B^2-Sampling: Fusing Balanced and Biased Sampling for Graph Contrastive Learning

Mengyue Liu
liumengyue@stu.xjtu.edu.cn
Xi'an Jiaotong University

Yun Lin*
lin_yun@sjtu.edu.cn
Shanghai Jiao Tong University

Jun Liu†
liukeen@xjtu.edu.cn
Xi'an Jiaotong University

Bohao Liu
bhaoliu@163.com
Xi'an Jiaotong University

Qinghua Zheng
qhzheng@mail.xjtu.edu.cn
Xi'an Jiaotong University

Jin Song Dong
dcsdjs@nus.edu.sg
National University of Singapore

ABSTRACT

Graph contrastive learning (GCL), aiming for an embedding space where semantically similar nodes are closer, has been widely applied in graph-structured data. Researchers have proposed many approaches to define positive and negative pairs (i.e., semantically similar and dissimilar pairs) on the graph, serving as labels to learn their embedding distances. Despite the effectiveness, those approaches usually suffer from two typical learning challenges. First, the number of candidate negative pairs is enormous. Thus, it is non-trivial to select representative ones to train the model in a more effective way. Second, the *heuristics* (e.g., graph views or metapath patterns) to define positive and negative pairs are sometimes less reliable, causing considerable noise for both "labelled" positive and negative pairs. In this work, we propose a novel sampling approach B^2-Sampling to address the above challenges in a unified way. On the one hand, we use *balanced* sampling to select the most representative negative pairs regarding both the topological and embedding diversities. On the other hand, we use *biased* sampling to learn and correct the labels of the most error-prone negative pairs during the training. The balanced and biased samplings can be applied iteratively for discriminating and correcting training pairs, boosting the performance of GCL models. B^2-Sampling is designed as a framework to support many known GCL models. Our extensive experiments on node classification, node clustering, and graph classification tasks show that B^2-Sampling significantly improves the performance of GCL models with acceptable runtime overhead. Our website [11] provides access to our codes and additional experiment results.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**.

*Corresponding author
†Corresponding author

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599262

KEYWORDS

graph contrastive learning, neural network, negative sampling

ACM Reference Format:
Mengyue Liu, Yun Lin, Jun Liu, Bohao Liu, Qinghua Zheng, and Jin Song Dong. 2023. B^2-Sampling: Fusing Balanced and Biased Sampling for Graph Contrastive Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599262

1 INTRODUCTION

Graph Contrastive Learning (GCL) has recently emerged as an important branch of self-supervised graph representation learning [27, 33]. GCL methods project nodes in graphs into an embedding space where semantically similar (positive) nodes are closer while semantically different (negative) nodes are farther. The learned embedding can widely facilitate many graph-based applications such as node classification [7, 21], graph classification [18], collaborative filtering [28], and community detection [23, 32].

Technically, typical GCL methods consist of contrasting heuristics and contrastive objectives design [33]. The contrasting heuristics define the positive and negative pairs to guide the contrastive training [7, 15, 35], and the contrastive objectives "pull" these positive pairs closer and "push" negative pairs farther in the embedding space. For example, graph augmentation is a representative way to generate contrasting pairs in homogeneous graphs. Based on the original graph G, its augmented graph G' can be generated by edge removing [7] or feature masking [35], etc. Given an anchor node $v \in G$, different *heuristics* define v's positive set $\mathcal{D}^+$ and negative set $\mathcal{D}^-$ ($\mathcal{D}^+, \mathcal{D}^- \subset G \cup G'$). On the other hand, some efforts focus on adjusting contrastive objectives such as Information Noise Contrastive Estimation (InfoNCE) [15, 29, 35], Jensen-Shannon Divergence (JSD) [7, 21], and Triplet Margin loss (TM) [24] to GCL.

While considerable contributions have been made to the above technical components in GCL, less attention has been paid to *GCL Sampling*, *i.e.*, *how to effectively sample positive and negative graph node pairs to further boost the performance.* Generally, GCL sampling has two problems, *i.e.*, training pair representativeness and training pair noise.

Training Pair Representativeness. Training pair representativeness indicates that among the sizeable negative pairs, some pairs are more useful than others in training the model in different training iterations [17, 24]. Thus, it is non-trivial to sample the most representative and useful pairs from the enormous negatives to

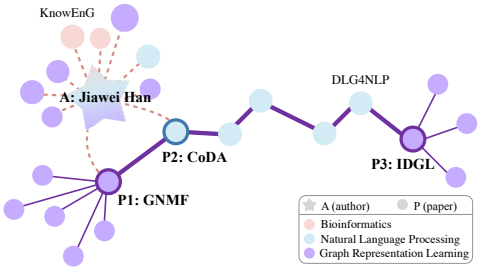


Figure 1: (1) P1 and P2 are (neighbouring) papers written by the same author Dr. *Jiawei Han*, but in different domains. (2) P1 and P3 are distant papers (i.e., their shortest path distance is 6), but in the same domain.

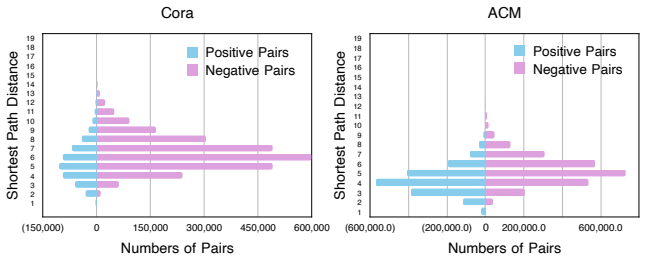


Figure 2: The numbers of node pairs with the same label (positive pairs) and different labels (negative pairs) at each shortest path distance in Cora and ACM. Distant nodes (with long shortest path distance) can have similar semantics while close nodes (with short shortest path distance) can also have different semantics.

update the model. Despite that some negative sampling techniques have been well studied in contrastive learning (CL) for visual representations [4, 8, 16], we found that borrowing those solutions has limited effectiveness to boost GCL methods (see experimental results in Section 4.2). Taking graph node pairs as the training samples, GCL sampling has its own challenge in defining the *diversity* and *representativeness* in the negative sampling, regarding both the topological properties and the used contrasting heuristics.

Training Pair Noise. Training pair noise indicates that *heuristics* of contrasting definitions for positive/negative pairs are sometimes unreliable. Some heuristics choose positive pairs via the connection strength between the neighbouring nodes [22], and some choose them by only considering the congruent nodes in augmented graphs [26, 32, 35]. However, our empirical studies show that, (1) many neighbouring nodes can have very different semantics (e.g., two connected Facebook users can just be *accidental* friends, sharing less common interests) and (2) many h-hop (h can be large) nodes can share similar semantics (e.g., a h-hop friend of a user is still in the same community or shares similar interests). Figure 1 shows a qualitative example where distant graph nodes can be semantically similar while the neighbouring nodes can be semantically different. More importantly, such cases are more prevalent than expected. Figure 2 shows a quantitative study where a considerable number of distant nodes share the same label (i.e., semantically similar) in graph-structured data. Given the complications of real-world graph semantics, any (topological) heuristics defining positive/negative pairs can only be *generally* correct but still suffer from considerable noise during the training.

In this work, we propose B^2-Sampling (Balanced and Biased Sampling) technique to address the above challenges in a unified way. We address the first challenge by sampling for *discrimination*. Specifically, we propose a *balanced* sampling technique regarding graph structures. We define topological diversity and runtime embedding diversity over the training negative pairs to choose representative pairs for model training. We address the second challenge by sampling for correction. Specifically, we propose a *biased* sampling technique regarding our observed *slow learning effect* of the noisy graph node pairs. The effect indicates that, during the training, embeddings of noisy pairs are usually harder for the model to fit than that of clean pairs, with the assumption that most pairs are clean. Based on such an observation, we design a noise-likelihood measurement based on how "smooth" the model can fit the embeddings of node pairs, and apply biased training on them to correct the potential noise. The balanced and biased sampling can be applied sequentially and interactively to further boost the performance of existing GCL models.

We apply B^2-Sampling on node-wise GCL methods such as GCA [35], GRACE [34] and HeCo [22] on eight datasets. Compared to the state-of-the-art (SOTA) CL/GCL negative sampling techniques (e.g., MoCHi [8], and ProGCL [26]), B^2-Sampling can significantly boost the performance of those GCL methods. Meanwhile, our ablation studies further confirm the effectiveness of both sampling strategies. Given that B^2-Sampling can universally be equipped to many GCL methods (e.g., node-wise and graph-wise), we have designed B^2-Sampling as a framework to be integrated with existing GCL models.

