## Acknowledgments and Disclosure of Funding

We thank anonymous reviewers for their time and effort in reviewing this paper. This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62192784, 62172052, 62002029, U1936014).

## References

[1] Self-supervised learning on graphs: Contrastive, generative, or predictive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and deep locally connected networks on graphs. In *ICLR*, 2014.

[4] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *ICLR*, 2018.

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

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, 2016.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[8] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 1960.

[9] Alan Frieze and Michał Karoński. *Introduction to random graphs*. Cambridge University Press, 2016.

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[12] Kaveh Hassani and Amir Hosein Khas Ahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, 2020.

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

[14] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2018.

[15] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning. *arXiv e-prints*, 2021.

[16] Jian Kang, Yan Zhu, Yinglong Xia, Jiebo Luo, and Hanghang Tong. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. In *WWW*, 2022.

[17] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[18] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2018.

[19] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip Yu. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[20] Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. Tail-gnn: Tail-node graph neural networks. In *KDD*, 2021.

[21] Zemin Liu, Wentao Zhang, Yuan Fang, Xinming Zhang, and Steven CH Hoi. Towards locality-aware meta-learning of tail node embeddings on networks. In *CIKM*, 2020.

[22] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *KDD*, 2020.

[23] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *NeurIPS Workshop on Relational Representation Learning*, 2018.

[24] Fan-Yun Sun, Jordon Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2020.

[25] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Investigating and mitigating degree-related biases in graph convoltuional networks. In *CIKM*, 2020.

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

[27] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[28] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, 2018.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[30] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *ICLR*, 2019.

[31] Ruijia Wang, Shuai Mou, Xiao Wang, Wanpeng Xiao, Qi Ju, Chuan Shi, and Xing Xie. Graph structure estimation neural networks. In *WWW*, 2021.

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

[33] Jun Wu, Jingrui He, and Jiejun Xu. Net: Degree-specific graph neural networks for node and graph classification. In *KDD*, 2019.

[34] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[35] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.

[36] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. In *NeurIPS*, 2021.

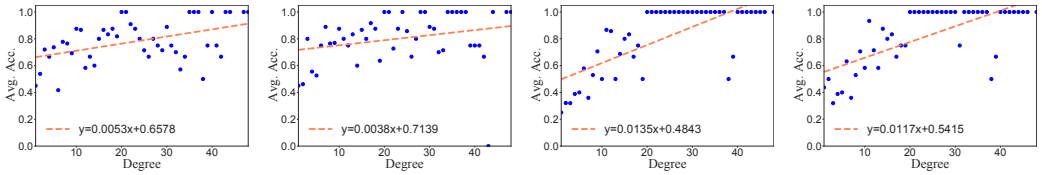
[37] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[38] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep Graph Contrastive Representation Learning. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

[39] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, 2021.

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3 and Appendix B.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 2, Section 5, Appendix A and C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 2 and Section 5.
   (b) Did you mention the license of the assets? [Yes] See Appendix C.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Appendix C.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix C.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]




(a) GCN on Photo (b) DGI on Photo (c) GCN on Computer (d) DGI on Computer

Figure 6: More results for the fairness of models to degree bias on Photo and Computer datasets.

# A Details of Section 2

## A.1 Implementation Details

We choose the commonly used Cora [17] , Citeseer [17] , Photo [23] and Computer [23] for evaluation. In Cora and Citeseer datasets, nodes represent papers, edges are the citation relationship between papers, node features comprise bag-of-words vector of keywords and labels represent the research field of papers. Photo and Computer datasets are segments of the Amazon co-purchase graph, where nodes represent products, edges indicate that two products are frequently bought together, node features are bag-of-words encoded product reviews and labels are given by the product category. The statistics of these datasets are summarized in Table 4. The above datasets are public and do not contain personally identifiable information and offensive content. The URL of our datasets is https://docs.dgl.ai/api/python/dgl.data.html and the license is Apache License 2.0.

We train DGI[2] [30] and GraphCL[3] [35] on these datasets with codes provided by authors. To compare with GCN[4] [17], our linear evaluation protocol deploys the semi-supervised split [17], where 20 labeled nodes per class form training set and test set composes of randomly sampled 1000 nodes with degree less than 50. GCN follows the standard training paradigm [17] with the above train-test split. All these methods are initialized as the corresponding papers and consist of two GCN layers, where their hyperparameters are carefully searched to achieve optimal performance on the test set.

