ReGCL: Rethinking Message Passing in Graph Contrastive Learning

Cheng Ji[1,2], Zixuan Huang[2], Qingyun Sun[1,2], Hao Peng[1,2], Xingcheng Fu[3], Qian Li[1,2], Jianxin Li[1,2]*

[1]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China
[2]School of Computer Science and Engineering, Beihang University, China
[3]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
{jicheng,penghao,liqian,lijx}@act.buaa.edu.cn, {huangzx,sunqy}@buaa.edu.cn, fuxc@gxnu.edu.cn

Abstract

Graph contrastive learning (GCL) has demonstrated remarkable efficacy in graph representation learning. However, previous studies have overlooked the inherent conflict that arises when employing graph neural networks (GNNs) as encoders for node-level contrastive learning. This conflict pertains to the partial incongruity between the feature aggregation mechanism of graph neural networks and the embedding distinction characteristic of contrastive learning. Theoretically, to investigate the location and extent of the conflict, we analyze the participation of message-passing from the gradient perspective of InfoNCE loss. Different from contrastive learning in other domains, the conflict in GCL arises due to the presence of certain samples that contribute to both the gradients of positive and negative simultaneously under the manner of message passing, which are opposite optimization directions. To further address the conflict issue, we propose a practical framework called ReGCL, which utilizes theoretical findings of GCL gradients to effectively improve graph contrastive learning. Specifically, two gradient-based strategies are devised in terms of both message passing and loss function to mitigate the conflict. Firstly, a gradient-guided structure learning method is proposed in order to acquire a structure that is adapted to contrastive learning principles. Secondly, a gradient-weighted InfoNCE loss function is designed to reduce the impact of false negative samples with high probabilities, specifically from the standpoint of the graph encoder. Extensive experiments demonstrate the superiority of the proposed method in comparison to state-of-the-art baselines across various node classification benchmarks.

1 Introduction

Inspired by recent advances in contrastive learning (CL) (Liu et al. 2021) in the fields of computer vision (CV) (Logeswaran and Lee 2018; He et al. 2020; Chen et al. 2020; Chuang et al. 2020) and natural language processing (NLP) (Oord, Li, and Vinyals 2018), graph contrastive learning (GCL) has emerged as a powerful self-supervised learning technique (Wu et al. 2021b; Liu et al. 2022b; Ji et al. 2023a; Liang et al. 2022). The combination of expressive power in graph neural networks (GNNs) (Kipf and Welling 2017; Veličković et al. 2018) and the effective self-supervised learning ability of contrastive learning have sparked significant interest in investigating various aspects of GCL, such as augmentation mechanisms (Yu et al. 2022; Zhang et al. 2023), negative sampling techniques (Xia et al. 2022), and contrastive loss functions (Liu et al. 2022a). Nevertheless, there is a noticeable gap in the literature that specifically focuses on the core problem of graph encoders in GCL. It has been observed that GNN and GCL present specific conflict issues in this paper.

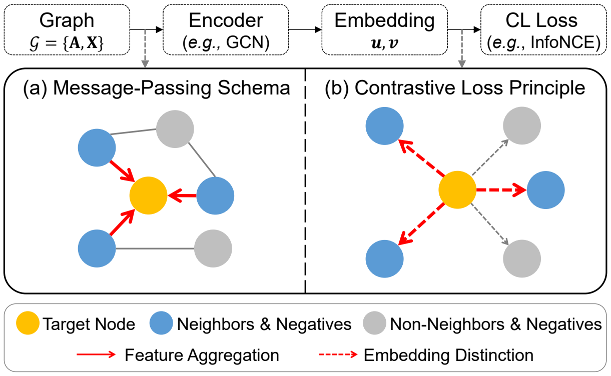

Graph
$\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$
Encoder
(*e.g.*, GCN)
Embedding
$\boldsymbol{u}, \boldsymbol{v}$
CL Loss
(*e.g.*, InfoNCE)
(a) Message-Passing Schema
(b) Contrastive Loss Principle
Target Node
Neighbors & Negatives
Non-Neighbors & Negatives
Feature Aggregation
Embedding Distinction

Figure 1: Conflict arises between the message-passing mechanism and the contrastive loss function. (a) The feature of neighbors is aggregated to the target node through a message-passing schema, resulting in close proximity between them. On the other hand, (b) the contrastive loss function aims to push them far apart, including the neighbors.

Most existing works in GCL primarily employ graph neural networks as encoders (Zhu et al. 2020, 2021; Tong et al. 2021; Wang et al. 2022), similar to semi-supervised node classification. GNNs employ aggregation operators within the local neighborhood to collect features from neighboring nodes, leading to the generation of embeddings that exhibit higher similarity within the neighbors. (Kipf and Welling 2017). GCL then optimizes the model using noise-contrastive estimation loss, such as the InfoNCE loss function (Oord, Li, and Vinyals 2018), which is a noise contrastive estimation (NCE) based objective and identifies each sample by contrasting the differences between the target node and its negatives, including its neighbors aggregated by the GNN encoder (Zhu et al. 2020). The aforementioned approach has

*Corresponding Author.

demonstrated encouraging outcomes, thereby prompting a surge in research endeavors within this field (Zhu et al. 2020, 2021; Tong et al. 2021; Wang et al. 2022). However, most previous studies have overlooked the investigation of whether directly employing GNNs as encoders in GCL is in line with the fundamental principles of contrastive learning.

An overlooked challenge is the conflict issue between the message-passing paradigm and noise contrastive estimation in node-level contrastive learning, as illustrated in Figure 1. Different from contrastive learning methods employed in other domains, GCL incorporates the step of neighborhood aggregation before the application of the contrastive loss function. The conflict stems from the disparity in approaches. The message-passing paradigm in GNNs attempts to propagate information between neighborhoods resulting in reducing the distances between adjacent nodes, thereby making them **close** to their neighbors. On the contrary, in accordance with the principle of InfoNCE, GCL employs a methodology where each node and its augmented version is considered a negative sample for all other samples. This effectively **widens** the distance between nodes in the latent space, enabling discrimination between samples. Consequently, a conflict arises between the feature aggregation of GNNs and the embedding distinction of GCL. Each node within the network undergoes partially contradictory optimization directions, as some nodes are encouraged by GNNs to move closer, while simultaneously being repelled from each other by GCL.

