# Revisiting Graph Contrastive Learning from the Perspective of Graph Spectrum

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Graph Contrastive Learning (GCL), learning the node representations by augmenting graphs, has attracted considerable attentions. Despite the proliferation of various graph augmentation strategies, some fundamental questions still remain unclear: what information is essentially encoded into the learned representations by GCL? Are there some general graph augmentation rules behind different augmentations? If so, what are they and what insights can they bring? In this paper, we answer these questions by establishing the connection between GCL and graph spectrum. By an experimental investigation in spectral domain, we firstly find the General grAph augMEntation (GAME) rule for GCL, i.e., the difference of the high-frequency parts between two augmented graphs should be larger than that of low-frequency parts. This rule reveals the fundamental principle to revisit the current graph augmentations and design new effective graph augmentations. Then we theoretically prove that GCL is able to learn the invariance information by contrastive invariance theorem, together with our GAME rule, for the first time, we uncover that the learned representations by GCL essentially encode the low-frequency information, which explains why GCL works. Guided by this rule, we propose a spectral graph contrastive learning module (SpCo), which is a general and GCL-friendly plug-in. We combine it with different existing GCL models, and extensive experiments well demonstrate that it can further improve the performances of a wide variety of different GCL methods.

## 1 Introduction

Graph Neural Networks (GNNs) learn the node representations in a graph mainly by message passing. GNNs have attracted significant interest and found many applications [11, 22, 14]. Training the high quality GNNs heavily relies on task-specific labels, while it is well known that manually annotating nodes in graphs is costly and time-consuming [10]. Therefore, Graph Contrastive Learning (GCL) is developed as a typical technique for self-supervised learning without the explicit usage of labels [27, 9, 35].

The traditional GCL framework (Fig. 1 (a)) mainly includes three components: graph augmentation, graph representation learning by an encoder, and contrastive loss. In essence, GCL aims to maximize agreement between augmentations to learn invariant representations [35]. Typical GCL methods have sought to elaborately design different graph augmentation strategies. For example, the heuristic based methods including node or edge dropping [31], feature masking [32], and diffusion [9]; and the learning based methods including InfoMin [25, 29], disentanglement [13], and adversarial training [30].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

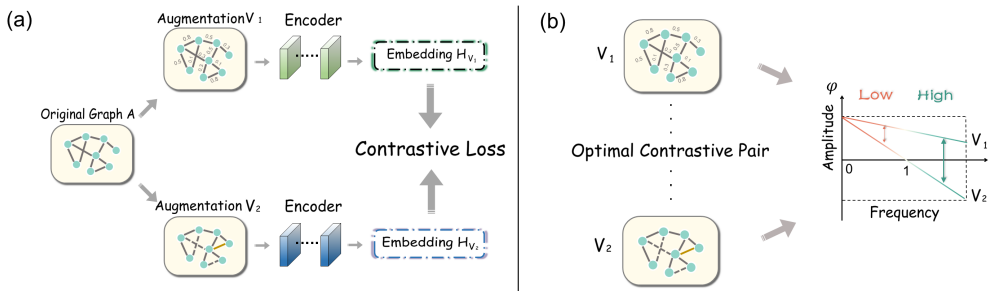



Figure 1: (a) The general framework of GCL. (b) The illustration of our findings in the empirical study (Section 3). If two contrasted graphs have a larger margin between high-frequency part than low-frequency part, they will boost the GCL. We call such two graphs as optimal contrastive pair.

Although various graph augmentation strategies are proposed, the fundamental augmentation mechanism is not well understood. *What information should we preserve or discard in an augmented graph? Are there some general rules across different graph augmentation strategies? How to use those general rules to validate and improve the current GCL methods?* This paper explores those questions.

