Does GCL Need a Large Number of Negative Samples? Enhancing Graph Contrastive Learning with Effective and Efficient Negative Sampling (Appendix)

Related Work

Graph Neural Networks. Graph Neural Networks (GNNs) have become a cornerstone in learning from graph-structured data, thanks to their ability to capture complex relational dependencies between nodes. The foundational work in this area, such as Graph Convolutional Networks (GCNs) (Kipf and Welling 2017), applies convolution operations to graph domains. GCNs aggregate information from a node's local neighborhood to learn effective node embeddings, making them particularly suited for tasks on homophilic graphs. However, the assumption of homophily and the risk of over-smoothing with deeper layers limit their expressiveness, particularly in complex graph structures.

To overcome some of these limitations, GraphSAGE (Hamilton, Ying, and Leskovec 2017) utilizes a sampling and aggregation strategy to create node representations. By sampling a fixed number of neighbors and aggregating their features. GraphSAGE also enables the model to handle unseen nodes effectively, addressing some of the scalability issues inherent in GCNs. Graph Attention Networks (GATs) (Veličković et al. 2018) marked another significant advancement by incorporating attention mechanisms into the aggregation process. which allow the model to assign different importance weights to different neighbors, enabling more fine-grained and context-aware representation learning. While these models have made significant strides in graph representation learning, they primarily focus on local neighborhood aggregation, often overlooking the broader topological structure of the graph. Unlike traditional methods that rely on fixed or learnable neighborhood aggregation, topological receptive fields consider the global topological context, providing a more holistic understanding of a node's position and role within the graph. These approaches aim to capture the nuanced topological features of graphs, enabling models to better understand and leverage the underlying structure, particularly in scenarios where local neighborhood information alone is insufficient.

Graph Contrastive Learning. Graph Contrastive Learning (GCL) currently has attracted widespread attention in the academic community. It mainly generates multiple augmented views through augmentation, and designs objective functions to train the model based on maximizing mutual information to reduce the model's dependence on label information. As a classic paradigm, GRACE (Zhu et al. 2020) trains the model by maximizing the similarity of nodes at the same position in two views and minimizing the similarity of nodes at other positions. On this basis, GCA (Zhu et al. 2021) designed an adaptive enhanced GCL framework to measure the importance of nodes and edges to protect the semantic information of graph data during augmentation. Local-GCL (Zhang et al. 2022) treats first-order neighbors as positive samples and employs a kernel-based contrastive loss to reduce complexity. ProGCL (Xia et al. 2022) utilizes a Beta mixture model to estimate the probability of a negative sample being true and devises methods to compute the weight of negative samples and synthesize new negative samples. B^2-Sampling (Liu et al. 2023) employs a balanced sampling strategy to select negative samples and corrects the noisy labels within these samples. HomoGCL (Li et al. 2023) expands the number of positive sample pairs using a Gaussian mixture model (GMM) and calculates the weight of positive samples through soft clustering. BGRL (Thakoor et al. 2021) adopts BYOL (Grill et al. 2020) as the backbone, where the online encoder is trained by predicting the target encoder to generate efficient node representations. Like BGRL, AFGRL (Lee, Lee, and Park 2022) also employs BYOL as the backbone to eliminate the dependency on negative samples. AFGRL determines positive samples using K Nearest Neighbor (KNN) and clustering algorithms. Unlike the GCLs mentioned above, DGI (Velickovic et al. 2019) learns embeddings by maximizing the mutual information between node representations and graph representations, while GGD (Zheng et al. 2022) divides nodes into positive and negative groups and uses simple binary cross-entropy to distinguish between these groups. MVGRL (Hassani and Ahmadi 2020) generates a global view using a diffusion matrix and contrasts it with a local view, combining global and local information.

Negative Sampling. Negative sampling plays a pivotal role in contrastive learning frameworks, where the objective is to distinguish between similar and dissimilar pairs. This concept builds on Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen 2010) , where the task is to differentiate true data points from noise introduced into the dataset. In

NCE, negative samples are drawn from a noise distribution and are essential for the model to learn meaningful representations by contrasting them with positive samples.

