Perfect Alignment May be Poisonous to Graph Contrastive Learning

Jingyu Liu [1] **Huayi Tang** [1] **Yong Liu** [1 2]

Abstract

Graph Contrastive Learning (GCL) aims to learn node representations by aligning positive pairs and separating negative ones. However, few of researchers have focused on the inner law behind specific augmentations used in graph-based learning. What kind of augmentation will help downstream performance, how does contrastive learning actually influence downstream tasks, and why the magnitude of augmentation matters so much? This paper seeks to address these questions by establishing a connection between augmentation and downstream performance. Our findings reveal that GCL contributes to downstream tasks mainly by separating different classes rather than gathering nodes of the same class. So perfect alignment and augmentation overlap which draw all intra-class samples the same can not fully explain the success of contrastive learning. Therefore, in order to understand how augmentation aids the contrastive learning process, we conduct further investigations into the generalization, finding that perfect alignment that draw positive pair the same could help contrastive loss but is poisonous to generalization, as a result, perfect alignment may not lead to best downstream performance, so specifically designed augmentation is needed to achieve appropriate alignment performance and improve downstream accuracy. We further analyse the result by information theory and graph spectrum theory and propose two simple but effective methods to verify the theories. The two methods could be easily applied to various GCL algorithms and extensive experiments are conducted to prove its effectiveness. The code is available at https://github.com/somebodyhh1/GRACEIS

[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China [2]Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China. Correspondence to: Yong Liu <liuyonggsai@ruc.edu.cn>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Graph Neural Networks (GNNs) have been successfully applied in various fields (Yang et al., 2022a) such as recommendation systems (He et al., 2020), drug discovery (Liu et al., 2018), and traffic analysis (Wu et al., 2019), etc (Yang et al., 2023; 2024). However, most GNNs require labeled data for training, which may not always be available. To address this issue, Graph Contrastive Learning (GCL), which does not rely on labels, has gained popularity as a new approach to graph representation learning (Veličković et al., 2018; You et al., 2020).

GCL often generates new graph views through data augmentation (Chen et al., 2020; Zhu et al., 2020; Jia & Zhang, 2022; Mumuni et al., 2024). GCL considers nodes augmented from the same as positive samples and others as negative samples. Subsequently, the model try to maximize similarity between positive samples and minimize similarity between negative ones (Wang & Isola, 2020; Hassani & Khasahmadi, 2020) to learn a better representation. So, data augmentation plays a vital role in graph contrastive learning, and data augmentation can be categorized into three types (Zhao et al., 2022): random augmentation (Veličković et al., 2018; Zhu et al., 2020), rule-based augmentation (Zhu et al., 2021; Wei et al., 2023; Liu et al., 2022), and learning-based augmentation (Suresh et al., 2021; Jiang et al., 2019). For instance, GRACE (Zhu et al., 2020) randomly masks node attributes and edges in graph data to obtain augmented graphs; GCA (Zhu et al., 2021) uses node degree to measure its importance and mask those unimportant with higher probability; And AD-GCL (Suresh et al., 2021) uses a model to learn the best augmentation and remove irrelevant information as much as possible. However, most data augmentation algorithms are designed heuristically, and there is a lack of theoretical analysis on how these methods will influence the downstream performance.

Some researchers have explored the generalization ability of contrastive learning (Arora et al., 2019; Wang & Isola, 2020; Huang et al., 2021). They propose that contrastive learning works by gathering positive pairs and separating negative samples uniformly. Wang et al. (2022b) argues that perfect alignment and uniformity alone cannot guarantee optimal performance. They propose that through stronger augmentation, there will be support overlap between dif-

ferent intra-class samples, which is called augmentation overlap (Saunshi et al., 2022; Huang et al., 2021). The augmentation overlap between two nodes mean that their corresponding augmented node could be the same, in this way aligning the anchor node with the the augmented node could also align two anchor nodes. Thus, the alignment of positive samples will also cluster all the intra-class samples together. And due to the limited inter-class overlap, inter-class nodes will not be gathered. However, Saunshi et al. (2022) points out that augmentation overlap may be relatively rare despite the excellent performance of contrastive learning methods. Hence, chances are that the success of contrastive learning cannot be solely attributed to alignment and augmentation overlap. It is of vital importance to figure out how augmentation works in the contrastive learning process, why the magnitude of augmentation matters so much and how to perform better augmentation. As data augmentation on graphs could be more customized and the magnitude of augmentation can be clearly represented by the number of modified edges/nodes (You et al., 2020), we mainly study the augmentation on graphs.

In this paper, we provide a new understanding of Graph Contrastive Learning and use a theoretical approach to analyze the impact of augmentation on contrastive learning process. We find that with a stronger augmentation, the model is performing better mainly because of inter-class separating rather than intra-class gathering brought by augmentation overlap. This aligns with the finding that augmentation overlap is actually quite rare in contrastive learning (Saunshi et al., 2022). Also, (Wang et al., 2022b) proposes that a stronger augmentation could help because of more augmentation overlap, then more intra-class nodes are gathered due to alignment. However, stronger augmentation naturally conflicts with better alignment, so does stronger augmentation helps intra-class gathering remains questionable. Moreover, stronger augmentation leads to better performance while the alignment is weaken, so we also question that does perfect alignment actually helps contrastive learning?

To further analyze the phenomena, we formulate a relationship between downstream accuracy, contrastive learning loss, and alignment performance, find that weak alignment performance caused by stronger augmentation can benefit the generalization. This explains why stronger augmentation will lead to better performance and reveals that perfect alignment is not the key to success, it may help to decrease contrastive loss, but also enlarge the gap between contrastive learning and downstream task, so specifically designed augmentation strategy is needed to achieve appropriate alignment and get the best downstream accuracy. This is why augmentation matters so much in contrastive learning.

Then, aiming to achieve better downstream accuracy, we need to figure out how to perform augmentation to achieve a better balance between contrastive loss and generalization. Therefore, we further analyze the contrastive process through information theory and graph spectrum theory. From the information theory perspective, we find augmentation should be stronger while keeping enough information, which is widely adopted explicitly or implicitly by designed algorithms (Zhu et al., 2021; 2020; Suresh et al., 2021). From the graph spectrum theory perspective, we analyze how the graph spectrum will affect the contrastive loss and generalization (Liu et al., 2022), finding that non-smooth spectral distribution will have a negative impact on generalization. Then we propose two methods based on the theories to verify our findings.

Our main contributions are as listed follows. (1) We reveal that when stronger augmentation is applied, contrastive learning benefits from inter-class separating more than intra-class gathering, and better alignment may not be helping as it conflicts with stronger augmentation. (2) We establish the relationship between downstream performance, contrastive learning loss, and alignment performance. Finds that better alignment would weaken the generalization, showing that why stronger augmentation helps, then we analyze the result from information theory and graph spectrum theory. (3) Based on the proposed theoretical results, we provide two simple but effective algorithms to verify the correctness of the theory. We also show that these algorithms can be extended to various contrastive learning methods to enhance their performance. (4) Extensive experiments are conducted on different contrastive learning algorithms and datasets using our proposed methods to demonstrate its effectiveness and verify our theory.

2. Augmentation and Generalization

2.1. Preliminaries

A graph can be represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the edge set. The feature matrix and the adjacency matrix are denoted as $\boldsymbol{X} \in \mathbb{R}^{N\times F}$ and $\boldsymbol{A} \in \{0,1\}^{N\times N}$, where F is the dimension of input feature, $\boldsymbol{x}_i \in \mathbb{R}^F$ is the feature of node v_i and $\boldsymbol{A}_{i,j} = 1$ iff $(v_i, v_j) \in \mathcal{E}$. The node degree matrix $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, ..., d_N)$, where d_i is the degree of node v_i.

In contrastive learning, data augmentation is used to create new graphs $\mathcal{G}^1, \mathcal{G}^2 \in \mathbb{G}^{\text{aug}}$, and the corresponding nodes, edges, and adjacency matrices are denoted as $\mathcal{V}^1, \mathcal{E}^1, \boldsymbol{A}^1, \mathcal{V}^2, \mathcal{E}^2, \boldsymbol{A}^2$. In the following of the paper, v is used to represent all nodes including the original nodes and the augmented nodes; v_i^+ is used to represent the augmented nodes including both v_i^1 and v_i^2; v_i^0 represents the original nodes only.

Nodes augmented from the same one, such as (v_i^1, v_i^2), are considered as positive pairs, while others are considered as

negative pairs. It is worth noting that a negative pair could come from the same graph, for node v_i^1, its negative pair could be $v_i^- \in \{v_j^+ | j \neq i\}$. Graph Contrastive Learning (GCL) is a method to learn an encoder that draws the embeddings of positive pairs similar and makes negative ones dissimilar (Chen et al., 2020; Wang & Isola, 2020). The encoder calculates the embedding of node v_i by $f(v_i)$, and we assume that $||f(v_i)|| = 1$.

2.2. How Does Augmentation Affect Downstream Performance

Previous work (Wang & Isola, 2020) proposes that effective contrastive learning should satisfy alignment and uniformity, meaning that positive samples should have similar embeddings, *i.e.,* $f(v_i^1) \approx f(v_i^2)$, and features should be uniformly distributed in the unit hypersphere. However, Wang et al. (2022b) pointed out that perfect alignment and uniformity does not ensure great performance. For instance, when $\{f(v_i^0)\}_{i=1}^N$ are uniformly distributed and $f(v_i^0) = f(v_i^+)$, there is a chance that the model may converge to a trivial solution that only projects very similar features to the same embedding, and projects other features randomly, then it will perform random classification in downstream tasks although it achieves perfect alignment and uniformity.

Wang et al. (2022b) argues that perfect alignment and intra-class augmentation overlap would be the best solution. The augmentation overlap means support overlap between different intra-class samples, and stronger augmentation is likely to bring more augmentation overlap. If two intra-class samples have augmentation overlap, then the best solution is projecting the two samples and their augmentation to the same embedding, which is called perfect alignment. For example, if two different nodes v_i^0, v_j^0 get the same augmentation v^+, then the best solution to contrative learning is $f(v_i^0) = f(v^+) = f(v_j^0)$. As the intra-class nodes are naturally closer, augmentation overlap often occurs between intra-class nodes, so perfect alignment and augmentation overlap could help intra-class gathering.

However, Saunshi et al. (2022) proposes that augmentation overlap is actually quite rare in practice, even with strong augmentation. Also augmentation overlap requires for strong augmentation, which makes alignment harder and conflicts with perfect alignment, so the success of contrastive learning may not be contributed to intra-class gathering brought by augmentation overlap. Therefore, it is important to understand how is contrastive learning working, and why stronger augmentation helps. More related work are introduced in Appendix E.

To begin with, we give an assumption on the label consistency between positive samples, which means the class label does not change after augmentation.

Assumption 2.1 (View Invariance)**.** For node v_i^0, the corresponding augmentation nodes v_i^+ get consistent labels, *i.e.,* we assume the labels are deterministic (one-hot) and consistent: $p(y|v_i^0) = p(y|v_i^+)$.

This assumption is widely adopted (Arora et al., 2019; Wang et al., 2022b; Saunshi et al., 2022) and reasonable. If the augmentation still keeps the basic structure and most of feature information is kept, the class label would not likely to change. Else if the augmentation destroys basic label information, the model tends to learn a trivial solution, so it is meaningless and we do not discuss the situation. The graph data keeps great label consistency under strong augmentation as discussed in Appendix C.3.

To further understand how is data augmentation working in contrastive learning, we use graph edit distance (GED) to denote the magnitude of augmentation, Trivedi et al. (2022) proposes that all allowable augmentations can be expressed using GED which is defined as minimum cost of graph edition (node insertion, node deletion, edge deletion, feature transformation) transforming graph $\mathcal{G}^0$ to $\mathcal{G}^+$. So a stronger augmentation could be defined as augmentation with a larger graph edit distance.

Assumption 2.2 (Augmentation Distance and Augmentation)**.** While Assumption 2.1 holds *i.e.,* $p(y|v_i^0) = p(y|v_i^+)$, as the augmentation getting stronger, the augmentation distance $\delta_{aug}^2 = \mathbb{E}_{p(v_i^0, v_i^+)}||f(v_i^0) - f(v_i^+)||^2$ will increase, *i.e.,* $\delta_{aug} \propto \text{GED}(\mathcal{G}^0, \mathcal{G}^+)$. $\text{GED}(\mathcal{G}^0, \mathcal{G}^+)$ indicates the graph edit distance between $\mathcal{G}^0$ and $\mathcal{G}^+$.

This is a natural assumption that is likely to hold because a bigger difference in input will lead to a bigger difference in output. Also we can notice that δ_{aug} is actually the distance between the anchor node and the augmented node, so δ_{aug} could naturally represent the alignment performance, a smaller δ_{aug} means a better alignment. Then Assumption 2.2 means that stronger augmentation would lead to larger δ_{aug}, *i.e.,* worse alignment. This phenomena is common in real practice as shown in Appendix C.3.

Definition 2.3. The class center is calculated by the expectation of all nodes belongs to the class, *i.e.,* $\mu_y = \mathbb{E}_{p(v,y)}[f(v_y)]$. We use δ_{y^+} and δ_{y^-} to represent intra-class and inter-class divergence respectively, and

$$\delta_{y^+}^2 = \mathbb{E}_{p(y,i,j)}||f(v_{y,i}^0) - f(v_{y,j}^0)||^2,$$
$$\delta_{y^-}^2 = \mathbb{E}_{p(y,y-,i,j)}||f(v_{y,i}^0) - f(v_{y^-,j}^0)||^2,$$

where y^- stands for a class different from y.

