Enhancing Contrastive Learning on Graphs with Node Similarity

Hongliang Chi
chih3@rpi.edu
Rensselaer Polytechnic Institute
Troy, New York, USA

Yao Ma
may13@rpi.edu
Rensselaer Polytechnic Institute
Troy, NewYork, USA

Abstract

Graph Neural Networks (GNN) have proven successful for graph-related tasks. However, many GNNs methods require labeled data, which is challenging to obtain. To tackle this, graph contrastive learning (GCL) have gained attention. GCL learns by contrasting similar nodes (positives) and dissimilar nodes (negatives). Current GCL methods, using data augmentation for positive samples and random selection for negative samples, can be sub-optimal due to limited positive samples and the possibility of false-negative samples. In this study, we propose an enhanced objective addressing these issues. We first introduce an ideal objective with all positive and no false-negative samples, then transform it probabilistically based on sampling distributions. We next model these distributions with node similarity and derive an enhanced objective. Comprehensive experiments have shown the effectiveness of the proposed enhanced objective for a broad set of GCL models[1].

CCS Concepts

• Computing methodologies → Machine learning; **• Machine learning → Machine learning approaches**; **Neural networks**.

Keywords

Graph Neural Networks, Self-Supervised Learning, Graph Contrastive Learning

ACM Reference Format:
Hongliang Chi and Yao Ma. 2024. Enhancing Contrastive Learning on Graphs with Node Similarity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3671898

1 Introduction

Graphs are regarded as a type of essential data structure to represent many real-world data, such as social networks [9, 11] and transportation networks [34] etc. Graph neural networks (GNNs) [21, 37, 39], which generalize deep neural networks to graphs, have demonstrated their great power in graph representation learning, thus facilitating many graph-related tasks from various fields including recommendations [8, 43], drug discovery [25, 31], and computer vision [10, 30]. Most GNN models are trained in a supervised setting, which receives guidance from labeled data. However, in real-world applications, labeled data are often difficult to obtain while unlabeled data are abundantly available [18]. Hence, to promote GNNs' adoption in broader real-world applications, it is of great significance to develop graph representation learning techniques that do not require labels. More recently, contrastive learning techniques [5, 13, 15], which are able to effectively leverage the unlabeled data, have been introduced for learning node representations with no labels available [41].

Graph contrastive learning (GCL) aims to map nodes into an embedding space where nodes with similar semantic meanings are embedded closely together while those with different semantic meanings are pushed far apart. More specifically, to achieve this goal, each node in the graph is treated as an anchor node. Then, nodes with similar semantics to this anchor are identified as positive samples while those with different semantic meanings are regarded as negative samples. The commonly used objective for GCL, based on InfoNCE, treats a single node as the positive sample, typically generated through data augmentation that alters the original graph, while negative samples are uniformly selected from the graph. The goal is to bring the positive sample closer to the anchor node and distance the negative samples from the anchor. However, this objective has two main shortcomings: (1) The set of negative samples often includes nodes that are semantically similar to the anchor (false-negative samples). Minimizing the contrastive objective can undesirably push these nodes away, negatively affecting the quality of the embeddings. Hence, removing false-negative samples has the potential to improve the performance of contrastive learning, which is also demonstrated in [6]; (2) The contrastive objective includes only a single positive sample derived from data augmentation, limiting its ability to group similar nodes effectively. Preferably, including more positive samples in the numerator benefits the contrastive learning process, which is verified in [20].

An ideal contrastive objective would include all positive samples and exclude any false-negative samples (see details in Section 3.1). However, this is unattainable without ground truth labels. In our study, we introduce an enhanced objective that approximates the ideal objective. In particular, we first transfer the ideal objective into a probabilistic form by modeling the anchor-aware distributions for sampling positive and negative samples. Intuitively, nodes with higher semantic similarity to the anchor node should have a higher probability to be selected as positive samples. Hence, we estimate these anchor-aware distributions by theoretically relating them with node similarity. Measuring node similarity is challenging since it involves both graph structure and node features, which interact with each other in a complicated way. Correspondingly, we propose a novel strategy to model the pairwise node similarity

[1] Code is available at https://github.com/frankhlchi/SimEnhancedGCL

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671898

by effectively utilizing both graph structure and feature information. With these estimated distributions, the probabilistic objective is then empirically estimated with samples, which makes the enhanced objective applicable. Our key contributions are summarised as follows:

- **Introduction of an Ideal GCL Objective**: We introduce an ideal contrastive objective for GCL that effectively incorporates all positive samples and eliminates false negatives.
- **Derivation of an Enhanced GCL Objective**: We probabilistically approximate the ideal objective, resulting in an enhanced objective that requires fewer samples for practical estimation. This enhancement is achieved through rigorous asymptotic analysis and by leveraging both graph and features information to accurately model anchor-aware distributions.
- **Comprehensive Experimental Validation**: Extensive experiments validate our enhanced objective's effectiveness and confirm the importance of our enhanced objective's two key components, positive & negative weights, and also the necessity of dual graph & feature information used in modelling anchor-aware distributions.
- **Application Beyond GCL**: We extend our methodology to enhance the neighborhood contrastive loss in the semi-supervised `Graph-MLP` model, further validating the effectiveness of our approach.

2 Preliminary

This section introduces basic notations and key concepts foundational to our discussions on GCL. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote a graph with $\mathcal{V}$ and $\mathcal{E}$ denoting its set of nodes and edges, respectively. The edges describe the connections between the nodes, which can be summarized in an adjacency matrix $\mathbf{A} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{V}|}$ with $|\mathcal{V}|$ denoting the number of nodes. The i, j-th element of the adjacency matrix is denoted as $\mathbf{A}[i, j]$. It equals 1 only when the nodes i and j connect to each other, otherwise 0. Each node $i \in \mathcal{V}$ is associated with a feature vector $\mathbf{x}_i$.

GCL aims to learn high-quality node representations by contrasting semantically similar and dissimilar node pairs. More specifically, given an anchor node $v \in \mathcal{V}$, those nodes with similar semantics as v are considered positive samples, while those with dissimilar semantics are treated as negative samples. The goal of GCL is to pull the representations of those semantically similar nodes close and push the semantically dissimilar ones apart. From the perspective of a single anchor node v, the goal can be achieved by minimizing the following objective $\mathcal{L}(v)$.

$$-\log \frac{e^{f(v)^\top f(v')/\tau}}{e^{f(v)^\top f(v')/\tau} + \sum_{v_s \in \mathcal{N}(v)} e^{f(v)^\top f(v_s)/\tau}}, \qquad (1)$$

where τ is the temperature hyper-parameter, v' is the positive sample, which is typically generated by data augmentation, $f(\cdot)$ is a function that maps a node v to its low-dimensional representation, and $\mathcal{N}(v)$ denotes the negative samples corresponding to the anchor v. The overall objective for all nodes is a summation of $\mathcal{L}(v)$ over all nodes in $\mathcal{V}$. Next, we briefly introduce the positive sample, the $f(\cdot)$ function, and the set of negative samples as follows. To create positive samples, graph augmentations like topology and feature transformations are employed. For example, `GRACE` [50] uses edge removal and feature masking for augmentation. Advanced methods, such as `GCA` [51], focus on adaptive augmentations prioritizing less important features and edges.