To further investigate and address the conflict issue, a theoretical analysis is conducted from the perspective of gradients. Specifically, the effects of different samples (*i.e.*, inter-view negative samples, intra-view negative samples, and positive samples) on both the positive and negative contributions to the gradients of GCL are explored. It is concluded that the conflict arises due to the simultaneous involvement of certain samples' features (the neighbors of the target sample and the positive sample) in both positive and negative gradients. To mitigate this conflict, we propose ReGCL, which consists of a gradient-guided structure learning method (GGSL) and a gradient-weighted InfoNCE loss function (GW-NCE). Specifically, a CL-adapted adjacency matrix is contained by the gradient estimator of GGSL, weakening the conflicts brought by GNNs. The embeddings are subsequently inputted into the gradient selector of GW-NCE in order to derive coefficients for positive and negative samples within the InfoNCE loss function. The main contributions are summarized as follows:

- We study the partial conflict issue between GNNs and node-level GCL under a theoretical analysis of gradients, exploring the location and extent of the conflict. To the best of our knowledge, it is the first attempt to study the conflict issue from the perspective of gradient.
- Building upon the theoretical findings, we propose a solution named ReGCL, which aims to alleviate the conflict by incorporating gradient-guided structure learning and gradient-weighted InfoNCE.
- Extensive experimental results demonstrate the superior performance of ReGCL in comparison to multiple state-of-the-art baselines on node classification benchmarks.

2 Preliminary

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ represents the set of nodes with a cardinality of N, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. Let $\mathbf{A} \in \{0,1\}^{N \times N}$ denote the adjacency matrix and $\mathbf{X} \in \mathbb{R}^{N \times F}$ be the feature matrix, where F denotes the dimension of features. Node-level graph contrastive learning methods first sample two augmentations $t \sim \mathcal{T}$ from a pool of augmentation functions $\mathcal{T}$. The augmentations generate two distinct views $\mathcal{G}_1$ and $\mathcal{G}_2$ of the original graph, where $\mathcal{G}_1 = (\mathbf{A}_1, \mathbf{X}_1)$ and $\mathcal{G}_2 = (\mathbf{A}_2, \mathbf{X}_2)$.

GCL subsequently employs message-passing neural networks to obtain the embeddings of nodes. Here, we focus on the single-layer graph convolution network (GCN). Consider a node $u_i \in \mathcal{G}_1$ as the target node:

$$\boldsymbol{u}_i = \boldsymbol{\Theta}^{\top} \sum_{j \in \mathcal{N}_i \cup \{i\}} \frac{e_{ji}}{\sqrt{d_j d_i}} \boldsymbol{x}_j, \quad i \in [1, N] \tag{1}$$

where $d_i = 1 + \sum_{j \in [1,N]} \mathbf{A}_{ji,1}$, e_{ji} is the edge weight from source node j to target node i, $\mathcal{N}_i$ is the neighbors of u_i in $\mathcal{G}_1$, $\boldsymbol{x}_i \in \mathbf{X}_1$ is the input feature of the node u_i, and $\boldsymbol{u}_i$ is the learned embedding.

After applying a projection function, graph contrastive learning seeks to identify the node u_i using an InfoNCE-based loss. This loss function aims to keep the embeddings of the same node in different views $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ close together (*i.e.*, positive pair), while simultaneously pushing other node pairs further apart (*i.e.*, negative pair):

$$\mathcal{L}_i = -\log \frac{f(\boldsymbol{u}_i, \boldsymbol{v}_i)}{f(\boldsymbol{u}_i, \boldsymbol{v}_i) + \sum_{k \neq i} f(\boldsymbol{u}_i, \boldsymbol{v}_k) + \sum_{k \neq i} f(\boldsymbol{u}_i, \boldsymbol{u}_k)}, \tag{2}$$

where $f(\cdot,\cdot) = \exp(\mathrm{sim}(\cdot,\cdot)/\tau)$, and $\mathrm{sim}(\boldsymbol{u}_i, \boldsymbol{v}_i) = \boldsymbol{u}_i \cdot \boldsymbol{v}_i / ||\boldsymbol{u}_i|| \cdot ||\boldsymbol{v}_i||$ is the cosine similarity, τ is the temperature.

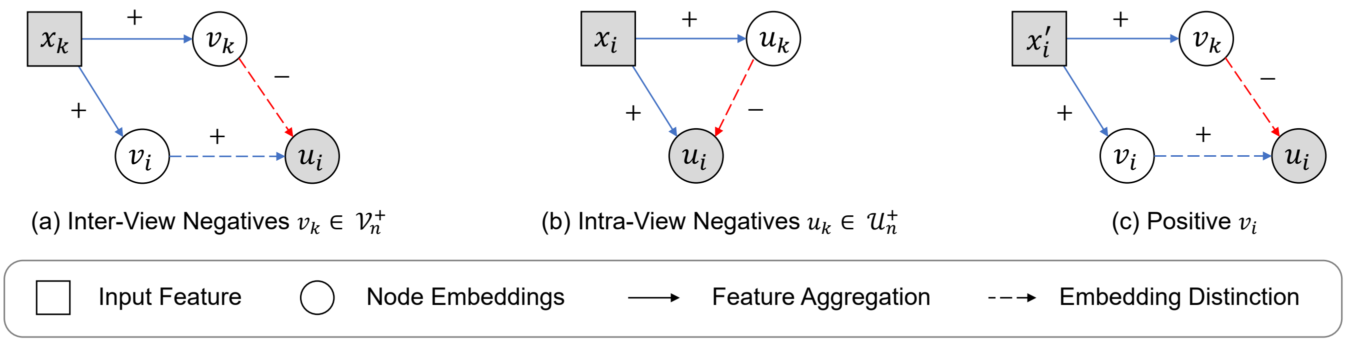
3 Theoretical Analysis: Conflicts between GNN and GCL

The problem of conflict in node-level GCL arises when specific samples are simultaneously included in both the aggregation of positive and negative samples. Consequently, for a given target node u_i, the same other samples may have opposite impacts on the optimization. This conclusion is reached by conducting a gradient analysis in this section, which is illustrated in Figure 2 showcasing three distinct conflicts.