## A.2 Additional Results

More comparison results between GCL methods and GCN on Photo and Computer datasets are shown in Figure 6. Please note that GraphCL has an out-of-memory issue on these datasets. From the figure, the gap between the slopes of DGI and GCN is relatively small. A reasonable hypothesis is that average node degrees of Photo and Computer datasets are much larger than those of Cora and Citeseer datasets, where the advantage of GCL to alleviate the neighborhood sparsity of tail nodes cannot be well exhibited.

# B Details of Section 3

**Theorem** 1 ***Intra-community Concentration.*** *Let pre-transformation representations $\tilde{L}X$ be sub-Gaussian random variable with variance $\sigma^2$. For all nodes $v_i \in S_\varepsilon$, if $\varepsilon^2 \leq \frac{\beta m}{6M^2\kappa}$, their representations $f(\mathcal{G}_i)$ fit sub-Gaussian distribution with variance $\sigma^2_{f,\varepsilon} \leq \frac{1}{\kappa}\sigma^2$.*

**Proof** For node $v_i$ in $S_\varepsilon$, we have $\|f(\mathcal{G}_i) - f(\hat{\mathcal{G}}_i)\|^2 \leq \varepsilon^2$. This implies that for nodes $v_i, v_j \in S_\varepsilon$ such that $\|\tilde{L}_i X - \tilde{L}_j X\|^2 \leq 2\beta m$, there exists a region of overlap so that $\|f(\mathcal{G}_i) - f(\mathcal{G}_j)\|^2 \leq \|f(\mathcal{G}_i) - f(\hat{\mathcal{G}}_i)\|^2 + \|f(\hat{\mathcal{G}}_i) - f(\mathcal{G}_j)\|^2 \leq 2\varepsilon^2$. That is, there are graph augmentations of $v_i$ which are sufficiently similar to graph augmentations of $v_j$ so that their representations should be similar, thereby driving $f(\mathcal{G}_i)$ and $f(\mathcal{G}_j)$ to be closer.

[2](MIT license) https://github.com/PetarV-/DGI
[3](MIT license) https://github.com/Shen-Lab/GraphCL
[4](MIT license) https://github.com/tkipf/pygcn

The variance of $\varepsilon$-close node representations in $f$ space is

$$\sigma_{f,\varepsilon}^2 = \frac{1}{2N^2(1-R_\varepsilon)^2} \sum_{v_i \in S_\varepsilon} \sum_{v_j \in S_\varepsilon} \|f(\mathcal{G}_i) - f(\mathcal{G}_j)\|^2. \tag{11}$$

The overlap $\beta m < \|\tilde{L}_i X - \tilde{L}_j X\|^2 \leq 2\beta m$ induces a graph where we say $v_j \in \mathcal{N}(i) \forall v_j$ s.t. $\|\tilde{L}_i X - \tilde{L}_j X\|^2 \leq 2\beta m$. For $N(1-R_\varepsilon)$ samples, we can decompose the variance as

$$\begin{aligned} \sigma_{f,\varepsilon}^2 &= \frac{1}{2N^2(1-R_\varepsilon)^2} \sum_{v_i \in S_\varepsilon} \sum_{v_j \in S_\varepsilon} \|f(\mathcal{G}_i) - f(\mathcal{G}_j)\|^2 \\ &= \frac{1}{2N^2(1-R_\varepsilon)^2} \sum_{v_i \in S_\varepsilon} \left( \sum_{v_j \in \mathcal{N}(i)} \|f(\mathcal{G}_i) - f(\mathcal{G}_j)\|^2 + \sum_{v_j' \notin \mathcal{N}(i)} \|f(\mathcal{G}_i) - f(\mathcal{G}_{j'})\|^2 \right). \end{aligned} \tag{12}$$

By the smoothness of $f$ we always have $\|f(\mathcal{G}_i) - f(\mathcal{G}_{j'})\|^2 \leq M^2 \|\tilde{L}_i X - \tilde{L}_{j'} X\|^2$. By the constraint we have that $\|f(\mathcal{G}_i) - f(\mathcal{G}_j)\|^2 \leq \frac{2\varepsilon^2 M^2}{\beta m} \|\tilde{L}_i X - \tilde{L}_j X\|^2 \forall v_j \in \mathcal{N}(i)$ and for $\eta = \frac{2\varepsilon^2 M^2}{\beta m} < 1$.