Many GCL frameworks utilize two augmented graphs of the original graph $\mathcal{G}$ as two views. Nodes from these views serve as anchor nodes, with the corresponding node in the other view as its positive sample. The negative sample set consists of nodes from both views.

3 Methodology

The objective in Eq.(1), despite its widespread use and strong performance, has inherent limitations. It only employs one positive sample for each anchor node, potentially restricting the quality of the learned representations. The uniform negative sampling can also introduce "false-negative" samples, undermining representation quality. To address these shortcomings, we introduce an enhanced objective. We first describe an ideal objective in Section 3.1, which incorporates more positive samples and eliminates "false-negative" samples. We then detail a strategy for practically estimating this ideal objective, emphasizing the need for modeling anchor-aware distributions for both positive and negative sampling. Using pairwise node similarity, we effectively model these distributions, as detailed in Section 3.3. Our proposed enhanced objective is discussed further in Section 3.4.

3.1 The Ideal Objective for GCL

To address the limitations of the conventional objective in Eq. (1), an ideal objective that enjoys the capability of learning high-quality representations would include all positive nodes in the numerator while only including true negative samples in the denominator. More specifically, such an ideal objective $\mathcal{L}_{ideal}(v)$ for an anchor node v could be formulated as follows.

$$-\log \frac{\sum_{v_j \in \mathcal{V}} \mathbb{1}_{[y=y_j]} e^{f(v)^\top f(v_j)/\tau}}{e^{f(v)^\top f(v')/\tau} + \sum_{v_j \in \mathcal{V}} \mathbb{1}_{[y\neq y_j]} e^{f(v)^\top f(v_j)/\tau}}, \qquad (2)$$

where y denotes the ground truth label for node v and y_j is the label of v_j, and $\mathbb{1}_{[a]}$ is an indicator function, which outputs 1 if and only if the argument a holds true, otherwise 0.

Nevertheless, the objective function in Eq. (2) is not achievable, as it is impossible to know the semantic classes of the downstream tasks in the contrastive training process, let alone the ground-truth labels. Hence, to make the objective more practical, in this paper, following the assumptions in [2, 6, 32], we assume there are a set of discrete latent classes C standing for the true semantics of each node. We use $h : \mathcal{V} \to C$ to denote the function mapping a given node to its latent class. For a node $v \in \mathcal{V}$, $h(v)$ denotes its latent class. Then, we introduce two types of anchor-aware sampling distributions over the entire node set $\mathcal{V}$. Specifically, for an anchor node v, we denote the probability of observing any node u sharing the same latent class (i.e., u is a positive sample corresponding to v) as $p_v^+(u) = p(u \mid h(u) = h(v))$. Similarly, $p_v^-(u) = p(u \mid h(u) \neq h(v))$ denotes the probability of observing u as a negative sample corresponding to v. Note that the subscript in p_v^+ and p_v^- indicates that they are specific to the anchor node v. With these distributions, we estimate the objective in Eq. (2) with positive and negative nodes sampled from the two distributions.

Specifically, we estimate the objective $\mathcal{L}_{est}(v)$ as follows.

$$\mathbb{E}_{\substack{\{v_j^+\}_{j=1}^m \sim p_v^+, \\ \{v_k^-\}_{k=1}^n \sim p_v^-}} \left[-\log \frac{\frac{l}{m}\sum_{j=1}^m e^{f(v)^\top f\left(v_j^+\right)/\tau}}{e^{f(v)^\top f(v')/\tau} + \frac{q}{n}\sum_k^n e^{f(v)^\top f\left(v_k^-\right)/\tau}} \right], \qquad (3)$$

where $\{v_j^+\}_{j=1}^m$ and $\{v_k^-\}_{k=1}^n$ denote the set of “positive nodes” and “negative nodes” sampled following p_v^+ and p_v^-, respectively; and m and n denotes the number positive and negative samples, respectively. As similar to [6], for the purpose of asymptotic analysis, we introduce two weight parameters l and q. When m and n are finite, we set $l = n$ and $q = m$, which ensures that Eq. (3) follows the same form as Eq. (2).

Though Eq. (3) is more practical than Eq. (2), its applicability is hindered by two main challenges:

- **Lack of Access to Anchor-Aware Distributions**: We do not have access to the two anchor-aware distributions p_v^+ and p_v^-
- **High Sampling Complexity for Accurate Estimation**: Even if we were to know these distributions, accurately estimating the expectation in the enhanced objective would require a significant number of samples.

To effectively address the identified challenges in estimating the ideal objective our method incorporates the following strategies:

- **Dual Information for Modeling Anchor-Aware Distributions**: To address the first issue, we propose leveraging both graph structure and feature information. Since directly modeling the distribution over all nodes in the graph is extremely difficult, we propose to connect the probabilities $p_v^+(u)$ and $p_v^-(u)$ of a specific node u with node similarity between node u and the anchor node v. This similarity is modeled using dual graph and feature information. More details on modeling anchor-aware distributions will be discussed in Section 3.3.
- **From Asymptotic Analysis to Practical Estimation**: For the second challenge, we adopt an asymptotic analysis, which leads to a new objective requiring fewer samples for estimation. The specifics of this analysis and the new objective are discussed in Section 3.2. Next, we first discuss how we address the second challenge assuming we are given the two sets of anchor-aware distributions p_v^+ and p_v^- in Section 3.2 and then discuss how we model the anchor-ware distributions p_v^+ and p_v^- in Section 3.3. The enhanced objective will be discussed in Section 3.4.

3.2 Efficient Estimation of the Objective

To allow a more efficient estimation of Eq. (3), we consider its asymptotic form by analyzing the case where m and n go to infinity, which is summarized in the following theorem.