Essentially, an augmented graph is obtained by changing some components in the original graph and thus strength of frequencies [19] in graph spectrum. This natural and intuitive connection between graph augmentation and graph spectrum inspires us to explore the effectiveness of augmentations from the spectral domain. We start with an empirical study (Section 3) to understand the importance of low-frequency and high-frequency information in GCL. Our findings indicate that both the lowest-frequency information and the high-frequency information are important in GCL. Retaining more high-frequency information is particularly helpful to improve the performance of GCL. However, as shown in Fig. 1 (b), the way of handling high-frequency information in two contrasted graphs $V_1$ and $V_2$ should be different, which can be finally summarized as a general graph augmentation (GAME) rule: the difference of amplitudes of high frequencies in two contrasted graphs should be larger than that of low frequencies.

To explain the GAME rule, we need to understand what information is encoded into the learned representations by GCL. We propose the contrastive invariance (Theorem 1), which, for the first time, theoretically proves that GCL is able to learn the invariance information from two contrasted graphs. Meanwhile, as can be seen in Fig. 1 (b), because the difference of amplitudes of lowest-frequency information is much smaller than that of high-frequency information, the lowest-frequency information will be the approximately invariant pattern between the two graphs $V_1$ and $V_2$. Therefore, with such two augmentations $V_1$ and $V_2$, we can conclude that the information learned by GCL is mainly the low-frequency information, whose usefulness has been well demonstrated [6]. This not only explains why GCL works, but also provides a clear and concise demonstration of which augmentation strategy is better, as verified by the experiments in Section 4.

Based on our findings and theoretical analysis, we define two augmentations satisfying the GAME rule are called an optimal contrastive pair. Then, we propose a novel **sp**ectral graph **co**ntrastive learning (SpCo), a general GCL framework, which is able to boost existing GCL methods with optimal contrastive pairs. Specifically, to ensure that the learned augmented graph is an optimal contrastive pair with the original adjacency matrix, we need to make the amplitude of its high frequency ascend while keeping the low frequency the same as the original structure. We model this process as an optimization objective based on matrix perturbation theory, which can be solved by Sinkhorn's Iteration [23] and finally obtain the augmented structure used for the following target GCL model.

Our contributions are summarized as follows. Firstly, to the best of our knowledge, we are the first attempt to fundamentally explore the augmentation strategies for GCL from spectral domain. We not only reveal the general graph augmentation rule behind different augmentation strategies, but

also explain why GCL works by proposing the contrastive invariance theorem. We show that the augmentation rule provides a novel insight to estimate the current augmentation strategies. Secondly, we propose a novel concept optimal contrastive pair and theoretically derive a general GCL framework SpCo, which is able to improve the performance of existing GCL methods. Last, we choose three typical GCL methods as target methods, and plug SpCo into them. Extensive experiments on five public datasets demonstrate that the plugged SpCo can consistently gain improvements compared with those original target methods.

## 2 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \xi)$ represent an undirected attributed graph, where $\mathcal{V}$ is the set of N nodes and $\xi \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. All edges formulate an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where $\mathbf{A}_{ij} \in \{0,1\}$ denotes the relation between nodes $i$ and $j$ in $\mathcal{V}$. The node degree matrix $\boldsymbol{D} = diag(d_1, \ldots . d_n)$, where $d_i = \sum_{j \in \mathcal{V}} \boldsymbol{A}_{ij}$ is the degree of node $i \in \mathcal{V}$. Graph $\mathcal{G}$ is often associated with a node feature matrix $\boldsymbol{X} = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{N \times d}$, where $x_i$ is a $d$ dimensional feature vector of node $i \in \mathcal{V}$. Let $\mathcal{L} = \boldsymbol{I_n} - \boldsymbol{A}$ be the unnormalized graph Laplacian of $\mathcal{G}$. If we set symmetric normalized adjacency matrix as $\hat{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}}$, then $\hat{\mathcal{L}} = \boldsymbol{I_n} - \hat{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}} (\boldsymbol{D} - \boldsymbol{A}) \boldsymbol{D}^{-\frac{1}{2}}$ is the symmetric normalized graph Laplacian.