Assuming that there is a constant proportion $0 \leq \lambda \leq 1$ of nodes in the set $\mathcal{N}(i) \forall v_i \in S_\varepsilon$, thus this graph is an Erdős-Renyi graph. From Theorem 4, if $\lambda \geq \frac{c \log N}{N}$ for $c > 1$ then with high probability, there are no unconnected components in graph. Every node is reachable from any other nodes in a finite number of steps. We can then decompose nodes in the graph into adjacent ones and those which are reachable within a certain number of steps. Let the shortest path between any two nodes be at most $D$, then we obtain the following inequality

$$\begin{aligned} \sigma_{f,\varepsilon}^2 &= \frac{1}{2N^2(1-R_\varepsilon)^2} \sum_{v_i \in S_\varepsilon} \sum_{v_j \in S_\varepsilon} \|f(\mathcal{G}_i) - f(\mathcal{G}_j)\|^2 \\ &\leq \lambda \eta \sigma_x^2 + (1-\lambda) D \eta \sigma_x^2. \end{aligned} \tag{13}$$

From Theorem 5, we have $3 \leq D \leq 4$ with high probability. So for $\sigma_{f,\varepsilon}^2 \leq \frac{1}{\kappa}\sigma_x^2$ with $\kappa \geq 1$, we require $\varepsilon^2 \leq \frac{\beta m}{2M^2\kappa(3-2\lambda)} \leq \frac{\beta m}{6M^2\kappa}$. □

**Theorem 4** *[8] If $p = \frac{c \log N}{N}$ where $c > 1$ with high probability, then the graph $G(N,p)$ has no unconnected components.*

**Theorem 5** *[9] Let $q \geq 2$ be a fixed positive integer. For $c > 0$ and*

$$p^q N^{q-1} = \log(\frac{N^2}{c}). \tag{14}$$

*Then $diam(G_{N,p}) \geq q$ with probability $e^{-\frac{c}{2}}$ and $diam(G_{N,p}) \leq q+1$ with probability $1 - e^{-\frac{c}{2}}$.*

**Definition** 1 **($\alpha, \gamma, \hat{d}$)-Augmentation.** *The augmentation set $\mathcal{T}$ is a $(\alpha, \gamma, \hat{d})$-augmentation, if for each community $C_k$, there exists a subset $C_k^0 \subset C_k$ such that the following two conditions hold*

1. $\mathbb{P}[v_i \in C_k^0] \geq \alpha \mathbb{P}[v_i \in C_k]$ *where* $\alpha \in (0,1]$,
2. $\sup_{v_i, v_j \in C_k^0} d_{\mathcal{T}}(v_i, v_j) \leq \gamma (\frac{B}{\hat{d}_{\min}^k})^{\frac{1}{2}}$ *where* $\gamma \in (0,1]$,

*where $\hat{d}_{\min}^k = \min_{v_i \in C_k^0, \hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i)} \hat{d}_i$, and $B$ is the feature dimension.*

**Remark** Since the node feature $X$ has been mapped to surface of the unit sphere $\mathbb{S}^{B-1} = \{X_i \in \mathbb{R}^B : \|X_i\| = 1\}$, there is a natural supremum for $d_{\mathcal{T}}(v_i, v_j)$ bounded by the node degree,

$$\begin{aligned} d_{\mathcal{T}}(v_i, v_j) &= \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \|(\frac{\hat{A}_i}{\hat{d}_i} - \frac{\hat{A}_j}{\hat{d}_j})X\| \\ &\leq \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \|\frac{\hat{A}_i}{\hat{d}_i} - \frac{\hat{A}_j}{\hat{d}_j}\| \cdot \sqrt{B} \\ &= \sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \|\frac{\hat{A}_i}{\hat{d}_i} - \frac{\hat{A}_j}{\hat{d}_i} + \frac{\hat{A}_j}{\hat{d}_i} - \frac{\hat{A}_j}{\hat{d}_j}\| \\ &\leq \sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \frac{\|\hat{A}_i - \hat{A}_j\|}{\hat{d}_i} + \|\hat{A}_j\| \left| \frac{1}{\hat{d}_i} - \frac{1}{\hat{d}_j} \right|. \end{aligned} \tag{15}$$