3.1 Gradient Analysis

In order to investigate the occurrence and magnitude of conflicts, a theoretical analysis is conducted on the gradient in graph contrastive learning. Unlike previous studies (Wang and Liu 2021; Wu et al. 2021a), we identify conflicts by analyzing the gradients with respect to the features $\boldsymbol{x}$ (*i.e.*, by taking the messaging-passing into account). It is imperative to perform this as conflicts cannot be identified only by analyzing the gradients with respect to representations or similarities. Formally, for a target u_i, the gradients of graph contrastive learning *w.r.t.*$\boldsymbol{x}_i$ is as follows:

$$\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{x}_i} = \boldsymbol{\Phi}(\underbrace{C(\boldsymbol{u}_i, \boldsymbol{v}_k)}_{\text{inter-view negatives}} + \underbrace{C(\boldsymbol{u}_i, \boldsymbol{u}_k)}_{\text{inter-view negatives}} + \underbrace{C(\boldsymbol{u}_i, \boldsymbol{v}_i)}_{\text{positives}}) \cdot \mathrm{c}_i, \tag{3}$$



(a) Inter-View Negatives $v_k \in \mathcal{V}_n^+$ (b) Intra-View Negatives $u_k \in \mathcal{U}_n^+$ (c) Positive v_i

Figure 2: Conflict Identification. (a) Conflict on inter-view negative samples. The feature of negative $\boldsymbol{x}_k$ is aggregated to the embeddings of itself $\boldsymbol{v}_k$ and the positive samples $\boldsymbol{v}_i$ in cases where the negative is also the neighbor of the positive sample. However, the action of contrastive loss to $\boldsymbol{v}_k$ and $\boldsymbol{v}_i$ are opposite, making $\boldsymbol{x}_k$ play a contradictory role in the optimization process. (b) Conflict on intra-view negative samples. Part of the embedding of the intra-view negative sample $\boldsymbol{u}_k$ has the participation of the target node feature $\boldsymbol{x}_i$ when the negative is adjacent to the target node, which leads to the opposite effect of $\boldsymbol{x}_i$ on optimization. (c) Conflict on inter-view negative samples. Similar to (a), the feature of positive sample $\boldsymbol{x}_i'$ also has the conflict issue.

where $\boldsymbol{\Phi} = \mathbf{W}^\top \boldsymbol{\Theta}^\top$ represents the parameters of GCN and projection function, $\mathrm{c}_i = \frac{\mathbf{I}-\tilde{\boldsymbol{u}}_i\tilde{\boldsymbol{u}}_i^\top}{\tau||\boldsymbol{u}_i||\sqrt{d_i d_i}}\boldsymbol{\Phi}$ is a constant *w.r.t.*$\boldsymbol{x}_i$. Specifically, $C(\boldsymbol{u}_i, \boldsymbol{v}_k)$ denotes the contribution of inter-view negative pairs, $C(\boldsymbol{u}_i, \boldsymbol{u}_k)$ denotes the contribution of intra-view negative pairs, and $C(\boldsymbol{u}_i, \boldsymbol{v}_i)$ donates the contribution of the positive pairs:

$$C(\boldsymbol{u}_i, \boldsymbol{v}_k) = \sum_{k\neq i}\sum_{j\in\mathcal{N}_k\cup\{k\}} P(\boldsymbol{u}_i, \boldsymbol{v}_k)\frac{e_{jk}}{\sqrt{d_k d_j}}\boldsymbol{x}_j, \quad (4)$$

$$C(\boldsymbol{u}_i, \boldsymbol{u}_k) = \sum_{k\neq i}\sum_{j\in\mathcal{N}_k\cup\{k\}} P(\boldsymbol{u}_i, \boldsymbol{u}_k)\frac{e_{jk}}{\sqrt{d_k d_j}}\boldsymbol{x}_j, \quad (5)$$

$$C(\boldsymbol{u}_i, \boldsymbol{v}_i) = \sum_{j\in\mathcal{N}_i\cup\{i\}} (P(\boldsymbol{u}_i, \boldsymbol{v}_i)-1)\frac{e_{ji}}{\sqrt{d_i d_j}}\boldsymbol{x}_j, \quad (6)$$

where $P(i,j) = \mathrm{softmax}(f(i,j)) \in [0,1]$ is the probability of i being identified as j. The proof is in the Appendix A. We have the following observations: (1) The gradient directions of each sample $\boldsymbol{x}_j$ in Eq. (4-5) and Eq. (6) exhibit opposite directions. (2) There are specific samples that simultaneously contribute to both the gradients of negatives and positive instances, resulting in conflicts between GNN and GCL.

3.2 Conflict Identification and Quantification

The emergence of conflicts is observed within specific samples that are engaged in both positive and negative gradients, as previously elucidated. To determine the location and nature of the conflict, we conducted a gradient analysis considering different types of conflicting samples, including inter-view negatives, intra-view negatives, and positives.

Conflict on Inter-View Negative Samples. Consider a set of inter-view negative samples $\mathcal{V}_n$ for the target node u_i, which can be divided into two disjoint subsets $\mathcal{V}_n = \mathcal{V}_n^+ \cup \mathcal{V}_n^-$ and $\mathcal{V}_n^+ \cap \mathcal{V}_n^- = \emptyset$. $\mathcal{V}_n^+$ represents the set of samples adjacent to the positive sample v_i, and $\mathcal{V}_n^-$ stands for nodes not adjacent to v_i. Specifically, the conflict occurs in $\mathcal{V}_n^+$ because these samples participate not only in Eq. (4) as well as being present in Eq. (6) as the neighbors of the positive sample. Formally, for $v_k \in \mathcal{V}_n^+$, the conflict in which $\boldsymbol{v}_k$ participates is measured by the weight coefficients of $\boldsymbol{x}_k$:

$$w(v_k, -) = \sum_{j\in\mathcal{N}_k\cup\{k\}} P(\boldsymbol{u}_i, \boldsymbol{v}_j)\frac{e_{kj}}{\sqrt{d_k d_j}}, \quad (7)$$

$$w(v_k, +) = (P(\boldsymbol{u}_i, \boldsymbol{v}_i)-1)\frac{e_{ki}}{\sqrt{d_i d_k}}, \quad (8)$$

where $w(v_k, -)$ is the weight of $\boldsymbol{x}_k$ in the gradients of inter-view negatives and $w(v_k, +)$ represents the weight in positives. The directions of the above two weights are also opposite, which causes conflict.