Note that we calculate the class center μ_y by averaging nodes from both original view and augmented views. As the augmentation on graphs are highly biased (Zhang et al., 2022), *i.e.,* the mean of augmented nodes are different from the original node, so the class center tends to be different. Also contrastive learning actually learns embedding on the

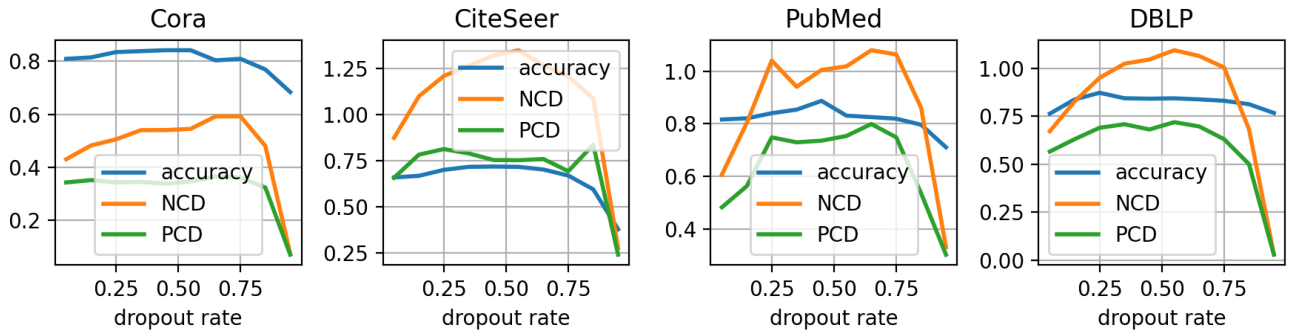

Figure 1. PCD means positive center distance ($\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_y||$), NCD means negative center distance ($\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_{y^-}||$) and accuracy is the downstream performance. X-axis stands for dropout rate of both edge and feature.

augmented view, so the class gathering result is largely affected by the augmentation, so it is more appropriate to include the augmented nodes when calculate the class center.

With the assumptions, we can get the theorem below:

Theorem 2.4 (Augmentation and Classification)**.** *If Assumption 2.1 holds, we know that:*

$$\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_y|| \leq \delta_{y^+} + \frac{2}{3}\delta_{aug}, \tag{1}$$

$$\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_{y^-}|| \leq \delta_{y^-} + \frac{2}{3}\delta_{aug}, \tag{2}$$

The proof can be found in Appendix A.1. This shows that the distance between a node and the class center could be represented by the augmentation distance δ_{aug} and the inter-class/intra-class divergence δ_{y^-}, δ_{y^+}. We then use positive and negative center distance to represent $\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_y||$ and $\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_{y^-}||$, respectively.

As we assumed in Assumption 2.2, when the augmentation becomes stronger, augmentation distance *i.e.,* δ_{aug} would increase. Also we notice that both positive and negative center distance are positively related to the magnitude of augmentation δ_{aug}. Therefore, stronger augmentation separates both inter-class and intra-class nodes, *i.e.,* it helps inter-class separating and hinders intra-class gathering. But the downstream performance tends to be better with stronger augmentation (Wang et al., 2022b; Zhu et al., 2020), so the performance gain may be brought by inter-class separating more than intra-class gathering.

The experiment shown in Figure 1 confirms our suspicion. We use dropout on edges and features to perform augmentation, and the dropout rate naturally represents the magnitude of augmentation *i.e.,* graph edit distance. We present the positive/negative center distance and downstream accuracy to show the changing tendency. Figure 1 shows that initially, as the dropout rate increases, positive center distance is not decreasing, but downstream performance could be enhanced as negative center distance increases sharply. So the better performance correlates to inter-class separating, and the intra-class nodes may not be gathered.

We show the results on more datasets including shopping graph, graph with heterophily and coauthor network in Appendix C.2. From those experiments, we can conclude that contrastive learning mainly contributes to downstream tasks by separating nodes of different classes (increasing negative center distance) rather than gathering nodes of the same class (non-decreasing positive center distance). This explains why contrastive learning can achieve satisfying performance with limited augmentation overlap and relatively weak alignment (Saunshi et al., 2022).

We can also understand the phenomena intuitively, The InfoNCE loss $\mathcal{L}_{\text{NCE}}$ can be written as below:

$$\mathcal{L}_{\text{NCE}} = \mathbb{E}_{p(v_i^1, v_i^2)}\mathbb{E}_{p(v_i^-)}\left[-\log \frac{\exp(f(v_i^1)^T f(v_i^2))}{\sum \exp(f(v_i^1)^T f(v_i^-))}\right].$$

The numerator stands for positive pair similarity, so stronger augmentation would make positive pair dissimilar and the numerator is harder to maximize. Then GCL would pay more attention to the minimize the denominator as shown in Appendix C.5. Minimizing the denominator is actually separating negative samples, and separating negative samples could effectively separate inter-class nodes as most negative samples are from the different classes. In contrast, with stronger augmentation augmentation overlap is still quite rare and positive pair are harder to be aligned, so intra-class nodes are hard to be gathered while the existence of intra-class negative nodes further weaken intra-class gathering. As a result intra-class nodes may not gather closer during contrastive learning. Also we can observe from Figure 1 that when we drop too much edges/features, downstream performance decreases sharply, and both positive and negative center similarity increases as too much information is lost and the basic assumption $p(y|v_i^0) = p(y|v_i^+)$ does not hold, then a trivial solution is learned.

2.3. Augmentation and Generalization

Although GCL with a stronger augmentation may help to improve downstream performance, why it works stays unclear. We need to figure out the relationship between augmentation distance, contrastive loss and downstream performance to further guide algorithm design. We first define the mean cross-entropy (CE) loss below, and use it to represent downstream performance.

Definition 2.5 (Mean CE loss)**.** For an encoder f and downstream labels $y \in [1, K]$, we use the mean CE loss $\hat{\mathcal{L}}_{\mathrm{CE}} = \mathbb{E}_{p(v^0,y)}\left[-\log \frac{\exp(f(v^0)^T \mu_y)}{\sum_{j=1}^{K} \exp(f(v^0)^T \mu_j)}\right]$ to evaluate downstream performance, where $\mu_j = \mathbb{E}_{p(v|y=j)}\left[f(v)\right]$.

It is easy to see that mean CE loss could indicate downstream performance as it requires nodes similar to their respective class center, and different from others class centers. Also it is an upper bound of CE loss $\mathcal{L}_{\mathrm{CE}} = \mathbb{E}_{p(v^0,y)}\left[-\log \frac{\exp(f(v^0)^T \omega_y)}{\sum_{i=1}^{K} \exp(f(v^0)^T \omega_i)}\right]$, where ω is parameter to train a linear classifier $g(z) = Wz, W = [\omega_1, \omega_2, ..., \omega_k]$. Arora et al. (2019) showed that the mean classifier could achieve comparable performance to learned weights, so we analyze the mean CE loss instead of the CE loss in this paper.

Theorem 2.6 (Generalization and Augmentation Distance)**.** *If Assumption 2.1 holds, and* ReLU *is applied as activation, then the relationship between downstream performance and* InfoNCE *loss could be represented as:*

$$\hat{\mathcal{L}}_{\mathrm{CE}} \geq \mathcal{L}_{\mathrm{NCE}} - 3\delta_{aug}^2 - 2\delta_{aug} - \log\frac{M}{K} - \frac{1}{2}\operatorname{Var}(f(v^+)|y) - \sqrt{\operatorname{Var}(f(v^0)|y)} - e\operatorname{Var}(\mu_y) - O(M^{-\frac{1}{2}}),$$

where M is number of negative samples[1]*, K is number of classes.*

The proof can be found in Appendix A.2. Theorem 2.6 gives a lower bound on the mean CE loss, we find that when we perform stronger augmentation, the lower bound would be smaller. The smaller lower bound does not enforce a better performance, but it shows a potential better solution. When the lower bound becomes smaller, the best solution is better so the model potentially performs better. For example, if there exists two models with $\hat{\mathcal{L}}_{\mathrm{CE}} \geq 0.7$ and $\hat{\mathcal{L}}_{\mathrm{CE}} \geq 0.3$ respectively, the latter one would be prefered as it is more likely to perform better, and the former one can never achieve performance better than 0.7. The upper bound instead shows the worst case, smaller upper bound means that the model could perform better at the worst case.

[1]the generalization are correlated with $-\log M - O(M^{-\frac{1}{2}})$, which is decreasing when M increases and M is large, so the theorem encourages large negative samples.

From experimental results shown in Appendix C.2, we can observe that the downstream performance tends to be better with stronger augmentation which corresponds to the decreasing lower bound, so the model is powerful enough to be close to the lower bound. Therefore, a smaller lower bound could lead to better performance.

Theorem 2.6 suggests a gap between $\hat{\mathcal{L}}_{\mathrm{CE}}$ and $\mathcal{L}_{\mathrm{NCE}}$, meaning that the encoders that minimize $\mathcal{L}_{\mathrm{NCE}}$ may not yield optimal performance on downstream tasks. Furthermore, it suggests that a higher augmentation distance δ_{aug} would make the bound smaller and enhance generalization, improve performance on downstream tasks. This aligns with previous finding that a stronger augmentation helps downstream performance. Also Inequality (1) demonstrates that the positive center distance is positively related to δ_{aug}, so better generalization correlates with higher positive center distance. This aligns with the experiments before that better downstream performance may come with a high positive center distance.

Theorem 2.6 also highlights the significance of augmentation magnitude in graph contrastive learning algorithms like GRACE (Zhu et al., 2020). A weak augmentation leads to better alignment but also a weak generalization, InfoNCE loss might be relatively low but the downstream performance could be terrible (Saunshi et al., 2022). When augmentation gets stronger, although perfect alignment cannot be achieved, it promotes better generalization and potentially leads to improved downstream performance. And when the augmentation is too strong, minimizing the InfoNCE loss becomes challenging (Li et al., 2022), leading to poorer downstream performance. Therefore, it is crucial to determine the magnitude of augmentation and how to perform augmentation as it directly affects contrastive performance and generalization.

3. Finding Better Augmentation

Previous sections have revealed that perfect alignment, may not help downstream performance. Instead a stronger augmentation that leads to larger δ_{aug} will benefit generalization while weakening contrastive learning process. Therefore, we need to find out how to perform augmentation to strike a better balance between augmentation distance and contrastive loss, leading to better downstream performance.

3.1. Information Theory Perspective

Due to the inherent connection between contrastive learning and information theory, we try to analyse it through information perspective. As shown by Oord et al. (2018), $\mathcal{L}_{\mathrm{NCE}}$ is a lower bound of mutual information. And, $\operatorname{Var}(f(v^0)|y)$, $\operatorname{Var}(f(v^+|y))$ and $\operatorname{Var}(\mu_y)$ can be represented by inherent properties of the graph and the augmentation distance δ_{aug}.

Thus, we can understand the process through information and augmentation, we can reformulate Theorem 2.6 as follows:

Corollary 3.1 (CE with Mutual Information)**.** *If Assumption 2.1 holds, the relationship between downstream performance, mutual information between views and augmentation distance could be represented as:*

$$\hat{\mathcal{L}}_{\mathrm{CE}} \geq \log(K) - I(v^1, v^2) - g(\delta_{aug}) - O(M^{-\frac{1}{2}}),$$

where $I(v^1, v^2)$ stands for the mutual information between v^1 and v^2, $g(\delta_{aug})$ is monotonically increasing with δ_{aug}, and is defined in Appendix A.3.

The proof can be found in Appendix A.3. Corollary 3.1 suggests that the best augmentation would be one that maximize the mutual information and augmentation distance. Tian et al. (2020) propose that a good augmentation should minimize $I(v^1, v^2)$ while preserve as much downstream related information as possible, *i.e.,* $I(v^1, y) = I(v^2, y) = I(v^0, y)$. However, downstream tasks is unknown while pretraining, so this is actually impossible to achieve. Our theory indicates that the augmentation should be strong while preserving as much information as possible, and the best augmentation should be the one satisfying InfoMin which means the augmentation gets rid of all useless information and keeps the downstream related ones. So InfoMin propose the ideal augmentation which can not be achieved, and we propose an actual target to train a better model.

To verify our theory, we propose a simple but effective method. We first recognize important nodes, features and edges, then leave them unchanged during augmentation to increase mutual information. Then for those unimportant ones, we should perform stronger augmentation to increase the augmentation distance.

We utilize gradients to identify which feature of node v is relatively important and carries more information. We calculate the importance of feature by averaging the feature importance across all nodes, the importance of node v could be calculated by simply averaging the importance of its features, and then use the average of the two endpoints to represent the importance of an edge:

$$\alpha_{v,p} = \frac{\partial \mathcal{L}_{\mathrm{NCE}}}{\partial x_{v,p}}, \quad \alpha_p = \mathrm{ReLU}\left(\frac{1}{|V'|}\sum_v \alpha_{v,p}\right),$$
$$\alpha_v = \mathrm{ReLU}\left(\frac{1}{|P'|}\sum_p \alpha_{v,p}\right), \quad \alpha_{e_{i,j}} = \left(\alpha_{v_i} + \alpha_{v_j}\right)/2,$$

where $\alpha_{v,p}$ is importance of the p^{th} feature of node v, α_p is the importance of p^{th} feature, α_v iss importance of node v, and $\alpha_{e_{i,j}}$ means the importance of edge (v_i, v_j).

For those edges/features with high importance, we should keep them steady and do no modification during augmentation. For those with relatively low importance, we can freely mask those edges/features, but we should make sure that the number of masked edges/features is greater than the number of kept ones to prevent δ_{aug} from decreasing. The process can be described by the following equation:

$$\tilde{\boldsymbol{A}} = \boldsymbol{A} * (\boldsymbol{M}_e \vee \boldsymbol{S}_e \wedge \boldsymbol{D}_e), \quad \tilde{\boldsymbol{F}} = \boldsymbol{F} * (\boldsymbol{M}_f \vee \boldsymbol{S}_f \wedge \boldsymbol{D}_f),$$

where $*$ is hadamard product, $\vee$ stands for logical OR, $\wedge$ stands for logical AND. $\boldsymbol{M}_e$, $\boldsymbol{M}_f$ represent the random mask matrix, which could be generated using any masking method, $\boldsymbol{S}_e$, $\boldsymbol{S}_f$ are the importance based retain matrix, it tells which edge/feature is of high importance and should be retained. For the top ξ important edges/features, we set $\boldsymbol{S}_e$, $\boldsymbol{S}_f$ to 1 with a probability of 50% and to 0 otherwise. $\boldsymbol{D}_e$, $\boldsymbol{D}_f$ show those edges/features should be deleted to increase δ_{aug}, for the least 2ξ important edges/features, we also set $\boldsymbol{D}_e$, $\boldsymbol{D}_f$ to 0 with a probability of 50% and to 1 otherwise.

This is a simple method, and the way to measure importance can be replaced by any other methods. It can be easily integrated into any other graph contrastive learning methods that require edge/feature augmentation. There are many details that could be optimized, such as how to choose which edges/features to delete and the number of deletions. However, since this algorithm is primarily intended for theoretical verification, we just randomly select edges to be deleted and set the number to be deleted as twice the number of edges kept.