Without loss of generality, we assume that $\hat{d}_i \geq \hat{d}_j$

$$\begin{aligned} d_{\mathcal{T}}(v_i, v_j) &= \sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \frac{\|\hat{A}_i - \hat{A}_j\|}{\hat{d}_i} + \|\hat{A}_j\| \left( \frac{1}{\hat{d}_j} - \frac{1}{\hat{d}_i} \right) \\ &\leq \sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \frac{\sqrt{\hat{d}_i + \hat{d}_j}}{\hat{d}_i} + \sqrt{\hat{d}_j} \left( \frac{1}{\hat{d}_j} - \frac{1}{\hat{d}_i} \right) \\ &= \sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \frac{\sqrt{\hat{d}_i + \hat{d}_j} - \sqrt{\hat{d}_j}}{\hat{d}_i} + \frac{1}{\sqrt{\hat{d}_j}} \\ &\leq \sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \frac{1}{\sqrt{\hat{d}_i}} + \frac{1}{\sqrt{\hat{d}_j}} \\ &\leq 2\sqrt{B} \cdot \min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \frac{1}{\sqrt{\hat{d}_j}}. \end{aligned} \tag{16}$$

Following this form, we define the RHS of the second condition to delineate the concentrated part.

**Lemma** 1 *For a $(\alpha, \gamma, \hat{d})$-augmentation with subset $C_k^0$ of each community $C_k$, if nodes belonging to $(C_1^0 \cup \cdots \cup C_K^0) \cap S_\varepsilon$ can be correctly assigned by the community indicator $F_f$, then the error of all nodes can be bounded by $(1-\alpha) + R_\varepsilon$, where $R_\varepsilon = \mathbb{P}[\overline{S_\varepsilon}]$ is the proportion of complement.*

**Proof** Since every node $v_i \in (C_1^0 \cup \cdots \cup C_K^0) \cap S_\varepsilon$ can be correctly assigned by $F_f$, the error rate

$$\begin{aligned} \mathrm{Err}(F_f) &= \sum_{k=1}^{K} \mathbb{P}[F_f(\mathcal{G}_i) \neq k, \forall v_i \in C_k] \\ &\leq \mathbb{P}\left[\overline{(C_1^0 \cup \cdots \cup C_K^0) \cap S_\varepsilon}\right] \\ &= \mathbb{P}\left[\overline{C_1^0 \cup \cdots \cup C_K^0} \cup \overline{S_\varepsilon}\right] \\ &\leq (1-\alpha) + \mathbb{P}[\overline{S_\varepsilon}] \\ &= (1-\alpha) + R_\varepsilon. \end{aligned} \tag{17}$$

□

**Lemma** 2 *For a $(\alpha, \gamma, \hat{d})$-augmentation and each $\ell \in [K]$, if*

$$\mu_\ell^\top \mu_k < r^2(1 - \rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon) - \sqrt{2\rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)} - \frac{\Delta_\mu}{2})$$

*holds for all $k \neq \ell$, then every node $v_i \in C_\ell^0 \cap S_\varepsilon$ can be correctly assigned by the community indicator $F_f$, where $\rho_\ell(\alpha, \delta, \varepsilon) = 2(1-\alpha) + \frac{2R_\varepsilon}{p_\ell} + \alpha(\frac{M\gamma\sqrt{B}}{r\sqrt{\hat{d}_{\min}^\ell}} + \frac{2\varepsilon}{r})$ and $\Delta_\mu = 1 - \min_{k \in [K]} \|\mu_k\|^2 / r^2$.*

**Proof** To show that every node $v_i \in C_\ell^0 \cap S_\varepsilon$ can be correctly assigned by $F_f$, we need to prove that for all $k \neq \ell$, $\|f(\mathcal{G}_i) - \mu_\ell\| < \|f(\mathcal{G}_i) - \mu_k\|$. It is equivalent to prove

$$f(\mathcal{G}_i)^\top \mu_\ell - f(\mathcal{G}_i)^\top \mu_k - (\frac{1}{2}\|\mu_\ell\|^2 - \frac{1}{2}\|\mu_k\|^2) > 0. \tag{18}$$

Let $\tilde{f}(\mathcal{G}_i) = \mathbb{E}_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i)}[f(\hat{\mathcal{G}}_i)]$. Then $\|\tilde{f}(\mathcal{G}_i)\| = \|\mathbb{E}_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i)}[f(\hat{\mathcal{G}}_i)]\| \leq \mathbb{E}_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i)}[\|f(\hat{\mathcal{G}}_i)\|] = r$.