Since $\hat{\mathcal{L}}$ is symmetric normalized, its eigen-decomposition is $\boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^\top$, where $\boldsymbol{\Lambda} = diag(\lambda_1, \ldots, \lambda_N)$ and $\boldsymbol{U} = [\boldsymbol{u_1}^\top, \ldots, \boldsymbol{u_N}^\top] \in \mathbb{R}^{N \times N}$ are the eigenvalues and eigenvectors of $\hat{\mathcal{L}}$, respectively. Without loss of generality, assume $0 \leq \lambda_1 \leq \cdots \leq \lambda_N < 2$ (where we approximate $\lambda_N \approx 2$ [11]). Denote by $\mathcal{F}_{\mathcal{L}} = \{\lambda_1, \ldots, \lambda_{\lfloor N/2 \rfloor}\}$ the amplitudes of *low-frequency components* and by $\mathcal{F}_{\mathcal{H}} = \{\lambda_{\lfloor N/2 \rfloor + 1}, \ldots, \lambda_N\}$ the amplitudes of *high-frequency components*. The **graph spectrum** is defined as these amplitudes of different frequency components of a given graph, denoted as $\phi(\lambda)$, indicating which parts of frequency are enhanced or weaken [19]. Additionally, we rewrite $\hat{\boldsymbol{\mathcal{L}}} = \lambda_1 \cdot \boldsymbol{u_1} \boldsymbol{u_1}^\top + \cdots + \lambda_N \cdot \boldsymbol{u_N} \boldsymbol{u_N}^\top$, where we define term $\boldsymbol{u_i} \boldsymbol{u_i}^\top \in \mathbb{R}^{N \times N}$ as the eigenspace related to $\lambda_i$, denoted as $\boldsymbol{S_i}$.

**Graph Contrastive Learning** (GCL) [27, 9, 32] aims to learn discriminative embeddings without supervision, whose pipeline is shown in Fig. 1(a). We summarize the representative GCL in Appendix E. Specifically, two augmentations are randomly extracted from $\boldsymbol{A}$ in a predefined way and are encoded by GCN [11] to obtain the node embeddings under these two augmentations. Then, for one target node, its embedding in one augmentation is learned to be close to the embeddings of its positive samples in the other augmentation and be far away from those of its negative samples. Models built in this way are capable of discriminating similar nodes from dissimilar ones. For example, some graph contrastive methods [35, 31, 34] use classical InfoNCE loss [18] as the optimization objective as follows.

$$\mathcal{L}(\boldsymbol{h_i^{V_1}}, \boldsymbol{h_i^{V_2}}) = \log \frac{\exp(\theta(\boldsymbol{h_i^{V_1}}, \boldsymbol{h_i^{V_2}})/\tau)}{\exp(\theta(\boldsymbol{h_i^{V_1}}, \boldsymbol{h_i^{V_2}})/\tau) + \sum\limits_{k \neq i} \exp(\theta(\boldsymbol{h_i^{V_1}}, \boldsymbol{h_k^{V_2}})/\tau)}, \tag{1}$$

where $\boldsymbol{h_i^{V_1}}$ and $\boldsymbol{h_i^{V_2}}$ are the embeddings of node $i$ under augmentations $V_1$ and $V_2$, respectively, $\theta$ is the similarity metric, such as cosine similarity, and $\tau$ is a temperature parameter. The total loss is $\mathcal{L}_{total} = \sum\limits_i \frac{1}{2} \Big( \mathcal{L}(\boldsymbol{h_i^{V_1}}, \boldsymbol{h_i^{V_2}}) + \mathcal{L}(\boldsymbol{h_i^{V_2}}, \boldsymbol{h_i^{V_1}}) \Big)$.

## 3 Impact of Graph Augmentation: An Experimental Investigation

In this section, we aim to explore what information should be considered for a better graph augmentation from the perspective of graph spectrum. Specifically, following the framework shown in Fig. 1(a), the augmentation $V_1$ is the adjacency matrix $\boldsymbol{A}$, and the augmentation $V_2$ is the generated $\boldsymbol{V}$. Then, we utilize a shared GCN with one layer as the encoder, and the contrastive loss is the InfoNCE loss as in Eq. (1).