In fact, most graph contrastive learning methods follow a similar framework as discussed in Appendix B.1.

3.2. Graph Spectrum Perspective

In this section, we attempt to analyze InfoNCE loss and augmentation distance from graph spectrum perspective as graph and GNNs are deeply connected with spectrum theory. We start by representing them using the spectrum of the adjacency matrix $\boldsymbol{A}$.

Theorem 3.2 (Theorem 1 of Liu et al. (2022) Restated)**.** *Given adjacency martix $\boldsymbol{A}$ and the generated augmentation $\boldsymbol{A}'$, $\boldsymbol{A}''$, the i^{th} eigenvalues of $\boldsymbol{A}'$ and $\boldsymbol{A}''$ are λ_i', λ_i'', respectively. The following upper bound is established:*

$$\mathcal{L}_{\mathrm{NCE}} \geq N \log N - (N+1)\sum_i \theta_i \lambda_i' \lambda_i'', \tag{3}$$

where θ_i is the adaptive weight of the i^{th} term, the detail of θ_i is discussed in Appendix C.1.

Corollary 3.3 (Spectral Representation of δ_{aug})**.** *If Assumption 2.1 holds, and λ_i', λ_i'' are i^{th} eigenvalues of $\boldsymbol{A}'$ and $\boldsymbol{A}''$,*

Table 1. Quantitative results on node classification, algorithm+I stands for the algorithm with information augmentation, and algorithm+S stands for the algorithm with spectrum augmentation. We show the error bar in Figure 12

	Methods	Cora	CiteSeer	PubMed	DBLP	Photo	Computers	mean	p-value
Baseline	Supervised GCN	83.31±0.07	69.81±0.98	85.36±0.09	81.26±0.01	93.28±0.03	88.11±0.14	81.93	-
	Supervised GAT	83.83±0.30	70.31±0.65	84.04±0.40	81.92±0.03	93.17±0.05	86.82±0.09	81.57	-
	GRACE+SpCo	83.45±0.79	69.9±1.24	OOM	83.61±0.14	91.56±0.19	83.37±0.38	81.01	-
	GCS	83.39±0.54	68.73±1.68	84.92±0.19	83.38±0.37	90.15±0.24	86.54±0.26	81.97	-
Unsupervised	GRACE	82.52±0.75	70.44±1.49	84.97±0.17	84.01±0.34	91.17±0.15	86.36±0.32	83.25	-
	GRACE+I	**83.78±1.08**	**72.89±0.97**	84.97±0.14	84.80±0.17	91.64±0.21	87.57±0.53	84.28	0.155
	GRACE+S	83.61±0.85	72.83±0.47	**85.45±0.25**	**84.83±0.18**	**91.99±0.35**	**87.67±0.33**	**84.40**	0.003
	GCA	83.74±0.79	71.09±1.29	85.38±0.20	83.99±0.21	91.67±0.38	86.77±0.31	83.77	-
	GCA+I	**84.93±0.81**	**72.74±1.05**	**85.73±0.13**	**84.79±0.28**	91.94±0.13	86.60±0.29	**84.46**	0.089
	GCA+S	84.51±0.89	72.38±0.86	85.35±0.09	84.49±0.24	**92.02±0.34**	**86.97±0.40**	84.29	0.147
	AD GCL	81.68±0.80	70.01±0.97	84.77±0.15	83.14±0.16	91.34±0.33	84.80±0.51	82.62	-
	AD GCL+I	**83.46±1.06**	71.06±0.91	**85.52±0.33**	**84.76±0.09**	91.71±0.78	86.02±0.53	**83.76**	0.003
	AD GCL+S	82.96±0.53	**71.35±0.47**	85.38±0.30	84.45±0.19	**91.79±0.33**	**86.49±0.26**	83.74	0.006

respectively, then:

$$2\delta_{aug} \geq \mathbb{E}_{p(v_i^1, v_i^2)} ||f(v_i^1) - f(v_i^2)|| \geq \sqrt{2 - \frac{2}{N} \sum_i \theta_i \lambda_i' \lambda_i''}. \tag{4}$$

Theorem 2.6 suggests that we should strive to make $\mathcal{L}_{\text{NCE}}$ small while increase δ_{aug}, but they are kindly mutually exclusive. As shown in Theorem 3.2, and Corallary 3.3 proved in Appendix A.4, when θ_i is positive, a small $\mathcal{L}_{\text{NCE}}$ requires for large $|\lambda_i|$ while a large δ_{aug} requires for small $|\lambda_i|$, and it works exclusively too when θ_i is negative. As contrastive learning is trained to minimize $\mathcal{L}_{\text{NCE}}$, θs are going to increase as the training goes, so we can assume that θs will be positive, the detailed discussion and exact definition of θ can be found in Appendix C.1. Therefore, to achieve a better trade-off, we should decrease $|\lambda_i|$ while keep InfoNCE loss also decreasing. In fact, reducing $|\lambda_i|$ actually reduces the positive λ_i and increases the negative λ_i, which is trying to smoothen the graph spectrum and narrow the gap between the spectrum. As suggested by Yang et al. (2022b), graph convolution operation with unsmooth spectrum results in signals correlated to the eigenvectors corresponding to larger magnitude eigenvalues and orthogonal to the eigenvectors corresponding to smaller magnitude eigenvalues. So with enough graph convolution operations, if $|\lambda_i| > |\lambda_j|$, then we can get the embedding $f(v)$ satisfying $\text{sim}(f(v), e_i) \gg \text{sim}(f(v), e_j)$ where e_i denotes the eigenvector corresponding to λ_i, causing all representations similar to e_i. Therefore, an unsmooth spectrum may lead to similar representations and result in over-smooth. This can also be observed from Inequality (4), where a higher $|\lambda_i|$ draws $f(v_i^1)$ and $f(v_i^2)$ more similar.

We now know that smoothing the graph spectrum can help with graph contrastive learning. The question is how to appropriately smooth the spectrum. We propose a simple method. As the training aims to minimize $\mathcal{L}_{\text{NCE}}$, the parameter θ_is are supposed to increase. Therefore, we can use θ_i as a symbol to show whether the model is correctly trained. When θ_i gradually increases, we can decrease λ as needed. However, when θ_i starts to decrease, it is likely that the change on the spectrum is too drastic, and we should take a step back. The process could be described as follows:

$$\lambda_i = \lambda_i + \text{direction}_i * \lambda_i * \alpha,$$
$$\text{direction}_i = \begin{cases} -1, & \text{cur}(\theta_i) - \text{pre}(\theta_i) \geq \epsilon \\ 1, & \text{cur}(\theta_i) - \text{pre}(\theta_i) \leq -\epsilon, \\ 0, & \text{otherwise} \end{cases}$$

where α is a hyperparameter that determines how much we should decrease/increase λ_i. ϵ is used to determine whether θ_i is increasing, decreasing, or just staying steady. $\text{cur}(\theta_i)$ and $\text{pre}(\theta_i)$ represents the current and previous θ_i.

In this way, the contrastive training will increase θ_i and result in a lower $\mathcal{L}_{\text{NCE}}$, while we justify λ_i to achieve a better augmentation distance, which promises a better generalization ability. Also some spectral augmentations implicitly decreases $|\lambda|$s as shown in Appendix B.2.

4. Experiments

In this section, we mainly evaluate the performance of the methods we proposed on six datasets: Cora, CiteSeer, PubMed, DBLP, Amazon-Photo and Amazon-Computer. We select 3 contrastive learning GNN, GRACE (Zhu et al., 2020), GCA (Zhu et al., 2021) and AD-GCL (Suresh et al., 2021), then we integrate those models with our proposed methods to verify its applicability and correctness of the theory. Details of datasets and baselines are shown in Appendix D.1. The results are summarized in Table 1. We further investigate the positive/negative center distance in Appendix D.4, the hyperparameter sensitivity is studied in

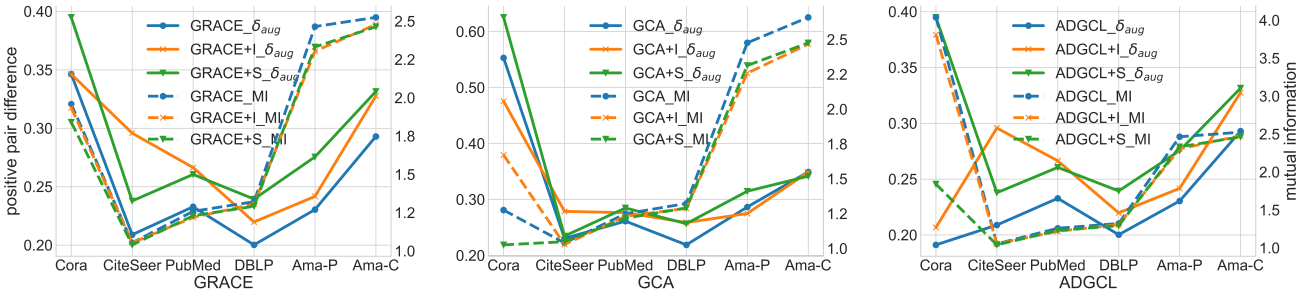

Figure 2. Augmentation distance and InfoNCE, GRACE+I stands for GRACE with information augmentation, and GRACE+S stands for GRACE with spectrum augmentation. GRACE+x_MI means mutual information between two views after training, and GRACE+x_δ_{aug} is augmentation distance caused by the method.

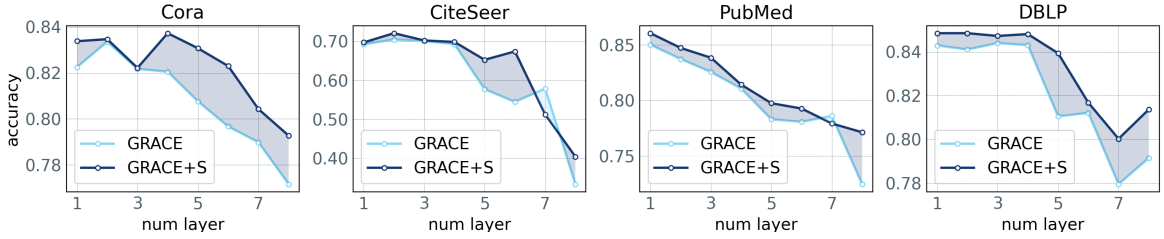

Figure 3. Accuracy on downstream tasks with different number of layers.

Appendix D.5, and the change of θ and the spectrum while training is shown in Appendix D.3.

From Table 1 shows that GRACE+I (GRACE with information augmentation) and GRACE+S (GRACE with spectrum augmentation) both improve the downstream performance. This improvement is significant for GRACE since it primarily performs random dropout, resulting in the loss of valuable information. But for GCA, the performance gain is relatively weak as GCA already drops the unimportant ones with a higher probability, allowing it to capture sufficient information. AD-GCL aggressively drops as much information as possible and some important ones are also dropped, so our methods help greatly. Overall, our methods improve the performance of original algorithm and helps downstream tasks, the p-value on the averaged performance shown in Table 1 also prove that our method is effective. We further discuss the two different methods and combine then in Appendix D.7. Also we conduct further discussion on some augmentation free methods in Appendix C.4.

4.1. Augmentation Distance

Figure 2 shows that for all three algorithms, our augmentation methodologies can conduct stronger augmentation while preserving similar mutual information. In this way, our methods achieve higher augmentation distance while capturing similar information of the original view. So our methods achieve similar contrastive loss with better generalization, resulting in improved downstream performance.

4.2. Over-smooth

While reducing $|\lambda_i|$, we obtain a graph with smoother spectrum, and could relieve the over-smooth by preventing nodes being too similar with the eigenvector corresponding to the largest eigenvalue. This enables the application of relatively more complex models. We can verify this by simply stacking more layers. As shown in Figure 3, if applied spectrum augmentation, the model tends to outperform the original algorithm especially with more layer, and the best performance may come with a larger number of layers, which indicates that more complicated models could be applied and our method successfully relieve over-smooth.

5. Conclusion

In this paper, we study the impact of contrastive learning on downstream tasks and propose that perfect alignment does not necessarily lead to better performance. Instead, we find that a relatively large augmentation distance is more beneficial for generalization by enlarging the distance of inter-class nodes. We further study how the augmentation influences contrastive learning by information theory and the graph spectrum theory and propose two effective methods.

Impact Statement

This work studies the theory and algorithm of Graph Contrastive Learning, which does not present any foreseeable societal consequence.

Acknowledgements

We thank the anonymous reviewers for their valuable and constructive suggestions and comments. This work is supported by the Beijing Natural Science Foundation (No.4222029); the National Natural Science Foundation of China (N0.62076234); the National Key Research and Development Project (No.2022YFB2703102); the "Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China"; the Beijing Outstanding Young Scientist Program (NO.BJJWZYJH012019100020098); the Public Computing Cloud, Renmin University of China; the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (NO.2021030199), the Huawei-Renmin University joint program on Information Retrieval: the Unicom Innovation Ecological Cooperation Plan; the CCF-Huawei Populus Grove Fund.

References

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Ash, J. T., Goel, S., Krishnamurthy, A., and Misra, D. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*, 2021.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

Budimir, I., Dragomir, S. S., and Pecaric, J. Further reverse results for jensen's discrete inequality and applications in information theory. *RGMIA research report collection*, 3 (1), 2000.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Guo, X., Wang, Y., Wei, Z., and Wang, Y. Architecture matters: Uncovering implicit mechanisms in graph contrastive learning. *arXiv preprint arXiv:2311.02687*, 2023.

Hassani, K. and Khasahmadi, A. H. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pp. 4116–4126. PMLR, 2020.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., and Wang, M. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Huang, W., Yi, M., and Zhao, X. Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*, 2021.

Jia, B.-B. and Zhang, M.-L. Multi-dimensional classification via selective feature augmentation. *Machine Intelligence Research*, 19(1):38–51, 2022.

Jiang, B., Zhang, Z., Lin, D., Tang, J., and Luo, B. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11313–11320, 2019.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Li, H., Cao, J., Zhu, J., Luo, Q., He, S., and Wang, X. Augmentation-free graph contrastive learning of invariant-discriminative representations. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Li, S., Wang, X., Zhang, A., Wu, Y., He, X., and Chua, T.-S. Let invariant rationale discovery inspire graph contrastive learning. In *International conference on machine learning*, pp. 13052–13065. PMLR, 2022.