One the one hand,

$$\begin{aligned}
f(\mathcal{G}_i)^\top \mu_\ell &= \frac{1}{p_\ell} f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j}[\tilde{f}(\mathcal{G}_j)\mathbb{I}(v_j \in C_\ell)] \\
&= \frac{1}{p_\ell} f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j}[\tilde{f}(\mathcal{G}_j)\mathbb{I}(v_j \in C_\ell \cap C_\ell^0 \cap S_\varepsilon)] + \frac{1}{p_\ell} f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j}\left[\tilde{f}(\mathcal{G}_j)\mathbb{I}\left(v_j \in C_\ell \cap \overline{C_\ell^0 \cap S_\varepsilon}\right)\right] \\
&= \frac{\mathbb{P}[C_\ell^0 \cap S_\varepsilon]}{p_\ell} f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon}[\tilde{f}(\mathcal{G}_j)] + \frac{1}{p_\ell}\mathop{\mathbb{E}}_{v_j}\left[f(\mathcal{G}_i)^\top \tilde{f}(\mathcal{G}_j) \cdot \mathbb{I}(v_j \in C_\ell \backslash C_\ell^0 \cap S_\varepsilon)\right] \\
&\geq \frac{\mathbb{P}[C_\ell^0 \cap S_\varepsilon]}{p_\ell} f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon}[\tilde{f}(\mathcal{G}_j)] - \frac{r^2}{p_\ell}\mathbb{P}[C_\ell \backslash C_\ell^0 \cap S_\varepsilon],
\end{aligned} \tag{19}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Note that

$$\mathbb{P}[C_\ell \backslash C_\ell^0 \cap S_\varepsilon] \leq \mathbb{P}[(C_\ell \backslash C_\ell^0) \cup \overline{S_\varepsilon}] = (1-\alpha)p_\ell + R_\varepsilon, \tag{20}$$

and

$$\mathbb{P}[C_\ell^0 \cap S_\varepsilon] = \mathbb{P}[C_\ell] - \mathbb{P}[C_\ell \backslash C_\ell^0 \cap S_\varepsilon] \geq p_\ell - ((1-\alpha)p_\ell + R_\varepsilon) = \alpha p_\ell - R_\varepsilon. \tag{21}$$

Plugging to Eq. (19), we have

$$\begin{aligned}
f(\mathcal{G}_i)^\top \mu_\ell &\geq \frac{\mathbb{P}[C_\ell^0 \cap S_\varepsilon]}{p_\ell} f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon}[\tilde{f}(\mathcal{G}_j)] - \frac{r^2}{p_\ell}\mathbb{P}[C_\ell \backslash C_\ell^0 \cap S_\varepsilon] \\
&\geq \left(\alpha - \frac{R_\varepsilon}{p_\ell}\right) f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon}[\tilde{f}(\mathcal{G}_j)] - r^2\left(1 - \alpha + \frac{R_\varepsilon}{p_\ell}\right).
\end{aligned} \tag{22}$$

Notice that $v_i \in C_\ell^0 \cap S_\varepsilon$. For any $v_j \in C_\ell^0 \cap S_\varepsilon$, we have $d_{\mathcal{T}}(v_i, v_j) \leq \gamma(\frac{B}{\hat{d}_{\min}^\ell})^{\frac{1}{2}}$. Let $(\hat{\mathcal{G}}_i^*, \hat{\mathcal{G}}_j^*) = \arg\min_{\hat{\mathcal{G}}_i \in \mathcal{T}(\mathcal{G}_i), \hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)} \|f(\hat{\mathcal{G}}_i) - f(\hat{\mathcal{G}}_j)\|$, thus $\|f(\hat{\mathcal{G}}_i^*) - f(\hat{\mathcal{G}}_j^*)\| \leq M\gamma(\frac{B}{\hat{d}_{\min}^\ell})^{\frac{1}{2}}$. Since $v_j \in S_\varepsilon$, for any $\hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)$, $\|f(\hat{\mathcal{G}}_j) - f(\hat{\mathcal{G}}_j^*)\| \leq \varepsilon$. Similarly, since $v_i \in S_\varepsilon$, we have $\|f(\hat{\mathcal{G}}_i) - f(\hat{\mathcal{G}}_i^*)\| \leq \varepsilon$. The first term of Eq. (22) can be bounded by