In the field of visual representation learning, several landmark studies have emphasized the importance of negative sampling. MoCo (He et al. 2020) introduces a dynamic memory bank that stores a large set of negative samples, which are then contrasted with positive pairs to learn robust embeddings. The approach allows for a more flexible and efficient retrieval of negative samples, which is critical in maintaining the quality of the learned representations. SimCLR (Chen et al. 2020) demonstrates that using larger batch sizes to generate numerous negative pairs during training significantly enhances performance. Both MoCo and SimCLR operate on the principle that the more negative samples a model is exposed to, the better it can distinguish between positive and negative instances, leading to more discriminative representations.

Further advancements in negative sampling strategies have been explored in methods like InfoNCE (van den Oord, Li, and Vinyals 2018) and PIRL (Misra and van der Maaten 2020). InfoNCE is designed to optimize the model by comparing each positive sample against multiple negatives. PIRL extends this by leveraging negative samples that are harder to distinguish from positives, thereby pushing the model to learn more nuanced representations.

Theoretical Supplement

In order to make the readers more clear about the motivation of our article, we provide the proof of our Theorem here.

Proof of Theorem 1

Proof: We assume the graph contains k semantic blocks $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_k\}$, each with a core semantic s_j. For an anchor node v_i belonging to semantic block $\mathcal{S}_j$, its feature can be decomposed as:

$$x_i = s_j + \epsilon_i, \tag{1}$$

where s_j is the core semantic of block $\mathcal{S}_j$, and ϵ_i represents the individual deviation of node v_i.

Intra-Block Difference. For two nodes v_i and v_j within the same semantic block $\mathcal{S}_p$, their features are:

$$x_i = s_p + \epsilon_i, \quad x_j = s_p + \epsilon_j, \tag{2}$$

and the difference between their features is:

$$\Delta\text{diff}_{\text{intra}} = \|x_i - x_j\| = \|\epsilon_i - \epsilon_j\|, \tag{3}$$

indicating that the intra-block difference is determined by the individual deviations from the core semantic.

Inter-Block Difference. For two nodes v_i and v_j from different semantic blocks $\mathcal{S}_p$ and $\mathcal{S}_q$, their features are:

$$x_i = s_p + \epsilon_i, \quad x_j = s_q + \epsilon_j, \tag{4}$$

and the difference between their features is:

$$\Delta\text{diff}_{\text{inter}} = \|(s_p + \epsilon_i) - (s_q + \epsilon_j)\|. \tag{5}$$

Using the triangle inequality, this can be simplified to:

$$\Delta\text{diff}_{\text{inter}} = \|s_p - s_q + (\epsilon_i - \epsilon_j)\| \geq \|s_p - s_q\| - \|\epsilon_i - \epsilon_j\|. \tag{6}$$

When $\|s_p - s_q\|$ is sufficiently large, the individual deviation $\|\epsilon_i - \epsilon_j\|$ can be neglected, leading to:

$$\Delta\text{diff}_{\text{inter}} \approx \|s_p - s_q\|, \tag{7}$$

indicating that the inter-block difference is primarily determined by the difference between the core semantics of the blocks. □

Proof of Theorem 2

Proof: Consider the InfoNCE loss function with respect to the similarity between negative samples:

$$\frac{\partial \mathcal{L}_{\text{InfoNCE}}}{\partial \theta(h_i, h_j)} = \frac{1}{\tau} \frac{e^{\theta(h_i, h_j)/\tau}}{e^{\theta(h_i, h_i')/\tau} + \sum_{k \neq i} e^{\theta(h_i, h_k)/\tau}}, \tag{8}$$

where $\theta(h_i, h_j)$ represents the similarity between any pair of negative samples. For a given anchor node v_i, the sum of gradients for intra-semantic block negatives $\text{SG}(i)_{\text{intra}}$ and inter-semantic block negatives $\text{SG}(i)_{\text{inter}}$ can be expressed as:

$$\begin{aligned} \text{SG}(i)_{\text{intra}} &= \sum_{j \in \mathbf{N}_{\text{intra}}} \frac{\partial \mathcal{L}_{\text{InfoNCE}}}{\partial \theta(h_i, h_j)}, \\ \text{SG}(i)_{\text{inter}} &= \sum_{j' \in \mathbf{N}_{\text{inter}}} \frac{\partial \mathcal{L}_{\text{InfoNCE}}}{\partial \theta(h_i, h_{j'})}. \end{aligned} \tag{9}$$