Lin, L., Chen, J., and Wang, H. Spectral augmentation for self-supervised learning on graphs. *arXiv preprint arXiv:2210.00643*, 2022.

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

Liu, Q., Allamanis, M., Brockschmidt, M., and Gaunt, A. Constrained graph variational autoencoders for molecule design. *Advances in neural information processing systems*, 31, 2018.

Mo, Y., Peng, L., Xu, J., Shi, X., and Zhu, X. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7797–7805, 2022.

Mumuni, A., Mumuni, F., and Gerrar, N. K. A survey of synthetic data augmentation methods in machine vision. *Machine Intelligence Research*, pp. 1–39, 2024.

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

Peng, Z., Huang, W., Luo, M., Zheng, Q., Rong, Y., Xu, T., and Huang, J. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, pp. 259–270, 2020.

Saunshi, N., Ash, J., Goel, S., Misra, D., Zhang, C., Arora, S., Kakade, S., and Krishnamurthy, A. Understanding contrastive learning requires incorporating inductive biases. In *International Conference on Machine Learning*, pp. 19250–19286. PMLR, 2022.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

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

Stewart, G. W. Matrix perturbation theory. 1990.

Suresh, S., Li, P., Hao, C., and Neville, J. Adversarial graph augmentation to improve graph contrastive learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15920–15933. Curran Associates, Inc., 2021.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

Trivedi, P., Lubana, E. S., and Koutra, D. Understanding self-supervised graph representation learning from a data-centric perspective. 2022.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

Wang, H., Guo, X., Deng, Z.-H., and Lu, Y. Rethinking minimal sufficient representation in contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16041–16050, 2022a.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Wang, Y., Zhang, Q., Wang, Y., Yang, J., and Lin, Z. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. *arXiv preprint arXiv:2203.13457*, 2022b.

Wei, C., Wang, Y., Bai, B., Ni, K., Brady, D., and Fang, L. Boosting graph contrastive learning via graph contrastive saliency. In *International Conference on Machine Learning*, pp. 36839–36855. PMLR, 2023.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

Xu, D., Cheng, W., Luo, D., Chen, H., and Zhang, X. Infogcl: Information-aware graph contrastive learning, 2021.

Yang, L., Kang, L., Zhang, Q., Li, M., He, D., Wang, Z., Wang, C., Cao, X., Guo, Y., et al. Open: Orthogonal propagation with ego-network modeling. *Advances in Neural Information Processing Systems*, 35:9249–9261, 2022a.

Yang, L., Zhang, Q., Shi, R., Zhou, W., Niu, B., Wang, C., Cao, X., He, D., Wang, Z., and Guo, Y. Graph neural networks without propagation. In *Proceedings of the ACM Web Conference 2023*, pp. 469–477, 2023.

Yang, L., Shi, R., Zhang, Q., Wang, Z., Cao, X., Wang, C., et al. Self-supervised graph neural networks via low-rank decomposition. *Advances in Neural Information Processing Systems*, 36, 2024.

Yang, M., Shen, Y., Li, R., Qi, H., Zhang, Q., and Yin, B. A new perspective on the effects of spectrum in graph neural networks. In *International Conference on Machine Learning*, pp. 25261–25279. PMLR, 2022b.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

Yuan, Y., Xu, B., Shen, H., Cao, Q., Cen, K., Zheng, W., and Cheng, X. Towards generalizable graph contrastive learning: An information theory perspective. *arXiv preprint arXiv:2211.10929*, 2022.

Zhang, Y., Zhu, H., Song, Z., Koniusz, P., and King, I. Costa: covariance-preserving feature augmentation for graph contrastive learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2524–2534, 2022.

Zhao, T., Jin, W., Liu, Y., Wang, Y., Liu, G., Günnemann, S., Shah, N., and Jiang, M. Graph data augmentation for graph machine learning: A survey. *arXiv preprint arXiv:2202.08871*, 2022.

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

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

A. Theoretical Proof

A.1. Proof of Theorem 2.4

If we set $\delta_{y^+}^2 = \mathbb{E}_{p(y,i,j)}||f(v_{y,i}^0) - f(v_{y,j}^0)||^2$, and $\delta_{y^+}^2 = \mathbb{E}_{p(y,y',i,j)}||f(v_{y,i}^0) - f(v_{y',j}^0)||^2$. Then with Assumption 2.1 and jensen inequality, we know that $\mathbb{E}_{p(v_i)}||f(v_i^0) - f(v_i^+)||^2 \leq \delta_{aug}^2$, $\mathbb{E}_{p(v_i)}||f(v_i^0) - f(v_i^+)|| \leq \delta_{aug}$ and $\mathbb{E}_{p(y,i,j)}||f(v_{y,i}^0) - f(v_{y,j}^0)|| \leq \delta_{y^+}$, $\mathbb{E}_{p(y,y',i,j)}||f(v_{y,i}^0) - f(v_{y',j}^0)|| \leq \delta_{y^-}$. Therefore, we can get the inequality below:

$$\begin{aligned}
\mathbb{E}_{p(v_{y,i},v_{y,j}|y)}||f(v_{y,i}^+) - f(v_{y,j}^0)||^2 &\leq \mathbb{E}_{p(v_{y,i},v_{y,j}|y)}||f(v_{y,i}^+) - f(v_{y,i}^0)||^2 + \mathbb{E}_{p(v_{y,i},v_{y,j}|y)}||f(v_{y,i}^0) - f(v_{y,j}^0)||^2 \\
&\quad + 2\mathbb{E}_{p(v_{y,i},v_{y,j}|y)}||f(v_{y,i}^+) - f(v_{y,j}^0)|| \cdot ||f(v_{y,i}^0) - f(v_{y,j}^0)|| \\
&\leq \delta_{aug}^2 + \delta_{y^+}^2 + 2\delta_{aug}\delta_{y^+} \\
&= (\delta_{aug} + \delta_{y^+})^2.
\end{aligned}$$

As $\mu_y = \mathbb{E}_{p(v_y|y)}[f(v_y)] = \frac{1}{3}\mathbb{E}_{p(v_y^0|y)}f(v_y^0) + \frac{2}{3}\mathbb{E}_{p(v_y^+|y)}f(v_y^+)$, we know that,

$$\begin{aligned}
\mathbb{E}_{p(v_y^{0'}|y)}||f(v_y^{0'}) - \mu_y|| &= \mathbb{E}_{p(v_y^{0'}|y)}||f(v_y^{0'}) - \frac{1}{3}\mathbb{E}_{p(v_y^0|y)}f(v_y^0) - \frac{2}{3}\mathbb{E}_{p(v_y^+|y)}f(v_y^+)|| \\
&= \mathbb{E}_{p(v_y^{0'}|y)}\left\|\frac{1}{3}\left(f(v_y^{0'}) - \mathbb{E}_{p(v_y^0|y)}f(v_y^0)\right) + \frac{2}{3}\left(f(v_y^{0'}) - \mathbb{E}_{p(v_y^+|y)}f(v_y^+)\right)\right\| \\
&\leq \mathbb{E}_{p(v_y^{0'}|y)}\left[\left\|\frac{1}{3}\left(f(v_y^{0'}) - \mathbb{E}_{p(v_y^0|y)}f(v_y^0)\right)\right\| + \left\|\frac{2}{3}\left(f(v_y^{0'}) - \mathbb{E}_{p(v_y^+|y)}f(v_y^+)\right)\right\|\right] \\
&\leq \mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^0|y)}\frac{1}{3}||(f(v_y^{0'}) - f(v_y^0))|| + \mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^+|y)}\frac{2}{3}||f(v_y^{0'}) - f(v_y^+)|| \\
&\leq \frac{1}{3}\delta_{y^+} + \frac{2}{3}(\delta_{aug} + \delta_{y^+}) \\
&= \delta_{y^+} + \delta_{aug}
\end{aligned}$$

Similarly, we know that $\mathbb{E}_{p(v_y^{0'}|y)}||f(v_y^{0'}) - \mu_{y^-}|| \leq \delta_{y^-} + \delta_{aug}$

Next we prove a bound for $\mathbb{E}_{p(v_y^0,y^-|y)}f(v_y^0)^T\mu_{y^-}$ for other use. As $\mu_y = \mathbb{E}_{p(v_y|y)}[f(v_y)] = \frac{1}{3}\mathbb{E}_{p(v_y^0|y)}f(v_y^0) + \frac{2}{3}\mathbb{E}_{p(v_y^+|y)}f(v_y^+)$, we know that,

$$\begin{aligned}
\mathbb{E}_{p(v_y^{0'}|y)}f(v_y^{0'})^T\mu_y &= \mathbb{E}_{p(v_y^{0'}|y)}f(v_y^{0'})^T(\frac{1}{3}\mathbb{E}_{p(v_y^0|y)}f(v_y^0) + \frac{2}{3}\mathbb{E}_{p(v_y^+|y)}f(v_y^0)) \\
&= \frac{1}{3}\mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^0|y)}f(v_y^{0'})^Tf(v_y^0) + \frac{2}{3}\mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^+|y)}f(v_y^{0'})^Tf(v_y^+).
\end{aligned}$$

assume that $\mathbb{E}_{p(a,b)}||a-b||^2 \leq c^2$, $||a|| = ||b|| = 1$, then

$$\begin{aligned}
&\mathbb{E}_{p(a,b)}(a^T - b^T)(a-b) \leq c^2 \\
&\mathbb{E}_{p(a,b)}[a^Ta - a^Tb - b^Ta + b^Tb] \leq c^2 \\
&\mathbb{E}_{p(a,b)}[2 - 2a^Tb] \leq c^2 \\
&\mathbb{E}_{p(a,b)}a^Tb \geq \frac{2-c^2}{2} = 1 - \frac{c^2}{2}.
\end{aligned}$$

As we already know that $\mathbb{E}_{p(y,y',i,j)}||f(v_{y,i}^0) - f(v_{y',j}^0)||^2 \leq \delta_{y^+}^2$ and $\mathbb{E}_{p(v_{y,i},v_{y,j}|y)}||f(v_{y,i}^+) - f(v_{y,j}^0)||^2 \leq (\delta_{aug} + \delta_{y^+})^2$. So $\mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^0|y)}f(v_y^{0'})^Tf(v_y^0) \geq 1 - \frac{\delta_{y^+}^2}{2}$ and $\mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^+|y)}f(v_y^{0'})^Tf(v_y^+) \geq 1 - \frac{(\delta_{aug}+\delta_{y^+})^2}{2}$.

Then, we can calculate $\mathbb{E}_{p(v_y^{0'}|y)} f(v_y^{0'})^T \mu_y$ as below:

$$\begin{aligned}\mathbb{E}_{p(v_y^{0'}|y)} f(v_y^{0'})^T \mu_y &= \frac{1}{3}\mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^{0}|y)} f(v_y^{0'})^T f(v_y^{0}) + \frac{2}{3}\mathbb{E}_{p(v_y^{0'}|y)}\mathbb{E}_{p(v_y^{+}|y)} f(v_y^{0'})^T f(v_y^{+}) \\ &\geq 1 - \frac{\delta_{aug}^2}{3} - \frac{2\delta_{aug}\delta_{y^+}}{3} - \frac{\delta_{y^+}^2}{2}.\end{aligned} \tag{5}$$

Similarly, we know that $\mathbb{E}_{p(v_y^0, y^-|y)} f(v_y^0)^T \mu_{y^-} \geq 1 - \frac{\delta_{aug}^2}{3} - \frac{2\delta_{aug}\delta_{y^-}}{3} - \frac{\delta_{y^-}^2}{2}$.

A.2. Proof of Theorem 2.6

$$\hat{\mathcal{L}}_{\text{CE}} = \underbrace{-\mathbb{E}_{p(v_i^0, y)} f(v_i^0)^T \mu_y}_{\Lambda_1} + \underbrace{\mathbb{E}_{p(v_i^0)} \log \sum_{i=j}^{K} \exp(f(v_i^0)^T \mu_j)}_{\Lambda_2}.$$

$$\begin{aligned}\Lambda_1 &= -\mathbb{E}_{p(v_i^0, y)} f(v_i^0)^T \mu_y \\ &= -\mathbb{E}_{p(v_i^0, y)} \left[f(v_i^0)^T f(v_i^+) + f(v_i^0)^T (\mu_y - f(v_i^+)) \right] \\ &\overset{(a)}{\geq} -\mathbb{E}_{p(v_i^0, v_i^+, y)} f(v_i^0)^T f(v_i^+) - \mathbb{E}_{p(v_i^+, y)} ||f(v_i^+) - \mu_y|| \\ &\geq -\mathbb{E}_{p(v_i^0, v_i^+, y)} f(v_i^0)^T f(v_i^+) - \mathbb{E}_{p(v_i^0, v_i^+, y)} ||f(v_y^+) - f(v_y^0)|| - \mathbb{E}_{p(v_i^0, v_i^+, y)} ||f(v_y^0) - \mu_y|| \\ &\overset{(b)}{\geq} -\mathbb{E}_{p(v_i^0, v_i^+, y)} f(v_i^1)^T f(v_i^2) - 3\delta_{aug}^2 - \delta_{aug} - \mathbb{E}_{p(v_i^0, v_i^+, y)} ||f(v_y^0) - \mu_y||.\end{aligned}$$

(a) $f(v_i^0)^T(\mu_y - f(v_i^+)) \leq (\frac{\mu_y - f(v_i^+)}{||\mu_y - f(v_i^+)||})^T(\mu_y - f(v_i^+)) = ||\mu_y - f(v_i^+)||$.