Conflict on Intra-View Negative Samples. Let $\mathcal{U}_n$ denote the collection of intra-view negative samples pertaining to the target node u_i. $\mathcal{U}_n$ can be divided into two disjoint subsets $\mathcal{U}_n = \mathcal{U}_n^+ \cup \mathcal{U}_n^-$ and $\mathcal{U}_n^+ \cap \mathcal{U}_n^- = \emptyset$, where $\mathcal{U}_n^+$ represents the set of samples that are adjacent to the target node u_i, while $\mathcal{U}_n^-$ refers to nodes that are not adjacent to u_i. Specifically, the conflict is in $\mathcal{U}_n^+$ because the target node participates in the message-passing of $\mathcal{U}_n^+$ in Eq. (5), which should not be included in the gradients of negatives. Formally, given the target node u_i, the conflict within $\mathcal{U}_n^+$ can be quantified by the weight of $\boldsymbol{x}_i$ in the gradients of intra-view negatives:

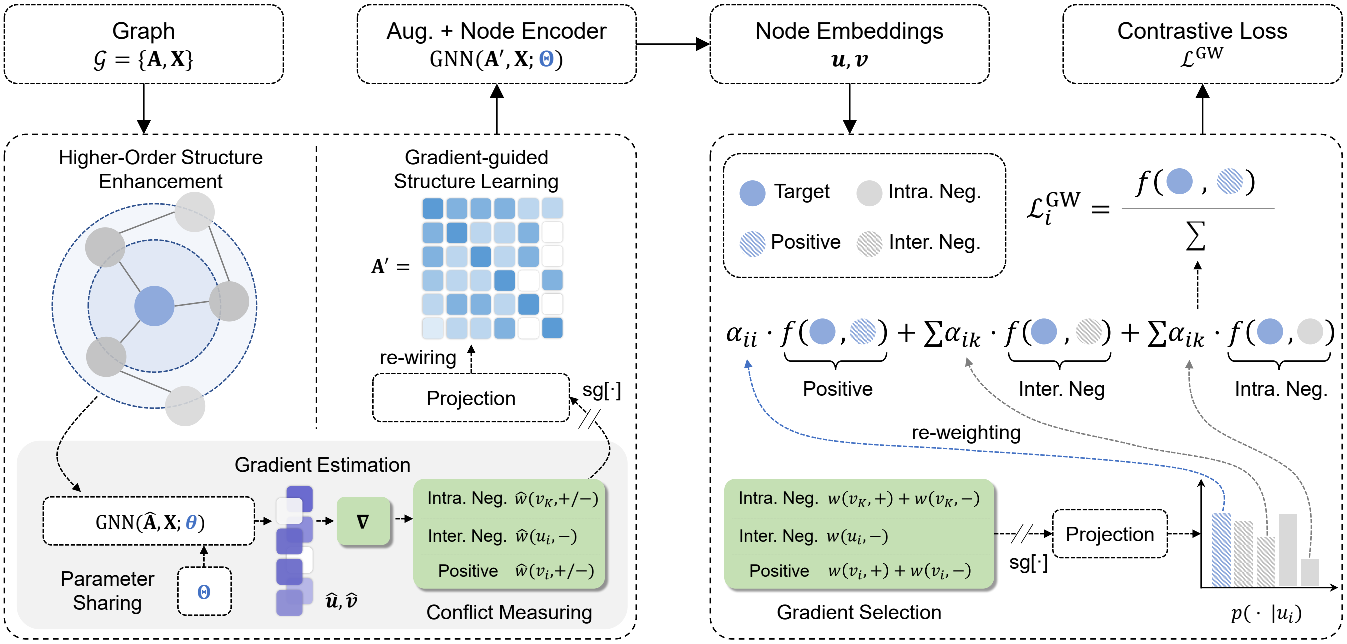
$$w(u_i, -) = \sum_{j\in\mathcal{U}_n^+} P(\boldsymbol{u}_i, \boldsymbol{u}_j)\frac{e_{ij}}{\sqrt{d_i d_j}}. \quad (9)$$

Conflict on Positive Samples. Denote v_i as the positive sample of the target node u_i. Similar to inter-view negatives, the conflict on the positive sample is caused by v_i participating in both the message-passing of negatives in Eq. (4) and positives in Eq. (6). Formally, the conflict of v_i is as follows:

$$w(\boldsymbol{v}_i, -) = \sum_{j\in\mathcal{V}_n^+} P(\boldsymbol{u}_i, \boldsymbol{u}_j)\frac{e_{ij}}{\sqrt{d_i d_j}}, \quad (10)$$

$$w(\boldsymbol{v}_i, +) = (P(\boldsymbol{u}_i, \boldsymbol{v}_i)-1)\frac{e_{ii}}{\sqrt{d_i d_i}}, \quad (11)$$

where $w(\boldsymbol{v}_i, -)$ and $w(\boldsymbol{v}_i, +)$ represent the conflict of v_i in the gradients of negatives and positives.


Graph
$\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$
Aug. + Node Encoder
GNN($\mathbf{A}', \mathbf{X}; \boldsymbol{\Theta}$)
Node Embeddings
$\boldsymbol{u}, \boldsymbol{v}$
Contrastive Loss
$\mathcal{L}^{\mathrm{GW}}$
Higher-Order Structure Enhancement
Gradient-guided Structure Learning
$\mathbf{A}' =$
re-wiring
Projection
sg[·]
Gradient Estimation
GNN($\hat{\mathbf{A}}, \mathbf{X}; \boldsymbol{\theta}$)
∇
Intra. Neg. $\hat{w}(v_K,+/-)$
Inter. Neg. $\hat{w}(u_i,-)$
Positive $\hat{w}(v_i,+/-)$
$\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}$
Parameter Sharing
$\boldsymbol{\Theta}$
Conflict Measuring
Target
Intra. Neg.
Positive
Inter. Neg.
$\mathcal{L}_i^{\mathrm{GW}} = \frac{f(\,,\,)}{\sum}$
$\alpha_{ii} \cdot f(\,,\,) + \sum\alpha_{ik} \cdot f(\,,\,) + \sum\alpha_{ik} \cdot f(\,,\,)$
Positive
Inter. Neg
Intra. Neg.
re-weighting
Intra. Neg. $w(v_K,+) + w(v_K,-)$
Inter. Neg. $w(u_i,-)$
Positive $w(v_i,+) + w(v_i,-)$
Projection
sg[·]
Gradient Selection
$p(\cdot\ |u_i)$

(a) Gradient-guided Structure Learning (GGSL)

(b) Gradient-weighted InfoNCE (GW-NCE)

Figure 3: ReGCL framework. (a) Gradient-guided structure learning (GGSL) learning a CL-adapted adjacency matrix $\mathbf{A}'$ through higher-order structure enhancement and the gradient estimation, leveraging the theoretical analysis on GCL gradients. (b) Gradient-weighted InfoNCE (GW-NCE) generates the coefficients on positive and negatives in the contrastive loss.

