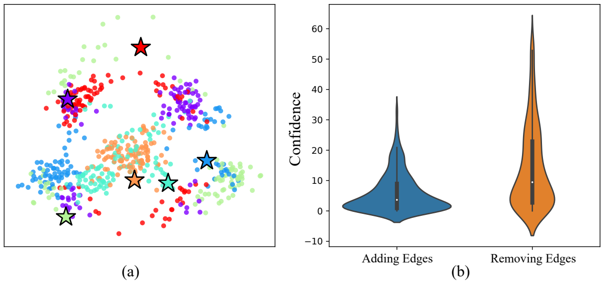


Figure 1: **(a)** Semantic bias analysis on DD dataset. The perturbed graph representations (●) deviate from the original graph representations (★). PCA Visualization of graph representations. **(b)** Confidence collapse analysis on MUT dataset. The confidence variation of the prediction results is up to 60% when removing some edges.

To improve the robustness of graph classification models, researchers have proposed various approaches, such as adversarial attacks based approaches [Zhang *et al.*, 2022; Xu *et al.*, 2022], graph representation learning based approaches [You *et al.*, 2020; Zheng *et al.*, 2020], graph structure learning based approaches [Luo *et al.*, 2021; Song *et al.*, 2022] and data augmentation based approaches [Wang *et al.*, 2021; Han *et al.*, 2022], etc. These approaches all rely on driving the model to converge on the prediction of original and perturbed graphs. In consequence, when confronted with noisy inputs, models trained using the above approaches do not jitter as much in predictive metrics (e.g., accuracy) as models learned using vanilla graph classification models.

Despite the outstanding performance, we empirically find that the existing graph classification approaches cannot guarantee semantic consistency across perturbed and original graphs, which we call the **semantic bias** issue. In detail, for a robust graph classification model, the semantic embedding of the perturbed and original graphs should be as close as possible while maintaining the stability of the metrics. This is crucial for applying graph classification models in some critical scenarios, such as bioinformatics and healthcare. As shown in Figure 1 (a), when we add 30% perturbations (adding edges) to the input graphs of the G-Mixup model [Han *et al.*, 2022], as observed, the semantic representations of the perturbed graphs deviates significantly from that of the original graphs.

To deal with the above semantic bias issue, we propose

[*] Corresponding author

an instance-view consistency representation learning method that performs semantic constraints through multi-granularity contrastive learning. Specifically, we rely on dropouts in the neural network to acquire multiple perturbed representations from the same original graph. Then, we employ unsupervised contrastive learning [Gao *et al.*, 2021] to maximise the semantic consistency of the multiple perturbed representations at both the node- and graph-level. Moreover, we introduce an instance weighting method to avoid semantic damage caused by false-negative representations in contrastive learning.

Another empirical finding is that the prediction confidence (i.e. prediction probability) of existing graph classification models can vary dramatically in response to data perturbations, even leading to a collapse of the prediction results, which we call the **confidence collapse** issue. Existing methods usually simulate perturbation scenarios to evaluate the robustness of graph classification models. The fact is that simulations of perturbation scenarios are restricted, and the overall metrics stability of the dataset does not guarantee that the confidence in the data would remain constant. When experiencing complex or varied disturbances in practical circumstances, predicting confidence in graph classification models collapses. As shown in Figure 1 (b), when we remove 30% edges to the input graphs, for the G-Mix model [Han *et al.*, 2022] , the prediction confidence changes by up to 60%.

To alleviate the confidence collapse issue, we propose a class-view discriminative representation learning method that adjusts intra- and inter-class distances utilizing the prototype-driven class distance optimization technique. Specifically, we employ class prototypes, i.e. class centres, as targets to drive graph representations closer to the class prototype of the same class and away from other class prototypes. In addition, we design prototype centre loss to further increase inter-class spacing by adjusting the distance between prototypes. In this way, the model can obtain discriminative representations, thereby increasing its tolerance to perturbed data and mitigating changes in prediction confidence.

Based upon the above, we effectively integrate these two solutions into a unified model, resulting in our approach, **M**ulti-View **G**raph **R**epresentation **L**earning (MGRL), where both solutions share the common goal: to learn a robust graph representation. The motivation behind this model is that a better representation can lead to more efficient and robust prediction results. We choose several relevant baselines from graph classification robustness studies for comparison on eight benchmark datasets, and the experimental results validate the effectiveness of MGRL.

Summarily, we make the following contributions: **(1)** We empirically discover that existing graph classification models suffer from semantic bias and confidence collapse issues when facing data perturbations. **(2)** We propose a novel model, MGRL, which alleviates the semantic bias issue via instance-view consistency representation learning, and mitigates the confidence collapse issue via class-view discriminative representation learning, respectively. **(3)** We demonstrate the effectiveness of our model through comparison, ablation, and visualization experiments on eight benchmark graph classification datasets.