(b) $\mathbb{E}_{p(v_i^0, v_i^+)} ||f(v_i^0) - f(v_i^+)||^2 \leq \delta_{aug}^2$, then:

$$\begin{aligned}\delta_{aug}^2 &\geq \mathbb{E}_{p(v_i^0, v_i^1)} (f(v_i^0) - f(v_i^1))^T \cdot (f(v_i^0) - f(v_i^1)) \\ &= \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} (f(v_i^0) - f(v_i^1))^T \cdot (f(v_i^0) - f(v_i^1) + f(v_i^2) - f(v_i^2)) \\ &= \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^0)^T f(v_i^0) - f(v_i^0)^T f(v_i^1) + f(v_i^0)^T f(v_i^2) - f(v_i^0)^T f(v_i^2) \\ &\qquad - f(v_i^1)^T f(v_i^0) + f(v_i^1)^T f(v_i^1) - f(v_i^1)^T f(v_i^2) + f(v_i^1)^T f(v_i^2) \\ &= 2 + \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} \left[-2 f(v_i^0)^T f(v_i^1) + f(v_i^0)^T f(v_i^2) - f(v_i^0)^T f(v_i^2) - f(v_i^1)^T f(v_i^2) + f(v_i^1)^T f(v_i^2) \right] \\ &\overset{(c)}{\geq} 2 - 2 + \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} \left[f(v_i^0)^T f(v_i^2) - 1 - f(v_i^1)^T f(v_i^2) + 1 - 2\delta_{aug}^2 \right] \\ &= \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} \left[f(v_i^0)^T f(v_i^2) - f(v_i^1)^T f(v_i^2) \right] - 2\delta_{aug}^2.\end{aligned}$$

So, we can get the relationship between $\mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^0)^T f(v_i^2)$ and $\mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^1)^T f(v_i^2) - 2\delta_{aug}^2$ as below:

$$\begin{aligned}\delta_{aug}^2 &\geq \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^0)^T f(v_i^2) - \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^1)^T f(v_i^2) - 2\delta_{aug}^2, \\ \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^0)^T f(v_i^2) &\leq \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^1)^T f(v_i^2) + 3\delta_{aug}^2.\end{aligned}$$

As v_i^2 is an augmented node, we can get that,

$$\mathbb{E}_{p(v_i^0, v_i^+)} f(v_i^0)^T f(v_i^+) \leq \mathbb{E}_{p(v_i^0, v_i^1, v_i^2)} f(v_i^1)^T f(v_i^2) + 3\delta_{aug}^2.$$

(c) $f(v_i^0)^T f(v_i^1) \leq 1$, $f(v_i^0)^T f(v_i^2) \leq 1$, and $\mathbb{E}_{p(v_i^1,v_i^2)} f(v_i^1)^T f(v_i^2) \geq \frac{2-\mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)||^2}{2} \geq 1 - \frac{\mathbb{E}_{p(v_i^1,v_i^2)}(||f(v_i^1)-f(v_i^0)||+||f(v_i^0)-f(v_i^2)||)^2}{2} \geq 1 - 2\delta_{aug}^2$.

Lemma A.1 ((Budimir et al., 2000) Corollary 3.5 (restated)). *Let $g : \mathbb{R}^m \to \mathbb{R}$ be a differentiable convex mapping and $z \in \mathbb{R}^n$. Suppose that g is L- smooth with the constant $L > 0$, i.e. $\forall x, y \in \mathcal{R}^m, \|\nabla g(x) - \nabla g(y)\| \leq L\|x - y\|$. Then we have*

$$0 \leq \mathbb{E}_{p(z)} g(z) - g\left(\mathbb{E}_{p(z)} z\right) \leq L\left[\mathbb{E}_{p(z)}\|z\|^2 - \|\mathbb{E}_{p(z)} z\|^2\right] = L\sum_{j=1}^{n} \mathrm{Var}(z^{(j)}),$$

where $z^{(j)}$ denotes the j-th dimension of v.

Lemma A.2 ((Wang et al., 2022b) Lemma A.2. restated). *For $\mathrm{LSE} := \log \mathbb{E}_{p(z)} \exp(f(v)^\top g(z))$, we denote its (biased) Monte Carlo estimate with M random samples $z_i \sim p(z), i = 1, \ldots, M$ as $\widehat{\mathrm{LSE}}_M = \log \frac{1}{M}\sum_{i=1}^{M} \exp(f(v)^\top g(z_i))$. Then the approvimation error $A(M)$ can be upper bounded in expectation as*

$$A(M) := \mathbb{E}_{p(v,z_i)} |\widehat{\mathrm{LSE}}(M) - \mathrm{LSE}| \leq \mathcal{O}(M^{-1/2}).$$

We can see that the approvimation error converges to zero in the order of $M^{-1/2}$.

$$\begin{aligned}
\Lambda_2 &= \mathbb{E}_{p(v_i^0)} \log \sum_{j=1}^{K} \exp(f(v_i^0)^T \mu_{y_j}) \\
&= \mathbb{E}_{p(v_i^0)} \log \frac{1}{K} \sum_{i=j}^{K} \exp(f(v_i^0)^T \mu_{y_j}) + \log K \\
&= \mathbb{E}_{p(v_i^0)} \log \mathbb{E}_{p(y_j)} \exp(f(v_i^0)^T \mu_{y_j}) + \log K \\
&\overset{(d)}{\geq} \mathbb{E}_{p(v_i^1)} \log \mathbb{E}_{p(y_j)} \exp(f(v_i^1)^T \mu_{y_j}) - \delta_{aug} - e\sum_{j=1}^{n} \mathrm{Var}(\mu_j) + \log K \\
&\overset{(e)}{\geq} \mathbb{E}_{p(v_i^1)} \mathbb{E}_{p(y_i)} \log \frac{1}{M} \sum_{j=1}^{M} \exp(f(v_i^1)^T \mu_{y_j}) - A(M) + \log K - \delta_{aug} - e\sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&= \mathbb{E}_{p(v_i^1)} \mathbb{E}_{p(y_i)} \log \frac{1}{M} \sum_{j=1}^{M} \exp(\mathbb{E}_{p(v_i^-|y_i^-)} f(v_i^1)^T f(v_i^-)) - A(M) + \log K - \delta_{aug} - e\sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&\overset{(f)}{\geq} \mathbb{E}_{p(v_i^1)} \mathbb{E}_{p(y_i)} \mathbb{E}_{p(v_i^-|y_i^-)} \log \frac{1}{M} \sum_{i=1}^{M} \exp(f(v_i^1)^T f(v_i^-)) \\
&\quad - \frac{1}{2}\sum_{j=1}^{m} \mathrm{Var}(f_j(v^-|y)) - A(M) + \log K - \delta_{aug} - e\sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&= \mathbb{E}_{p(v_i^1)} \mathbb{E}_{p(y_i)} \mathbb{E}_{p(v_i^-|y_i^-)} \log \sum_{i=1}^{M} \exp(f(v_i^1)^T f(v_i^-)) \\
&\quad - \log M - \frac{1}{2}\sum_{j=1}^{m} \mathrm{Var}(f_j(v^-|y)) - A(M) + \log K - \delta_{aug} - e\sum_{j=1}^{n} \mathrm{Var}(\mu_j).
\end{aligned}$$

(d) We can show that: $\exp(\left[f(v)^T \mu_{y_j}\right]$ is convex, and u_{y_j} satisfy e-smooth,

$$||\frac{\partial \exp(f(v)^T a)}{\partial a} - \frac{\partial \exp(f(v)^T b)}{\partial b}||$$

$$
\begin{aligned}
&= ||\exp(f(v)^T a)f(v) - \exp(f(v)^T b)f(v))|| \\
&= |\exp(f(v)^T a) - \exp(f(v)^T b)| \cdot ||f(v)|| \\
&\leq |\exp(f(v)^T a) - \exp(f(v)^T b)| \\
&\leq e||(f(v)^T)(a-b)|| \quad (f(v)^T a, f(v)^T b \leq 1, \text{so the biggest slope is } e) \\
&\leq e||a-b||.
\end{aligned}
$$

So according to Lemma A.1, we get,

$$
\begin{aligned}
\mathbb{E}_{p(y_j)} \exp(\left[f(v_i^1)^T \mu_{y_j}\right]) &\leq \exp(\left[f(v_i^1)^T \mathbb{E}_{p(y_j)} \mu_{y_j}\right]) + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&= \exp(f(v_i^1)^T \mu) + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j).
\end{aligned}
$$

Then, we can calculate the difference between $\log \mathbb{E}_{p(y_j)} \exp(\left[f(v_i^0)^T \mu_{y_j}\right])$ and $\log \mathbb{E}_{p(y_j)} \exp(\left[f(v_i^1)^T \mu_{y_j}\right])$ by applying reversed Jensen and Jensen inequality, respectively.

$$
\begin{aligned}
&\log \mathbb{E}_{p(y_j)} \exp(\left[f(v_i^1)^T \mu_{y_j}\right]) - \log \mathbb{E}_{p(y_j)} \exp(\left[f(v_i^0)^T \mu_{y_j}\right]) \\
&\leq \log \mathbb{E}_{p(y_j)} \exp(\left[f(v_i^1)^T \mu_{y_j}\right]) - \left[f(v_i^0)^T \mu\right] \\
&\leq \log \left[\exp(f(v_i^1)^T \mu) + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j)\right] - \left[f(v_i^0)^T \mu\right] \\
&= \log \left[\exp(f(v_i^1)^T \mu)\right] + \log \left[1 + \frac{e \sum_{j=1}^{n} \mathrm{Var}(\mu_j)}{\exp(f(v_i^1)^T \mu)}\right] - \left[f(v_i^0)^T \mu\right] \\
&\leq f(v_i^1)^T \mu - f(v_i^0)^T \mu + \log \left[1 + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j)\right] \quad (e^2 \sum_{j=1}^{n} \mathrm{Var}(\mu_j), \text{if not ReLU}) \\
&\leq (f(v_i^1)^T - f(v_i^0)^T)\mu + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&\leq (f(v_i^1) - f(v_i^0))^T \frac{||\mu||}{||f(v_i^1) - f(v_i^0)||} (f(v_i^1) - f(v_i^0)) + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&\leq (f(v_i^1) - f(v_i^0))^T \frac{1}{||f(v_i^1) - f(v_i^0)||} (f(v_i^1) - f(v_i^0)) + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j) \\
&\leq \delta_{aug} + e \sum_{j=1}^{n} \mathrm{Var}(\mu_j).
\end{aligned}
$$

(e) We adopt a Monte Carlo estimation with M samples from $p(y)$ and bound the approximation error with Lemma A.2.

(f) We also uses Lemma A.1, and as proof by Wang et al. (2022b), logsumexp is L-smooth for $L = \frac{1}{2}$.

$$
\begin{aligned}
\mathcal{L}_{\mathrm{CE}} &= \Lambda_1 + \Lambda_2 \\
&\geq -\mathbb{E}_{p(v,y)} f(v_i^1)^T f(v_i^2) - 3\delta_{aug}^2 - \delta_{aug} - \mathbb{E}_{p(v^0,y)} ||f(v_y^0) - \mu_y|| \\
&\quad + \mathbb{E}_{p(v_i^1)} \mathbb{E}_{p(y_i)} \mathbb{E}_{p(v_i^-|y_i)} \log \sum_{i=1}^{M} \exp(f(v_i^1)^T f(v_i^-))
\end{aligned}
$$

$$
\begin{aligned}
&- \log M - \frac{1}{2}\sum_{j=1}^{m}\mathrm{Var}(f_j(v^-|y)) - A(M) + \log K - \delta_{aug} - e\sum_{j=1}^{n}\mathrm{Var}(\mu_j) \\
=& \left[-\mathbb{E}_{p(v_i^1,v_i^2)} f(v_i^1)^T f(v_i^2) + \mathbb{E}_{p(v_i^-)} \log \sum_{i=1}^{M} \exp(f(v_i^1) f(v_i^-))\right] - 3\delta_{aug}^2 - \delta_{aug} - \mathbb{E}_{p(v^0,y)}||f(v_y^0) - \mu_y|| \\
&- \log M - \frac{1}{2}\sum_{j=1}^{m}\mathrm{Var}(f_j(v^-|y)) - A(M) + \log K - \delta_{aug} - e\sum_{j=1}^{n}\mathrm{Var}(\mu_j) \\
=& \mathcal{L}_{\mathrm{NCE}} - 3\delta_{aug}^2 - 2\delta_{aug} - \log\frac{M}{K} - \frac{1}{2}\sum_{j=1}^{m}\mathrm{Var}(f_j(v^-|y)) - A(M) - e\sum_{j=1}^{n}\mathrm{Var}(\mu_j) - \mathbb{E}_{p(v^0,y)}||f(v_y^0) - \mu_y|| \\
\overset{(g)}{\geq}& \mathcal{L}_{\mathrm{NCE}} - 3\delta_{aug}^2 - 2\delta_{aug} - \log\frac{M}{K} - \frac{1}{2}\mathrm{Var}(f(v^+)|y) - \sqrt{\mathrm{Var}(f(v^0)|y)} - O(M^{-\frac{1}{2}}) - e\,\mathrm{Var}(\mu_y).
\end{aligned}
$$

(g) This holds because, v^- is randomly selected from v^+ and,

$$
\begin{aligned}
&\sum_{j=1}^{m}\mathrm{Var}(f_j(v^-|y)) \\
=& \sum_{j=1}^{m}\mathbb{E}_{p(y)}\mathbb{E}_{p(v|y)}(f_j(v^+) - \mathbb{E}_{p(v'|y)}f_j(v'))^2 \\
=& \mathbb{E}_{p(y)}\mathbb{E}_{p(v|y)}\sum_{j=1}^{m}(f_j(v^+) - \mathbb{E}_{v'}f_j(v')) \\
=& \mathbb{E}_{p(y)}\mathbb{E}_{p(v|y)}||f(v) - \mathbb{E}_{v'}f(v')||^2 \\
=& \mathrm{Var}(f(v^+)|y).
\end{aligned}
$$

And similarly, we can get $\sum_{j=1}^{n}\mathrm{Var}(\mu_j) = \mathrm{Var}(\mu_y)$. So the lower bound is proved.