In summary, the main contributions of our work are as follows:

- **Technique:** We propose B^2-Sampling, a novel GCL-oriented sampling technique to boost the performance from the perspective of contrasting pairs diversity (by *balanced* sampling) and noise (by *biased* sampling).
- **Tool:** We build B^2-Sampling framework based on our technique to integrate with any GCL models, facilitating the practical use.
- **Experiment:** We conducted extensive experiments on GCL methods on various baseline techniques on eight datasets, evaluating the effectiveness of our B^2-Sampling in node-level and graph-level downstream tasks.

2 PROBLEM DEFINITION

Given a graph $G = \{V, E\}$ consisting of $|V| = N$ nodes and $|E| = M$ edges, we denote $\mathbf{X} \in \mathbb{R}^{N \times d} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ as the node attributes matrix, where $\mathbf{x}_i \in \mathbb{R}^d$ is the attribute information of node v_i. Ideally, we shall have a ground-truth semantic labelling function $f^*_{sd} : V \times V \rightarrow \{0, 1\}$, so that we have a ground-truth distribution P^* on all the node pairs $\mathcal{D}$, i.e., $f^*_{sd}(v_i, v_j) \sim P^*$. However, the ground-truth function $f^*_{sd}(\cdot, \cdot)$ is hardly available in practice, if not impossible. Thus, many GCL methods use pre-defined heuristic semantic labelling function $f_{sd} : V \times V \rightarrow \{0, 1\}$ to have $f_{sd}(v_i, v_j) \sim P$ (P is an estimated semantic labelling distribution).

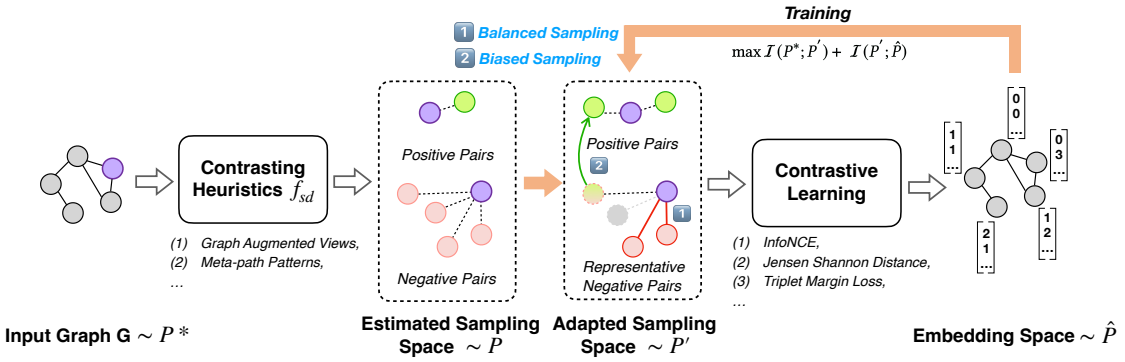


Figure 3: B^2-Sampling serves as a plugin in the overall graph contrastive learning paradigm, which adaptively re-adjusts and corrects the node pairs, regarding the graph structure and runtime node embedding, from the sampling space.

Table 1: Notation Descriptions.

Notation	Description
$G = \{V, E\}$	graph G with node set V and edge set E
$\mathbf{X}$	node attribute matrix of graph G
P^*	ground-truth labelling distribution
P	labelling distribution by pre-defined heuristics
$\hat{P}$	learned labelling distribution by GCL
P'	labelling distribution adjusted from P
f^*_{sd}	ground-truth labelling function
f_{sd}	pre-defined heuristic semantic labelling function
f'_{sd}	adjusted semantic labelling function
f_p	projection function to encode nodes
g	similarity measuring function
$\mathcal{D}^-, \mathcal{D}^+$	negative, positive pairs set defined by f_{sd}
$\mathcal{D}^{-'}, \mathcal{D}^{+'}$	negative, positive pairs set adjusted by B^2-Sampling

We denote positive set as $\mathcal{D}^+ = \{(v_i, v_j)|f_{sd}(v_i, v_j) = 1\}$, and negative set as $\mathcal{D}^- = \{(v_i, v_k)|f_{sd}(v_i, v_k) = 0\}$. Then, a projection function $f_p : V \to \mathbb{R}^n$ encodes each graph node to an n-dimensional embedding space, and the semantic similarities of node pairs are measured[1] by $g : \mathbb{R}^n \times \mathbb{R}^n \to [0, 1]$, and thus we have a learned distribution $\hat{P}$ on node embedding pairs, i.e., $g(f_p(v_i), f_p(v_j)) \sim \hat{P}$. Existing works use graph augmentation or meta-path definition to implement $f_{sd}(\cdot, \cdot)$, and use InfoNCE, JSD etc., to train $f_p(\cdot)$. For a clearer explanation, we list the important notations in Table 1.

Given a contrasting heuristics $f_{sd}(\cdot, \cdot)$, and a contrastive objective, we aim for (1) *a correction function* $f_c : V \times V \to \{0, 1\}$ to adjust the distribution P of the estimated sampling space to an adapted distribution P' (i.e., adjust $f_{sd}(\cdot, \cdot)$ to $f'_{sd}(\cdot, \cdot)$); and (2) *a sampling function* $f_s(\cdot)$ on the adapted distribution P', i.e., $f_s : \{V \times V\} \to \{V \times V\}$ to have representative negative pairs set $\mathcal{D}^{-'} \subset \mathcal{D}^-$ to train f_p. Intuitively, we correct P to P' for a distribution closer to ground truth P^* and then learn $\hat{P}$ against P' with an improved sampling strategy to achieve better performance. Thus we maximize:

$$\mathcal{I} = \underbrace{\mathcal{I}(P^*; P')}_{f_c(\cdot,\cdot)\to P'} + \underbrace{\mathcal{I}(P'; \hat{P})}_{f_s(\cdot)\to \mathcal{D}^{-'}}, \quad (1)$$

where $\mathcal{I}(\cdot;\cdot)$ represents the mutual information (MI) between two distributions, $\mathcal{D}^{-'} = f_s(\mathcal{D}^-) = \{(v_i, v_k)|f'_{sd}(v_i, v_k) = 0\}$ is the representative negatives set, and $\mathcal{D}^{+'} = f_c(\mathcal{D}^+) = \{(v_i, v_j)|f'_{sd}(v_i, v_j) = 1\}$ is the adjusted positives set.

In this work, we design biased sampling to re-adjust the distribution of the sampling space to make P' closer to P^*, so as to maximize $\mathcal{I}(P^*; P')$.[2] We sample representative contrasting pairs through balanced sampling to train the model and learn a $\hat{P}$ closer to P', so as to maximize $\mathcal{I}(P'; \hat{P})$. Balanced and biased sampling interacts with each other to achieve our research goal.

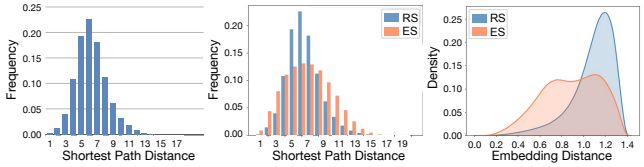
3 METHOD

Overview. Figure 3 shows how our B^2-Sampling can serve as a plugin into the general GCL paradigm. Given a graph G, different heuristics are designed to derive positive and negative node pairs. Moreover, a contrastive objective is used to measure the distance between the embedding of a pair of nodes, and compares it against the estimated "label" (i.e., positive or negative) in the sampling space. B^2-Sampling fits in-between the contrasting heuristics and contrastive object modules, and (1) selectively sample representative (negative) pairs in the space to train the model and (2) adaptively re-adjust the sampling space.

In practice, the two-phase sampling strategy can be applied repetitively. Given the distribution of existing sampling space P_i and its resulted embedding $\mathbf{Z}_i$ (i indicates the training iteration), we apply balanced sampling technique on P_i regarding $\mathbf{Z}_i$ to select representative negatives subset $\mathcal{D}^{-'}$ in the first phase. During the training, we collect the runtime learning dynamics for the training pairs to guide the correction of the sampling space, reassigning sets of positive pairs $\mathcal{D}^+$ and negative pairs $\mathcal{D}^-$. As a result, we will have P_{i+1} and $\mathbf{Z}_{i+1}$. By this means, we evolve the sampling space iteratively and boost the learning performance finally.

[1] We usually use normalized Euclidean distance and a cosine function to this end.

[2] $\mathcal{I}(P^*; P') = \mathcal{H}(P^*) - \mathcal{H}(P^*|P')$. $\mathcal{H}$ represents the entropy, and $\mathcal{I}(P^*; P')$ reaches the maximum when $P^* = P'$.