4 Methodology

In this section, we present our proposed model, ReGCL, as depicted in Figure 3. To address the conflict issues between GNN and GCL, ReGCL comprises two primary components: (1) *Gradient-guided Structure Learning* (GGSL) to weaken the impact of feature smoothing of latent negative samples in the message passing stage, and (2) *Gradient-weighted InfoNCE* (GW-NCE) is employed to decrease the weight assigned to potential false negatives in contrastive loss.

4.1 Gradient-guided Structure Learning

To improve message passing and enhance its adaptability to graph contrastive learning, we propose a gradient-guided structure learning (GGSL) to learn new edges and weights, thereby mitigating the adverse effects of feature smoothing.

Gradient Estimation. As gradients are not accessible in the context of message-passing, we propose using a gradient estimator prior to the graph encoder in order to acquire the aforementioned weights. To obtain the accurate gradient, we first employ a $\mathrm{GNN}(\mathbf{A}, \mathbf{X}; \boldsymbol{\theta})$ as the gradient estimator to obtain the embeddings before feeding them into the encoder. In particular, the gradient estimator is associated with the same parameter as the graph encoder $\mathrm{GNN}(\mathbf{A}, \mathbf{X}; \boldsymbol{\Theta})$, rather than being updated through back-propagation (*i.e.*, $\boldsymbol{\theta} \leftarrow \boldsymbol{\Theta}$). Subsequently, we can calculate the estimated weights $\hat{w}$ as:

$$\hat{\boldsymbol{u}} = \mathrm{GNN}(\mathbf{A}_1,\mathbf{X}_1; \boldsymbol{\theta}),\ \hat{\boldsymbol{v}} = \mathrm{GNN}(\mathbf{A}_2, \mathbf{X}_2; \boldsymbol{\theta}), \tag{12}$$

$$\hat{w} = \Omega(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}, \mathbf{A}_1, \mathbf{A}_2), \tag{13}$$

where $\Omega(\cdot)$ represents the functions of Eq. (7-11).

Gradient-guided Structure Learning. It is observed that there are three weights, $\{\hat{w}(v_k, +), \hat{w}(u_i, -), \hat{w}(v_i, -)\}$, which are deemed unsuitable for graph contrastive learning due to the impact of message-passing. The absolute values of these measurements indicate the intensity of the conflicts. To mitigate such conflicts, one can rebuild the edge weight based on the given values. Due to the correctness of GNN and GCL, we aim to find a trade-off solution. On one hand, it is necessary to attenuate all of their effects, resulting in a reduction of the corresponding edge weight in $\hat{w}$. Moreover, a higher value of $\hat{w}$ indicates a greater likelihood of being considered as neighbors by the encoder compared to negative samples. Therefore, we propose to utilize an increasing function that has a range spanning from 0 to 1 as the function for projecting edge weights.

$$\mathbf{A}'_{ij} = \frac{1}{n_{ij}} \sum \frac{1}{1 + \exp(-\operatorname{sg}[\tilde{w}_{ij}])}, \tag{14}$$

where $\mathbf{A_{ij}}'$ represents the edge weight from node i to node j, n_{ij} represents the total number of times the edge appears in $\{\hat{w}(v_k, +), \hat{w}(u_i, -), \hat{w}(v_i, -)\}$, and $\tilde{w}_{ij}$ is the normalization of $\hat{w}_{ij}$, which represents the term that contains e_{ij} in the sum of $\hat{w}$. $\operatorname{sg}[\cdot]$ means the stop-gradient operator as the parameter is updated by copying the encoder. We discard the edges with low weights by a ratio of the original edges ($\delta\%$ times the number of edges $|\mathcal{E}|$).

Furthermore, to counteract that GGSL weakens the amount of information during the message passing, we introduce the higher-order neighbors to enhance the encoder. The k-th order structure can be formally defined as follows:

$$\hat{\mathbf{A}} = \mathbf{A}^1 + \mathbf{A}^2 + \cdots + \mathbf{A}^k. \tag{15}$$

The higher-order structure is incorporated into the model, thereby replacing the original graph structure. It is important to note that, unlike other existing methods that also incorporate higher-order structure or multiple hops (Abu-El-Haija et al. 2019; Klicpera, Bojchevski, and Günnemann 2019), the input of the encoder in GGSL maintains a similar sparsity to the original graph, which is controlled by a threshold parameter δ. The incorporation of the higher-order structure in GGSL aims to enhance the acquisition of a more precise adjacency matrix by expanding the pool of potential neighboring candidates. This aspect is particularly beneficial as the range of negative samples in contrastive learning encompasses the entire dataset.

4.2 Gradient-weighted InfoNCE

In addition to adapting GNN to GCL, it is necessary to appropriately adjust the penalty of negative samples based on the characteristics of GNNs. This adjustment aims to reduce the occurrence of false negative samples. Therefore, we propose a gradient-weighted InfoNCE (GW-NCE) that incorporates the gradient weight w as a guiding factor for re-weighting the positive and negative samples within the original InfoNCE loss function.

Gradient Selection. In contrast to GGSL, the gradients are utilized in the computation of the loss function. We thus propose a mechanism for selecting gradients in order to obtain the variable w. The procedure for gradient estimation in GGSL is similar, with the exception that the graph encoder $\mathrm{GNN}(\mathbf{A}, \mathbf{X}; \mathbf{\Theta})$ is employed as the representation learner to obtain node embeddings $\boldsymbol{u}/\boldsymbol{v}$ and w.

Gradient-weighted InfoNCE. As stated previously, the conflict within the set $\{w(v_k, +), w(u_i, -), w(v_i, -)\}$ arises from the approach used by the GNN. This also suggests the potential for a negative sample to be incorrectly classified as such (for instance, the likelihood of this occurring increases with higher values of w). Thus, one can reduce the weight of a negative sample with larger w. Furthermore, there are two additional weights, denoted as $\{w(v_k), w(v_i)\}$, where $w(v_k) = w(v_k, -) + w(v_k, +)$ and $w(v_i) = w(v_i, -) + w(v_i, +)$. The two weights illustrate the comparative scale of the conflict, which should correspond to the extent of the loss's impact. Specifically, we propose to use a decreasing function to project w into a weight within InfoNCE:

$$\alpha_{ij} = \frac{1}{n_{ij}} \sum \frac{1}{1 + \exp(\mathrm{sg}[\dot{w}_{ij}])}, \tag{16}$$

where $\dot{w}_{ij}$ including the corresponding term in weights $\{w(v_k, +), w(u_i, -), w(v_i, -), -w(v_k), -w(v_i)\}$. Finally, the GW-NCE is formed as follows:

$$\mathcal{L}_i = -\log \frac{f_{ii}^{+}}{\alpha_{ii} f_{ii}^{+} + \sum_{k \neq i} \alpha_{ik} f_{ik}^{\text{inter}} + \sum_{k \neq i} \alpha_{ik} f_{ik}^{\text{intra}}}, \tag{17}$$

where $f_{ii}^{+} = f(\boldsymbol{u}_i, \boldsymbol{v}_i)$, $f_{ik}^{\text{inter}} = f(\boldsymbol{u}_i, \boldsymbol{v}_k)$, and $f_{ik}^{\text{intra}} = f(\boldsymbol{u}_i, \boldsymbol{u}_k)$. The detailed algorithm and complexity analysis of ReGCL can be found in Appendix B.