A.3. Proof of Corollary 3.1

For $\mathrm{Var}(f(v_y^0|y))$, we can use augmentation distance and the intrinsic property of model and data to express.

$$
\begin{aligned}
\mathrm{Var}(f(v_y^0|y)) &= \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^0|y)}||f(v_y^0) - \mu_y||^2 \\
&= \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^0|y)}\left[(f(v_y^0) - \mu_y)^T(f(v_y^0) - \mu_y))\right] \\
&= \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^0|y)}\left[f(v_y^0)^T f(v_y^0) + \mu_y^T\mu_y - 2f(v_y^0)^T\mu_y\right] \\
&\leq \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^0|y)}\left[2 - 2f(v_y^0)^T\mu_y\right] \\
&\overset{(h)}{\leq} \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^0|y)}\left[2 - 2(1 - \frac{1}{3}\delta_{aug}^2 - \frac{2}{3}\delta_{aug}\delta_{y^+} - \frac{1}{2}\delta_{y^+}^2)\right] \\
&= \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^0|y)}\left[\frac{2}{3}\delta_{aug}^2 + \frac{4}{3}\delta_{aug}\delta_{y^+} + \delta_{y^+}^2)\right] \\
&\leq \frac{2}{3}\delta_{aug}^2 + \frac{4}{3}\delta_{aug}L\epsilon_0 + L^2\epsilon_0^2,
\end{aligned}
$$

where $\epsilon_0 = \mathbb{E}_{p(y)}\mathbb{E}_{p(v_i^0,v_j^0|y)}||v_i^0 - v_j^0||$ and L is the Lipschitz constant, so $\delta_{y^+}^2 = \mathbb{E}_{p(y,i,j)}||f(v_{y,i}^0) - f(v_{y,j}^0)||^2 \leq (L\epsilon_0)^2$.

Then we can easily get that,

$$\begin{aligned}
\mathrm{Var}(f(v_y^+)|y) &= \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^-|y)}||f(v_y^+) - \mu_y||^2 \\
&\le \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^+|y)}(||f(v_y^+) - f(v_y^0)|| + ||f(v_y^0) - \mu_y||)^2 \\
&= \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^+|y)}||f(v_y^+) - f(v_y^0)||^2 + \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^+|y)}||f(v_y^0) - \mu_y||)^2 \\
&\quad + 2\mathbb{E}_{p(y)}\mathbb{E}_{p(v_y^+|y)}||f(v_y^+) - f(v_y^0)|| \cdot ||f(v_y^0) - \mu_y|| \\
&\le \delta_{aug}^2 + \mathrm{Var}(f(v_y^0)|y) + 2\delta_{aug}\sqrt{\mathrm{Var}(f(v_y^0)|y)} \\
&= (\delta_{aug} + \sqrt{\mathrm{Var}(f(v_y^0)|y)})^2.
\end{aligned}$$

(h) We use Theorem 2.4.

And $\mathrm{Var}(\mu_y)$ can also be expressed by intrinsic properties.

$$\begin{aligned}
\mathrm{Var}(\mu_y) &= \mathbb{E}_{p(y)}||\mu_y - \mu||^2 \\
&= \mathbb{E}_{p(y)}||\mu_y - f(v_y^*) + f(v_y^*) - \mu||^2 \\
&\le \mathbb{E}_{p(y)}(||\mu_y - f(v_y^*)|| + ||f(v_y^*) - \mu||)^2 \\
&= \mathbb{E}_{p(y)}||\mathbb{E}_{p(v_y|y)}f(v_y) - f(v_y^*)||^2 + \mathbb{E}_{p(y)}||f(v_y^*) - \mathbb{E}_{p(v)}f(v)||^2 \\
&\quad + 2\mathbb{E}_{p(y)}(||\mathbb{E}_{p(v_y|y)}f(v_y) - f(v_y^*)|| \cdot ||f(v_y^*) - \mathbb{E}_{p(v)}f(v)||) \\
&= \mathbb{E}_{p(y)}||\mathbb{E}_{p(v_y|y)}[f(v_y) - f(v_y^*)]||^2 + \mathbb{E}_{p(y)}||\mathbb{E}_{p(v)}[f(v_y^*) - f(v)]||^2 \\
&\quad + 2\mathbb{E}_{p(y)}(||\mathbb{E}_{p(v_y|y)}[f(v_y) - f(v_y^*)]|| \cdot ||\mathbb{E}_{p(v)}[f(v_y^*) - f(v)]||) \\
&\le \mathbb{E}_{p(y)}\mathbb{E}_{p(v_y|y)}||f(v_y) - f(v_y^*)||^2 + \mathbb{E}_{p(y)}\mathbb{E}_{p(v)}||f(v_y^*) - f(v)||^2 \\
&\quad + 2\mathbb{E}_{p(y)}(\mathbb{E}_{p(v_y|y)}||f(v_y) - f(v_y^*)|| \cdot ||f(v_y^*) - f(v)||) \\
&\le L^2\epsilon_1^2 + L^2\epsilon_2^2 + 2L^2\epsilon_1\epsilon_2 \\
&= L^2(\epsilon_1 + \epsilon_2)^2,
\end{aligned}$$

where v_y^* could be any node of class y, and $\epsilon_1 = \mathbb{E}_{p(v,y)}||v_y - v_y^*||$, $\epsilon_2 = \mathbb{E}_{p(y)}\mathbb{E}_{p(v)}||v - v_y^*||$.

$$\begin{aligned}
\hat{\mathcal{L}}_{\mathrm{CE}} &\ge \mathcal{L}_{\mathrm{NCE}} - 3\delta_{aug}^2 - 2\delta_{aug} - \log\frac{M}{K} - \frac{1}{2}\mathrm{Var}(f(v^-)|y) - \sqrt{\mathrm{Var}(f(v^0)|y)} - O(M^{-\frac{1}{2}}) - e\,\mathrm{Var}(\mu_y) \\
&\ge \mathcal{L}_{\mathrm{NCE}} - 3\delta_{aug}^2 - 2\delta_{aug} - \log\frac{M}{K} - \frac{1}{2}(\delta_{aug} + \sqrt{\mathrm{Var}(f(v_y^0)|y)})^2 - \sqrt{\mathrm{Var}(f(v_y^0)|y)} - O(M^{-\frac{1}{2}}) - eL^2(\epsilon_1 + \epsilon_2)^2 \\
&= \mathcal{L}_{\mathrm{NCE}} - 3\delta_{aug}^2 - 2\delta_{aug} - log\frac{M}{K} - \frac{1}{2}\delta_{aug}^2 - (\delta_{aug} + 1)\sqrt{\mathrm{Var}(f(v_y^0)|y))} \\
&\quad - \frac{1}{2}\mathrm{Var}(f(v_y^0)|y)) - O(M^{-\frac{1}{2}}) - eL^2(\epsilon_1 + \epsilon_2)^2 \\
&= \mathcal{L}_{\mathrm{NCE}} - g(\delta_{aug}) - \log\frac{M}{K} - O(M^{-\frac{1}{2}}),
\end{aligned}$$

where $g(\delta_{aug}) = \frac{23}{6}\delta_{aug}^2 + \frac{1}{2}L^2\epsilon_0^2 + eL^2(\epsilon_1 + \epsilon_2)^2 + 2\delta_{aug} + \frac{2}{3}\delta_{aug}L\epsilon_0 + (\delta_{aug} + 1)\sqrt{\frac{2}{3}\delta_{aug}^2 + \frac{4}{3}\delta_{aug}L\epsilon_0 + L^2\epsilon_0^2}$.

According to Oord et al. (2018), we get,

$$\begin{aligned}
I(f(v_i^1), f(v_i^2)) &\ge \log(M) - \mathcal{L}_{\mathrm{NCE}}, \\
\mathcal{L}_{\mathrm{NCE}} &\ge \log(M) - I(f(v_i^1), f(v_i^2)).
\end{aligned}$$

Therefore, we can reformulate Theorem 2.6 as below:

$$\begin{aligned}
\hat{\mathcal{L}}_{\mathrm{CE}} &\ge \log(M) - I(f(v_i^1), f(v_i^2)) - g(\delta_{aug}) - \log\frac{M}{K} - O(M^{-\frac{1}{2}}) \\
&= \log(K) - I(f(v_i^1), f(v_i^2)) - g(\delta_{aug}) - O(M^{-\frac{1}{2}}).
\end{aligned}$$

A.4. Proof of Corollary 3.3

Corallary 3.3 could be simply proved below:

$$\begin{aligned}\mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)||^2 &= \mathbb{E}_{p(v_i^1,v_i^2)}[(f(v_i^1)^T-f(v_i^2)^T)(f(v_i^1)-f(v_i^2))]\\ &= \mathbb{E}_{p(v_i^1,v_i^2)}[2-2f(v_i^1)^T f(v_i^2)]\\ &= 2-\frac{2}{N}tr((H^1)^T H^2)\\ &\overset{(1)}{=} 2-\frac{2}{N}\sum_i \theta_i\lambda_i'\lambda_i''.\end{aligned}$$

So $(\mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)||)^2 \le \mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)||^2 = 2-\frac{2}{N}\sum_i \theta_i\lambda_i'\lambda_i''$, then $\mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)|| \le \sqrt{2-\frac{2}{N}\sum_i \theta_i\lambda_i'\lambda_i''}$.

(1) is suggested by Liu et al. (2022), $tr((H^1)^T H^2)$ could be represented as $\sum_i \theta_i\lambda_i'\lambda_i''$.

As we know that,

$$\mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)|| \le \mathbb{E}_{p(v_i^1,v_i^2)}(||f(v_i^1)-f(v_i^0)|| + ||f(v_i^0)-f(v_i^2)||) \le 2\delta_{aug}.$$

Then, we can get:

$$2\delta_{aug} \ge \mathbb{E}_{p(v_i^1,v_i^2)}||f(v_i^1)-f(v_i^2)|| \ge \sqrt{2-\frac{2}{N}\sum_i \theta_i\lambda_i'\lambda_i''}. \tag{6}$$

A.5. Proof of Lemma B.1

From Stewart (1990), we know the following equation:

$$\Delta\lambda_i = \lambda_i^{'} - \lambda_i = u_i^T\Delta A u_i - \lambda_i u_i^T \Delta D u_i + O(||\Delta A||).$$

So we can calculate the difference between λ_i', λ_i'' and λ_i,

$$\begin{aligned}\Delta\lambda_i &= \sum_m(\sum_n u_i[n]\Delta A[m][n])u_i[m] - \lambda_i\sum_m u_i[m]\Delta D[m]u_i[m] + O(||\Delta A||)\\ &= \sum_{m,n} u_i[m]u_i[n]\Delta A[m][n] - \lambda_i\sum_{m,n} u_i[m]\Delta A[m][n]u_i[m] + O(||\Delta A||).\end{aligned}$$

And we can directly calculate $\lambda_i' - \lambda_i''$ as below:

$$\begin{aligned}\lambda_i' - \lambda_i'' &= \Delta\lambda_i' - \Delta\lambda_i''\\ &= \sum_{m,n} u_i[m]u_i[n]\Delta\widehat{A}[m][n] - \lambda_i\sum_{m,n} u_i[m]\Delta\widehat{A}[m][n]u_i[m]\\ &= \sum_{m,n} u_i[m]\Delta\widehat{A}[m][n](u_i[n] - \lambda_i u_i[m]).\end{aligned}$$

B. GCL Methods with Spatial and Spectral Augmentation

B.1. Spatial Augmentation

Most augmentation methods are applied to explicitly or implicitly increase mutual information while maintain high augmentation distance. GRACE simply adjusts this by changing the drop rate of features and edges. AD-GCL (Suresh et al., 2021) directly uses the optimization objective $min_{\{aug\}}max_{\{f\in F\}}I(f(v), f(aug(v)))$ to search for a stronger augmentation.

And GCA (Zhu et al., 2021) could always perform better than random drop. This is mainly because GCA calculates node importance and masks those unimportant to increase mutual information. Also they use p_τ as a cut-off probability, so for those unimportant features/edges, all of them share the same drop probability p_τ. By setting a large p_τ, GCA can reduce the drop probability for the least important features/edges and drop more relatively important ones to achieve a trade-off between mutual information and augmentation distance.

From Figure 4, we could clearly see that, as p_τ increases, augmentation distance and $\mathcal{L}_{\text{NCE}}$ are increasing, and leads to a better downstream performance, than when p_τ becomes too large, we got a trivial solution. And in the code provided by the author, p_τ is set to 0.7. So GCA performances well on downstream tasks not only because its adaptive augmentation, but also its modification on augmentation distance.

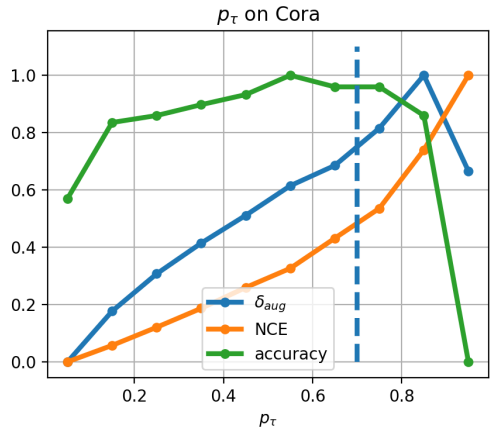

Figure 4. influence of p_τ on Cora (all the data are normalized for better visualization)

B.2. Spectral Augmentation

Furthermore, we can demonstrate that lots of spectral augmentations follow this schema to improve downstream performance. Liu et al. (2022) proposes that increasing the number of high-frequency drops leads to better performance. This is because high-frequency parts are associated with higher coefficients λ_i, so increasing the number of high-frequency drops can have a stronger increasement on δ_{aug}, resulting in better performance.

Lemma B.1 (Change of Spectrum)**.** *if we assume that $A' = A + \Delta A_1$, $A'' = A + \Delta A_2$, λ_i', λ_i'' is the i^{th} eigenvalue of A' and A'', respectively. $\Delta\widehat{A} = A' - A''$, and u_i is the corresponding eigenvector.*

$$\lambda_i' - \lambda_i'' = \sum_{m,n} u_i\left[m\right] \Delta\widehat{A}\left[m\right]\left[n\right]\left(u_i\left[n\right] - \lambda_i u_i\left[m\right]\right).$$

Lemma B.1 is proved in Appendix A.5. Lin et al. (2022) propose to maximize the spectral difference between two views, but Lemma B.1 shows that difference between spectrum is highly correlated with the original magnitude, so it is actually encouraging more difference in large $|\lambda_i|$. But rather than just drop information, they try to improve the spectrum of first view, and decrease the other view. if we simply assume $\lambda_i' = \lambda_i + n$, $\lambda_i'' = \lambda_i - n$, then $\lambda_i'\lambda_i'' = \lambda_i^2 - n^2 \leq \lambda_i^2$, so this could also be explained by augmentation distance increasement.

C. Further Explanation

C.1. Value of θs

As defined by Liu et al. (2022), θs are actually linear combination of the eigenvalues of adjacency matrix $\boldsymbol{A}$. To demonstrate what θs actually are, we first focus on the assumption below.