$$\begin{aligned}
f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon}[\tilde{f}(\mathcal{G}_j)] &= \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon} \mathop{\mathbb{E}}_{\hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)}[f(\mathcal{G}_i)^\top f(\hat{\mathcal{G}}_j)] \\
&= \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon} \mathop{\mathbb{E}}_{\hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)}[f(\mathcal{G}_i)^\top (f(\hat{\mathcal{G}}_j) - f(\mathcal{G}_i) + f(\mathcal{G}_i))] \\
&\geq r^2 + \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon} \mathop{\mathbb{E}}_{\hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)}[f(\mathcal{G}_i)^\top (f(\hat{\mathcal{G}}_j) - f(\mathcal{G}_i))] \\
&= r^2 + \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon} \mathop{\mathbb{E}}_{\hat{\mathcal{G}}_j \in \mathcal{T}(\mathcal{G}_j)}[f(\mathcal{G}_i)^\top (\underbrace{f(\hat{\mathcal{G}}_j) - f(\hat{\mathcal{G}}_j^*)}_{\|\cdot\| \leq \varepsilon} + \underbrace{f(\hat{\mathcal{G}}_j^*) - f(\hat{\mathcal{G}}_i^*)}_{\|\cdot\| \leq M\gamma(\frac{B}{\hat{d}_{\min}^\ell})^{\frac{1}{2}}} \\
&\quad + \underbrace{f(\hat{\mathcal{G}}_i^*) - f(\hat{\mathcal{G}}_i)}_{\|\cdot\| \leq \varepsilon})] \\
&= r^2 - r(M\gamma(\frac{B}{\hat{d}_{\min}^\ell})^{\frac{1}{2}} + 2\varepsilon).
\end{aligned} \tag{23}$$

Therefore, Eq. (22) turns to

$$\begin{aligned} f(\mathcal{G}_i)^\top \mu_\ell &\geq \left(\alpha - \frac{R_\varepsilon}{p_\ell}\right) f(\mathcal{G}_i)^\top \mathop{\mathbb{E}}_{v_j \in C_\ell^0 \cap S_\varepsilon} [\tilde{f}(\mathcal{G}_j)] - r^2 \left(1 - \alpha + \frac{R_\varepsilon}{p_\ell}\right) \\ &\geq \left(\alpha - \frac{R_\varepsilon}{p_\ell}\right) (r^2 - r(M\gamma(\frac{B}{\hat{d}_{\min}^\ell})^{\frac{1}{2}} + 2\varepsilon)) - r^2 \left(1 - \alpha + \frac{R_\varepsilon}{p_\ell}\right) \\ &= r^2 \left(1 - 2(1-\alpha) - \frac{2R_\varepsilon}{p_\ell} - \left(\alpha - \frac{R_\varepsilon}{p_\ell}\right) \left(\frac{M\gamma\sqrt{B}}{r\sqrt{\hat{d}_{\min}^\ell}} + \frac{2\varepsilon}{r}\right)\right) \\ &= r^2(1 - \rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)). \end{aligned} \tag{24}$$

On the other hand,

$$\begin{aligned} f(\mathcal{G}_i)^\top \mu_k &= (f(\mathcal{G}_i) - \mu_\ell)^\top \mu_k + \mu_\ell^\top \mu_k \\ &\leq \|f(\mathcal{G}_i) - \mu_\ell\| \cdot \|\mu_k\| + \mu_\ell^\top \mu_k \\ &\leq r\sqrt{\|f(\mathcal{G}_i)\|^2 - 2f(\mathcal{G}_i)^\top \mu_\ell + \|\mu_\ell\|^2} + \mu_\ell^\top \mu_k \\ &\leq r\sqrt{2r^2 - 2f(\mathcal{G}_i)^\top \mu_\ell} + \mu_\ell^\top \mu_k \\ &\leq \sqrt{2\rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)} r^2 + \mu_\ell^\top \mu_k. \end{aligned} \tag{25}$$