To achieve balance, we assume $\text{SG}(i)_{\text{intra}} = \text{SG}(i)_{\text{inter}}$:

$$\sum_{j \in \mathbf{N}_{\text{intra}}} e^{\theta(h_i, h_j)/\tau} = \sum_{j' \in \mathbf{N}_{\text{inter}}} e^{\theta(h_i, h_{j'})/\tau}. \tag{10}$$

Assuming $\theta(h_i, h_j) = 1$ for intra-block pairs and letting $\hat{\theta}(h_i, h_{j'})$ denote the mean similarity for inter-block pairs, we get:

$$Pe^{1/\tau} = \sum_{j'=1, j' \neq i}^{N-P} e^{\theta(h_i, h_{j'})/\tau}. \tag{11}$$

This defines the threshold at which the model balances its focus between distinguishing intra- and inter-semantic block negatives. □

Datasets

In our experiments, we adopt six widely-used datasets, including *PubMed* (Yang, Cohen, and Salakhutdinov 2016), *Amazon-Photo*, *Amazon-Computers* (Shchur et al. 2018), *Coauthor-CS*, *Coauthor-Physics* (Sinha et al. 2015) and *Wiki-CS* (Mernyei and Cangea 2020). The detailed introduction of these datasets are as follows:

- **PubMed** (Yang, Cohen, and Salakhutdinov 2016) is citation network datasets where nodes mean papers and edges mean citation relationships. Each dimension of the feature corresponds to a word. The nodes are labeled by the categories of the paper.
- **Amazon-Photo and Amazon-Computers**(Shchur et al. 2018) are two networks based on Amazon co-purchase graphs, where nodes mean goods and edges mean that two goods are frequently bought together. Node features are bag-of-words vector generated from product reviews, and the nodes are labeled by product categories.

Table 1: Hyperparameters specifications

Dataset	Learning rate	Weight decay	Hidden_dim	Num epoch	Cluster	Neighbors
PubMed	0.00005	0.0005	4096	1500	30	100
CS	0.0001	0.00005	2048	1500	50	100
Photo	0.00001	0.00001	4096	600	10	100
Computers	0.00005	0.00001	4096	200	30	100
Physics	0.00001	0.00005	2048	600	15	100
Wiki-CS	0.00001	0.00005	512	200	15	10

Table 2: Code links of various baseline methods.

Methods	Source Code
BGRL	https://github.com/nerdslab/bgrl
MVGRL	https://github.com/kavehhassani/mvgrl
DGI	https://github.com/PetarV-/DGI
GBT	https://github.com/pbielak/graph-barlow-twins
GRACE	https://github.com/CRIPAC-DIG/GRACE
GCA	https://github.com/CRIPAC-DIG/GCA
ProGCL	https://github.com/junxia97/ProGCL

- **Coauthor-CS and Coauthor-Physics** (Sinha et al. 2015) are two co-authorship networks based on the Microsoft Academic Graph from the KDD Cup 2016 challenge, where nodes mean authors and edged mean co-authorship between two authors. Node features are paper keywords of each author's papers, and class labels correspond to the most active fields of each author.
- **Wiki-CS** (Mernyei and Cangea 2020) is a Wikipedia-based dataset where nodes mean Computer Science articles and edges mean hyperlinks between the articles. Node features are calculated as the average of pre-trained GloVe word embeddings (Pennington, Socher, and Manning 2014) and class labels are different branches of the field.