(a) Cora (b) RS v.s. SS (c) RS v.s. ES

Figure 4: (a) shows the empirical distribution of shortest path distances on Cora. (b) and (c) show effects of *shortest path distance-weighted sampling* (SS) and *embedding distance-weighted sampling* (ES), compared to random sampling (RS).

3.1 Balanced Sampling

Similar to traditional contrastive learning, GCL has the challenge of selecting representative negative pairs. We formalize the problem of balanced sampling as follows. Given a limited budget B ($B > 0$) and the negative pair set $\mathcal{D}^-$, we aim to sample a subset $\mathcal{D}^{-'}$, i.e.,

$$\mathcal{D}^{-'} = \underset{\mathcal{D}^{-'} \subset \mathcal{D}^-}{\arg\max}\, \mathcal{K}(\mathcal{D}^{-'}) \quad s.t. \quad |\mathcal{D}^{-'}| \leq B. \tag{2}$$

Here, $\mathcal{K}$ represents the information diversity over graph node pairs regarding topological diversity and embedding diversity. Appendix A.1 gives its theoretical explanation. The topological diversity measures how representative a pair of nodes resemble in the graph (input space), and the embedding diversity measures how diverse the embeddings of sampled node pairs are distributed in the embedding space (output space). To achieve the goal defined in Equation 2, we define and measure topological or embedding diversity of the training pairs, as the estimation for $\mathcal{K}(\cdot)$.

We adopt the evenness measure [1] to sample negative pairs. Intuitively, we require that the sampled negative pairs are uniformly distributed over diversified topological and embedding distances.

Topological Diversity. We use the shortest path distance between a pair of nodes as an indicator of topological diversity (Appendix A.2 gives explanations). The shortest path distance preserves a balance between computational cost and structural informativeness, compared to other structures such as loops and triangles.

Figure Figure 4(a) shows the empirical distribution over the shortest path distance of node pairs on the dataset of Cora[3]. The empirical frequency of different shortest path distances are generally imbalanced, which can hardly be mitigated by the random sampling strategy (the distribution in blue in Figure 4(b)). In contrast, our shortest path distance-weighted sampling results in a "flatten" distribution (in orange) as shown in Figure 4(b), leading us to sample a balanced number of negative pairs with more diversified shortest path distances.

We achieve balanced sampling with topological diversity as follows: Given an anchor node v_i, its negative nodes set $\mathcal{D}_i^-$, and a negative node $v_k \in \mathcal{D}_i^-$, assume that the shortest path distance $d_s(\cdot)$ between v_i and v_k equals to dis, *i.e.*, $d_s(v_i, v_k) = dis$, then the shortest path distance-weighted sampling probability p_s to select v_k is estimated by:

$$p_s(v_i, v_k) = \frac{1}{|S(v_i)|} \times \frac{1}{|N(v_i, d_s(v_i, v_k))|}, \tag{3}$$

where $S(v_i) = \{d \mid d = d_s(v_i, v_m), v_m \in \mathcal{D}_i^-\}$, and the first term $\frac{1}{|S(v_i)|}$ is a coefficient to normalize probability. $N(v_i, d_s(v_i, v_k)) = N(v_i, dis) = \{v_m \mid d_s(v_i, v_m) = dis, v_m \in \mathcal{D}_i^-\}$, and the second term $\frac{1}{|N(v_i, d_s(v_i, v_k))|}$ denotes the probability to sample a node from a set of nodes whose shortest path distance to v_i equals to dis. The shortest path distance between unconnected nodes is set to -1.

Embedding Diversity. To select the negative samples uniformly distributed over the embedding space, we estimate the distribution of high-dimensional vectors during the training and sample the pairs regarding their high-dimensional embedding diversity.

The probability distribution $p_{\text{norm}}(\cdot)$ of an Euclidean distance d_e between any two normalized n-dimensional embeddings is [6]:

$$p_{\text{norm}}(d_e) = \frac{d_e^{n-2}}{c(n)} \left[1 - \left(\frac{d_e}{2r}\right)^2\right]^{\frac{n-3}{2}}, \tag{4}$$

where the coefficient $c(n) = \sqrt{\pi} \cdot \Gamma\left(\frac{n-1}{2}\right) / \Gamma\left(\frac{n}{2}\right)$, and $\Gamma(\cdot)$ is a Gamma function defined as $\Gamma(n) = (n-1)!$ for $n \in \mathbb{N}$. Also, it is proved that $P_{\text{norm}}(d_e)$ follows a normal distribution $\mathcal{N}\left(\sqrt{2}, \frac{1}{2n}\right)$ in an n-dimensional space when $n \geq 128$ [24, 31].

It means that, given an anchor node v_i, random sampling its negative nodes is more likely to have the nodes around $\sqrt{2}$-away (in embedding space). Thus, for the anchor node v_i, we define the embedding distance-weighted sampling probability p_e for selecting its negative node v_k as:

$$p_e(v_i, v_k) = \min\left(p_{\text{norm}}^{-1}(d_e(\mathbf{z}_i, \mathbf{z}_k)), \lambda\right), \tag{5}$$

where λ is a parameter to provide the least probability to sample any negative nodes [24], $\mathbf{z}_i = f_p(v_i)$ and $\mathbf{z}_k = f_p(v_k)$ are the normalized embeddings of node v_i and v_k, $d_e(\mathbf{z}_i, \mathbf{z}_k)$ is their embedding distance. By this means, we are able to sample negative pairs uniformly regarding the embedding distance during the whole training process. Finally, we sample a negative node v_k by:

$$p_{\text{balanced}}(v_i, v_k) = \alpha p_s(v_i, v_k) + (1 - \alpha) p_e(v_i, v_k), \tag{6}$$

where parameter α is a coefficient to balance the effect of two distance-weighted sampling strategies. Moreover, Figure 4(c) shows the shifted sampling distribution (in orange) from the original sampling distribution.

Thus, given an anchor node v_i, we obtain its ***representative negative node set*** $\mathcal{D}_i^{-'}$ through balanced sampling. The InfoNCE-based balanced sampling (B1_S for short) loss[4] is defined as:

$$\mathcal{L}_{\text{B1\_S}}(v_i) = -\frac{1}{|\mathcal{D}_i^+|} \sum_{v_j \in \mathcal{D}_i^+} \log \frac{e^{g(\mathbf{z}_i, \mathbf{z}_j)/\tau}}{e^{g(\mathbf{z}_i, \mathbf{z}_j)/\tau} + \sum_{v_k \in \mathcal{D}_i^{-'}} e^{g(\mathbf{z}_i, \mathbf{z}_k)/\tau}}, \tag{7}$$

[3] We adopt sampling without replacement to strike a good balance between overwhelmingly large categories and duplicated sampling.

[4] In this study, we use the InfoNCE loss as an example to illustrate each sampling phase. Our sampling strategies can be easily applied to other GCL loss functions as well.

where $\mathcal{D}_i^+ = \{v_j | f_{sd}(v_i, v_j) \sim P\}$ contains congruent nodes of v_i in other graph augmented views, and $\mathcal{D}_i^{-'} = \{v_k | (v_i, v_k) \sim p_{\text{balanced}}\}$, $\mathcal{D}_i^{-'} \subset \mathcal{D}_i^-$. τ is the temperature parameter [3].

3.2 Biased Sampling

Biased sampling is designed to mitigate the *noisy negative pairs* introduced by heuristics $f_{sd}(\cdot)$, i.e., maximizing $\mathcal{I}(P^*; P')$ (see Equation 1). The challenge lies in that the ground-truth labels (i.e., true positive and negative pairs) are not available in practice. Different from some researchers de-noising by fitting the overall distribution with mixed distributions [26], we investigate the discrepancies between the noisy pairs (i.e., false negative pairs) and clean pairs (i.e., true negative pairs) from a dynamic perspective.

Our rationale is that the model fits the noisy and clean pairs in a different manner during the training process: Noisy pairs are usually in-distribution pairs with incorrect labels, which means that they produce similar signals as the majority of the normal training pairs but diverge with different labels. Therefore, they can cause a contrary learning effect. Specifically, assume that we have a clean pairs set $\mathcal{D}_c = \{(v_i, v_j) | f_{sd}(v_i, v_j) \sim P^*\}$ and a noisy pairs set $\mathcal{D}_n = \{(v_i, v_j) | f_{sd}(v_i, v_j) \nsim P^*\}$. Given that $\mathcal{D}_c$ and $\mathcal{D}_n$ can provide conflicting signals to the model, it is harder for the model to fit $\mathcal{D}_n$, compared to fitting $\mathcal{D}_c$, when the training set is $\mathcal{D}_c \cup \mathcal{D}_n$ and $|\mathcal{D}_c| \gg |\mathcal{D}_n|$. Metaphorically, noisy pairs make the model more "struggled" to learn their embeddings.