**Generating augmentation $\boldsymbol{V}$** We construct the augmented graph by extracting information with different frequencies from the original graph, so that we can analyze the effect of different information. This process is shown in Fig. 2. Specifically, we divide the eigenvalues of $\boldsymbol{\mathcal{L}}$ into $\mathcal{F}_{\mathcal{L}}$ and $\mathcal{F}_{\mathcal{H}}$ parts, and conduct augmentations in these two parts, respectively. Taking the augmentation in $\mathcal{F}_{\mathcal{L}}$ for example, we keep the high-frequency part as $\boldsymbol{u_{N/2}u_{N/2}^\top} + \cdots + \boldsymbol{u_N u_N^\top}$. Then, we gradually add the eigenspaces in $\mathcal{F}_{\mathcal{L}}$ back with rates $[20\%, 40\%, 60\%, 80\%]$, starting from the lowest frequency. Therefore, $\boldsymbol{V}$ augmenting 20% in $\mathcal{F}_{\mathcal{L}}$ is $\boldsymbol{u_1 u_1^\top} + \cdots + \boldsymbol{u_{0.2N} u_{0.2N}^\top} + \boldsymbol{u_{N/2} u_{N/2}^\top} + \cdots + \boldsymbol{u_N u_N^\top}$. Similarly, $\boldsymbol{V}$ augmenting 20% in $\mathcal{F}_{\mathcal{H}}$ is $\boldsymbol{u_1 u_1^\top} + \cdots + \boldsymbol{u_{N/2} u_{N/2}^\top} + \boldsymbol{u_{(N+1)/2} u_{(N+1)/2}^\top} + \cdots + \boldsymbol{u_{0.7N} u_{0.7N}^\top}$. Therefore, the graph spectrum of $\boldsymbol{V}$, $\phi_{\boldsymbol{V}}(\lambda) = 1, \forall \lambda \in [0, 2]$.

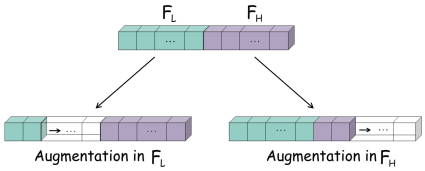
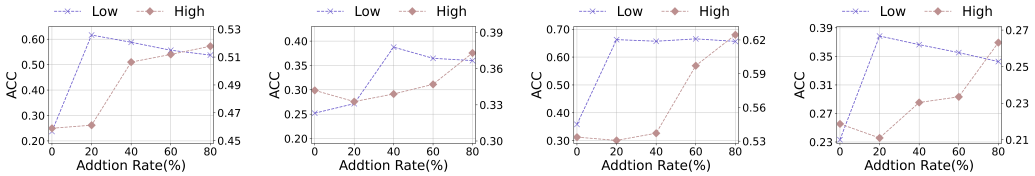



Figure 2: The generation of $\boldsymbol{V}$.




(a) Cora (b) Citeseer (c) BlogCatalog (d) Flickr

Figure 3: The results of case study on four datasets. The x-axis means different addition rate of different frequency interval, and y-axis means the performance on ACC. The performance of augmentations in $\mathcal{F}_{\mathcal{L}}$ are plotted on the left y-axis, and in $\mathcal{F}_{\mathcal{H}}$ are plotted on the right y-axis.

**Results and observation** We conduct the node classification on four datasets: Cora, Citeseer [11], BlogCatalog, and Flickr [16]. The accuracy is shown in Fig. 3. For each dataset, (1) when augmentation occurs in $\mathcal{F}_{\mathcal{L}}$, the performance achieves the best only when the lowest part of $\mathcal{F}_{\mathcal{L}}$ is maintained; (2) with more high-frequency information added, the performance generally rises. With the observations, we plot the graph spectrums of $\boldsymbol{A}$ and $\boldsymbol{V}$ in Fig. 4. We can conclude that in generated $\boldsymbol{V}$, when more lowest frequencies are kept, the difference of amplitude, i.e., the graph specturm, in $\mathcal{F}_{\mathcal{L}}$ between $\boldsymbol{A}$ and $\boldsymbol{V}$ becomes smaller; when more frequencies in $\mathcal{F}_{\mathcal{H}}$ are involved, the margin of graph spectrum in $\mathcal{F}_{\mathcal{H}}$ between $\boldsymbol{A}$ and $\boldsymbol{V}$ becomes larger. Inspired by the findings, we propose the general **G**raph **A**ug**ME**ntation rule, called GAME rule, as follow.