Theorem 3.1. *For fixed l and q, as $m, n \longrightarrow \infty$, it holds that:*

$$\mathbb{E}_{\substack{\{v_j^+\}_{j=1}^m \sim p_v^+ \\ \{v_k^-\}_{k=1}^n \sim p_v^-}} \left[-\log \frac{\frac{l}{m}\sum_{j=1}^m e^{f(v)^\top f(v_j^+)/\tau}}{e^{f(v)^\top f(v')/\tau} + \frac{q}{n}\sum_{k=1}^n e^{f(v)^\top f(v_k^-)/\tau}} \right]$$

$$\longrightarrow -\log \frac{l\mathbb{E}_{v^+ \sim p_v^+(v^+)}\left[e^{f(v)^\top f(v^+)/\tau}\right]}{e^{f(v)^\top f(v')/\tau} + q\mathbb{E}_{v^- \sim p_v^-(v^-)}\left[e^{f(v)^\top f(v^-)/\tau}\right]}. \qquad (4)$$

Proof. As τ is a nonzero scalar, the contrastive objective is bounded. Thus, we could apply the Dominated Convergence Theorem to prove the theorem above as follows:

$$\lim_{m\to\infty} \lim_{n\to\infty} \mathbb{E}\left[-\log \frac{\frac{l}{m}\sum_{j=1}^m e^{f(v)^\top f(v_j^+)/\tau}}{e^{f(v)^\top f(v')/\tau} + \frac{q}{n}\sum_{k=1}^n e^{f(v)^\top f\left(v_k^-\right)/\tau}} \right]$$

$$= \mathbb{E}\left[\lim_{m\to\infty} \lim_{n\to\infty} -\log \frac{\frac{l}{m}\sum_{j=1}^m e^{f(v)^\top f(v_j^+)/\tau}}{e^{f(v)^\top f(v')/\tau} + \frac{q}{n}\sum_{k=1}^n e^{f(v)^\top f\left(v_k^-\right)/\tau}} \right]$$

$$= \mathbb{E}\left[-\log \frac{l\mathbb{E}_{v^+ \sim p_v^+(v^+)}\left[e^{f(v)^\top f(v^+)/\tau}\right]}{e^{f(v)^\top f(v')/\tau} + q\mathbb{E}_{v^- \sim p_v^-(v^-)}\left[e^{f(v)^\top f(v^-)/\tau}\right]} \right]$$

$$= -\log \frac{l\mathbb{E}_{v^+ \sim p_v^+(v^+)}\left[e^{f(v)^\top f(v^+)/\tau}\right]}{e^{f(v)^\top f(v')/\tau} + q\mathbb{E}_{v^- \sim p_v^-(v^-)}\left[e^{f(v)^\top f(v^-)/\tau}\right]}.$$

□

As demonstrated in Theorem 1, the objective of Eq. (4) is an asymptotic form of Eq. (3). In this work, we aim to empirically estimate Eq. (4) instead of Eq. (3). Specifically, Eq. (4) contains two expectations to be estimated. Compared to Eq. (3), the sampling complexity is significantly reduced, as we disentangled the joint distribution in Eq. (3), and only need to estimate these two expectations independently. More specifically, to estimate $\mathbb{E}_{v^+ \sim p_v^+(v^+)}\left[e^{f(v)^\top f(v^+)/\tau}\right]$, a straightforward way is to randomly draw samples from p_v^+ and calculate its empirical mean. However, it is typically inefficient and inconvenient to obtain samples directly from p_v^+, since p_v^+ itself needs to be estimated (this will be discussed in Section 3.3) and we cannot obtain a simple analytical form to perform the sampling. The same reason applies to the estimation of $\mathbb{E}_{v^- \sim p_v^-(v^-)}\left[e^{f(v)^\top f(v^-)/\tau}\right]$. Therefore, in this work, we adopt the importance sampling strategy [12] to estimate the two expectations using samples from the uniform distribution p as follows.

$$\begin{aligned} \mathbb{E}_{v^+ \sim p_v^+}\left[e^{f(v)^\top f(v^+)/\tau}\right] &= \mathbb{E}_{v^+ \sim p}\left[\frac{p_v^+(v^+)}{p(v^+)} e^{f(v)^\top f(v^+)/\tau}\right] \\ &\approx \frac{1}{M}\sum_{v_j \in \mathcal{V}_M}\left[\frac{p_v^+(v_j)}{p(v_j)} e^{f(v)^\top f(v_j)/\tau}\right]; \qquad (5) \end{aligned}$$

$$\begin{aligned} \mathbb{E}_{v^- \sim p_v^-}\left[e^{f(v)^\top f(v^-)/\tau}\right] &= \mathbb{E}_{v^- \sim p}\left[\frac{p_v^-(v^-)}{p(v^-)} e^{f(v)^\top f(v^-)/\tau}\right] \\ &\approx \frac{1}{N}\sum_{v_j \in \mathcal{V}_N}\left[\frac{p_v^-(v_j)}{p(v_j)} e^{f(v)^\top f(v_j)/\tau}\right], \qquad (6) \end{aligned}$$

where $\mathcal{V}_M = \{v_j\}_{j=1}^M \sim p$ contains M nodes sampled from p and $\mathcal{V}_N = \{v_j\}_{j=1}^N \sim p$ contains N nodes sampled from p, which are utilized for estimation. To obtain the final empirical form of Eq. (4), the two sets of anchor-aware distributions p_v^+ and p_v^- remain to be estimated, which is discussed in the next section.

3.3 Modeling and Estimating Anchor-Aware Distributions

Here, we discuss the modeling details of the anchor-aware distributions p_v^+ and p_v^-. As discussed earlier in Section 3.1, for an anchor v, the positive sample distribution is a conditional distribution relying on the agreement of the latent classes of v and any other sample u, which can be formulated as $p_v^+(u) = p_v(u|h(v) = h(u))$. Direct modeling this distribution is impossible, since we do not have access to the latent semantic class. Here, we propose to model $p_v^+(u)$ with the node similarity between the anchor node v and a given sample u (Section 3.3 and Section 3.3.2). We then discuss the process to evaluate node similarity with both graph structure and node feature information in Section 3.3.3.

3.3.1 Modeling Anchor-Aware Distributions with Node Similarity. Based on Bayes' theorem, we have

$$p_v^+(u) \propto p_v(h(v) = h(u)|u)p(u), \tag{7}$$

where p is a uniform distribution over all nodes, and $p_v(h(v) = h(u))$ is the probability that u shares the same latent semantic class of v. Therefore, to obtain $p_v^+(u)$, it is essential to model $p_v(h(v) = h(u)|u)$ as p is already known. Intuitively, if v and u are more "similar" to each other, they are more likely to share the same semantic class. Assuming that we are given a function $\text{sim}(\cdot,\cdot)$ that measures the pair-wise similarity of any two nodes, then we further assume that the probability $p_v(h(v) = h(u)|u)$ is positively correlated with $\text{sim}(v, u)$, which can be formulated as

$$p_v(h(v) = h(u)|u) \propto \mathcal{T}(\text{sim}(v, u)), \tag{8}$$

where $\mathcal{T}$ is a monotonic increasing transformation. We will discuss the details of the transformation and the similarity function in Section 3.3.2 and Section 3.3.3, respectively. Together with Eq. (7), we have

$$p_v^+(u) \propto \mathcal{T}(\text{sim}(v, u))p(u), \tag{9}$$

which intuitively expresses that those samples that are more similar to v are more likely to be sampled as positive samples. We then formulate the probability $p_v^+(u)$ with $\text{sim}(v, u)$ as follows.

$$p_v^+(u) = \frac{\mathcal{T}(\text{sim}(v, u))p(u)}{\int \mathcal{T}(\text{sim}(v, v^s))p(v^s)dv^s} = \frac{\mathcal{T}(\text{sim}(v, u))p(u)}{\mathbb{E}_{v^s \sim p}[\mathcal{T}(\text{sim}(v, v^s))]}. \tag{10}$$