2 Related Work

The recent advances in deep learning have enabled graph classification models to achieve superior performance on various graph classification tasks [You *et al.*, 2020; Sun *et al.*, 2020; Hassani and Ahmadi, 2020; Xu *et al.*, 2021; Chu *et al.*, 2021; Li *et al.*, 2022b; Ma *et al.*, 2022a; Wang *et al.*, 2022; Ma *et al.*, 2022b]. These approaches typically rely on a well-designed model architecture to understand the semantics of graph data. However, the message aggregation mechanism of graph classification models makes them highly sensitive to the structure of graph data [Zhang and Zitnik, 2020; Zhu *et al.*, 2021], resulting in severe challenges to the robustness of graph classification models

Based on the above considerations, researchers have proposed various methods to address the robustness of these graph models [Geisler *et al.*, 2021; Luo *et al.*, 2021; Wang *et al.*, 2021; Li *et al.*, 2022a; Xu *et al.*, 2022; Han *et al.*, 2022]. For example, [Zheng *et al.*, 2020] and [Luo *et al.*, 2021] enhances the robustness of graph classification models by removing task-irrelevant edges from graph data. However, these previous approaches simply sought to maintain the overall data metric when facing data perturbations. Different from previous work, we experimentally verify that existing graph robust classification models are suffering from semantic bias and confidence collapse problems. Then, we explore the robustness of graph classification models for the first time from these two perspectives .

3 Methodology

3.1 Problem Formulation

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, we denote the node set and edge set as $\mathcal{V} = \{v_1, \cdots, v_N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ respectively. The associated node feature matrix is represented as $X = \{x_v | v \in \mathcal{V}\} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where x_v denotes feature for node v and d is the dimension of input feature. Also, we leverage $A \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ to denote the adjacency matrix, where $A_{i,j} = 1$ if $(v_i, v_j) \in \mathcal{E}$. In this paper, we aim to learn a graph neural encoder $\mathcal{F}(\cdot)$ to obtain more robust representations for downstream graph-level classification task, which can be formalized as : $\hat{y} = \mathcal{F}(\mathcal{G}(X, A))$.

3.2 Overview

This paper proposes a multi-view graph representation learning model (named MGRL), which utilizes the instance-view consistency representation learning method and class-view discriminative representation learning method to alleviate the semantic bias and confidence collapse problems, thereby enhancing the effectiveness and robustness of graph classification models. As illustrated in Figure 2, our model primarily contains a perturbed graph encoder and two representation learning modules with different views.

3.3 Perturbed Graph Encoder

In this part, we employ the perturbed graph encoder to obtain different perturbed representations. For graph data perturbations, most existing works rely on human priors or expert knowledge [You *et al.*, 2020] to design some specific perturbation strategies. Due to the diversity of data distribution and

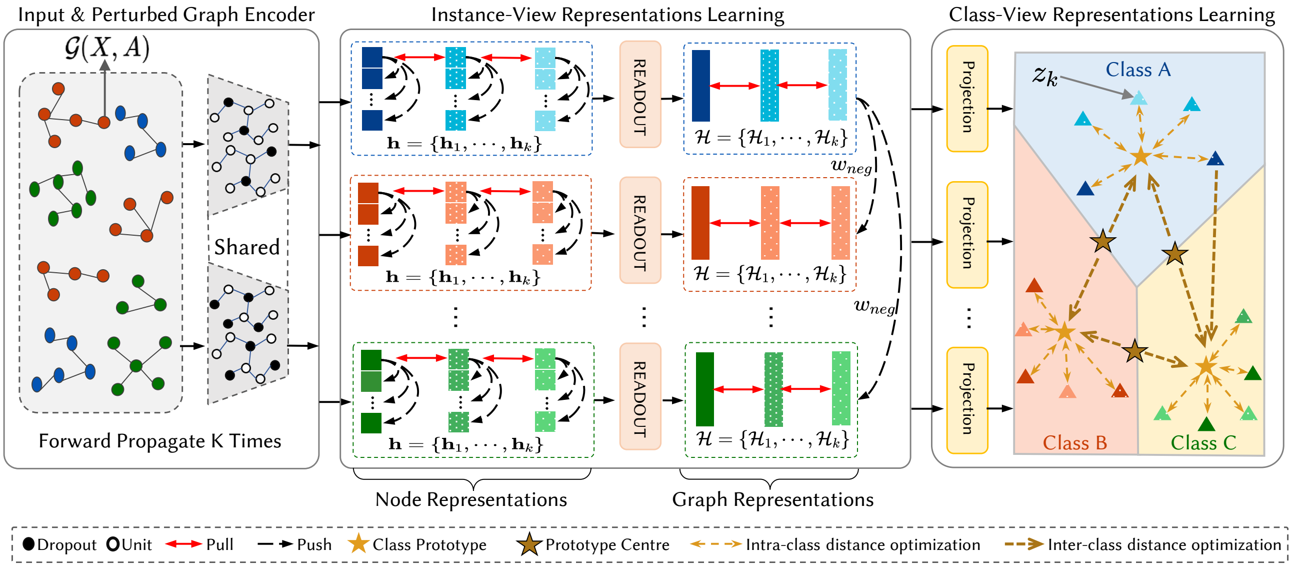

Input & Perturbed Graph Encoder
$\mathcal{G}(X, A)$
Shared
Forward Propagate K Times
Instance-View Representations Learning
$\mathbf{h} = \{\mathbf{h}_1, \cdots, \mathbf{h}_k\}$
READOUT
$\mathcal{H} = \{\mathcal{H}_1, \cdots, \mathcal{H}_k\}$
w_{neg}
Node Representations
Graph Representations
Class-View Representations Learning
Projection
z_k
Class A
Class B
Class C
● Dropout
○ Unit
Pull
Push
Class Prototype
Prototype Centre
Intra-class distance optimization
Inter-class distance optimization

Figure 2: An illustration of our proposed model named MGRL. MGRL performs representation learning from two views to alleviate the semantic bias and confidence collapse issues, improving the effectiveness and robustness of graph classification models.

downstream tasks, these methods are not always effective and may even destroy the graph's semantics [Xia *et al.*, 2022]. To circumvent this difficulty, we introduce the standard dropout as perturbations to obtain different graph embeddings. We refer to these embeddings as perturbed representations, which share the same gold label. This operation is the effective data perturbation method capable of preserving the samples' semantics [Gao *et al.*, 2021].