5 Experiments

In this section, we verify the effectiveness of the proposed ReGCL[1] by comparing the SOTA methods in graph learning.

5.1 Experimental Settings

Datasets. We evaluate the proposed ReGCL on five node classification datasets, including the citation networks, co-purchase networks, and co-authorship networks. Cora and Citeseer are citation networks that are widely used as node classification benchmarks (Kipf and Welling 2017), Amazon Photo is the Amazon co-purchase network (Shchur et al. 2018), and Coauthor CS includes the co-authorships of the academic graph (Shchur et al. 2018).

Baselines. We compare ReGCL with representative graph learning methods: (1) semi-supervised GNNs: GCN (Kipf and Welling 2017) and GAT (Veličković et al. 2018), (2) unsupervised graph representation learning: DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and GAE (Kipf and Welling 2016), and (3) self-supervised graph contrastive learning: DGI (Velickovic et al. 2019), GRACE (Zhu et al. 2020), MVGRL (Hassani and Khasahmadi 2020), BGRL (Thakoor et al. 2021), GCA (Zhu et al. 2021), CCA-SSG (Zhang et al. 2021), GRADE (Wang et al. 2022).

Evaluation Protocol. We adhere to the commonly employed evaluation procedure (Velickovic et al. 2019; Zhang et al. 2021). We initially train the model using all nodes without labels and subsequently proceed to train an additional classifier using the fixed node embeddings. For the baseline results, we use the public-reported results if their experimental setting is the same as ours. Otherwise, we reproduce them with the authors' code. Please refer to Appendix C for more details on the dataset split and hyperparameter settings.

5.2 Comparison with State-of-the-art Methods

We report the experimental results in Table 1. It is observed that ReGCL demonstrates superior performance compared to the state-of-the-art baselines, including both supervised and unsupervised methods. Specifically, ReGCL achieves an enhancement of 5.1% on average. Compared to GRACE, which can be regarded as an ablation version of ReGCL, the observed improvements amount to a 2.9% increase, thereby indicating the efficacy of the proposed gradient-guided structure learning (GGSL) and gradient-weighted InfoNCE (GW-NCE). Furthermore, in comparison to more powerful data/model augmentation (*e.g.*, GRADE), our proposed ReGCL demonstrates its efficacy. The results consistently indicate that the implementation of well-designed conflict mitigation mechanisms for the encoder leads to a higher level of performance. This suggests that careful attention to such mechanisms is crucial for achieving effective conflict mitigation. Compared to the baselines that employ alternative loss functions instead of InfoNCE (*e.g.*, CCA-SSG), our proposed GW-InfoNCE demonstrates greater competitiveness. Please consult the ablation study in the subsequent section for a more comprehensive analysis.

[1] https://github.com/RingBDStack/ReGCL

Methods	Input	Cora	Citeseer	Pubmed	Photo	CS
Supervised GCN (Kipf and Welling 2017)	**X**,**A**,**Y**	82.5±0.4	71.2±0.3	79.2±0.3	92.4±0.2	93.0±0.3
Supervised GAT (Veličković et al. 2018)	**X**,**A**,**Y**	83.0±0.7	72.5±0.7	79.0±0.3	92.6±0.4	92.3±0.2
Raw Features (Velickovic et al. 2019)	**X**	47.9±0.4	49.3±0.2	69.1±0.3	78.5±0.0	90.4±0.0
DeepWalk (Perozzi, Al-Rfou, and Skiena 2014)	**A**	70.7±0.6	51.4±0.5	74.3±0.9	89.4±0.1	84.6±0.2
GAE (Kipf and Welling 2016)	**X**,**A**	71.5±0.4	65.8±0.4	72.1±0.5	91.6±0.1	90.0±0.7
DGI (Velickovic et al. 2019)	**X**,**A**	82.3±0.6	71.8±0.7	76.8±0.6	91.6±0.2	92.2±0.6
GRACE (Zhu et al. 2020)	**X**,**A**	81.9±0.4	71.2±0.5	80.6±0.4	92.2±0.2	92.9±0.0
MVGRL (Hassani and Khasahmadi 2020)	**X**,**A**	83.5±0.4	73.3±0.5	80.1±0.7	91.7±0.1	92.1±0.1
BGRL (Thakoor et al. 2021)	**X**,**A**	81.7±0.5	72.1±0.5	80.2±0.4	92.6±0.3	93.0±0.2
GCA (Zhu et al. 2021)	**X**,**A**	83.4±0.3	72.3±0.1	80.2±0.4	92.5±0.2	93.1±0.0
CCA-SSG (Zhang et al. 2021)	**X**,**A**	84.2±0.4	73.1±0.3	81.6±0.4	**93.1±0.1**	93.3±0.2
GRADE (Wang et al. 2022)	**X**,**A**	83.3±0.5	68.2±0.6	81.5±0.5	92.6±0.3	93.2±0.3
ReGCL (Ours)	**X**,**A**	**84.8±0.1**	**74.3±0.3**	**83.9±0.3**	92.6±0.3	**93.7±0.3**

Table 1: Test accuracy (%±standard deviation) of node classification task. (bold: best results, underlined: runner-ups.)

5.3 Ablation Study

To further investigate the effectiveness of each component of the proposed ReGCL, we conduct the ablation study with the following ReGCL variants:

1. ReGCL *w/o* GGSL: we discard the gradient-guided structure learning and directly input the graph $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ into the augmentation and encoder to obtain the node embeddings while preserving the GW-NCE.
2. ReGCL *w/o* GW-NCE: we replace the gradient-weighted InfoNCE as a normal InfoNCE as Eq. (2). The GGSL is still used before the augmentation.
3. ReGCL *w/o* Both: we ablate both the two main components of ReGCL (*i.e.*, GGSL and GW-NCE), which is the same as the architecture of GRACE model.
4. ReGCL *w/o* Higher-Order: for a finer ablation, we perform the gradient-guided structure learning without any higher-order structure enhancement (*i.e.*, $k = 1$).
5. ReGCL *w/o* Re-wiring: to explore whether the validity should be attributed to the learned edges or their weights, we fix the edges as the original graphs.