Note that, in practice, $\mathbb{E}_{v^s \sim p}[\mathcal{T}(\text{sim}(v, v^s))]$ can be empirically estimated using the set of samples $\mathcal{V}_M$ in Eq. (5) as follows.

$$\mathbb{E}_{v^s \sim p}[\mathcal{T}(\text{sim}(v, v^s))] \approx \frac{1}{M} \sum_{v_j \in \mathcal{V}_M} \mathcal{T}(\text{sim}(v, v_j)). \tag{11}$$

Then, we can estimate $p^+(u)$ as follows

$$\hat{p}_v^+(u) = \frac{\mathcal{T}(\text{sim}(v, u))p(u)}{\frac{1}{M} \sum_{v_j \in \mathcal{V}_M} \mathcal{T}(\text{sim}(v, v_j))}, \tag{12}$$

where $\hat{p}_v^+(u)$ is the empirical estimate of $p_v^+(u)$. Intuitively, given $\hat{p}_v^+(u)$, we can directly estimate $\hat{p}_v^-(u)$ as $1 - \hat{p}_v^+(u)$. However, this is typically not optimal for the purpose of contrastive learning for several reasons: 1) first, the samples in $\mathcal{V}_M$(in Eq. (5)) and $\mathcal{V}_N$ (in Eq. (6)) are likely different, which makes it infeasible to directly model p_v^- using $1 - p_v^+$ for all selected nodes; 2) second, we prefer different properties of the estimations for the two distributions p_v^+ and p_v^- for the purpose of contrastive learning. Specifically, we prefer a relatively conservative estimation for p_v^+ to reduce the impact of "false positives" (i.e, avoid assigning high p_v^+ for real negative samples). In contrast, a more aggressive estimation of p_v^- is acceptable. Modeling a conservative p_v^+ and aggressive p_v^- at the same time cannot be achieved if we constrain $p_v^+(u) + p_v^-(u) = 1$. Due to the above reasons, in this work, we relax this constraint and model p_v^- flexibly using node similarity $\text{sim}(\cdot,\cdot)$ as follows.

$$p_v^-(u) = \frac{\mathcal{D}(\text{sim}(v, u))p(u)}{\mathbb{E}_{v^s \sim p}[\mathcal{D}(\text{sim}(v, v^s))]}, \tag{13}$$

where $\mathcal{D}$ is a monotonic decreasing function, indicating that p_v^- is negatively correlated with the similarity. Similar to Eq. (12), $p_v^-(u)$ can be empirically estimated with N samples in $\mathcal{V}_N$ (described in Eq. (6)) as follows.

$$\hat{p}_v^-(u) = \frac{\mathcal{D}(\text{sim}(v, u))p(u)}{\frac{1}{N} \sum_{v_j \in \mathcal{N}} \mathcal{D}(\text{sim}(v, v_j))}. \tag{14}$$

Next, we first discuss the details of the monotonic increasing transformation function $\mathcal{T}$ and the monotonic decreasing transformation $\mathcal{D}$ in Section 3.3.2. We then discuss the similarity function $\text{sim}(v, u)$ in Section 3.3.3.

3.3.2 Transformations. To flexibly adjust the two estimated anchor-aware distributions $\hat{p}_v^+$ and $\hat{p}_v^-$ between conservative estimation to aggressive estimation, we utilize exponential function with temperature [1] to model the transformation functions as follows

$$\mathcal{T}(\text{sim}(v_i, v_j)) = \exp(\text{sim}(v_i, v_j)/\tau_p) - 1; \tag{15}$$

$$\mathcal{D}(\text{sim}(v_i, v_j)) = \exp(-\text{sim}(v_i, v_j)/\tau_n), \tag{16}$$

where τ_p and τ_n are two temperature parameters. We could adjust the estimation of the two distributions $\hat{p}_v^+$ in Eq. (12) and $\hat{p}_v^-$ in Eq. (14) by varying τ_p and τ_n, respectively. More specifically, for $\hat{p}_v^+$, we could make the distribution more conservative by decreasing τ_p, which increases the probability mass for those samples with high similarity. In the extreme case, when τ_p goes to 0, the probability mass concentrates in the sample with the largest similarity. On the other hand, when τ_p reaches infinity, $\hat{p}_v^+$ converges to a distribution proportional to similarity. Note that without the "−1" in Eq. (15), $\hat{p}_v^+$ converges to a uniform distribution as τ_p goes to infinity, which leads to model collapse as all samples in Eq. (5) will be treated equally (all treated as positive samples). Thus, we include "−1" in Eq. (15) to avoid such cases. Similarly, $\hat{p}_v^-$ can be adjusted from a uniform distribution to a distribution with mass concentrated on the sample with the smallest similarity by varying τ_n. Specifically, when τ_n goes to 0, the estimated $\hat{p}_v^-$ converges to the uniform distribution and the estimation in Eq. (6) reduces to the same result as the convectional negative sampling strategy.

3.3.3 Modeling Node Similarity. Here, we delve into modeling node similarity, considering both graph structure and node features. We first outline methods for capturing each type of similarity and then detail their integration for a comprehensive similarity function.

Graph Structure Similarity Personalized Page Rank (PPR) is a widely adopted tool for measuring the relevance between nodes in graph mining [24, 28, 29]. More recently, it has also been adopted to improve graph representation learning [22, 23]. In this work, we utilize the PPR score to model the structural node similarity. Specifically, the personalized PageRank matrix is defined as $\mathbf{P} = \alpha(\mathbf{I} - (1-\alpha)\hat{\mathbf{A}})^{-1}$, where $\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, $\mathbf{D}$ is the degree matrix, and $\alpha \in (0,1)$ is a hyper-parameter. The i,j-th element of the PPR matrix $\mathbf{P}$ denoting as $\mathbf{P}[i,j]$ measures the structural similarity between node v_i and node v_j. However, calculating the matrix $\mathbf{P}$ is computationally expensive, especially for large-scale graphs, as it involves a matrix inverse. In this work, we adopt the iterative approximation of the PPR matrix for measuring the node similarity as $\hat{\mathbf{P}} = (1-\alpha)^K \hat{\mathbf{A}}^K + \sum\limits_{k=0}^{K-1} \alpha(1-\alpha)^k \hat{\mathbf{A}}^k$, where K is the number of iterations. Note that, $\hat{\mathbf{P}}$ converges to $\mathbf{P}$ as K goes infinity [22].

Feature Similarity. To better mine the pairwise node similarity from features, we adopt the classic cosine similarity. Specifically, feature similarity between nodes v_i and v_j is evaluated by $\text{sim}_F(v_i, v_j) = \cos(\mathbf{x}_i, \mathbf{x}_j)$, where $\mathbf{x}_i, \mathbf{x}_j$ are the original input features of node v_i and v_j, respectively.