Specifically, given a graph $\mathcal{G}$ with N nodes, we pass it to the graph neural encoder and forward propagate K times. Since the dropout in the neural network can randomly discard different neurons, the K perturbed representations we obtain have some differences. The process can be formalized as:

$$\{\boldsymbol{h}_k\}_{k=0}^{K} = \textbf{Graph Encoder}\ (\mathcal{G}) \tag{1}$$

where $\boldsymbol{h}_k = \{h_k^n \in \mathbb{R}^{d_1}\}_{n=0}^{N}$ denotes the node representations set of the k^{th} perturbation, h_k^n denotes one node representation from $\boldsymbol{h}_k$, d_1 denotes the dimension of node representation and K denotes the number of perturbation.

3.4 Instance-View Consistency Representation Learning

In this section, we elaborate on our instance-view consistency representation learning approach. It employs a multi-granularity contrastive learning technique to perform semantic consistency learning at both node and graph levels, alleviating the semantic bias issue in graph classification models.

Node-Level Semantic Consistency Learning. We argue that a node representation obtained by a robust graph classification model possesses the following qualities: On the one hand, each node ought to maintain its semantic stability. That is, its semantic representation should not change significantly in response to data perturbations. On the other hand, each node representation should keep its semantic identity, meaning that different node representations are required to retain a specific semantic distance in the embedding space.

To achieve the above purpose, inspired by deep graph contrastive representation learning [Zhu *et al.*, 2020; Hassani and Ahmadi, 2020], we introduce the prevalent unsupervised contrastive learning [Gao *et al.*, 2021] to constrain node representation semantics. In detail, we select a node representation h_k^n from the node representation set $\boldsymbol{h}_k$ as an anchor. Accordingly, the corresponding node representations from other $K-1$ perturbed representation set are employed as positive representations, denoted as h_j. Also, we regard the node representations other than h_k^n and h_j among $\{\boldsymbol{h}_k\}_{k=0}^{K}$ as negative node representations, denoted as h_{neg}. Note that our negative representations contain intra- and inter-graph node representations simultaneously.

We aim to maximize the semantic consistency between the anchor and positive node representations while minimizing the semantic consistency between the anchor and negative node representations. Therefore, the node-level consistency representation learning loss is defined as follows:

$$\mathcal{L}_{nodel} = -\log \frac{\sum_{j=1}^{K} \mathbb{I}_{[j\neq k]} e^{s(h_k^n,h_j)/\tau_1}}{\underbrace{\sum_{neg=1}^{N-1} e^{s(h_k^n,h_{neg})/\tau_1}}_{\text{intra-graph}} + \underbrace{\sum_{neg=1}^{(K-1)(N-1)} e^{s(h_k^n,h_{\text{neg}})/\tau_1}}_{\text{inter-graph}}} \tag{2}$$

where $s(,)$ is a metric function to measure the semantic similarity between different node representations, the $\mathbb{I}$ indicates an indicator function and τ_1 is a temperature parameter. In this paper, we choose cosine similarity as the metric function following the previous works [Zhu *et al.*, 2020].

Graph-Level Semantic Consistency Learning. With the help of node representations, we can employ a READOUT function to obtain perturbed graph representations:

$$\{\mathcal{H}_k\}_{k=0}^{K} = \mathbf{READOUT}(\{\boldsymbol{h}_k\}_{k=0}^{K}) \tag{3}$$

where the $\mathcal{H}_k \in \mathbb{R}^{d_2}$ represents the graph representation for the k^{th} perturbation. We adopt global average pooling as the READOUT function.

To maintain the semantic consistency of the model at the graph level, we can directly adopt a similar learning approach to the node level (Equation 2) for our purpose. Differently, the anchor is represented as a graph representation $\mathcal{H}_k$, and the positive representations are the other $K-1$ perturbed graph representations from the same original graph, marked as $\mathcal{H}_j$. In addition, we refer to all perturbed graphs from other original graph representations within the mini-batch as negative representations, marked as $\mathcal{H}_{neg}$. We can derive the optimisation objective of the graph-level semantic consistency:

$$\mathcal{L}_{graph} = -\log \frac{\sum_{j=1}^{K-1} \mathbb{I}_{[j\neq k]} e^{s(\mathcal{H}_k,\mathcal{H}_j)/\tau_2}}{\sum_{\text{neg}=1}^{(M-1)K} e^{s(\mathcal{H}_k,\mathcal{H}_{\text{neg}})/\tau_2}} \tag{4}$$

where M indicates the number of graph in a mini-batch .

As observed in Equation 4, the unsupervised contrastive learning adopted can push away all other negative graph representations, regardless of whether they have the same gold label as the anchor representation. We refer to these incorrectly pushed-away representations as false negative representations. We argue that this indiscriminate learning mechanism would disrupt the semantic representation learning of the graph encoder. Note that, the above case exists only for graph-level consistency representation learning. The negative representations of the anchor representation come from one mini-batch, which may contain graph representations with the same label as the anchor. In contrast, the negative representations of a node all come from an original graph and share the same label. Considering this, we introduce an instance weighting method to avoid the detriment caused by false negative representations for graph-level consistency learning.