Slow Learning Effect. Given a training pair (v_i, v_j), the training process starts from the epoch e_{ini} and ends at epoch e_{end}, we measure its *learning speed* by:

$$ls((v_i, v_j)) = \frac{d_e(\mathbf{z}_i^{e_{\text{end}}}, \mathbf{z}_j^{e_{\text{end}}}) - d_e(\mathbf{z}_i^{e_{\text{ini}}}, \mathbf{z}_j^{e_{\text{ini}}})}{e_{\text{end}} - e_{\text{ini}}}, \tag{8}$$

where $\mathbf{z}_\cdot^{e_{\text{ini}}}$ and $\mathbf{z}_\cdot^{e_{\text{end}}}$ are the learned embeddings of node $v_\cdot$ at epoch e_{ini} and e_{end}, respectively.

Figure 5 shows our empirical investigation on the model learning efficiency of the noisy and clean samples on two datasets (i.e., Cora and ACM), we draw learning speeds of clean and noisy pairs under different shortest path distance. We can see that the model is "slower" to learn on the noisy pairs, compared to the clean pairs. We call such a phenomenon as *slow learning effect* of noisy data, introduced by any contrasting heuristics. We include its tests of statistical significance in Appendix A.3.

Based on such an empirical effect, we track the learning speed of the training pairs and correct their "labels" when some pairs manifest the slow learning effect. Specifically, we introduce a hyperparameter β to sample the pairs of the slowest learning effect. Thus, given an anchor node v_i, its new ***positive node set*** $\mathcal{D}_i^{+'} = \{v_j \mid ls(v_i, v_j) < \beta \vee (v_j \in \mathcal{D}_i^+)\}$. As hard negative samples have been proved helpful for learning [16], we can have a conservative[5] β to avoid selecting them. The biased sampling (B2_S for short) loss is defined as:

$$\mathcal{L}_{\text{B2\_S}}(v_i) = -\frac{1}{|\mathcal{D}_i^{+'}|} \sum_{v_j \in \mathcal{D}_i^{+'}} \log \frac{e^{g(\mathbf{z}_i, \mathbf{z}_j)/\tau}}{e^{g(\mathbf{z}_i, \mathbf{z}_j)/\tau} + \sum_{v_k \in V / \mathcal{D}_i^{+'}} e^{g(\mathbf{z}_i, \mathbf{z}_k)/\tau}}, \tag{9}$$

[5] β can be simply set as 0 or values of first five percent of learning speed (sorted in the ascending order) at the first three shortest path distance length.

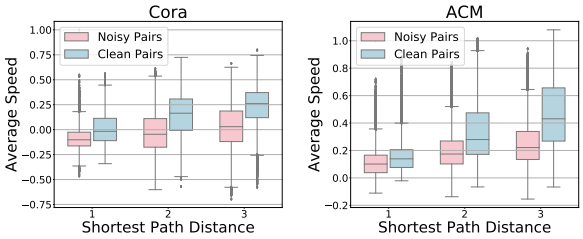


Figure 5: Slow Learning Effect: Box plots of the average learning speed of noisy and clean pairs (at three kinds of shortest path distances) in Cora and ACM datasets.

where V is the full node set. After biased sampling, the positive and negative pairs are reassigned, and an adapted sampling distribution P' is adjusted.

3.3 Complexity Analysis

B^2-Sampling requires calculating the shortest path distance in the pre-computation process and the balanced and biased sampling during training. With N nodes and M edges, we use SPFA [13] to calculate the shortest path distance among nodes in a parallel mode (with c processes). The average time complexity is $O_{sp}(NM/c) \approx O_{sp}(N^2)$, where we set c larger than the average degree of the input graph[6] and $M/c < N$. During the training, our sampling strategy first reuses the pairwise similarity between all node embeddings in base models (without extra computation), and then sample contrasting pairs in time $O_{sample}(N^2)$. In GCL base models, they perform GNN-based encoder with time $O_{enc}(N+M)$ and compute InfoNCE-based contrastive loss in time $O_{loss}(N^2 n)$ [2]. Then, equipped with our B^2-Sampling, the total time complexity is $O_{sp}(N^2) + O_{sample}(N^2) + O_{enc}(N+M) + O_{loss}(N^2 n) \approx O(N+M+N^2 n)$, which approximately equals to the time complexity of GCL base models while increasing their effectiveness.

4 EXPERIMENTS

4.1 Experimental Setup

We evaluate B^2-Sampling with the following research questions:

- **RQ1 (Overall Experiment)**: How effective is our B^2-Sampling compared to the popular negative sampling techniques in the known CL and GCL?
- **RQ2 (Ablation Study)**: How balanced sampling and biased sampling can contribute to the overall performance?
- **RQ3 (Sensitivity Analysis)**: How the runtime configuration of B^2-Sampling affects the overall performance?
- **RQ4 (Applicability Study)**: Whether B^2-Sampling can also boost the performance of graph-level GCL methods?

4.1.1 Base Methods: 3 Representative GCL methods. GRACE [34], GCA [35] and HeCo [22] are three representative GCL methods in representing nodes of homogeneous/heterogeneous graphs. Table 2 shows their key components in GCL paradigm designing, and detailed descriptions are in Appendix B.1. We equip these base methods with B^2-Sampling and other six popular CL/GCL negative mining techniques to see how can they boost their performances.

[6] In our experiment, $c = 24$ and $M/c < N$ holds on all datasets.

Table 2: General descriptions of the GCL base models: GRACE, GCA, and HeCo.

Method	Target Graph Type	Data augmentations		Contrasting Architecture		Evaluation Metric	
		Topology	Feature	Positive Samples	Negative Samples	Classification	Clustering
GRACE	Homogeneous Graphs	Edge Removing	Feature Dropout	a congruent node in the other view	the other nodes in intra-view and inter-view	ACC	–
GCA	Homogeneous Graphs	Adaptive Edge Removing	Adaptive Feature Masking	a congruent node in the other view	the other nodes in intra-view and inter-view	ACC	–
HeCo	Heterogeneous Graphs	Meta-path-based Augmentations	Feature Dropout	n nodes with the most meta-paths	all the remaining nodes	Macro-F1 Micro-F1 AUC	NMI ARI

4.1.2 Datasets. As GRACE and GCA are applicable to homogeneous graphs, and HeCo to heterogeneous graphs, we choose five homogeneous benchmarks and three heterogeneous benchmarks (underlined) following their respective papers, covering four kinds of networks: **(1)** Academic Coauthor Networks: *DBLP*, *Coauthor-CS*; **(2)** Academic Reference Networks: *Wiki-CS*, *Cora*, *ACM*; **(3)** E-commerce Networks: *Amazon-Computers (Computers)*, *Amazon-Photo (Photo)*; **(4)** Movie Knowledge Base: *Freebase-Movie*. Their statistic details and descriptions are in Appendix B.2.

4.1.3 Baselines. We compare B^2-Sampling with six popular CL/GCL negative mining strategies: *DCL* [4], *HCL* [16], *Ring* [25] and *MoCHi* [8] are hard negative mining strategies for computer vision. They try to debiase (DCL), utilize hard negatives (HCL) or semi-hard negatives (Ring), or synthesize new negative points (MoCHi) to improve the quality of visual representations; *ProGCL* [26] aims to eliminate the bias in graph-structured data by fitting a beta mixture model (BMM); *Random sampling* (RS for short) is a simple baseline which uniformly samples negative pairs with a fixed probability.

4.1.4 Implementation details. B^2-Sampling is implemented with PyTorch. We adopt the same evaluation metrics (shown in Table 2) and experimental settings (e.g., multiple runs and random seeds) used in GRACE, GCA and HeCo to perform the node classification and clustering task. Models are trained in an unsupervised manner. The obtained embeddings are fed to a simple Logistic Regression classifier (for a node classification task) or clustered by the K-means algorithm (for a node clustering task). For homogeneous datasets: *Wiki-CS*, *Cora* are from public splits [12, 20], and *Coauthor-CS*, *Amazon-Photo*, *Amazon-Computers* follow a 1:1:8 training/validation/testing set split. For heterogeneous datasets used in HeCo, 40 labeled nodes per class are chosen for training, 1000 nodes for validating, and 1000 for testing. GRACE and GCA are reproduced on a Tesla V100-PCIE-32GB GPU, and HeCo on a GeForce RTX 2080 Ti. Hyperparameters α and β of B^2-Sampling vary from datasets and are empirically set as 0.3 and 0, respectively. The negative sampling ratio k is set as 0.2. Reproducibility details are in Appendix B.3.