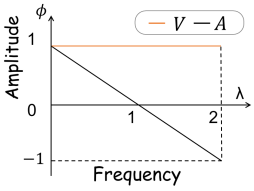



Figure 4: The spectrum of $\boldsymbol{A}$ and $\boldsymbol{V}$.

**The General Graph Augmentation Rule**

Given two random augmentations $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$, their graph spectrums are $\phi_{V_1}(\lambda)$ and $\phi_{V_2}(\lambda)$. Then, $\forall\ \lambda_m \in [1,2]$ and $\lambda_n \in [0,1]$, $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ are an effective pair of graph augmentations if the following condition is satisfied:

$$|\boldsymbol{\phi_{V_1}(\lambda_m)} - \boldsymbol{\phi_{V_2}(\lambda_m)}| > |\boldsymbol{\phi_{V_1}(\lambda_n)} - \boldsymbol{\phi_{V_2}(\lambda_n)}|.$$

We define such pair of augmentations as optimal contrastive pair.

## 4 Analysis of The General Graph Augmentation Rule

In this section, we verify the GAME rule with both of experiments and theoretical analysis.

**Experimental analysis** We substitute existing augmentations proposed by MVGRL [9], GCA [35] and GraphCL [31] for augmentation $\boldsymbol{V}$ in the case. Specifically, MVGRL proposes PPR matrix, heat diffusion matrix and pair-wise distance matrix. GCA mainly randomly drops edges based on

Degree, Eigenvector and PageRank. GraphCL adopts random node dropping, edge perturbation and subgraph sampling. The nine augmentations almost cover the mainstream augmentations in GCL. To accurately depict the change of the amplitude after these augmentations for some $\lambda_i$, we turn to matrix perturbation theory [24]:

$$\Delta\lambda_i = \lambda_i' - \lambda_i = \boldsymbol{u}_i^\top \Delta\boldsymbol{A}\boldsymbol{u}_i - \lambda_i \boldsymbol{u}_i^\top \Delta\boldsymbol{D}\boldsymbol{u}_i + \mathcal{O}(||\Delta\boldsymbol{A}||), \tag{2}$$

where $\lambda_i'$ is the eigenvalue after change, $\Delta\boldsymbol{A} = \boldsymbol{A'} - \boldsymbol{A}$ represent the modification of edges after augmentation, and $\Delta\boldsymbol{D}$ is the respective change in degree matrix. With Eq. (2), we calculate the eigenvalues on Cora after each augmentation, and plot their graph spectrums in Fig. 5. Simultaneously, we use the GCL framework in Section 3 to test the performances of these augmentations, and results are shown in Table 1. As shown in figure and table, PPR matrix, Heat diffusion matrix and Distance matrix better accord with GAME rule, where they have small difference with $\boldsymbol{A}$ in $\mathcal{F}_{\mathcal{L}}$, while have a large difference in $\mathcal{F}_{\mathcal{H}}$. Therefore, they outperform other augmentations.