Fusing Graph and Feature Similarity. Given the structure similarity $\text{sim}_G(v_i, v_j)$ and feature similarity $\text{sim}_F(v_i, v_j)$, it is vital to define an adaptive function $\text{sim}(\cdot,\cdot)$ to fuse them and output a combined similarity score capturing information from both sources. Specifically, we propose to combine the two similarities to form the overall similarity as $\text{sim}(v_i, v_j) = \beta \cdot \text{sim}_F(v_i, v_j) \cdot \gamma + (1-\beta) \cdot \text{sim}_G(v_i, v_j)$, where γ is the scaling factor to control the relative scale between the two similarity scores, and β is a hyper-parameter balancing the two types of similarity. In general, γ could also be treated as a hyper-parameter. In this work, we fix $\gamma = \sum sim_G(v_i, v_j) / \sum sim_F(v_i, v_j)$ such that the two types of similarity are at the same scale.

3.4 The Proposed Enhanced Objective

With the estimation of $\hat{p}_v^+$ in Eq. (12) and $\hat{p}_v^-$ in Eq. (14), we propose an enhanced objective $\mathcal{L}_{EN}(v)$ as follows.

$$-\log \frac{\sum\limits_{v_j \in \mathcal{V}_M} \left[w_v^+(v_j) e^{f(v)^\top f(v_j)/\tau} \right]}{e^{f(v)^\top f(v')/\tau} + \sum\limits_{v_j \in \mathcal{V}_N} \left[w_v^-(v_j) e^{f(v)^\top f(v_j)/\tau} \right]}, \tag{17}$$

where $w_v^+(v_j)$ and $w_v^-(v_j)$ are defined as follows.

$$\begin{aligned} w_v^+(v_j) &= \frac{\mathcal{T}(\text{sim}(v, v^j))}{\frac{1}{M} \sum\limits_{v_j \in \mathcal{V}_M} \mathcal{T}(\text{sim}(v, v_k))}; \\ w_v^-(v_j) &= \frac{\mathcal{D}(\text{sim}(v, v_j))}{\frac{1}{N} \sum\limits_{v_j \in \mathcal{V}_N} \mathcal{D}(\text{sim}(v, v_k))}. \end{aligned} \tag{18}$$

$\mathcal{V}_M$ and $\mathcal{V}_N$ are the two sets of nodes introduced in Eq. (5) and Eq. (6). If we set $\mathcal{V}_M = \mathcal{V}_N = \mathcal{V}$, we can make a direct comparison between the enhanced objective in Eq. (17) and the ideal objective in Eq. (2). Specifically, the enhanced objective can be considered as a soft version of the ideal objective, where the weights $w_v^+(v_j)$ and $w_v^-(v_j)$ in Eq. (17) reflect the likelihood of v_j being a positive sample or a negative sample, respectively.

4 Beyond Graph Contrastive Learning

The philosophy of contrastive learning has inspired other frameworks for graph representation learning. In `Graph-MLP` [16], an auxiliary neighborhood contrastive loss is proposed to enhance the performance of MLP on the semi-supervised node classification task. As indicated by its name, the key idea of the neighborhood contrastive loss is to treat the "neighboring nodes" as positive samples and contrast them with their corresponding anchor nodes. Since the neighbors are defined through graph structure, such a loss helps incorporate the graph information into the representation learning process of MLP. It has been shown that the MLP model equipped with the neighborhood contrastive loss is capable of achieving performance comparable to or even stronger than graph neural network models. In this section, we briefly describe the `Graph-MLP` model with neighborhood contrastive loss and discuss how the proposed techniques discussed in Section 3 can be utilized to further enhance this loss.

4.1 Neighborhood Contrastive Loss and `Graph-MLP`

For an anchor node $v_i \in \mathcal{V}$, the neighborhood contrastive loss is defined as follows:

$$\mathcal{L}_{NC}(v_i) = -\log \frac{\sum\limits_{v_j \in \mathcal{V}_b} \mathbb{1}_{[v_j \neq v_i]} \hat{\mathbf{A}}^r[i,j] e^{f(v_i)^\top f(v_j)/\tau}}{\sum\limits_{v_k \in \mathcal{V}_b} \mathbb{1}_{[v_k \neq v_i]} e^{f(v_i)^\top f(v_k)/\tau}}, \tag{19}$$

where $\mathcal{V}_b$ is a set of nodes uniformly sampled from $\mathcal{V}$, $\hat{\mathbf{A}}^r$ is the r-th power of the normalized adjacency matrix $\hat{\mathbf{A}}$. The i,j-th element $\hat{\mathbf{A}}^r[i,j]$ is only non-zero when node v_j is within the r-hop neighborhood of node v_i, otherwise $\hat{\mathbf{A}}^r[i,j] = 0$. Hence, in the numerator of Eq. (19), only the r-hop neighbors are treated as positive samples. The denominator is similar to that in contrastive learning. Overall, the neighborhood contrastive loss for all nodes in the graph can be formulated as follows.

$$\mathcal{L}_{NC} = \sum_{i \in \mathcal{V}} \mathcal{L}_{NC}(v_i). \tag{20}$$

In `Graph-MLP`, the neighborhood contrastive loss is combined with the cross-entropy loss for conventional semi-supervised node classification as $\mathcal{L}_{train} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{NC}$, where $\alpha > 0$ is a hyper-parameter that balances the cross-entropy loss $\mathcal{L}_{CE}$ and the neighborhood contrastive loss $\mathcal{L}_{NC}$. When the graph is large, the neighborhood contrastive loss can be calculated in a batch-wise way, where $\mathcal{L}_{NC}$ can be calculated over a batch of nodes as $\mathcal{L}_{NC} =$

$\sum_{v_i \in \mathcal{B}} \mathcal{L}_{NC}(v_i)$ with $\mathcal{B}$ denoting a specific sampled batch. Correspondingly, in this scenario, $\mathcal{V}_b$ in Eq. (19) can be replaced by $\mathcal{B}$.

4.2 Enhanced Objective for Graph-MLP

Following the same philosophy as in Section 3, we propose the following enhanced neighborhood contrastive loss.

$$\mathcal{L}_{EN-NC}(v_i) = -\log \frac{\sum_{v_j \in \mathcal{V}_b} \mathbb{1}_{[v_j \neq v_i]} w^+_{v_i}(v_j) e^{f(v_i)^\top f(v_j)/\tau}}{\sum_{v_k \in \mathcal{V}_b} \mathbb{1}_{[v_k \neq v_i]} w^-_{v_i}(v_k) e^{f(v_i)^\top f(v_k)/\tau}}, \quad (21)$$

where $w^+_{v_i}(v_j)$ and $w^-_{v_i}(v_k)$ are the positive weight between nodes v_i, v_j and negative weight between nodes v_i, v_k as defined in Eq. (18). We can replace $\mathcal{L}_{NC}(v_i)$ in Eq. (20) with $\mathcal{L}_{EN-NC}(v_i)$ to form an enhanced training framework for MLP models. We name such a framework as Graph-MLP+. Its superiority is empirically verified in the experiments section (Section 5.3).