4.2 RQ1: Effectiveness

We evaluate **GCL base models + CL/GCL sampling strategies** on node classification and node clustering tasks. The best results are shown in bold, and the second-best results are underlined. "↑" and "↓" refer to performance improvement and drop compared with base models respectively. Overall, our B^2-Sampling performs the best and consistently improves the performances of three base models on different node-level tasks on all datasets. While other baselines are unable to provide continuous improvements over the base models and even worsen them.

Node Classification. As shown in Table 3 and Table 4, we see that B^2-Sampling always performs better than all baselines on all datasets. Generally, CL negative mining strategies (*i.e.*, DCL, HCL, MoCHi, Ring) bring limited improvements or degrade the performances of GCL-based models on most datasets. Since they are designed for mining negatives in vision, failing to leverage the topological structures. RS performs better even than some well-designed negative mining strategies, perhaps because it selects negatives following the exact distribution of the whole negatives. ProGCL improves the performances of base models on most datasets but sometimes with a margin. Our investigation on the debiased sampling approach of ProCL shows that it usually introduces the risks of misrecognizing false negatives (~40%). In contrast, the biased sampling strategy based on the slow learning effect in our B^2-Sampling carries a misrecognizing risk of 0%-20%. Our B^2-Sampling enhances the ACC of GCA by 0.3% to 1.8%, and ACC on Cora even outperforms some supervised node representation learning methods. It enhances HeCo by 0.3% to 2.8% in Micro-F1 scores. Moreover, for a test of statistical significance, we conduct two-sample t-tests on SOTA baselines (underlined) and our B^2-Sampling. Alternative hypothesis is H_1:metric(SOTA)<metric(B^2-Sampling). The p-value in the last line shows that all the p-values are smaller than 0.05, indicating that B^2-Sampling outperforms SOTA baselines with statistical significance. Furthermore, we verified that our B^2-Sampling also achieves strong robustness performance as the noise ratio of negative pairs increases. Please see appendix B.4 for details.

Node Clustering. Table 5 shows the results of HeCo enhanced by baselines and B^2-Sampling. We can see that most baselines improve HeCo on ACM but worsen it on DBLP and Freebase, indicating they fail to handle the noisy-label problem (especially in DBLP and Freebase). ProGCL performs well but is inapplicable to Freebase since it cannot distinguish the positive and negative distributions according to the similarity of embeddings (i.e., '-' in Table 5 without reasonable results). Our B^2-Sampling recognizes the positive and negative pairs by their learning speeds but not the similarity of embeddings, achieving significant improvements over HeCo on all datasets. Specifically, B^2-Sampling improves NMI of HeCo by 4.1%

Table 3: ACC (%±std) of *GCA/GRACE* and *GCA/GRACE + CL/GCL sampling strategies* on node classification.

Method / Dataset	GCA					GRACE				
	Cora	Photo	Computers	Wiki-CS	Coauthor-CS	Cora	Photo	Computers	Wiki-CS	Coauthor-CS
Base Model	82.68±0.04	92.04±0.35	88.00±0.22	78.24±0.06	93.00±0.02	82.13±0.05	91.50±0.53	86.98±0.15	77.13±0.10	92.78±0.01
+RS	82.47±0.05↓	91.88±0.34↓	88.06±0.20↑	77.99±0.09↓	93.09±0.02↑	82.02±0.05↓	91.64±0.47↑	87.38±0.28↑	77.20±0.08↑	92.77±0.01↓
+DCL	82.74±0.05↑	91.33±0.36↓	87.10±0.38↓	78.03±0.08↓	92.91±0.09↓	80.72±0.05↓	91.02±0.29↓	86.99±0.37↑	77.86±0.15↑	93.01±0.07↑
+HCL	79.20±0.04↓	90.65±0.65↓	87.24±0.32↓	77.14±0.14↓	92.36±0.11↓	76.63±0.14↓	90.55±0.51↓	87.01±0.38↑	77.12±0.15↓	92.82±0.03↑
+MoCHi	79.76±0.07↓	91.34±0.05↓	87.96±0.37↓	78.12±0.16↓	93.13±0.07↑	75.19±0.15↓	90.75±0.72↓	88.20±0.14↑	76.53±0.14↓	93.03±0.05↑
+Ring	79.59±0.03↓	91.63±0.62↓	88.23±0.29↑	75.73±0.10↓	92.98±0.10↓	79.82±0.04↓	90.25±0.65↓	88.15±0.21↑	74.23±0.16↓	92.75±0.06↓
+ProGCL	73.31±1.20↓	92.75±0.10↑	87.81±0.18↓	78.25±0.08↑	93.32±0.13↑	82.12±0.08↓	91.85±0.19↑	86.88±0.47↓	77.22±0.10↑	92.95±0.06↑
+B^2-Sampling	**84.43±0.20**↑	**93.15±0.19**↑	**89.28±0.12**↑	**79.18±0.05**↑	**93.34±0.15**↑	**83.88±0.02**↑	**92.26±0.10**↑	**88.80±0.13**↑	**78.16±0.08**↑	**93.07±0.04**↑
(p-value)	*(1.81e-28)*	*(9.50e-10)*	*(2.01e-23)*	*(2.18e-31)*	*(4.90e-4)*	*(4.77e-47)*	*(5.07e-16)*	*(5.92e-20)*	*(2.75e-13)*	*(1.30e-4)*

Table 4: Results (%±std) of *HeCo* and *HeCo + CL/GCL sampling strategies* on node classification.

Method / Dataset	ACM			DBLP			Freebase		
	Macro-F1	Micro-F1	AUC	Macro-F1	Micro-F1	AUC	Macro-F1	Micro-F1	AUC
HeCo	87.31±0.31	86.75±0.47	96.30±0.32	90.48±0.23	90.80±0.28	98.09±0.08	60.95±0.65	63.95±0.97	78.16±1.16
+RS	85.43±0.51↓	85.27±0.76↓	95.56±0.25↓	90.52±0.22↑	90.89±0.26↑	98.14±0.08↑	60.91±0.86↓	63.72±1.07↓	78.24±1.61↑
+DCL	88.92±0.13↑	88.75±0.12↑	96.89±0.48↑	88.46±0.23↓	88.86±0.23↓	97.62±0.12↓	58.15±0.41↓	60.65±0.71↓	74.25±0.86↓
+HCL	87.28±0.11↓	86.83±0.13↑	96.62±0.05↑	87.77±0.31↓	88.11±0.31↓	97.32±0.10↓	59.51±0.47↓	61.35±0.50↓	75.05±0.10↓
+MoCHi	88.62±0.30↑	87.89±0.59↑	96.98±0.16↑	86.00±0.46↓	87.07±0.45↓	96.00±0.17↓	57.00±0.70↓	59.58±0.76↓	73.34±1.14↓
+Ring	88.22±0.38↑	88.09±0.38↑	95.78±0.28↓	90.06±0.43↓	90.42±0.44↓	97.51±0.14↓	61.00±0.70↑	63.58±0.76↓	78.34±1.14↑
+ProGCL	88.26±0.38↑	88.12±0.43↑	96.15±0.24↓	90.17±0.39↓	90.47±0.43↓	98.02±0.12↓	–	–	–
+B^2-Sampling	**89.86±0.22**↑	**89.55±0.36**↑	**97.23±0.11**↑	**90.74±0.41**↑	**91.05±0.41**↑	**98.18±0.12**↑	**62.42±0.41**↑	**65.06±0.58**↑	**78.74±1.00**↑
(p-value)	*(1.87e-20)*	*(1.91e-14)*	*(4.55e-10)*	*(1.54e-05)*	*(1.37e-4)*	*(1.60e-4)*	*(6.88e-12)*	*(2.54e-07)*	*(1.16e-3)*

Table 5: NMI and ARI (%) of *HeCo and HeCo + CL/GCL sampling strategies* on node clustering.

Method / Dataset	ACM		DBLP		Freebase	
	NMI	ARI	NMI	ARI	NMI	ARI
HeCo	60.79	59.82	71.03	76.56	13.40	15.18
+RS	57.01↓	52.11↓	72.12↑	77.29↑	11.47↓	11.58↓
+DCL	63.01↑	66.93↑	61.36↓	67.23↓	12.75↓	13.58↓
+HCL	57.84↓	57.92↓	61.59↓	67.61↓	10.58↓	11.44↓
+MoCHi	63.82↑	67.98↑	58.72↓	64.33↓	9.37↓	10.02↓
+Ring	61.95↑	66.48↑	72.05↑	77.67↑	15.15↑	17.57↑
+ProGCL	61.36↑	59.94↑	73.76↑	78.90↑	–	–
+B^2-Sampling	**66.00**↑	**69.38**↑	**75.14**↑	**80.23**↑	**18.82**↑	**20.28**↑

to 6.8%, and ARI by 3.7% to 9.6%, demonstrating its effectiveness in detecting the community structure of graphs.