Assumption C.1 (High-order Proximity)**.** $\boldsymbol{M} = w_0 + w_1\boldsymbol{A} + w_2\boldsymbol{A^2} + \cdots + w_q\boldsymbol{A^q}$, where $\boldsymbol{M} = X^1W \cdot W^T(X^2)^T$, $\boldsymbol{A^i}$ means matrix multiplications between i $\boldsymbol{A}$s, and w_i is the weight of that term.

Where X^1, X^2 indicates the feature matrix of graph $\mathcal{G}^1, \mathcal{G}^2$, W stands for the parameter of the model, so $\boldsymbol{M} = X^1W \cdot W^T(X^2)^T$ means embedding similarity between two views, and could be roughly represented by the weighted sum of different orders of $\boldsymbol{A}$. Furthermore, we have that:

$$\begin{cases} w_0 + w_1\lambda_1 + \cdots + w_q\lambda_1^q = \theta_1 \\ w_0 + w_1\lambda_2 + \cdots + w_q\lambda_2^q = \theta_2 \\ \cdots \\ w_0 + w_1\lambda_N + \cdots + w_q\lambda_N^q = \theta_N, \end{cases}$$

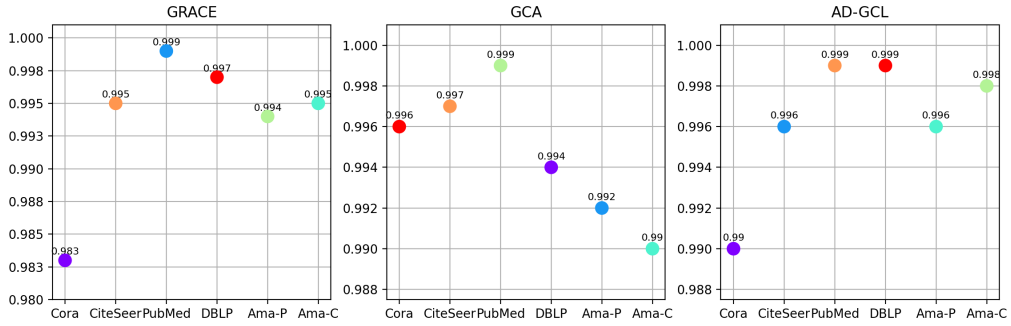

Figure 5. Percentage of positive θ

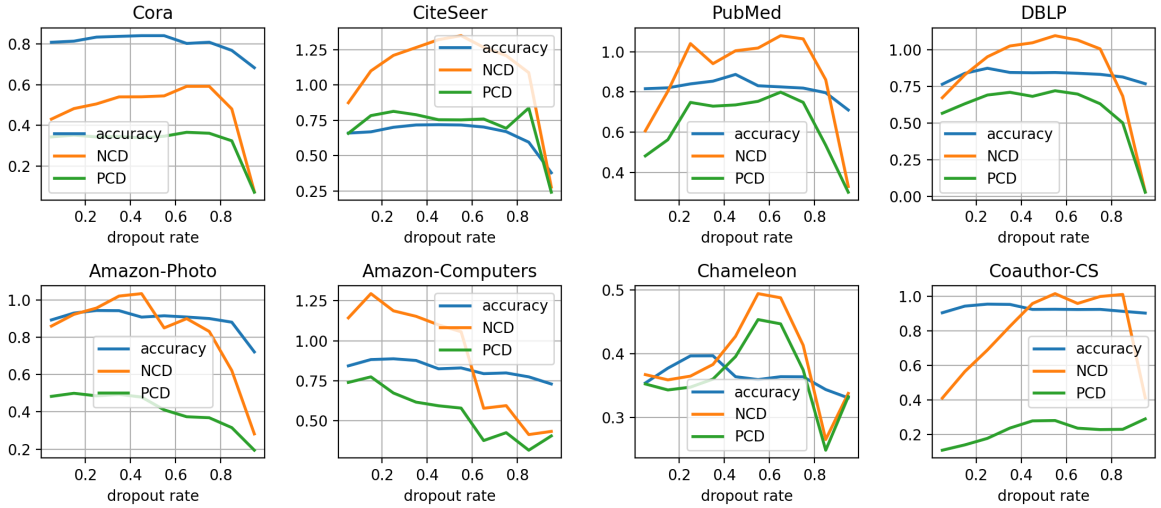

Figure 7. More experiments on PCS and NCS, the detailed data is slightly different due to randomness, but it shows similar tendency

where $\lambda_1, ..., \lambda_N$ is N eigenvalues of the adjacency matrix $\boldsymbol{A}$.

So we know that θs are actually linear combination of λs. As the model is trained to minimize $\mathcal{L}_{\text{NCE}}$, θs are expected to increase, and we can simply set $w_0, w_2, ..., w_{2(q/2)}$ to be positive and other w_i to 0, then we can get θs that are all positive, and the model would easily find better ws.

We can say that in the training process, θs are mostly positive, and the experiments shown in Figure 10 indicate it true.

C.2. PCS, NCS and Downstream Performance

More experiments are conducted on various of datasets to show that our finding could be generalized rather that limited to few datasets in Figure 7. They show similar tendency that with the dropout rate increasing, the downstream accuracy increases first and decreases when the augmentation is too strong. And those experiments show that when the downstream accuracy increases, the positive center distance are sometimes increasing, and the better downstream performance is mainly caused by the increasing distance of negative center.

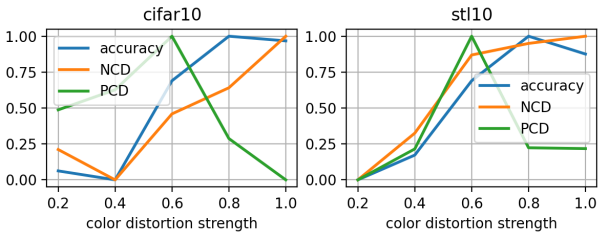

Figure 6. relationship of PCD, NCD and performance on images

We also conduct experiments on images to verify our

theory, we control the magnitude of augmentation by adjusting the color distortion strength, and the results are normalized by Min-Max normalization. From Figure 6, we can observe that the downstream performance is also closely correlated with negative center distance especially when the color distortion strength changes from 0.2 to 0.6 the positive center distance increases while downstream performance is increasing, but when color distortion is greater than 0.6 the positive center distance also tends to decrease. This aligns with our finding in Theorem 2.4 that with the augmentation gets stronger the negative center distance is increasing while the positive center distance does not change in specific pattern. Also the color distortion is not strong enough to change the label information, so the downstream performance keeps increasing with stronger augmentation.

C.3. Change of δ_{aug} and Label Consistency

To verify how is δ_{aug} changing with stronger augmentation, we use drop rate of edges/features as data augmentation, and find that when the drop rate increases, δ_{aug} also tends to increase. Also to verify the view invariance assumption, we first train a well conditioned model and use its prediction as $p(v_i)$, then we change the drop rate and calculate new $p'(v_i)$, then we can observe from Figure 8 that though the KL divergence is increasing with drop rate, it remains quite small magnitude, so the label is consistent with data augmentation.

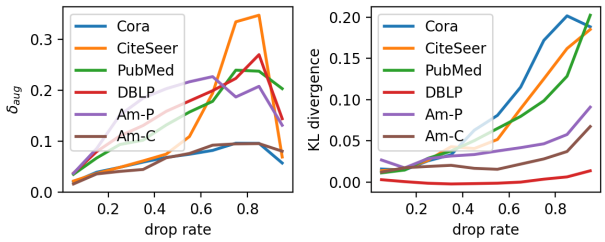

Figure 8. relationship between δ_{aug}, KL divergence and augmentation

C.4. Augmentation Free Methods

In this paper, we mainly discuss how the augmentation will affect the contrastive performance, but actually, GCL methods with or without augmentation aim for the same, they both try to align intra-class nodes and separate inter-class nodes. However, during contrastive learning, label information is not accessible, so they use different methods to get intra-class nodes.

- GCL methods with augmentation create intra-class nodes by data augmentation, so it is necessary to control the strength of augmentation to ensure label consistency. But augmentation brings more flexibility, you can freely change the topology and feature of the graph, so a good GCL method with augmentation always require a well-designed data augmentation. This could lead to great performance, but the they require more time consumption and overlook the unique properties of graphs.

- GCL methods without augmentation instead find intra-class nodes by other methods. For example, GMI (Peng et al., 2020) and iGCL (Li et al., 2023) try to align the anchor with its neighbors and similar nodes (which are more likely to hold the same label), and SUGRL (Mo et al., 2022) create intra-class nodes by two different embedding generation methods. Label based methods like SupCon (Khosla et al., 2020) directly align samples with the same class. These methods take advantage of the inherent property of the dataset such as homophily and the similarity between intra-class samples but the positive sample construction is not as flexible as augmentation.

Therefore, GCL methods with or without methods are inherently the same, they both align positive samples, and they create the positive samples differently. Our analysis focus on the difference between two positive samples, so the analysis can also be employed on those methods.

C.5. change on positive/negative pair similarity

The InfoNCE loss $\mathcal{L}_{\text{NCE}}$ can be written as $\mathcal{L}_{\text{NCE}} = \mathbb{E}_{p(v_i^1, v_i^2)} \mathbb{E}_{p(v_i^-)} \left[-\log \frac{\exp(f(v_i^1)^T f(v_i^2))}{\sum_{\{v_i^-\}} \exp(f(v_i^1)^T f(v_i^-))} \right]$, and when we perform stronger augmentation, $f(v_i^1)^T f(v_i^2)$ would be hard to maximize, and the model will try to minimize $f(v_i^1)^T f(v_i^-)$ harder. From Figure 9, when the augmentation gets stronger, negative and positive pair similarity both decreases, so the class separating performance is enhanced.

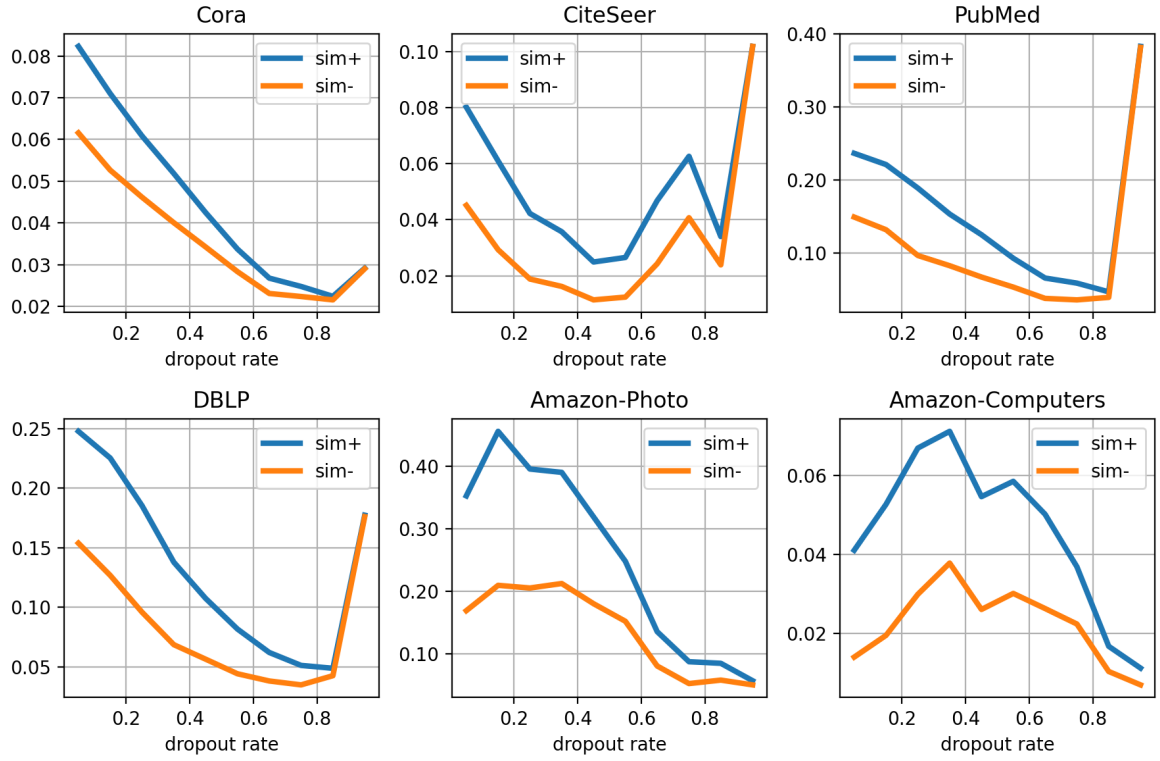

Figure 9. $\mathrm{sim}+$ represents the positive pair similarity $f(v_i^1)^T f(v_i^2)$, and $\mathrm{sim}-$ is negative pair similarity $f(v_i^1)^T f(v_i^-)$, the x-axis stands for dropout rate on edges

D. Experiments

D.1. Datasets and Experimental Details

We choose the six commonly used Cora, CiteSeer, PubMed, DBLP, Amazon-Photo and Amazon-Computer for evaluation. The first four datasets are citation networks (Sen et al., 2008; Yang et al., 2016; Bojchevski & Günnemann, 2017), where nodes represent papers, edges are the citation relationship between papers, node features are comprised of bag-of-words vector of the papers and labels represent the fields of papers. In Amazon-Photos and Amazon-Computers (Shchur et al., 2018), nodes represent the products and edges means that the two products are always bought together, the node features are also comprised of bag-of words vector of comments, labels represent the category of the product.

We use 2 layers of GCNConv as the backbone of encoder, we use feature/edge drop as data augmentation, the augmentation is repeated randomly every epoch, and InfoNCE loss is conducted and optimized by Adam. After performing contrastive learning, we use logistic regression for downstream classification the solver is liblinear, and in all 6 datasets we randomly choose 10% of nodes for training and the rest for testing.