5 Experiment

In this section, we conduct experiments to verify the effectiveness of the enhanced objectives. We also perform an ablation study to provide a deep understanding of the proposed objectives. Specifically, we first introduce the datasets we adopt for experiments in Section 5.1. Then, we present the significant enhanced results for GRACE, GCA, and Graph-MLP with discussions in Section 5.2 and Section 5.3, respectively. The ablation study is presented in Section 5.4.

5.1 Datasets

Here, we introduce the datasets we adopt for the experiments. Following previous papers [50, 51, 51], we adopt 8 datasets including Cora [33], Citeseer [33], Pubmed [33], DBLP [44], A-Photo [35], A-Computers [35], Co-CS [35], and Wiki-CS [26] for evaluating the performance. The details of the dataset statistics are shown in the Table 1.

Table 1: Summary of Datasets.

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
DBLP	17,716	105,734	1,639	4
A-Computers	13,752	245,861	767	10
A-Photo	7,650	119,081	745	8
Wiki-CS	11,701	216,123	300	10
Co-CS	18,333	81,894	6,805	15

5.2 Performance of Enhanced GCL models

The enhanced objective proposed in Eq. (17) is quite flexible and can be utilized to improve the performance of various frameworks that adopt the conventional graph contrastive learning objective. In this work, we adopt GRACE [50], a recently proposed representative GCL framework and its updated version, GCA [51], as base models (check Section 2 for a brief description of GRACE and GCA). We denote the GRACE framework with the enhanced objective as GRACE+ and use GCA+ to represent the enhanced GCA. Next, we first present results for GCA and then describe results for GCA.

5.2.1 GRACE. Following [50], we conduct the experiment with GRACE and GRACE+ on the first 4 citation datasets as introduced in Section 5.1. To evaluate the effectiveness of GRACE+, we adopt the same linear evaluation scheme as in [38, 50].

Here, the experiments are conducted in two stages. In the first stage, we learn node representations with the graph contrastive learning frameworks (GRACE and GRACE+) in a self-supervised fashion. Then, in the second stage, we evaluate the quality of the learned node representations through the node classification task. Specifically, a logistic regression model with the obtained node representations as input is trained and tested. To comprehensively evaluate the quality of the representations, we adopt different training-validation-test splits. Specifically, we first randomly split the node sets into three parts: 80% for testing, 10% for validation, and the rest of 10% is utilized to further build the training sets. With the remaining 10% of nodes, we build 5 different training sets that consist of 2%, 4%, 6%, 8%, 10% of nodes in the entire graph. The training set is randomly sampled from the 10% of data for building the training subset. In each setting, following the official published code of [50], the logistic regression model is trained for 3 runs with different random initializations. Furthermore, we repeat the entire experiment including both stages for 30 times and report the average performance of 90 runs with standard deviation.

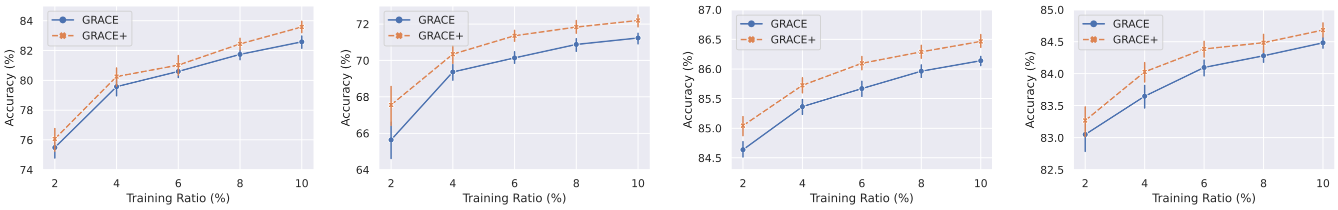
The results of GRACE and GRACE+ are summarized in Figure 1. From these figures, it is clear that GRACE+ consistently outperforms GRACE on all datasets under various training ratios. These results validate the effectiveness of the proposed enhanced objective.

Table 2: Node classification results (%) of GCA (original model), ProGCL (advanced baseline method) and GCA+.

Model	A-Computers	A-Photo	Wiki-CS	Co-CS
GCA	87.49±0.39	92.03±0.39	76.46±1.30	92.73±0.21
ProGCL	87.58±0.55	92.36±0.33	76.40±0.73	**93.00±0.21**
GCA+	**88.15±0.40**	**92.52±0.45**	**78.64±0.18**	92.82±0.29

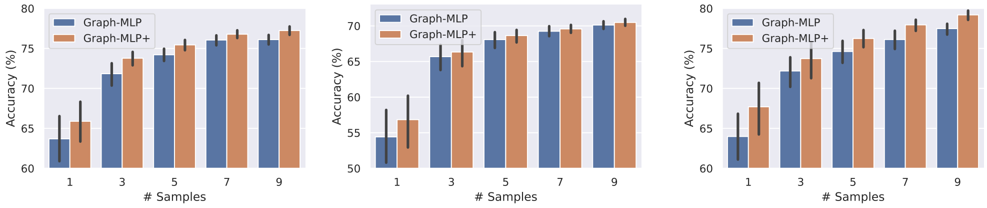
5.2.2 GCA. Following [51], we evaluate GCA and GCA+ on 4 datasets including A-Photo, A-Computers, Co-CS, and Wiki-CS. In addition, we also compare GCA+ with ProGCL [42], which is a recent solid GCL baseline aiming to enhance the loss of GCA by addressing the issue of "false negative" samples. We adopt the hyper-parameters provided in [42] for running ProGCL.

In our experimental setup, we randomly split the nodes into three parts: 80% for testing, 10% for validation, and 10% for training. We train the logistic regression classifier for 20 runs. Furthermore, we repeat the experiment including both stages for 10 times with different random seeds. We report the average performance of the 200 runs together with the standard deviation in the performance table in the main body of the paper. For GCA, we adopt the GCA-DE variant since it achieves the best performance overall among the three variants proposed in [51]. The results of GCA are produced using the official code and exact parameter settings provided in [51].



(a) Cora (b) Citeseer (c) Pubmed (d) DBLP

Figure 1: Node classification results of GRACE and GRACE+ with various training ratios.



(a) Cora (b) Citeseer (c) Pubmed

Figure 2: Node classification results of Graph-MLP and Graph-MLP+ with limited training samples.

For ProGCL, we adopt the ProGCL-reweight variant. The results of ProGCL are produced using the official code and exact parameter settings provided in [42]. The results of GCA and ProGCL are different from those reported in [42, 51] as the experiment settings are different. In particular, we repeat the entire process of experiments for 10 times resulting a total of 200 runs, while, in [51], the entire process was executed once with 20 runs.

The performances of GCA, GCA+, and ProGCL are reported in Table 2. As demonstrated in the Table, GCA+ surpasses GCA on all datasets, which further illustrates the effectiveness of the proposed enhanced objective. Also, it suggests that the enhanced objective is general and can be utilized to advance various GCL methods. Furthermore, GCA+ outperforms the advanced ProGCL on 3 out of 4 datasets, which further illustrates the effectiveness of the proposed enhanced objective.