4.3 RQ2: Ablation Study

B^2-Sampling consists of *balanced sampling* (B1_S) and *biased sampling* (B2_S). We respectively disable them to evaluate their contributions to the overall performance. We conduct the ablation study based on HeCo with Micro-F1 measurement to evaluate the performance on the node classification task, and NMI and ARI measurement on the node clustering task.

As shown in Table 6, B1_S performs better on DBLP while B2_S performs better on ACM. The difference largely lies in different datasets suffering from imbalanced and noisy-label problems to different degrees. The positive pairs in the DBLP are much more abundant than those in the ACM. Thus, the boosting performance of B2_S in the DBLP is less significant than that in the ACM. Overall, both of the sampling components are helpful for training.

To further compare the effects of B^2-Sampling, B1_S, and B2_S, we visualize the embedding distance distribution of positive and negative pairs in Figure 6. We compare embeddings learned by HeCo, HeCo+B1_S, HeCo+B2_S, and HeCo+B^2-Sampling in a pairwise way. From a visual point of view, the less the overlapping area between two distributions, the better one performs than the other. Overall, our B^2-Sampling achieves less overlap between positive and negative pairs distributions (see Figure 6(a)), demonstrating its strong ability to discriminate positive and negative pairs. Figure 6(b) and 6(c) show that both B1_S and B2_S can draw positive pairs closer and pushes negative pairs farther on the embedding distance compared with the original HeCo. In addition, B1_S and B2_S show respective advantages in learning positive pairs and negative pairs (see Figure 6(f)). B^2-Sampling learns the best embeddings based on the advantages from B1_S and B2_S.

4.4 RQ3: Sensitivity Analysis

We perform sensitivity analysis on three critical hyper-parameters in B^2-Sampling: the sampling ratio k for selecting representative negative samples, the coefficient α to balance the shortest path distance weight and embedding distance weight in balanced sampling, and the threshold β in biased sampling to correct labels of noisy samples. We report the Micro-F1 and NMI values on ACM

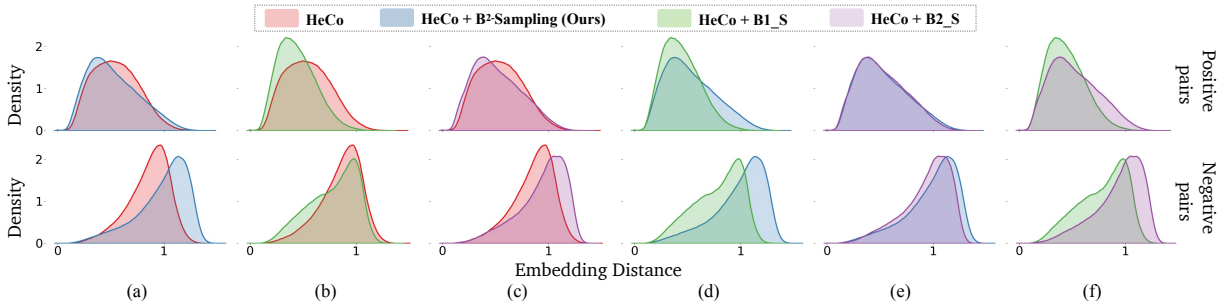


Figure 6: Densities of embedding distance between nodes with the same label (positive pairs at the top) and different labels (negative pairs at the bottom) for embeddings trained on ACM with four different training strategies.

Table 6: Ablation study for B^2-Sampling on node classification and node clustering. B1_S and B2_S are short for balanced sampling and biased sampling respectively.

Method \ Dataset	ACM			DBLP		
	Micro-F1	NMI	ARI	Micro-F1	NMI	ARI
HeCo	86.75	61.05	59.82	90.80	71.03	76.56
+B1_S	88.04	62.44	62.87	91.01	74.81	80.14
+B2_S	88.54	64.17	66.53	90.86	71.22	76.81
+B^2-Sampling	**89.55**	**66.00**	**69.38**	**91.05**	**75.14**	**80.23**

Table 7: ACC (%±std) gains by applying biased sampling to GraphCL/MVGRL on different datasets in the graph classification task.

Method	MUTAG	NCI1	PROTEINS	DD
GraphCL	87.4 ± 1.4	77.6 ± 0.4	74.6 ± 0.5	78.6 ± 0.4
+Biased Sampling	**91.5 ± 1.2**	**79.9 ± 0.5**	**75.4 ± 0.3**	**80.7 ± 0.5**
Method	MUTAG	PTC_MR	IMDB-BIN	IMDB-MULTI
MVGRL	89.7 ± 1.1	62.5 ± 1.7	74.2 ± 0.7	51.2 ± 0.5
+Biased Sampling	**90.4 ± 1.3**	**64.5 ± 1.4**	**74.9 ± 0.8**	**51.7 ± 0.3**

and Freebase by varying k, α in Figure 7(a) and Figure 7(b), and β in Figure 7(c) and have the following conclusion:

(1) *Balanced sampling strategy is robust against a variety of k and α.* We observe that the performance drops slightly with k increasing, showing more negative samples does not mean better performance.
(2) *A smaller β is more practical choice for B^2-Sampling.* We can see that, when $\beta < 0$, the values of Micro-F1 and NMI are stable with the increase of β; In contrast, when $\beta > 0$, Micro-F1 and NMI decrease with the increase of β. Overall, when the learning speed is not "that slow", the sampled pairs are likely to have the correct label.

4.5 RQ4: Applicability Study

We probe into the applicability of B^2-Sampling on graph-level learning tasks. Since our balanced sampling leverages the local topological property (shortest path distance) among node pairs, which is inadaptability among graphs, we take biased sampling as a light version of B^2-Sampling and apply it to graph-level GCL models. GraphCL [29] and MVGRL [7] are two well-known GCL models based on graph-level contrastive losses, we adopt seven datasets from their works and conduct graph classification tasks. As shown in Table 7, our biased sampling consistently enhances the performance of base models, showing its effectiveness and applicability.

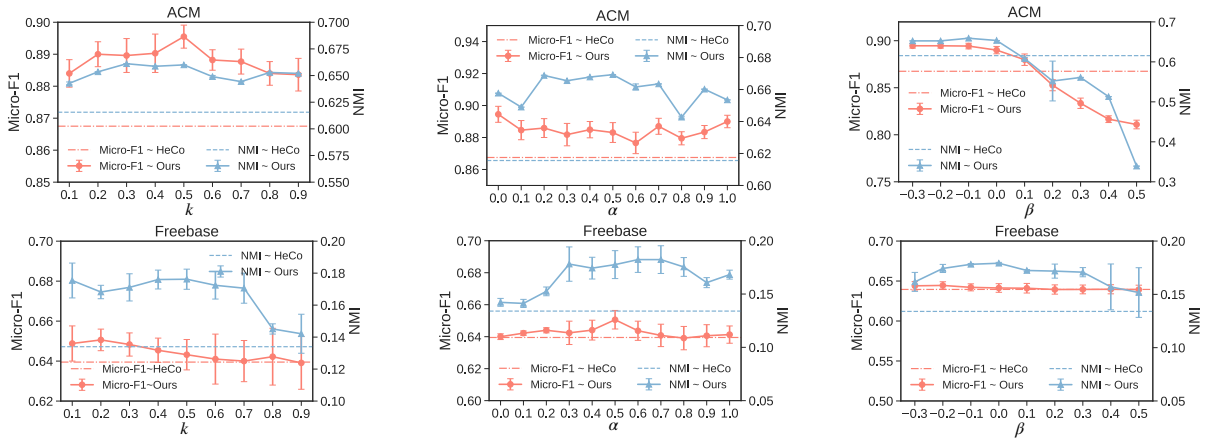
5 RELATED WORKS

5.1 Graph Contrastive Learning

Graph contrastive learning is an increasingly popular self-supervised learning approach [7, 19, 21, 26, 29, 34, 35]. They usually apply topological augmentation to generate multiple views and adopt different contrasting modes to maximize similarities of positive pairs while minimizing similarities of negative pairs with different contrastive loss. For a more comprehensive overview, readers may check out here [27, 33]. Most existing GCL methods are negative-sample-based and can be categorized into three contrasting modes: local-local (L-L), global-global (G-G), and global-local (G-L).