Hyperparameters Settings

In this section, we present the hyperparameter specifications used for training the E2Neg model on various datasets. Table 1 details the hyperparameters employed for different datasets.

Pseudo Code of E2Neg

The following pseudo code outlines the E2Neg training algorithm, which combines spectral clustering with graph augmentation to improve negative sampling in GCL. The algorithm clusters nodes, identifies cluster centers, and reconstructs the graph topology. During training, it computes embeddings for both the original and augmented graphs, and updates the model using the InfoNCE loss, focusing on the cluster centers to train the embeddings.

Algorithm 1: The E2Neg training algorithm

Input: Original Graph $\mathcal{G}$, Encoder f, Projector g.
Parameter: parameters of various datasets in Table 1.

Get cluster $C = \{C_1, C_2, \cdots, C_K\}$ using K-means
Get cluster center $c = \{c_1, c_2, \cdots, c_k\}$ via spectual centrality
Get $\hat{\mathcal{G}}$ by topology reconstruction of $\mathcal{G}$
for epoch = 0, 1, 2, . . . **do**
    Generate an augmentation function t
    Get augmented graph via $\tilde{\mathcal{G}} = t(\hat{\mathcal{G}})$
    Get node embedding $\hat{\boldsymbol{H}}$, $\tilde{\boldsymbol{H}}$ of $\hat{\mathcal{G}}$, $\tilde{\mathcal{G}}$ through the same encoder f.
    Compute InfoNCE loss $\mathcal{L}$ only using the representations of cluster centers.
    Update the parameters of f via $\mathcal{L}$
end for
Return node embedding $\boldsymbol{H}$, trained encoder f.

Reproducibility

Table 2 presents the GitHub links to the source codes of various contrastive methods used in our evaluation.

References

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, 1597–1607.

Grill, J.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. Á.; Guo, Z.; Azar, M. G.; Piot, B.; Kavukcuoglu, K.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *JMLR Proceedings*, 297–304. JMLR.org.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30*, 1024–1034.

Hassani, K.; and Ahmadi, A. H. K. 2020. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the 37th International Conference on Machine Learning*, 4116–4126.

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

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations*.

Lee, N.; Lee, J.; and Park, C. 2022. Augmentation-Free Self-Supervised Learning on Graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence*, 7372–7380. AAAI Press.

Li, W.; Wang, C.; Xiong, H.; and Lai, J. 2023. HomoGCL: Rethinking Homophily in Graph Contrastive Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1341–1352.

Liu, M.; Lin, Y.; Liu, J.; Liu, B.; Zheng, Q.; and Dong, J. S. 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*, 1489–1500.

Mernyei, P.; and Cangea, C. 2020. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. *CoRR*, abs/2007.02901.

Misra, I.; and van der Maaten, L. 2020. Self-Supervised Learning of Pretext-Invariant Representations. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 6706–6716. Computer Vision Foundation / IEEE.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543. ACL.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of Graph Neural Network Evaluation. *CoRR*, abs/1811.05868.

Sinha, A.; Shen, Z.; Song, Y.; Ma, H.; Eide, D.; Hsu, B. P.; and Wang, K. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of the 24th International Conference on World Wide Web Companion*.

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

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR*, abs/1807.03748.

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

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations*.

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

Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *Proceedings of the 33nd International Conference on Machine Learning*, volume 48, 40–48. JMLR.org.

Zhang, H.; Wu, Q.; Wang, Y.; Zhang, S.; Yan, J.; and Yu, P. S. 2022. Localized Contrastive Learning on Graphs. *CoRR*, abs/2212.04604.

Zheng, Y.; Pan, S.; Lee, V. C. S.; Zheng, Y.; and Yu, P. S. 2022. Rethinking and Scaling Up Graph Contrastive Learning: An Extremely Efficient Approach with Group Discrimination. In *Advances in Neural Information Processing Systems*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep Graph Contrastive Representation Learning. *CoRR*, abs/2006.04131.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *WWW '21: The Web Conference*, 2069–2080.