5.3 Graph-MLP

Here, we investigate how the enhanced objective helps improve the performance of Graph-MLP by comparing its performance with Graph-MLP+. A brief introduction of Graph-MLP and Graph-MLP+ can be found in Section 4.

Table 3: Node classification results of Graph-MLP and Graph-MLP+.

Model	Cora	Citeseer	Pubmed
Graph-MLP	79.7± 1.15	72.99±0.54	79.62±0.67
Graph-MLP+	**80.51±0.69**	**74.03±0.51**	**81.42±0.92**

Following [16], we adopt three datasets including Cora, Citeseer, and Pubmed for comparing Graph-MLP+ with Graph-MLP. As described in Section 4, Graph-MLP runs in a semi-supervised setting. The classification model is trained in an end-to-end way. We adopt the conventional public splits of the datasets [21] to perform the experiments. The experiments are repeated for 30 times with different random initialized parameters, and the average performance is reported in Table 3. As shown in Table 3, Graph-MLP+ outperforms Graph-MLP by a large margin on all three datasets. Graph-MLP+ even outperforms message-passing methods such as GCN by a large margin, especially on Citeseer and Pubmed, which indicates that the proposed objective can effectively incorporate the graph structure information and improve the performance of MLPs. Note that, in the Graph-MLP+ framework, only the MLP model is utilized for inference after training, which is more efficient than message-passing methods in terms of both time and complexity.

We further compare Graph-MLP+ with Graph-MLP under the setting where labeled nodes are extremely limited. Specifically, we keep the test and validation set fixed, and only use s samples per class for training, where s is set to 1, 3, 5, 7, and 9. When creating these various training sets, the s samples per class are sampled from the training set in the public split setting. The results are presented in Figure 2. Graph-MLP+ achieves stronger performance than Graph-MLP under all settings over all three datasets. Furthermore, Graph-MLP+ performs extremely well when labels are limited. This indicates that the proposed enhanced objective can more effectively utilize the graph structure and feature information, which leads to high-quality node representations even when labels are scarce.

5.4 Ablation Study

We assess the impact of key elements in our enhanced objective through ablation studies. We first evaluate the contributions of the positive weights $w_v^+(v_j)$ and negative weights $w_v^-(v_j)$. Then, we analyze the influence of both graph and feature similarities on the model's performance. We only conduct ablation studies based on GRACE and Graph-MLP as GCA is a variant of GRACE. For GRACE, we adopt the 10%/10%/80% training, validation, and testing split. For Graph-MLP, we use the conventional public splits.

5.4.1 Positive and Negative Weights in the Enhanced Objective. In Section 3.4, two weights $w_v^+(v_j)$ and $w_v^-(v_j)$ are introduced to increase the variety of positive samples and alleviate the effect of false negative samples, respectively. Here, we aim to investigate how these two kinds of weights contribute to the model performance. For this investigation, we introduce two variants of the proposed objective that only incorporate the positive weights or negative weights. The results for GRACE+ and Graph-MLP+ and their corresponding variants are summarized in Table 4 and Table 5, respectively. Specifically, in Table 4, we denote the variant of GRACE+ with only $w_v^+(v_j)$ as GRACE+(P) and the one with only $w_v^-(v_j)$ is denoted as GRACE+(N). Likewise, the two variants for Graph-MLP+ are denoted as Graph-MLP+(P) and Graph-MLP+(N) in Table 5. In Table 4, both GRACE+(P) and GRACE+(N) consistently outperform GRACE on all datasets. Similarly, in Table 5, both Graph-MLP+(P) and Graph-MLP+(N) consistently outperform Graph-MLP on all datasets. These results clearly illustrate that both positive weights and negative weights are important for improving the enhanced objectives. Furthermore, these two types of weights contribute to the objectives in a complementary way since Graph-MLP+ outperforms all variants on most datasets.

Table 4: Node classification results (%) of GRACE+ and its variants (↑: increase ≤ 0.5%; ↑↑: increase > 0.5%).

	Cora	Citeseer	Pubmed	DBLP
GRACE	82.56±1.21	71.23±0.86	86.12±0.23	84.43±0.25
GRACE+(P)	83.59±0.98 ↑↑	71.53±0.97 ↑	86.41±0.29 ↑	84.76±0.23 ↑
GRACE+(N)	83.20±1.26 ↑↑	72.19±0.78 ↑↑	86.31±0.22 ↑	84.89±0.26 ↑
GRACE+(G)	83.19±1.31 ↑↑	71.40±1.03 ↑	86.34±0.23 ↑	84.55±0.26 ↑
GRACE+(F)	82.71±1.29 ↑	71.90±0.87 ↑↑	86.11±0.26	84.65±0.26 ↑
GRACE+	**83.62±1.13** ↑↑	**72.26±0.82** ↑↑	**86.45±0.29** ↑	**84.77±0.27** ↑

Table 5: Node classification results (%) of Graph-MLP+ and its variants (↑: increase ≤ 0.5%; ↑↑: increase > 0.5%).

	Cora	Citeseer	Pubmed
Graph-MLP	79.7 ± 1.15	72.99 ± 0.54	79.62 ± 0.67
Graph-MLP+(P)	80.33 ± 0.96 ↑↑	73.69 ± 0.45 ↑↑	79.79 ± 0.89 ↑
Graph-MLP+(N)	79.80 ± 0.86 ↑	73.41 ± 0.44 ↑	79.76 ± 0.82 ↑
Graph-MLP+(G)	80.32 ± 0.73 ↑↑	73.4 ± 0.36 ↑↑	79.96 ± 0.94 ↑
Graph-MLP+(F)	61.80 ± 0.61	64.04 ± 0.73	75.92 ± 0.90
Graph-MLP+	**80.51±0.69** ↑↑	**74.03±0.51** ↑↑	**81.42±0.92** ↑↑

5.4.2 Similarity Measure. We examine the impact of graph and feature similarities detailed in Section 3.3.3. Two enhanced objective variants are introduced: one using only graph similarity and the other, only feature similarity. Results for GRACE+, Graph-MLP+ and their variants are presented in Tables 4 and 5. In these tables, GRACE+(G) represents the GRACE+ variant using only graph similarity, and GRACE+(F), the one using only feature similarity. For Graph-MLP+, the variants are denoted as Graph-MLP+(G) and Graph-MLP+(F). Our observations are as follows:

- In Table 4, both GRACE+(G) and GRACE+(F) outperforms GRACE on most datasets, which demonstrates that both graph and feature similarity contain important information about node similarity and they can be utilized for effectively modeling the anchor-aware distributions. GRACE+ outperforms the two variants and the base model GRACE on all datasets, which indicates that the graph similarity and feature similarity are complementary to each other, and properly combining them results in better similarity estimation leading to strong performance.
- In Table 5, Graph-MLP+(G) significantly outperforms Graph-MLP while Graph-MLP+(F) does not perform well. This is potentially due to the lack of graph information in MLP models. Different from GRACE+(F) which incorporates graph information in the encoder, Graph-MLP+(F) only utilizes feature information, which leads to inferior performance. On the other hand, the strong performance of Graph-MLP+ suggests that the enhanced objective effectively incorporates the graph structure information. However, this does not mean the feature similarity is not important. Graph-MLP+ outperforms Graph-MLP+(G) on all three datasets, which suggests that the feature similarity brings additional information than graph similarity, and properly combining them is important.