GCL methods in L-L contrasting mode define positive and negative pairs on node level, *i.e.*, the positive and negative samples are node pairs. For example, given an anchor node, GCC [15] designs its positives and negatives in other networks to learn transferable structural node representations; GRACE [34] treats its congruent node from another augmented graph as the positive one and all left nodes as negatives; GCA [35] adopts the same designation as GRACE but further equips GRACE with adaptive data augmentation, learning important patterns underneath the input graph. Our B^2-Sampling can be easily applied to GCL methods in L-L contrasting mode and make a further improvement.

GCL methods in G-G/G-L contrasting modes define positive and negative pairs on graph-level, in $(graph, node)$ or $(graph, graph)$-shaped. For example, given a graph G, DGI [21] and MVGRL [7] apply graph augmentation to it to get another mutant graph G', and then take nodes in G as positives while in G' as negatives; GraphCL [29] takes multiple generated augmented graphs based on G as positives, and other graphs in the same minibatch are negatives. InfoGraph [18] encodes multiple graphs and maximizes the MI of the graph-node, graph-edges, and graph-context pairs to obtain representations of substructures of different scales. They output graph embeddings for graph-level tasks. Since our balanced



(a) Micro-F1/NMI v.s. Negative sampling ratio k (b) Micro-F1/NMI v.s. Balance coefficient α (c) Micro-F1/NMI v.s. Threshold β

Figure 7: Sensitivity w.r.t Hyper-parameters k, α, β.

sampling in B^2-Sampling considers the local topological property (shortest path distance) among node pairs, which is inadaptability among graphs, we can take biased sampling as a light version of B^2-Sampling and apply it to GCL methods in these two modes.

5.2 CL/GCL Sampling

There are two kinds of noteworthiness negative samples, false negative samples and hard negative samples, which guide a CL/GCL method to correct its mistakes more quickly [4, 14]. False negative samples are samples with the same labels but are treated as dissimilar pairs because of the contrasting heuristics in CL/GCL methods. Hard negative samples are those pairs that are mapped nearby in embedding space but should be far apart.

Some CL sampling strategies develop debasing terms to avoid contrasting false negative pairs. For example, DCL [4] decomposes the data distribution into positive and negative distributions and develops a debiased contrastive objective to relieve the sampling bias. Meanwhile, plenty of CL sampling strategies are interested in hard negative samples: they adopt different hard negative mixing strategies [8, 10] or build a tunable sampling distribution that prefers hard samples [16] to generate diverse and informative negative samples. In spite of their promising performance in the field of computer vision, they bring limited improvement or even performance drop when applied to graphs [26, 33].

Recently, some researchers pay more attention to hard negative samples in graph-structured data. ProGCL [26] distinguishes true and false negatives by fitting a beta mixture model on the similarities of embeddings and proposes two strategies based on this: ProGCL-weight re-weights positive and negative terms in the denominator of loss; ProGCL-mix synthesizes more hard negatives. However, ProGCL-weight may allocate more weights to negative samples which are easy to train. Moreover, our empirical experiment shows that once two distributions are mixed, especially if one distribution is the minority, there is limited information to decompose the overall distribution. M-Mix [30] follows i-Mix [10] and dynamically assigns different mixing weights when generating hard negatives. One of its modules, M-Mix-up, utilizes an adjacency matrix for denoising in graphs. Different from them, our B^2-Sampling samples informative negatives by measuring topology and embedding diversities and further corrects the labels of false negatives by our slow learning effect observation.

6 CONCLUSION

In this work, we propose B^2-Sampling, a two-phase sampling strategy applicable to a class of GCL methods for further boosting performance. Balanced sampling in phase one selects representative negative pairs with diversified shortest path distances and embedding distances to consistently provide information for training. Biased sampling in phase two corrects the potential false negative pairs regarding their slow learning effect to denoising. Through extensive experiments on different node-level and graph-level downstream tasks, our B^2-Sampling performs the best compared to various baselines. Our evaluation shows that B^2-Sampling is easily compatible with node-wise GCL methods in local-local contrasting mode, and its light version (biased sampling) is also applicable to GCL methods in G-G, G-L contrasting modes, showing its superiority.

7 ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (2022YFC3303600), National Natural Science Foundation of China (62137002, 62293553), Innovative Research Group of the National Natural Science Foundation of China (61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Natural Science Basic Research Program of Shaanxi (2023-JC-YB-293), the Youth Innovation Team of Shaanxi Universities, Minister of Education, Singapore (MOET32020-0004), the National Research Foundation, Singapore, and Cyber Security Agency of Singapore under its National Cybersecurity Research and Development Programme (Award No. NRF-NCRI_TAU_2021-0002) and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN).

REFERENCES

[1] Luis Bulla. 1994. An index of evenness and its associated diversity measure. *Oikos* (1994), 167–171.

[2] Ming Chen, Zhewei Wei, Bolin Ding, Yaliang Li, Ye Yuan, Xiaoyong Du, and Ji-Rong Wen. 2020. Scalable graph neural networks via bidirectional propagation. *Advances in neural information processing systems* 33 (2020), 14556–14566.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[4] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *Advances in neural information processing systems* 33 (2020), 8765–8775.

[5] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[6] John M Hammersley. 1950. The distribution of distance in a hypersphere. *The Annals of Mathematical Statistics* (1950), 447–452.

[7] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*. PMLR, 4116–4126.

[8] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 21798–21809.

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[10] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. 2021. I-mix: A domain-agnostic strategy for contrastive representation learning. *ICLR* (2021).

[11] Mengyue Liu, Yun Lin, Jun Liu, Bohao Liu, Qinghua Zheng, Jin Song Dong. 2023. B2-Sampling Website. [Online; accessed 2 Feb 2023]. https://sites.google.com/view/b2-sampling/home.

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

[13] Edward F Moore. 1959. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959*. 285–292.

[14] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2016. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4004–4012.

[15] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160.

[16] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive Learning with Hard Negative Samples. In *ICLR*.

[17] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[18] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2019. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000* (2019).

[19] Puja Trivedi, Ekdeep Singh Lubana, Yujun Yan, Yaoqing Yang, and Danai Koutra. 2022. Augmentations in graph contrastive learning: Current methodological flaws & towards better practices. In *Proceedings of the ACM Web Conference 2022*. 1538–1549.

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

[21] 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.

[22] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1726–1736.

[23] Yanling Wang, Jing Zhang, Haoyang Li, Yuxiao Dong, Hongzhi Yin, Cuiping Li, and Hong Chen. 2022. ClusterSCL: Cluster-Aware Supervised Contrastive Learning on Graphs. In *Proceedings of the ACM Web Conference 2022*. 1611–1621.

[24] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 2840–2848.

[25] 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).

[26] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z Li. 2022. ProGCL: Rethinking Hard Negative Mining in Graph Contrastive Learning. In *International Conference on Machine Learning*. PMLR, 24332–24346.

[27] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2021. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757* (2021).

[28] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2021. Enhanced graph learning for collaborative filtering via mutual information maximization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 71–80.

[29] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems* 33 (2020), 5812–5823.

[30] Shaofeng Zhang, Meng Liu, Junchi Yan, Hengrui Zhang, Lingxiao Huang, Xiaokang Yang, and Pinyan Lu. 2022. M-Mix: Generating Hard Negatives via Multi-sample Mixing for Contrastive Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2461–2470.

[31] Deli Zhao, Jiapeng Zhu, and Bo Zhang. 2019. Latent Variables on Spheres for Sampling and Spherical Inference. (2019).

[32] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. 2021. Graph debiased contrastive learning with joint representation clustering. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 3434–3440.

[33] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116* (2021).

[34] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep graph contrastive representation learning. *GRL + @ICML* (2020).

[35] 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*. 2069–2080.

A METHOD

A.1 Explanation of Balanced Sampling

Assume that the samples in $\mathcal{D}^-$ follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, and after balanced sampling, the selected samples in $\mathcal{D}^{-'}$ follows $\mathcal{N}(\mu', \sigma'^2)$. We use the information entropy $\mathcal{H}$ to represent the diversity $\mathcal{K}$, and the entropy $\mathcal{H}$ of $\mathcal{N}(\mu, \sigma^2)$ is:

$$\begin{aligned}\mathcal{H}\left[\mathcal{N}\left(\mu,\sigma^2\right)\right] &= -\int_x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= -\int_x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left[-\frac{1}{2}\log 2\pi\sigma^2 - \frac{(x-\mu)^2}{2\sigma^2}\right] dx \\ &= \frac{1}{2}\log 2\pi\sigma^2 + \frac{1}{2\sigma^2}\int_x (x-\mu)^2 \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= \frac{1}{2}\log 2\pi\sigma^2 + \frac{\sigma^2}{2\sigma^2} \\ &= \frac{1}{2}\log 2\pi e\sigma^2.\end{aligned} \tag{10}$$

As we claims in Section 3.1, after balanced sampling, $\sigma' > \sigma$, and the entropy of selected samples in $\mathcal{D}^{-'}$ is larger than samples in $\mathcal{D}^-$. Therefore, we obtain more diverse samples after balanced sampling.