The results are in Figure 4 with the following observations.

Effect of GGSL. After excluding the proposed gradient-guided structure learning module (*i.e.*, ReGCL *w/o* GGSL), the performance experiences a decrease of up to 1.1%. The observed decrease in performance demonstrates the impact of GGSL which serves to alleviate the conflict in GNN for graph contrastive learning. Furthermore, when comparing the outcomes presented in Table 1 with those derived from GRACE, it is evident that there is still an observed improvement of 1.4% in the performance of ReGCL without the use of GW-NCE. The only difference between this approach and GRACE lies in the encoding (*i.e.*, whether to utilize GGSL). The observed effectiveness of the proposed gradient-guided structure learning mechanism is noteworthy.

Effect of GW-NCE. It is evident that, in the absence of the GW-NCE, there is a decrease in performance by up to 1.7%. The results indicate that the proposed gradient-weighted approach effectively mitigates the conflict within the original InfoNCE loss function. Specifically, when compared to the GRACE (ReGCL *w/o* Both), ReGCL still achieves an 1.0% improvements when removing GGSL. The aforementioned two models exhibit variation solely in the design of the loss function, thereby demonstrating the superiority of the proposed gradient-guided InfoNCE. Additionally, the ReGCL *w/o* GGSL also outperforms CCA-SSG in the majority of cases. Therefore, the implementation of a well-designed objective aimed at alleviating conflicts can enhance the effectiveness of contrastive learning on graphs.

In addition to conducting ablations on the main components of ReGCL, we also explore the effectiveness within the GGSL for a more comprehensive analysis.

Effect of Higher-Order Structure Learning. We further assess the efficacy of higher-order structure enhancement by maintaining the order at $k = 1$. ReGCL with higher-order neighbors results in a 2.0% improvement compared to the absence of higher-order neighbors. The aforementioned statement illustrates that the higher-order structure enhancement proves beneficial for GGSL by mitigating the impact of first-order neighbors. Please refer to the subsequent section for further elaboration. It has been observed that the removal of higher-order structure learning in the Photo and CS datasets leads to a significant decrease in effectiveness, even more so than removing both components. This highlights the importance of higher-order structure learning.

Effect of CL-adapted matrix. The proposed GGSL algorithm has the capability to simultaneously learn new edges and re-weight edges. To determine the effectiveness of GGSL, we keep the structure of the graphs constant and only learn the edge weights (*i.e.*, ReGCL *w/o* Re-wiring). The result findings indicate that the learning of new edges contributes to a 2.7% average improvement in GGSL. Similar to higher-order structure learning, re-wiring plays a significant role in both Photo and CS datasets.

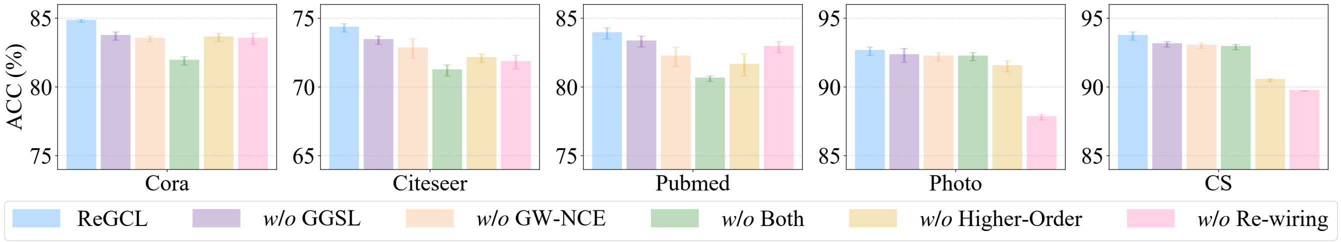


Figure 4: Ablation study of ReGCL.

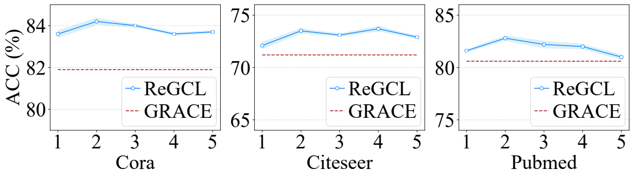


Figure 5: Sensitivity of the number of orders k.

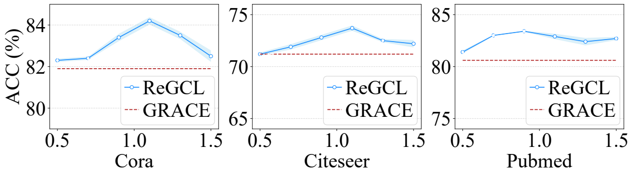


Figure 6: Sensitivity of the threshold δ.

5.4 Hyperparameter Sensitivity

In the ReGCL framework, two crucial hyperparameters: the order of neighbors k which determines the extent of message-passing, and the threshold δ decides the number of edges in the learned structure. Thus, the effects of the two hyperparameters are shown in Figure 5-6.

Number of Orders k. From Figure 5, we find that appropriately increasing the order helps the performance of ReGCL while the first-order and overly higher-order will damage the effectiveness of the model. Specifically, an average 1.7% improvement is observed when increasing the value of k beyond the first order. However, given a larger k, the accuracy of the model declines which suggests that the number of orders should not be too large due to the over-smoothing issue of GNNs. Note that, in contrast to higher-order or multi-hop models, the sparsity of the structure input into the encoder after GGSL remains similar to that of the original graph. This suggests that the effectiveness of GGSL can be attributed to the learned new edges and their weights, which are based on the higher-order neighbors, rather than an increase in the number of edges. Therefore, it is crucial to select an optimal higher-order number for the dataset.

Threshold δ. The parameter δ denotes the ratio between the number of edges in the graph generated by GGSL and the number of edges in the original graph. We vary the parameter δ within the range of 0.5 to 1.5, representing a variation of 50% to 150% of the edges. We have made the following observations based on the experimental results in Figure 6. Firstly, a structure that includes a larger number of edges is advantageous for GCL, which is beneficial for most datasets. The observed phenomena indicate that the performance is enhanced when there are more CL-adapted edges. Secondly, excessively increasing the number of edges in GCL is not beneficial, as it will lead to a higher frequency of conflicts.