More specifically, given an anchor graph representation $\mathcal{H}_k$, and its label y_k, one of anchor's negative graph representations h_{neg}, the negative representation's label y_{neg}, we can depend on their gold labels to generate the weight of this negative representation:

$$w_{neg} = \begin{cases} 0, & if \;\; y_{neg} = y_k \\ 1, & if \;\; y_{neg} \neq y_i \end{cases} \tag{5}$$

This way, if a graph representation is the anchor's false negative representation, its weight will be set to 0. Conversely, the weight is set to 1. To this point, the Equation 4 can be rewritten as follows:

$$\mathcal{L}_{graph} = -\log \frac{\sum_{j=1}^{K-1} \mathbb{I}_{[j\neq k]} e^{s(\mathcal{H}_k,\mathcal{H}_j)/\tau_2}}{\sum_{\text{neg}=1}^{(M-1)K} w_{neg} e^{s(\mathcal{H}_k,\mathcal{H}_{\text{neg}})/\tau_2}} \tag{6}$$

where w_{neg} represents the weights of negative representations.

Datasets	Graphs	Avg.Nodes	Avg.Edges	Classes
IB	1000	19.22	96.53	2
IM	1500	65.94	89	3
RB	2000	497.63	497.75	2
CO	5000	2475.78	74.49	3
PR	1113	72.82	39.06	2
DD	1178	715.66	284.32	2
NC	4110	32.30	29.87	2
MUT	4337	30.77	30.32	2

Table 1: The statistics of datasets.

Eventually, our instance-view consistency representation learning loss can be defined as follows:

$$\mathcal{L}_{instance\ view} = \alpha_1 \mathcal{L}_{node} + \alpha_2 \mathcal{L}_{graph} \tag{7}$$

where α_1 and α_2 are trade-off parameters.

3.5 Class-View Discriminative Representation Learning

In this section, we describe our class-view discriminative representation learning approach. It utilizes a prototype-driven class distance optimization technique to adjust intra- and inter-class distances, thereby mitigating the confidence collapse issue of graph classification models.

To be specific, given K perturbed graph representations $\{\mathcal{H}_k\}_{k=0}^{K}$, we first map these representations to the class space through a nonlinear projection network, a one-layer MLP followed by an activation function:

$$\{z_k\}_{k=0}^{K} = \mathbf{Projection}(\{\mathcal{H}_k\}_{k=0}^{K}) \tag{8}$$

where the $z_k \in \mathbb{R}^{d_3}$ represents the class space embedding for the k^{th} perturbed graph representation and d_3 denotes the dimension of the representation.

Then, with the help of the gold label, we can directly obtain the class prototypes by averaging the representation vectors with the same label, which can be denoted as:

$$c_i = \frac{1}{\mathcal{N}_i} \sum_{i=0}^{\mathcal{N}_i} \mathbb{I}\left[y_k = y_i\right] z_k \tag{9}$$

where y_i and $\mathcal{N}_i$ denote the label and number belonging to class i respectively. y_k denotes the gold label of z_k.

Since the calculation of class prototypes involves all graph representations, resulting in huge computational expense. Therefore, we introduce a moving average method [Ge *et al.*, 2023] to update the class prototypes. This method can reduce the computational cost while stabilising training:

$$c_{i,t} = \lambda * c_{i,t} + (1-\lambda) * c_{i,(t-1)} \tag{10}$$

where $c_{i,t}$ denotes the prototype of class i in the t step and the $\lambda \in (0,1)$ is the moving average coefficient.

To optimize intra- and inter-class distances, we urge graph representations from the same class to approach the class prototype and move away from other class prototypes. We can learn a more compact class representation by reducing intra-class distances and increasing inter-class distances. The loss function can be defined as follows:

Methods	IB	IM	RB	CO	PR	DD	NC	MUT
MVGRL [2020]	$72.20_{\pm 1.86}$	$53.73_{\pm 1.61}$	-	$79.24_{\pm 1.32}$	$62.14_{\pm 1.20}$	$65.71_{\pm 4.09}$	$69.24_{\pm 1.26}$	$74.38_{\pm 0.86}$
InfoGraph [2020]	$70.20_{\pm 2.56}$	$52.93_{\pm 1.61}$	$74.40_{\pm 2.35}$	$79.00_{\pm 0.66}$	$61.78_{\pm 1.18}$	$69.91_{\pm 3.58}$	$67.25_{\pm 1.96}$	$76.09_{\pm 1.32}$
GraphCL [2020]	$73.20_{\pm 2.03}$	$54.00_{\pm 2.34}$	$75.59_{\pm 2.35}$	$79.12_{\pm 1.18}$	$61.96_{\pm 3.41}$	$69.55_{\pm 3.62}$	$66.71_{\pm 1.46}$	$77.10_{\pm 1.14}$
CSSL [2021]	$72.40_{\pm 1.85}$	$52.79_{\pm 2.71}$	$75.40_{\pm 3.12}$	$80.48_{\pm 0.95}$	$60.71_{\pm 1.59}$	$70.58_{\pm 2.60}$	$67.29_{\pm 0.56}$	$76.78_{\pm 1.22}$
VIB-GSL [2022]	$72.60_{\pm 1.85}$	$53.33_{\pm 1.43}$	-	$79.12_{\pm 1.00}$	$61.60_{\pm 0.89}$	$70.15_{\pm 1.91}$	$65.10_{\pm 2.00}$	$69.63_{\pm 1.88}$
CAL [2022]	$72.02_{\pm 1.72}$	$53.73_{\pm 2.65}$	$73.60_{\pm 1.93}$	$81.04_{\pm 0.46}$	$60.53_{\pm 2.21}$	$70.92_{\pm 2.52}$	$\mathbf{70.41}_{\pm 0.71}$	$76.91_{\pm 0.47}$
NodeSam [2022]	$72.40_{\pm 2.87}$	$53.46_{\pm 0.97}$	$74.20_{\pm 2.11}$	$80.80_{\pm 1.17}$	$61.78_{\pm 1.04}$	$70.42_{\pm 3.37}$	$66.08_{\pm 1.31}$	$75.67_{\pm 1.34}$
G-Mixup [2022]	$73.00_{\pm 1.28}$	$51.33_{\pm 2.14}$	$74.70_{\pm 0.74}$	$78.36_{\pm 0.40}$	$60.71_{\pm 1.95}$	$70.50_{\pm 1.97}$	$66.87_{\pm 0.54}$	$76.32_{\pm 1.72}$
MGRL (Ours)	$\mathbf{73.79}_{\pm 2.19}$	$\mathbf{55.20}_{\pm 2.28}$	$\mathbf{75.80}_{\pm 1.53}$	$\mathbf{81.96}_{\pm 0.85}$	$\mathbf{62.85}_{\pm 1.33}$	$\mathbf{72.60}_{\pm 0.85}$	$68.32_{\pm 0.23}$	$\mathbf{77.56}_{\pm 0.47}$

Table 2: Summary of graph classification results. '-' denotes out of memory to complete the experiment.

$$\mathcal{L}_{pro} = \frac{-1}{KM} \sum_{j=1}^{KM} \cdot \mathbb{I}_{y=y_i} \cdot \log \frac{e^{(d(c_{i,t}, z_i)/\tau_3)}}{\sum_{o=1}^{|C|} \mathbb{I}_{y_i \neq y_o} \cdot e^{(d(z_i, c_{o,t})/\tau_3)}} \tag{11}$$

where z_i belongs to class i, $c_{i,t}$ is the class prototype of class i, $c_{o,t}$ denotes a class prototype of other class, $d(\cdot)$ denotes the distance measure function, $|C|$ is the number of class and τ_3 denotes the distance scaling factor.

Intuitively, we can adjust the class distances for discriminative representation learning via Equation 11, where the denominator represents the distance between class representations z_i and its corresponding class prototype $c_{i,t}$, and the numerator represents the distance between class representations z_i and other class prototypes $c_{o,t}$. Generally, the distance between the z_i and $c_{o,t}$ is much larger than the distance between the z_i and $c_{i,t}$, resulting in $e^{d(z_i, c_{o,t})}$ (denominator) being significantly smaller than $e^{d(c_{i,t}, z_i)}$ (numerator). In particular, as training proceeds, the former keep decreasing and the latter keep increasing to approach e. This causes the gradient signal produced by Equation 11 to maintain decreasing, slowing or even ending training.

To handle this issue, we design prototype centres, the average vector of the two class prototypes, as virtual representations. The prototype centre is closer to the class representations than other class prototypes $c_{o,t}$, so it can provide more gradient signal for class distance optimisation.

Intuitively, we can directly replace the $c_{o,t}$ in Equation 11 by the prototype centre. However, we experimentally find that utilizing the prototype centre to adjust the prototype distance can achieve better results. Therefore, we further adjust the inter-class distance by the following prototype centre loss:

$$\mathcal{L}_{cen} = \frac{1}{|C|} \sum_{j=1}^{|C|} \mathbb{I}_{[j \neq i]} \; d(c_{i,t}, c_{cen}^{i,j}) \tag{12}$$

where $c_{cen}^{i,j}$ denotes the prototype centre of class prototype $c_{i,t}$ and class prototype $c_{j,t}$ Since the cosine function is used as the distance metric function, we minimize the Equation 12 equivalent to maximizing the distance between the prototype centre $c_{cen}^{i,j}$ and the prototype $c_{i,t}$.

Eventually, our loss for class-view discriminative representation learning can be defined as follows:

$$\mathcal{L}_{class\ view} = \alpha_3 \mathcal{L}_{pro} + \alpha_4 \mathcal{L}_{cen} \tag{13}$$

where α_3 and α_4 are trade-off parameters.

3.6 Training Objective

To the end, we obtain prediction results for the graph classification task employing a forward propagation network, which consists of a layer of perceptrons, and the standard cross-entropy loss can be defined as:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{|C|} y log(\hat{y}) \tag{14}$$

where the y denotes the gold label.

Finally, the whole objective training loss for our model can be expressed as follows:

$$\mathcal{L}_{total}(\theta) = \mathcal{L}_{instance\ view} + \mathcal{L}_{class\ view} + \mathcal{L}_{ce} \tag{15}$$

where θ denotes the parameters of the model.