Note that $\Delta_\mu = 1 - \min_k \|\mu_k\|^2/r^2$, the LHS of Eq. (18) is

$$\begin{aligned} f(\mathcal{G}_i)^\top \mu_\ell - f(\mathcal{G}_i)^\top \mu_k - (\frac{1}{2}\|\mu_\ell\|^2 - \frac{1}{2}\|\mu_k\|^2) &\geq f(\mathcal{G}_i)^\top \mu_\ell - f(\mathcal{G}_i)^\top \mu_k - \frac{1}{2} r^2 \Delta_\mu \\ &\geq r^2(1 - \rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)) - \sqrt{2\rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)} r^2 - \mu_\ell^\top \mu_k - \frac{1}{2} r^2 \Delta_\mu \\ &= r^2 \left(1 - \rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon) - \sqrt{2\rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)} - \frac{1}{2}\Delta_\mu\right) - \mu_\ell^\top \mu_k > 0. \end{aligned} \tag{26}$$

□

**Theorem** 2 ***Inter-community Scatter.*** *For a $(\alpha, \gamma, \hat{d})$-augmentation, if*

$$\mu_\ell^\top \mu_k < r^2(1 - \rho_{\max}(\alpha, \gamma, \hat{d}, \varepsilon) - \sqrt{2\rho_{\max}(\alpha, \gamma, \hat{d}, \varepsilon)} - \frac{\Delta_\mu}{2}) \tag{27}$$

*holds for any pair of $(\ell, k)$ with $\ell \neq k$, then the error of the community indicator $F_f$ can be bounded by $(1-\alpha) + R_\varepsilon$, where $\rho_{\max}(\alpha, \gamma, \hat{d}, \varepsilon) = 2(1-\alpha) + \max_\ell \left(\frac{2R_\varepsilon}{p_\ell} + \frac{M\alpha\gamma\sqrt{B}}{r\sqrt{\hat{d}_{\min}^\ell}}\right) + \frac{2\alpha\varepsilon}{r}$) and $\Delta_\mu = 1 - \min_{k\in[K]} \|\mu_k\|^2/r^2$.*

**Proof** Since the augmentation $\mathcal{T}$ is $(\alpha, \gamma, \hat{d})$-augmentation, there exists a subset $C_k^0$ for each community $C_k$ such that $\mathbb{P}[C_k^0] \geq \alpha p_k$ and $\sup_{v_i, v_j \in C_k^0} d_{\mathcal{T}}(v_i, v_j) \leq \gamma(\frac{B}{\hat{d}_{\min}^k})^{\frac{1}{2}}$. Since for any $\ell \neq k$, we have $\mu_\ell^\top \mu_k < r^2(1 - \rho_{\max}(\alpha, \gamma, \hat{d}, \varepsilon) - \sqrt{2\rho_{\max}(\alpha, \gamma, \hat{d}, \varepsilon)} - \frac{\Delta_\mu}{2}) \leq r^2(1 - \rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon) - \sqrt{2\rho_\ell(\alpha, \gamma, \hat{d}, \varepsilon)} - \frac{\Delta_\mu}{2})$. According to Lemma 2, every node $v_i \in C_\ell^0 \cap S_\varepsilon$ can be correctly assigned by $F_f$. Therefore, every node $v_i \in (C_1^0 \cup \cdots \cup C_K^0) \cap S_\varepsilon$ can be correctly assigned by $F_f$. According to Lemma 1, the error rate $\mathrm{Err}(F_f) \leq (1-\alpha) + R_\varepsilon$. □

**Theorem** 3 *The term $R_\varepsilon$ is upper bounded by*

$$R_\varepsilon \leq \frac{[C(N-1, m)]^2}{\varepsilon} \mathbb{E}_{v_i} \mathbb{E}_{\hat{\mathcal{G}}_i^1, \hat{\mathcal{G}}_i^2 \in \mathcal{T}(\mathcal{G}_i)} \|f(\hat{\mathcal{G}}_i^1) - f(\hat{\mathcal{G}}_i^2)\|. \tag{28}$$

**Proof** For any given node $v_i$, we have

$$\sup_{\hat{\mathcal{G}}_i^1, \hat{\mathcal{G}}_i^2 \in \mathcal{T}(\mathcal{G}_i)} \|f(\hat{\mathcal{G}}_i^1) - f(\hat{\mathcal{G}}_i^2)\| \geq [C(N-1, m)]^2 \mathop{\mathbb{E}}_{\hat{\mathcal{G}}_i^1, \hat{\mathcal{G}}_i^2 \in \mathcal{T}(\mathcal{G}_i)} \|f(\hat{\mathcal{G}}_i^1) - f(\hat{\mathcal{G}}_i^2)\|. \tag{29}$$