6 Related Work

Inspired by the powerful self-supervised learning ability in CV (Chen et al. 2020; Fang et al. 2023a,b) and NLP (Oord, Li, and Vinyals 2018; Fang et al. 2022), there are multiple studies on graph contrastive learning (GCL) (Wu et al. 2021b; Liu et al. 2022b; Ji et al. 2023b; Liang et al. 2023). DGI (Velickovic et al. 2019) maximizes the mutual information between local and global representations. MVGRL (Hassani and Khasahmadi 2020) uses graph diffusion as a means to generate two distinct views of graphs utilized for contrastive learning. GRACE (Zhu et al. 2020) proposes the use of the InfoNCE loss function (Oord, Li, and Vinyals 2018) on graphs. Inspired by the aforementioned works, there exist several studies that center on node-level GCL, such as BGRL (Thakoor et al. 2021), CCA-SSG (Zhang et al. 2021), and GRADE (Wang et al. 2022). Different from the node-level GCL, GraphCL (You et al. 2020) focuses on graph-level tasks. The above GCL methods primarily use graph neural networks (GNNs) (Kipf and Welling 2017; Veličković et al. 2018) as encoders. Recently, there has been an increase in efforts to identify the underlying issues through augmentation mechanisms (Yu et al. 2022; Zhang et al. 2023), negative sampling techniques (Xia et al. 2022). However, GCL still faces the conflict issue proposed in this paper.

7 Conclusion

We present ReGCL, a graph contrastive learning framework to mitigate the conflict issue between GNN and GCL. Theoretically, an analysis is performed on gradients to identify the specific locations and mechanisms of conflict occurrence. Leveraging the theoretical findings, we design two gradient-based strategies. Gradient-guided structure learning enables the acquisition of a graph structure adapted to CL, thereby mitigating conflicts within the GNN. Gradient-weighted InfoNCE mitigates the occurrence of false negatives in the context of GNN by integrating the coefficients derived from the gradients. ReGCL achieves the SOTA results.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and suggestions. The corresponding author is Jianxin Li. The authors of this paper were supported by the NSFC through grant No.62225202.

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Steeg, G. V.; and Galstyan, A. 2019. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 21–29.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 1597–1607.

Chuang, C.-Y.; Robinson, J.; Lin, Y.-C.; Torralba, A.; and Jegelka, S. 2020. Debiased contrastive learning. *Advances in Neural Information Processing Systems*, 33: 8765–8775.

Fang, X.; Liu, D.; Zhou, P.; and Hu, Y. 2022. Multi-modal cross-domain alignment network for video moment retrieval. *IEEE Transactions on Multimedia*.

Fang, X.; Liu, D.; Zhou, P.; and Nan, G. 2023a. You Can Ground Earlier than See: An Effective and Efficient Pipeline for Temporal Sentence Grounding in Compressed Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2448–2460.

Fang, X.; Liu, D.; Zhou, P.; Xu, Z.; and Li, R. 2023b. Hierarchical local-global transformer for temporal sentence grounding. *IEEE Transactions on Multimedia*.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, 4116–4126.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Computer Vision and Pattern Recognition*, 9729–9738.

Ji, C.; Li, J.; Peng, H.; Wu, J.; Fu, X.; Sun, Q.; and Yu, P. S. 2023a. Unbiased and Efficient Self-Supervised Incremental Contrastive Learning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 922–930.

Ji, C.; Zhao, T.; Sun, Q.; Fu, X.; and Li, J. 2023b. Higher-Order Memory Guided Temporal Random Walk for Dynamic Heterogeneous Network Embedding. *Pattern Recognition*, 109766.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*.

Liang, K.; Liu, Y.; Zhou, S.; Tu, W.; Wen, Y.; Yang, X.; Dong, X.; and Liu, X. 2023. Knowledge Graph Contrastive Learning Based on Relation-Symmetrical Structure. *IEEE Transactions on Knowledge and Data Engineering*, 1–12.

Liang, K.; Meng, L.; Liu, M.; Liu, Y.; Tu, W.; Wang, S.; Zhou, S.; Liu, X.; and Sun, F. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767*.

Liu, N.; Wang, X.; Bo, D.; Shi, C.; and Pei, J. 2022a. Revisiting graph contrastive learning from the perspective of graph spectrum. *Advances in Neural Information Processing Systems*, 35: 2972–2983.

Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; and Tang, J. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 35(1): 857–876.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Philip, S. Y. 2022b. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6): 5879–5900.

Logeswaran, L.; and Lee, H. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv:1807.03748*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Thakoor, S.; Tallec, C.; Azar, M. G.; Munos, R.; Veličković, P.; and Valko, M. 2021. Bootstrapped representation learning on graphs. In *International Conference on Learning Representations 2021 Workshop on Geometrical and Topological Representation Learning*.

Tong, Z.; Liang, Y.; Ding, H.; Dai, Y.; Li, X.; and Wang, C. 2021. Directed graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 19580–19593.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *International Conference on Learning Representations*.

Wang, F.; and Liu, H. 2021. Understanding the Behaviour of Contrastive Loss. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2495–2504.

Wang, R.; Wang, X.; Shi, C.; and Song, L. 2022. Uncovering the Structural Fairness in Graph Contrastive Learning. *Advances in Neural Information Processing Systems*, 35: 32465–32473.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021a. Self-supervised Graph Learning for Recommendation. In *SIGIR Conference on Research and Development in Information Retrieval*, 726–735.

Wu, L.; Lin, H.; Tan, C.; Gao, Z.; and Li, S. Z. 2021b. Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*.

Xia, J.; Wu, L.; Wang, G.; Chen, J.; and Li, S. Z. 2022. ProGCL: Rethinking Hard Negative Mining in Graph Contrastive Learning. In *International Conference on Machine Learning*, 24332–24346.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823.

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 ACM SIGIR conference on research and development in information retrieval*, 1294–1303.

Zhang, H.; Wu, Q.; Yan, J.; Wipf, D.; and Yu, P. S. 2021. From canonical correlation analysis to self-supervised graph neural networks. *Advances in Neural Information Processing Systems*, 34: 76–89.

Zhang, Y.; Zhu, H.; Song, Z.; Koniusz, P.; and King, I. 2023. Spectral feature augmentation for graph contrastive learning and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11289–11297.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference*, 2069–2080.