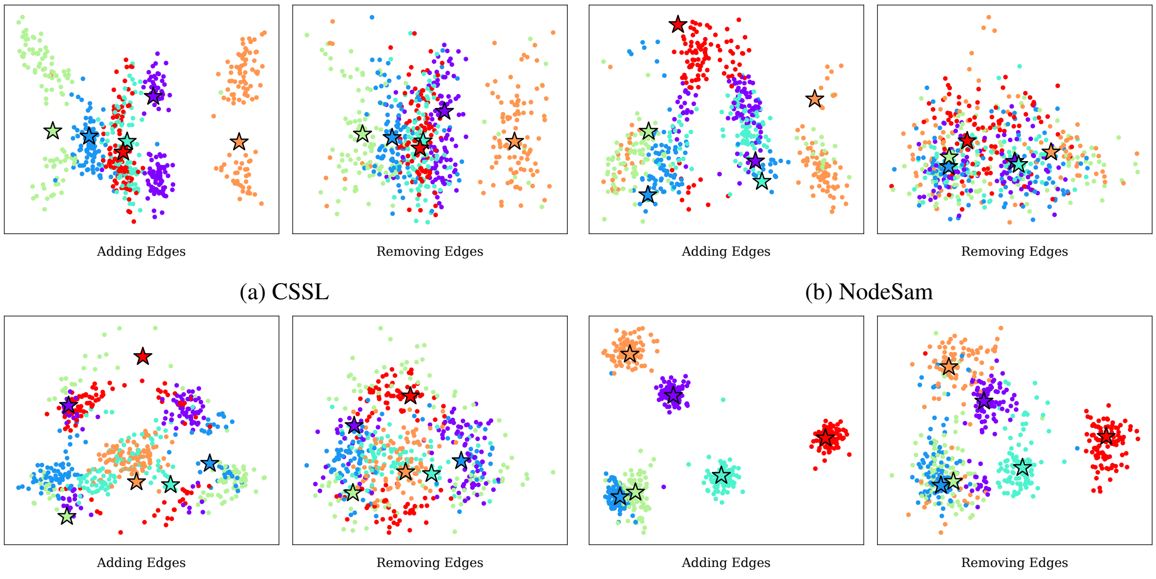
4 Experiment

4.1 Datasets

We evaluate our approach on eight benchmark datasets from TUDataset [Morris *et al.*, 2020], including four social networks datasets, such as IMDB-BINARY (IB), IMDB-MULTI (IM), COLLAB (CO) and REDDIT-BINARY (RB), and four bioinformatics datasets, such as PROTEINS (PR), DD, NCI1 (NC) and Mutagenicity (MUT). The details of the statistical results of the datasets are shown in Table 1.

4.2 Implementation Details

We adopt accuracy as an evaluation metric for the graph classification task and employ GCN [Kipf and Welling, 2017] as the graph encoder. We randomly select 80% of the data for the training set, 10% for the validation set, and the remaining 10% for the test set. During the practical implementation, the batch size is set to 32, the dropout is set to 0.5, the moving average coefficient λ is set to 0.0001, and the α_1, α_2, α_3, α_4 are set to 0.001, 1, 1, 1, respectively. We utilize a grid search technique to select the other best hyper-parameters with the learning rate selected from {0.01, 0.05, 0.001, 0.005}, the temperature parameter τ_1, τ_2 and τ_3 selected from {0.07, 0.1, 0.3, 0.5, 0.7, 0.9} and the K selected from {2,4,6,8,10}. We



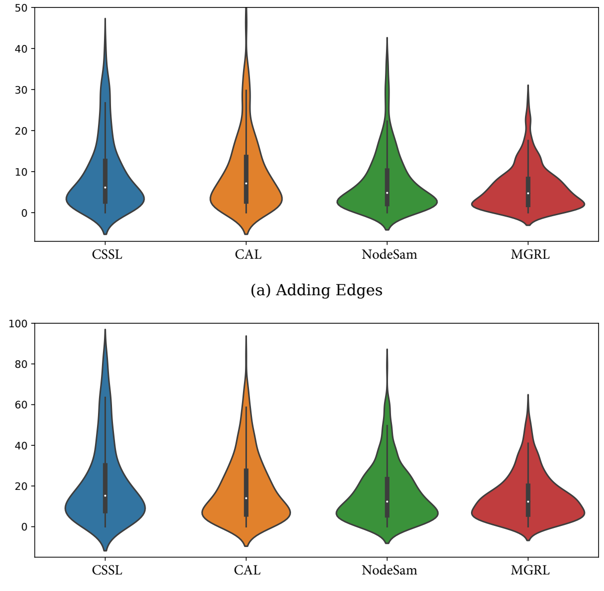
(a) CSSL

(b) NodeSam

(c) G-Mixup

(d) MGRL

Figure 3: Robustness to semantic consistency on DD. The closer the original graph representations (★) and the perturbed graph representations (●), the better the semantic consistency. Different colors denote different graph samples. PCA Visualization of graph representations.



(a) Adding Edges

(b) removing Edges

Figure 4: Robustness to confidence variation on the MUT dataset. The smaller the variation of confidence, the better the robustness.

train our model using the Adam optimizer [Kingma and Ba, 2015] and adopt the validation set to perform model tuning. For comparison with related works, we leverage the open-source implementation of their original papers. We use PyTorch to implement our model on a Linux machine with a GPU device Tesla V100 SXM2 32 GB. All experimental results are from an average of five runs.