A.2 Topological Diversity

During balanced sampling, we need to measure the topological distance between nodes within the negative pairs to explore the topological diversity. We find that the shortest path distance is strongly associated with topological diversity. As shown in Figure 8, the x-axis indicates the indices of shortest path distance, and the y-axis indicates the ratio of negative pairs. We can see that the negative pairs lie in all shortest path distances. Following the shortest path distance, we can sample various negative pairs with different distances, capturing the topological diversity of the graph and measuring topological diversity from local to global.

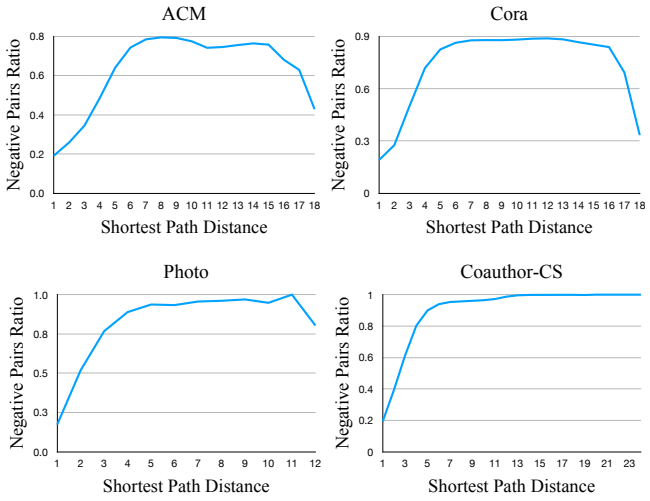


Figure 8: Shortest path distance reflects the topological diversity of graphs.

A.3 Tests of Statistical Significance

For a tests of statistical significance of Figure 5, we adopt a two-sample t-test on noisy and clean pairs in different shortest path distances. Our alternative hypothesis is $H_1 : ls_{noisy} < ls_{clean}$, and the null hypothesis is $H_0 : ls_{noisy} > ls_{clean}$. The results are shown in Table 8, showing that the learning speed of noisy samples is slower than that of clean samples with statistical significance.

Table 8: A two-sample t-test on noisy and clean pairs.

Dataset	Shortest Path Distance	Statistic	p-vlaue
Cora	1	-30.57371189	5.34e-197
	2	-115.0578437	<1e-200
	3	-253.2544935	<1e-200
ACM	1	-44.80810994	<1e-200
	2	-221.7181418	<1e-200
	3	-602.5348482	<1e-200

B DETAILS OF EXPERIMENT

B.1 Representative GCL Base Models.

GRACE [34], GCA [35] and HeCo [22] are three representative methods for representing nodes of homogeneous information networks and heterogeneous information networks with GCL paradigms, respectively. They adopt variants of InfoNCE loss and design *node-node* level positive and negative pairs:

- **GRACE** focuses on contrasting embeddings at the node level. It generates two graph views by corruption and learn node representations by maximizing the agreement of node representations in these two views. For a given node in homogeneous graphs, the congruent node in the augmented graph is defined as its positive sample, and the remaining nodes in two views (the original graph and augmented graph) are negative samples.
- **GCA** enhances GRACE by adopting adaptive edge removing (ER) and adaptive feature masking (FM) for graph augmentation. Adaptive graph augmentations help GCA learn representations that are insensitive to perturbation on unimportant nodes and edges. Its definition for the contrasting pairs is the same as GRACE's.
- **HeCo** selects *meta-path* view and *network schema* view according to structure characteristics of heterogeneous information networks for the graph augmentation. For a given node, its positive and negative samples are determined by the number of meta-paths connecting them. If two nodes are connected by many meta-paths, they are positive samples (i.e., a node can have multiple positive samples), and all left nodes are negative samples. Nodes embedding in a pair are from different views, realizing cross-view self-supervision.

B.2 Datasets

The eight datasets used in this paper are from four kinds of networks:

Table 10: Statistics of datasets. "-" indicates the node features are not provided and needed.

Dataset	# Nodes	Dimensions of Node Features	# Edges	# Classes
Wiki-CS	11,701	300	216,123	10
Computers	13,752	767	245,861	10
Photo	7,650	745	119,081	8
Coauthor-CS	18,333	6,805	81,894	15
Cora	2,708	1,433	5,429	7
ACM	Paper (P): 4,019 Author (A): 7,167 Subject (S): 60	P: 1,902 A: 7,167 S: 60	P-A: 13,407 P-S: 4,019	3
DBLP	Paper (P): 14,328 Author (A): 4,075 Conference (C):20 Term (T):7,723	P: 14,328 A: 344 C: – T: –	P-A: 19,645 P-C: 14,328 P-T: 85,810	4
Freebase	Movie (M): 3,492 Actor (A): 33,401 Director (D): 2,502 Writer (W): 4,459	M: 3,492 A: 2,502 D: 33,401 W: 4,459	M-A: 65,341 M-D: 3,762 M-W: 6,414	3

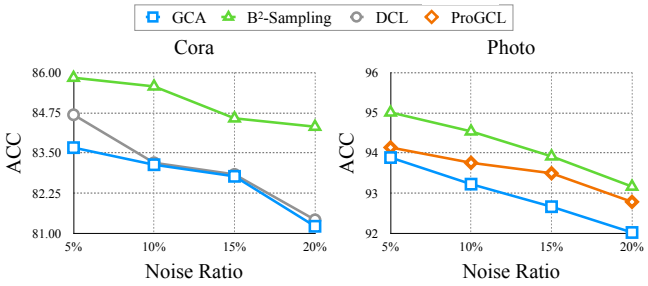


Figure 9: Robustness against Noise Negative Pairs.

- **Academic Coauthor Networks**: *DBLP*[7] contains heterogeneous nodes (author, paper, conference, term) and edges, while *Coauthor-CS*[8] is a homogeneous academic network. Both of their target nodes are authors, linked by co-author relationships.
- **Academic Reference Networks**: *Wiki-CS*[9] and Cora[10] are reference networks, in which nodes represent articles and edges represent their citations. *ACM*[11] is a heterogeneous network and the target nodes, papers, are linked by authors and subjects.
- **E-commerce Networks**: *Amazon-Computers* and *Amazon-Photo*[12] are homogeneous information networks where nodes represent goods and edges represent co-purchase relation.
- **Movie Knowledge Base**: *Freebase-Movie*[13] is a heterogeneous information network where nodes represent movies labeled by genres, and edges represent the relations among actors, directors, and producers.

Their statistic details are shown in Table 10.

B.3 Hyper-parameters Setting

Table 9: Hyperparameter Specifications

Dataset	k	α	β	sp
Wiki-CS	0.2	0.3	0.4	1,2
Amazon-Computers	0.2	0	0.2	1,2,3
Amazon-Photo	0.2	0	0.2	1,2,3
Coauthor-CS	0.2	0.3	-0.5	-1,1,2
Cora	0.2	0	-0.2	1,2,3,4
ACM	0.2	0.3	0	1,2,3
DBLP	0.02	0.3	-1.0	1,2
Freebase	0.2	0.3	-0.75	1,2

All model parameters are initialized with Glorot initialization [5], and trained using Adam SGD optimizer [9] on all datasets. We list four crucial hyper-parameters: the negative sampling ratio k, balance coefficient α in the first phase, and threshold β and the used shortest path distance sp in the second phase, in Table 9.

B.4 Robustness against Noise Negative Pairs

To explore the robustness against noisy negative pairs, we further conduct a robustness experiment on the node classification task (based on the experiments in Table 3). Specifically, we test our B^2-Sampling on Cora and Amazon-Photo (Photo) when the noise ratio of negative pairs increases from 5% to 20%. As shown in Figure 9, we can observe that (1) our B^2-Sampling can still achieve significant improvements over the base model GCA across different noise ratios; and (2) B^2-Sampling also consistently outperforms the SOTA baselines (i.e., underlined methods in Table 3).

[7] https://github.com/cynricfu/MAGNN
[8] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/ms_academic_cs.npz
[9] https://github.com/pmernyei/wiki-cs-dataset/tree/master/dataset
[10] https://github.com/tkipf/gcn
[11] https://github.com/Andy-Border/NSHE
[12] https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_%20computers.npz
[13] https://github.com/dingdanhao110/Conch