6 Related Work

Contrastive Learning. Contrastive learning (CL) aims to learn latent representations by discriminating positive from negative samples. The instance discrimination loss is introduced in [41] without data augmentation. In [3], it is proposed to generate multiple views by data augmentation and learn representations by maximizing mutual information. Momentum Contrast (MoCo) [15] maintains a memory bank of negative samples, which significantly increases the number of negatives used in the contrastive loss. In [5], it is discovered that the composition of data augmentations plays a critical role in CL. BYOL and Barlow Twins [45] [13] achieves strong CL performance without using negative samples. Recently, a series of tricks such as debiased negative sampling [6], and hard negative mining [19, 32, 40], positive mining [7] have been proved effective. Efforts have also been made to extend CL to supervised setting [20].
Graph Contrastive Learning. Deep Graph Infomax (DGI) [38] takes a local-global comparison mode by maximizing the mutual information between patch representations and high-level summaries of graphs. MVGRL [14] contrasts multiple structural views of graphs generated by graph diffusion. GRACE [50] utilizes edge removing and feature masking to generate two views for node-level contrastive learning. Based upon GRACE, GCA [51] adopts adaptive augmentations by considering the topological and semantic aspects of graphs. Several works investigate the bias in the negative sampling [42, 49]. MERIT [17] leverages Siamese GNNs to learn high-quality node representations. Most of these contrastive learning frameworks utilize negative samples in their training. BGRL [36]

and Graph Barlow Twins [4] are GCL methods without requiring negative samples. CCA-SSG [46] leverages Canonical Correlation Analysis to optimize correlation between two views. COSTA [47] tackles biases inherent in graph augmentation through the implementation of feature augmentation. SUGRL [27] is designed to enhance differences between classes while minimizing differences within the same class. Recently, SFA [48] leverages spectral feature augmentation as its augmentation scheme.

7 Conclusion

In this paper, we propose an effective enhanced contrastive objective to approximate the ideal contrastive objective for graph contrastive learning. The proposed objective leverages node similarity to model the anchor-aware distributions for sampling positive and negative samples. Also, the objective is designed to be flexible and general, which could be adopted for any graph contrastive learning framework that utilizes the traditional InfoNCE-based objective. Furthermore, the proposed enhancing philosophy generally applies to other contrasting-based models such as `Graph-MLP` which includes an auxiliary contrastive loss. Extensive experiments have demonstrated the significant improvement on various GCL models with the application of the enhanced objectives.

8 ACKNOWLEDGEMENT

This research is supported by the National Science Foundation (NSF) under grant numbers NSF-2406647 and NSF-2406648.

References

[1] J. S. Rowlinson *. 2005. The Maxwell–Boltzmann distribution. *Molecular Physics* (2005). https://doi.org/10.1080/002068970500044749

[2] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. 2019. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229* (2019).

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *NeurIPS* (2019).

[4] Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V Chawla. 2022. Graph barlow twins: A self-supervised representation learning framework for graphs. *Knowledge-Based Systems* 256 (2022), 109631.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

[6] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *NeurIPS* 33 (2020).

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2021. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*.

[8] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*.

[9] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep social collaborative filtering. In *Proceedings of the 13th ACM RecSys*.

[10] Victor Garcia and Joan Bruna. 2017. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043* (2017).

[11] Dmitri Goldenberg. 2021. Social network analysis: From graph theory to applications with python. *arXiv preprint arXiv:2102.10014* (2021).

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS* 33 (2020).

[14] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *ICML*.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

[16] Yang Hu, Haoxuan You, Zhecan Wang, Zhicheng Wang, Erjin Zhou, and Yue Gao. 2021. Graph-MLP: Node Classification without Message Passing in Graph. https://doi.org/10.48550/ARXIV.2106.04051

[17] Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. 2021. Multi-scale contrastive siamese networks for self-supervised graph representation learning. *arXiv preprint arXiv:2105.05682* (2021).

[18] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. 2020. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141* (2020).

[19] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *NeurIPS* 33 (2020), 21798–21809.

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *NeurIPS* 33 (2020).

[21] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[22] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).

[23] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion improves graph learning. *arXiv preprint arXiv:1911.05485* (2019).

[24] Andre Lamurias, Pedro Ruas, and Francisco M Couto. 2019. PPR-SSM: personalized PageRank and semantic similarity measures for entity linking. *BMC bioinformatics* 20, 1 (2019).

[25] Zhen Li, Mingjian Jiang, Shuang Wang, and Shugang Zhang. 2022. Deep learning methods for molecular representation and property prediction. *Drug Discovery Today* 27, 12 (2022), 103373.

[26] Péter Mernyei and Cătălina Cangea. 2020. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. *arXiv preprint arXiv:2007.02901* (2020).

[27] Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. 2022. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7797–7805.

[28] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford.

[29] Sungchan Park, Wonseok Lee, Byeongseo Choe, and Sang-Goo Lee. 2019. A survey on personalized PageRank computation algorithms. *Access* (2019).

[30] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. 2019. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 142–147.

[31] Prakash Chandra Rathi, R Frederick Ludlow, and Marcel L Verdonk. 2019. Practical high-quality electrostatic potential surfaces for drug discovery using a graph-convolutional deep neural network. *Journal of medicinal chemistry* (2019).

[32] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

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

[34] Behrooz Shahsavari and Pieter Abbeel. 2015. Short-term traffic forecasting: Modeling and learning spatio-temporal relations in transportation networks using graph neural networks. *University of California at Berkeley, Technical Report No. UCB/EECS-2015-243* (2015).

[35] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

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

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[38] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. *ICLR (Poster)* 2, 3 (2019), 4.

[39] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *ICML*. PMLR.

[40] Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. 2020. Conditional negative sampling for contrastive learning of visual representations. *arXiv preprint arXiv:2010.02037* (2020).

[41] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978* (2018).

[42] Jun Xia, Lirong Wu, Ge Wang, and Stan Z. Li. 2022. ProGCL: Rethinking Hard Negative Mining in Graph Contrastive Learning. In *International conference on machine learning*. PMLR.

[43] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. 2020. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International CIKM*.

[44] Jaewon Yang and Jure Leskovec. 2015. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems* (2015).

[45] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*. PMLR.

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

[47] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. 2022. COSTA: covariance-preserving feature augmentation for graph contrastive learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2524–2534.

[48] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. 2023. Spectral feature augmentation for graph contrastive learning and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 11289–11297.

[49] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph debiased contrastive learning with joint representation clustering. In *Proc. IJCAI*. 3434–3440.

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

[51] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*.