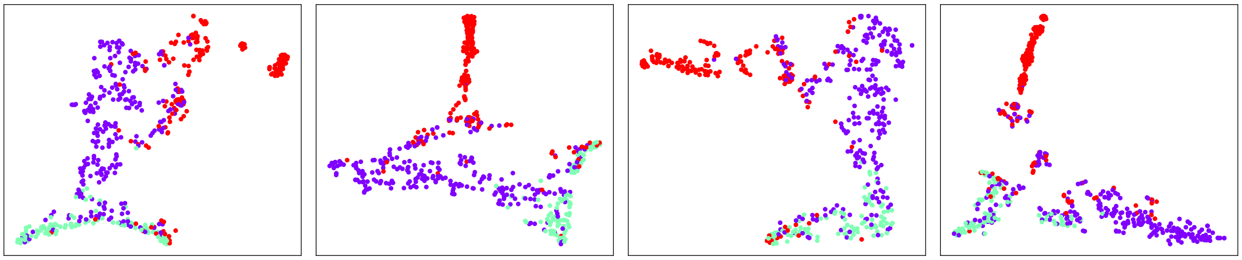
4.3 Performance Comparison

To evaluate the effectiveness and robustness of our model, we select some related works for comparison, including three graph representation learning approaches: **MVGRL** [Hassani and Ahmadi, 2020], **InfoGraph** [Sun *et al.*, 2020], **GraphCL** [You *et al.*, 2020], one graph structure learning approach **VIB-GSL** [Sun *et al.*, 2022], one graph causal learning approach **CAL** [Sui *et al.*, 2022] and three graph data augmentation approaches **CSSL** [Zeng and Xie, 2021], **NodeSam** [Yoo *et al.*, 2022] and **G-Mixup** [Han *et al.*, 2022].

As observed in Table 2, our MGRL shows the best results on most datasets, e.g. 2.84% higher accuracy than the competitive method GraphCL on the CO dataset and 2.14% higher accuracy than the model G-Mixup on the PR dataset, respectively. GraphCL develops four graph-specific data perturbation strategies to improve the model's semantic encoding capability. Unfortunately, these perturbation strategies may corrupt the original semantic structure of the graph. G-Mixup employs a mixup method for data expansion but ignores class-level representation learning. In contrast, our approach not only maintains the semantic integrity of the graph data but also learns the class discriminative representations through a prototype-driven distance optimisation technology.

4.4 Robustness Studies

Robustness to semantic consistency. As shown in Figure 3, our model MGRL demonstrates a strong advantage in semantic consistency under various perturbation scenarios. For instance, when removing edges, the perturbed graph representations of the NodeSam and G-Mixup models completely deviate from the original samples since they only focus on the overall metric of the data. In contrast, our MGRL utilises multi-granularity contrastive learning to obtain a consistent



(a) CSSL

(b) NodeSam

(c) G-Mixup

(d) MGRL

Figure 5: Comparison of the results of four different model graph-level representation. T-SNE visualizations on the CO dataset.

	Model	20%	40%	60%	80%
Removing Edges	NodeSam	71.60	66.10	**60.00**	54.00
	G-Mixup	72.40	65.69	56.60	51.29
	MGRL	**73.40**	**66.50**	**60.00**	**54.49**
Adding Edges	NodeSam	71.90	71.20	70.00	68.30
	G-Mixup	74.20	73.20	73.30	73.00
	MGRL	**75.40**	**75.29**	**74.20**	**74.00**

Table 3: Robustness to overall metric with different topology corruption ratios on the RB dataset.

Methods	IB	CO	PR	NC
MGRL	**73.79**	**81.96**	**62.85**	**68.32**
MGRL$_{w/o \text{ nodel level}}$	72.60	81.40	61.60	67.88
MGRL$_{w/o \text{ graph level}}$	72.60	81.04	60.00	67.88
MGRL$_{w/o \text{ instance view}}$	71.60	80.60	59.57	67.61
MGRL$_{w/o \text{ prototype center}}$	72.40	81.24	61.42	68.12
MGRL$_{w/o \text{ class view}}$	71.40	80.16	60.07	67.13

Table 4: Ablation studies on model components.

representation from the instance perspective.

Robustness to confidence variation. To assess the model's ability to stabilise confidence, we compare the absolute value of the difference in prediction confidence between the original and perturbed graphs for different models. As shown in the Figure 4, MGRL has the smallest jitter in prediction confidence for the two different perturbation scenarios compared to the other three models. The reason is that our prototype-driven class distance optimisation method learns a class-view discriminative representation, thereby improving the tolerance of the model to perturbed samples. This can effectively mitigate the confidence collapse issue.

Robustness to overall data metric. We compare the change in model performance against the perturbations to assess its robustness to the overall data metric. As shown in Table 3, our method achieves the best results when faced with different proportions of perturbations. For example, when removing 20% of the edges, our method obtains an accuracy of 73.40%, while the two different models NodeSam and G-Mixup yield an accuracy of 71.60% and 72.40%.

4.5 Ablation Studies

To explore the effectiveness of each model component, we design five variants of MGRL, including: (1) MGRL$_{w/o \text{ nodel level}}$ removes the nodel-level semantic consistency learning, (2) MGRL$_{w/o \text{ graph level}}$ cuts out the graph-level semantic consistency learning, (3) MGRL$_{w/o \text{ instance view}}$ does not use the instance-view representational learning module, (4) MGRL$_{w/o \text{ prototype center}}$ deletes prototype center loss and (5) MGRL$_{w/o \text{ class view}}$ removes class-view representation learning module.