Table 4: Statistics of datasets.

| Dataset | # Nodes | # Edges | # Features | # Classes | # Avg. Degree |
|---|---|---|---|---|---|
| **Cora** | 2,708 | 10,556 | 1,433 | 7 | 3.89 |
| **Citeseer** | 3,327 | 9,228 | 3,703 | 6 | 2.77 |
| **Photo** | 7,650 | 238,163 | 745 | 8 | 31.13 |
| **Computer** | 13,752 | 491,722 | 767 | 10 | 35.75 |

Therefore, the following set $S$ is a subset of $S_\varepsilon$,

$$S = \left\{ v_i : \mathop{\mathbb{E}}_{\hat{\mathcal{G}}_i^1, \hat{\mathcal{G}}_i^2 \in \mathcal{T}(\mathcal{G}_i)} \|f(\hat{\mathcal{G}}_i^1) - f(\hat{\mathcal{G}}_i^2)\| \leq \frac{\varepsilon}{[C(N-1,m)]^2} \right\} \subseteq S_\varepsilon. \tag{30}$$

By Markov's inequality, we have

$$\begin{aligned} R_\varepsilon = \mathbb{P}[\overline{S_\varepsilon}] &\leq \mathbb{P}[\bar{S}] \\ &\leq \frac{\mathbb{E}_{v_i} \mathbb{E}_{\hat{\mathcal{G}}_i^1, \hat{\mathcal{G}}_i^2 \in \mathcal{T}(\mathcal{G}_i)} \|f(\hat{\mathcal{G}}_i^1) - f(\hat{\mathcal{G}}_i^2)\|}{\frac{\varepsilon}{[C(N-1,m)]^2}} \\ &= \frac{[C(N-1,m)]^2}{\varepsilon} \mathbb{E}_{v_i} \mathbb{E}_{\hat{\mathcal{G}}_i^1, \hat{\mathcal{G}}_i^2 \in \mathcal{T}(\mathcal{G}_i)} \|f(\hat{\mathcal{G}}_i^1) - f(\hat{\mathcal{G}}_i^2)\|. \end{aligned} \tag{31}$$

□

# C Details of Section 5

**Baselines** We compare GRADE with state-of-the-art GCL models DGI [30], GraphCL [35], GRACE[5] [38], MVGRL[6] [12] and CCA-SSG[7] [36] and semi-supervised GCN [17] with their original codes. For GCL models, we follow the linear evaluation scheme introduced in [30], where each model is firstly trained in an unsupervised manner and node representations are subsequently fed into a simple logistic regression classifier. We adopt two universally accepted splits for full evaluation: 1) semi-supervised split [30, 35] that 20 labeled nodes per class are for training and 1000 nodes are for testing, 2) supervised split [38, 36] that 1000 nodes are for testing and the rest of nodes form the training set. It is worth noting that 1000 nodes in the test set are randomly sampled with degrees less than 50 to provide an appropriate degree range for analysis. GCN is trained by the original paradigm [17] with the above train-test split. All these methods are initialized as the corresponding papers and consist of two GCN layers, where their hyperparameters are carefully searched to achieve optimal performance on the test set.

**Implementation** For GRADE, we also utilize two GCN layers as the encoder. For hyperparameter settings, we vary the temperature $\tau$ in range $[0.5, 2]$, and the threshold $\zeta$ is searched in $[5, 15]$. The edge drop rate $p_{edr}$ and feature drop rate $p_{fdr}$ are tested in $[0.1, 0.4]$. We randomly initialize model parameters and use the Adam optimizer. Additionally, we employ random augmentation as a warmup since our graph augmentation relies on the quality of node representations. The number of epochs for a warmup is 200. The environment where we run experiments is:

- Operating system: Linux version 3.10.0-693.el7.x86_64
- CPU information: Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz
- GeForce RTX 3090

[5](MIT license) https://github.com/CRIPAC-DIG/GRACE
[6](MIT license) https://github.com/kavehhassani/mvgrl
[7](MIT license) https://github.com/hengruizhang98/CCA-SSG