Table 2. Dataset statistics

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
DBLP	17,716	105,734	1,639	4
Amazon-Photo	7,650	119,081	745	8
Amazon-Computers	13,752	245,861	767	10

Table 3. Dataset download links

Dataset	Download Link
Cora	https://github.com/kimiyoung/planetoid/raw/master/data
Citeseer	https://github.com//kimiyoung/planetoid/raw/master/data
Pubmed	https://github.com/kimiyoung/planetoid/raw/master/data
DBLP	https://github.com/abojchevski/graph2gauss/raw/master/data/dblp.npz
Amazon-Photo	https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_photo.npz
Amazon-Computers	https://github.com/shchur/gnn-benchmark/raw/master/data/npz/amazon_electronics_computers.npz

And the publicly available implementations of Baselines can be found at the following URLs:

- GCN: https://github.com/tkipf/gcn
- GAT: https://github.com/PetarV-/GAT
- GRACE: https://github.com/CRIPAC-DIG/GRACE
- GCA: https://github.com/CRIPAC-DIG/GCA
- AD-GCL: https://github.com/susheels/adgcl
- GCS: https://github.com/weicy15/GCS
- SpCo: https://github.com/liun-online/SpCo

D.2. Hyperparameter Setting

Table 4. Hyperparameters settings

Dataset	Learning rate	Weight decay	num layers	τ	Epochs	Hidden dim	Activation
Cora	5^{-4}	10^{-6}	2	0.4	200	128	ReLU
Citeseer	10^{-4}	10^{-6}	2	0.9	200	256	PReLU
Pubmed	10^{-4}	10^{-6}	2	0.7	200	256	ReLU
DBLP	10^{-4}	10^{-6}	2	0.7	200	256	ReLU
Amazon-Photo	10^{-4}	10^{-6}	2	0.3	200	256	ReLU
Amazon-Computers	10^{-4}	10^{-6}	2	0.2	200	128	RReLU

The hyperparameter settings is shown in Table 4, other hyperparameter correlated to only one algorithm are set the same as the original author. The hyperparameter in our methods retain rate ξ and spectrum change magnitude α, we select them from 0.05 to 0.45 and from -0.1 to 0.01, respectively.

D.3. Changes on the Spectrum

From Figure 10(a), we can see that, when the algorithm is training, θs are mostly increasing gradually, and when we perform spectrum augmentation, θs will not increase as before, increasing number of θ is close even smaller to decreasing ones. Then we take a step back on those decreasing ones, result in increasing θs again in the next epoch. Therefore, what we do is actually perform augmentation to maximize augmentation distance first, then maximize the mutual information after spectrum augmentation. The idea is actually similar AD-GCL, but we use θs to indicate whether the augmentation is too much, so we get a more reasonable result. Figure 10(b) and (c) shows that as the training goes, the change on larger magnitude eigenvalues are also more significant, causing the spectrum to be smoother.

Also there is one thing to notice that when we perform spectrum smoothen method, we are indirectly changing the edge weights, causing the augmentation being weaker or stronger as drop an edge with weight of 1 is different than drop an edge with weight $1 + noise$. To reduce its influence, we conduct extra augmentation or recovery based on the average weight change.

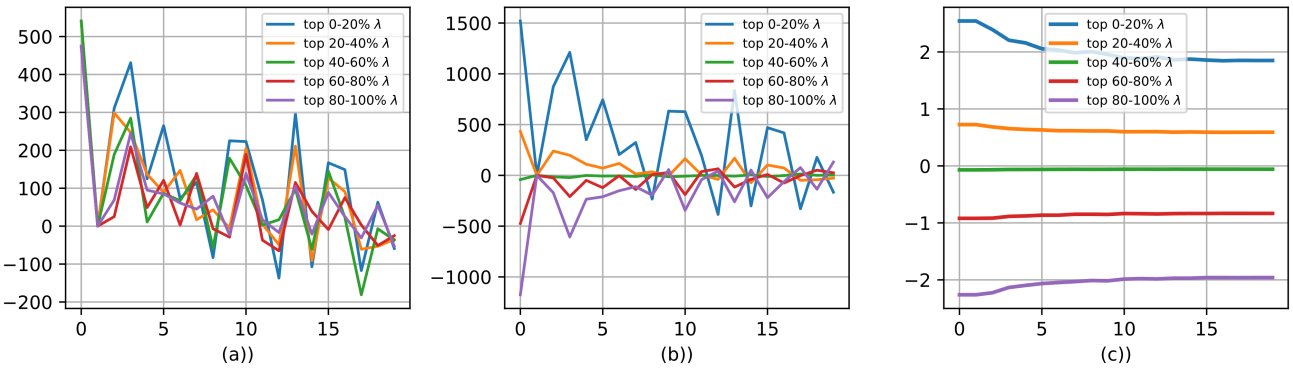

Figure 10. As we perform spectrum augmentation each 10 epochs, the x-axis is epoch/10, the y-axis of the left figure is number of decreasing λs minus number of increasing λs; for the middle one, y-axis stands for how much λs averagely decreases; and the right one is the average value of λ.

D.4. Center Distance

As we mentioned earlier, GCL mainly contributes to downstream tasks by increasing the negative center distance while maintaining a relatively small distance to the positive center. We propose two methods: one that increases mutual information between two views while keeping a high augmentation distance by masking more unimportant edges or features. This allows the model to learn more useful information, which forces nodes close to its positive center. The other method tries to increase augmentation distance while maintaining a relatively high mutual information, so it may not learn as much useful information. However, by increasing the augmentation distance, it forces the model to separate nodes from different classes further apart. In summary, the first method brings nodes of the same class closer together, while the second method separates nodes from different classes further apart just as shown in Figure 11.

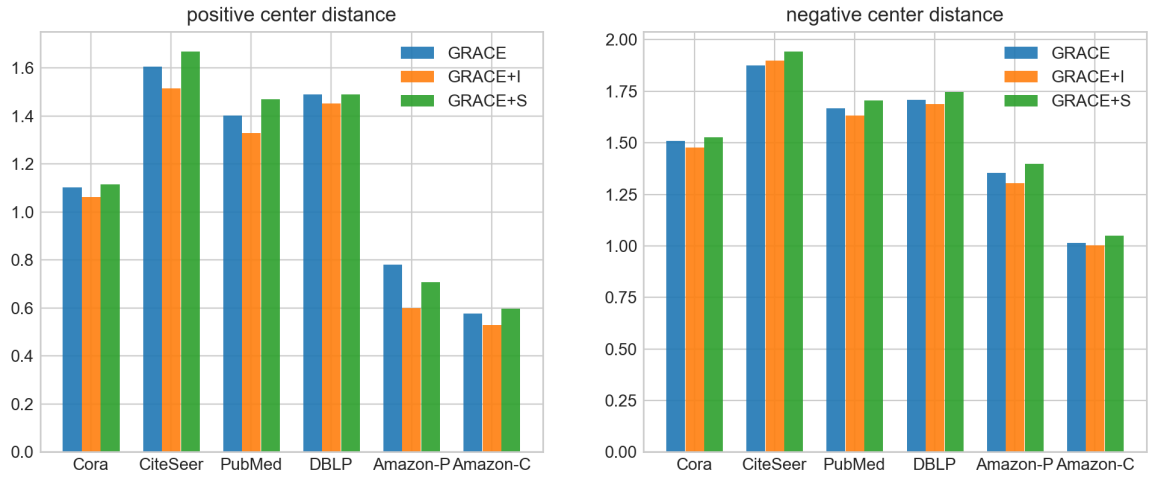

Figure 11. distance of nodes between its positive center and negative center, GRACE stands for the pure GRACE, GRACEI stands for GRACE with information augmentation, and GRACES stands for GRACE with spectrum augmentation

D.5. Hyperparameter Sensitivity

Analysis of retain rate. Retain rate controls how many important features/edges we kept, and how many unimportant ones dropped. We can see from Figure 13 that AD-GCL benefits from a larger retain rate as it is designed to minimize the mutual information, and lots of vital structures are dropped. And large datasets like PubMed, DBLP benefits less, it is mainly because a graph with more edges are more likely to maintain enough information than graph with little edges. For example,

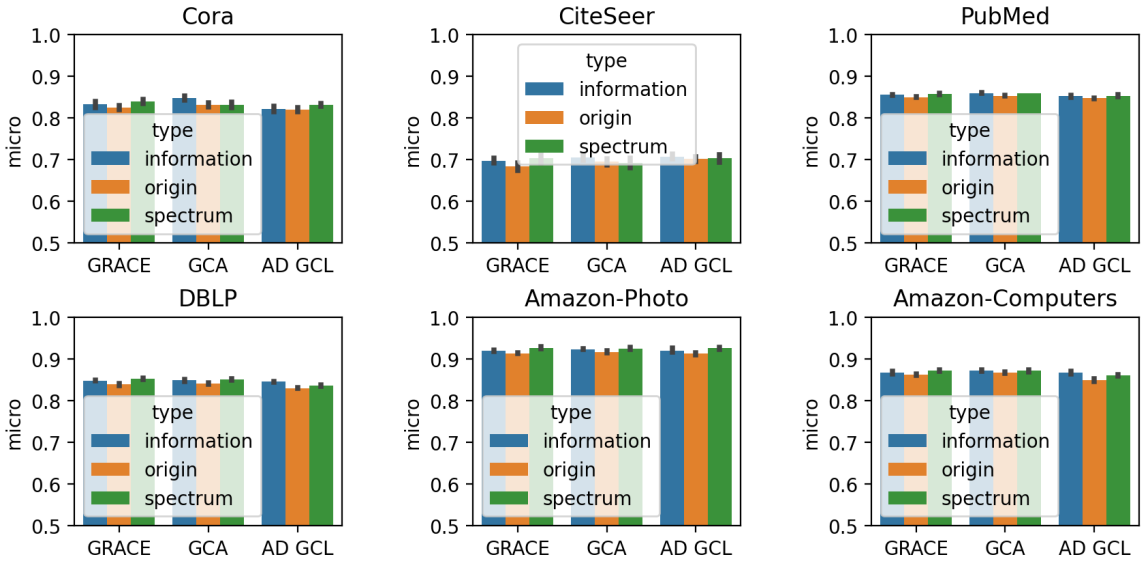

Figure 12. The error bar of algorithms

after a 30% dropout on edges, a graph with 1000 edges would still kept enough information for downstream tasks, but a graph with 10 edges would probably lose some vital information.

Analysis of α. α controls how much $|\lambda|$ will decrease, as we take a step back when the $|\lambda|$ decreases too much, the hyperparameter α does not matter so much. But as shown in Figure 14, it still performs more steady on large graphs as a wrong modification on a single λ matters less than on small graphs.

D.6. Time Complexity and Error Bar

Table 5. The time consumption (seconds) of algorithms

	Cora	CiteSeer	PubMed	DBLP	Amazon-P	Amazon-C
GRACE	8.02	10.08	62.37	56.89	19.05	28.71
GRACE+I	10.74	13.49	68.97	62.8	22.67	29.61
GRACE+S	9.61	12.46	78.11	69.44	21.13	36.94

From Table 5, we can observe that the information augmentation method achieve better performance with only few more time consuming, this is mainly because we do not calculate the importance of features/edges every epoch like GCS (Wei et al., 2023), we only calculate it once and use the same importance for the following training. However, the spectrum augmentation method consumes more time on large graphs like PubMed and DBLP, this is mainly we explicitly change the spectrum and calculate the new adjacency matrix, which could be replaced by some approximation methods but to prevent interference from random noise and show that Theorem 3.2 is meaningful, we still conduct eigen decomposition, but it is worth mentioning that the time complexity could be reduced by some approximation methods (Liu et al., 2022).

The error bar is reported in Figure 12, the experiments are conducted repeatedly for 10 times, we can observe that both the information augmentation and spectrum augmentation achieve better results, and they performs stably.

D.7. Combination of Information&Spectrum Augmentation

We combine the information augmentation and spectrum augmentation methods and show the result in Table 6. We can observe that combine the two methods achieve the best performance. We can observe that for larger and denser graphs, the information could still be well-preserved even after strong augmentation, rendering the augmentation less powerful compared to smaller graphs. And the spectrum augmentation modify the spectrum based on InfoNCE loss which will be

Figure 13. accuracy on downstream tasks with different retain rate

Figure 14. Accuracy on downstream tasks with different α

Table 6. Combine the information&Spectrum Augmentation (GRACE+IS)

	Cora	CiteSeer	PubMed	DBLP	Amazon-P	Amazon-C
GRACE	82.52±0.75	70.44±1.49	84.97±0.17	84.01±0.34	91.17±0.15	86.36±0.32
GRACE+I	83.78±1.08	72.89±0.97	84.97±0.14	84.80±0.17	91.64±0.21	84.54±0.53
GRACE+S	83.61±0.85	72.83±0.47	85.45±0.25	84.83±0.18	91.99±0.35	87.67±0.33
GRACE+IS	**84.58±0.79**	**72.94±0.52**	**85.62±0.17**	**84.87±0.25**	**92.04±0.32**	**87.73±0.41**

more stable on larger graphs, so it help more significantly in larger graphs.

E. Related Work

Graph Contrastive Learning. Graph Contrastive Learning has shown its superiority and lots of researcher are working on it. DGI (Veličković et al., 2018) contrasts between local node embeddnig and the global summary vector; GRACE (Zhu et al., 2020), GCA (Zhu et al., 2021) and GraphCL (You et al., 2020) randomly drop edges and features; AD-GCL (Suresh et al., 2021) and InfoGCL Xu et al. (2021) learn an adaptive augmentation with the help of different principles. In theoretical perspective, Liu et al. (2022) correlates the InfoNCE loss with graph spectrum, and propose that augmentation should be more focused on high frequency parts. Guo et al. (2023) further discuss that contrastive learning in graph is different with images. Lin et al. (2022) thinks that augmentation maximize the spectrum difference would help, and Yuan et al. (2022) analyse GCL with information theory.

Contrastive Learning Theory. By linking downstream classification and contrastive learning objectives, Arora et al. (2019) propose a theoretical generalization guarantee. Ash et al. (2021) further explore how does the number of negative samples influence the generalization. And Tian et al. (2020); Wang et al. (2022a) further discuss what kind of augmentation is better for downstream performance. Then Wang & Isola (2020) propose that perfect alignment and uniformity is the key to success while Wang et al. (2022b) argues augmentation overlap with alignment helps gathering intra-class nodes by stronger augmentation. However, Saunshi et al. (2022) show that augmentation overlap is actually quite rare while the downstream performance is satisfying. So the reason why contrastive learning helps remains a mystery, in this paper we propose that the stronger augmentation mainly helps contrastive learning by separating inter-class nodes, and different from previous works (Wang et al., 2022b; Wang & Isola, 2020; Huang et al., 2021), we do not treat perfect alignment as key to success, instead a stronger augmentation that draw imperfect alignment could help.