As observed in Table 4, the removal of any model component results in a decrease in model performance. For example, compared to MGRL, the accuracy of MGRL$_{w/o \text{ nodel level}}$ and MGRL$_{w/o \text{ graph level}}$ decrease by approximately 0.56% and 0.92% on CO dataset respectively, with MGRL$_{w/o \text{ instance view}}$ drops even more. We observe the largest degradation for MGRL$_{w/o \text{ class view}}$, suggesting that class-view representation learning is more important to the classification performance of the model.

4.6 Graph-Level Representation Studies

To demonstrate that our model can learn discriminative representations, we visualize the graph embedding obtained by the graph encoders of different models on the CO dataset. As observed in Figure 5, the different class representations of CSSL, NodeSam and G-Mixup have almost no spacing, and the representations within the same class are more dispersed. In contrast, the graph representations learned via MGLR are more compact and have greater inter-class distances, indicating that our model can achieve more discriminative representations. The reason is that we use the prototype-driven class distance optimization approach to simultaneously force samples away from other class prototypes and different class prototypes away from each other.

5 Conclusion

This paper proposes the MGRL to improve the performance and robustness of graph classification models simultaneously. MGRL utilizes an instance-view consistency representation learning method and a class-view discriminative representation learning method to alleviate semantic bias and confidence collapse issues. The experiments on eight benchmark datasets illustrate the effectiveness of our MGRL.

Acknowledgements

We sincerely thank the reviewers for their insightful comments and valuable suggestions. Thanks for the computing infrastructure provided by Beijing Advanced Innovation Center for Big Data and Brain Computing.

References

[Chu *et al.*, 2021] Guanyi Chu, Xiao Wang, Chuan Shi, and Xunqiang Jiang. Cuco: Graph representation with curriculum contrastive learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 2300–2306. ijcai.org, 2021.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics, 2021.

[Ge *et al.*, 2023] Ling Ge, Chunming Hu, Guanghui Ma, Hong Zhang, and Jihong Liu. Prokd: An unsupervised prototypical knowledge distillation network for zero-resource cross-lingual named entity recognition. *CoRR*, abs/2301.08855, 2023.

[Geisler *et al.*, 2021] Simon Geisler, Tobias Schmidt, Hakan Sirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 7637–7649, 2021.

[Han *et al.*, 2022] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 8230–8248. PMLR, 2022.

[Hassani and Ahmadi, 2020] Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 2020.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Li *et al.*, 2022a] Kuan Li, Yang Liu, Xiang Ao, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Reliable representations make A stronger defender: Unsupervised structure refinement for robust GNN. In Aidong Zhang and Huzefa Rangwala, editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 925–935. ACM, 2022.

[Li *et al.*, 2022b] Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. Let invariant rationale discovery inspire graph contrastive learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 13052–13065. PMLR, 2022.

[Luo *et al.*, 2021] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. Learning to drop: Robust graph neural network via topological denoising. In Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich, editors, *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 779–787. ACM, 2021.

[Ma *et al.*, 2022a] Guanghui Ma, Chunming Hu, Ling Ge, Junfan Chen, Hong Zhang, and Richong Zhang. Towards robust false information detection on social networks with contrastive learning. In Mohammad Al Hasan and Li Xiong, editors, *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1441–1450. ACM, 2022.

[Ma *et al.*, 2022b] Guanghui Ma, Chunming Hu, Ling Ge, and Hong Zhang. Open-topic false information detection on social networks with contrastive adversarial learning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2911–2923. Association for Computational Linguistics, 2022.

[Morris *et al.*, 2020] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020.

[Song *et al.*, 2022] Zixing Song, Yifei Zhang, and Irwin King. Towards an optimal asymmetric graph structure for robust semi-supervised node classification. In Aidong Zhang and Huzefa Rangwala, editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1656–1665. ACM, 2022.

[Sui *et al.*, 2022] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In Aidong Zhang and Huzefa Rangwala, editors, *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 1696–1705. ACM, 2022.

[Sun *et al.*, 2020] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[Sun *et al.*, 2022] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and Philip S. Yu. Graph structure learning with variational information bottleneck. In *Proc. of AAAI*, 2022.

[Wang *et al.*, 2021] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia, editors, *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 3663–3674. ACM / IW3C2, 2021.

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

[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 Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1070–1079. ACM, 2022.

[Xu *et al.*, 2021] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11548–11558. PMLR, 2021.

[Xu *et al.*, 2022] Jiarong Xu, Yang Yang, Junru Chen, Xin Jiang, Chunping Wang, Jiangang Lu, and Yizhou Sun. Unsupervised adversarially robust representation learning on graphs. In *Proc. of AAAI*, 2022.

[Yoo *et al.*, 2022] Jaemin Yoo, Sooyeon Shim, and U Kang. Model-agnostic augmentation for accurate graph classification. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1281–1291. ACM, 2022.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[Zeng and Xie, 2021] Jiaqi Zeng and Pengtao Xie. Contrastive self-supervised learning for graph classification. In *Proc. of AAAI*, 2021.

[Zhang and Zitnik, 2020] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[Zhang *et al.*, 2022] Mengmei Zhang, Xiao Wang, Meiqi Zhu, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. Robust heterogeneous graph neural networks against adversarial attacks. In *Proc. of AAAI*, 2022.

[Zheng *et al.*, 2020] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. Robust graph representation learning via neural sparsification. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11458–11468. PMLR, 2020.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *CoRR*, abs/2006.04131, 2020.

[Zhu *et al.*, 2021] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Qiang Liu, Shu Wu, and Liang Wang. Deep graph structure learning for robust representations: A survey. *CoRR*, abs/2103.03036, 2021.