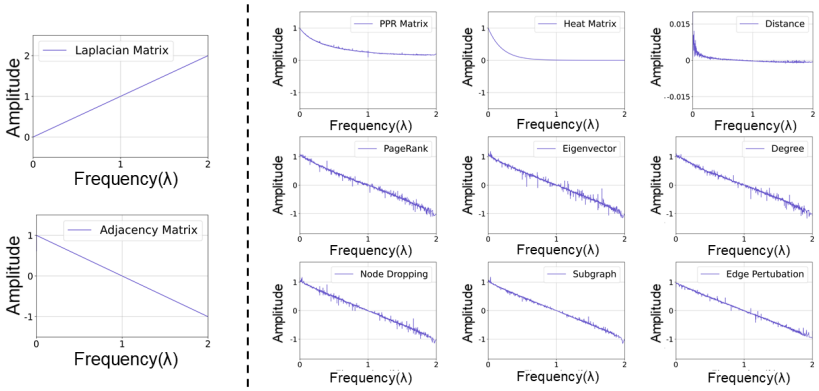



Figure 5: The graph spectrums of laplacian, adjacency matrix and nine existing augmentations.

Table 1: Performance of different existing augmentations to verify the GAME rule.

| Methods | GraphCL | | | GCA | | | MVGRL | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | Subgraph | Node dropping | Edge perturbation | Degree | PageRank | Eigenvector | PPR | Heat | Distance |
| Results | 34.9±3.5 | 29.8±2.3 | 37.7±4.4 | 40.2±4.1 | 38.5±5.0 | 42.1±4.9 | **58.0**±1.6 | **49.9**±4.2 | **46.1**±7.5 |

We also test the GAME rule in another circumstance, where we contrast among three cases: $\boldsymbol{A}$ and $\boldsymbol{A}^2$ (two-hop of $\boldsymbol{A}$), $\boldsymbol{A}$ and $\boldsymbol{A}$, and $\boldsymbol{A}^2$ and $\boldsymbol{A}^2$, and the results are given in Appendix C.

**Theoretical analysis** We have the following theorem to depict the learning process of the GCL.

**Theorem 1.** ***(Contrastive Invariance)*** *Given $\boldsymbol{A}$ and the augmented $\boldsymbol{V}$, the amplitudes of $i$-th frequency of $\boldsymbol{A}$ and $\boldsymbol{V}$ are $\lambda_i$ and $\gamma_i$, respectively. With the optimization of loss* (1)*, we have,*

$$\mathcal{L}_{total} \leq \frac{1+N}{2} \sum_i \theta_i \left[2 - (\lambda_i - \gamma_i)^2\right],$$

*where $\theta_i$ is an adaptive weight of the $i$th term with limited value.*

The proof is given in the Appendix A.1. Theorem 1 indicates an upper bound of GCL loss, implying that maximizing the contrastive loss equals to maximize the upper bound. Considering the total values of $[\theta_1, \ldots, \theta_N]$ are constrained, larger $\theta$ will be assigned to the smaller $(\lambda - \gamma)^2$. In other words, the invariance of two augmentations is enhanced from the point of view of graph spectrum. To our best knowledge, theorem 1, for the first time, theoretically proves that GCL can capture the invariance between two augmentations. Please recall that GAME rule suggests that the difference between two augmentations in $\mathcal{F}_{\mathcal{L}}$ is smaller. Thus, under the guidance of GAME rule, GCL tempts to capture the common low-frequency information of two augmentations. Thus, GAME rule points out a general augmentation strategy to manipulate encoder to capture low-frequency information, which achieves a better performance.

# 5 Spectral Graph Contrastive Learning

## 5.1 Theoretical derivation and formulation of SpCo

In this section, we tempt to learn an augmented graph from original graph based on the GAME rule. Based on eigenvalue perturbation formula (2) removing the high-order term $\mathcal{O}(||\Delta \boldsymbol{A}||)$, we have:

$$\begin{aligned}\Delta\lambda_i &\approx \boldsymbol{u}_i^\top \Delta \boldsymbol{A} \boldsymbol{u}_i - \lambda_i \boldsymbol{u}_i^\top \Delta \boldsymbol{D} \boldsymbol{u}_i \\ &= \sum_{m,n} [(\boldsymbol{u_i} \boldsymbol{u_i}^\top) * \Delta \boldsymbol{A}]_{m,n} - \lambda_i \boldsymbol{u_i}^\top \Delta \boldsymbol{D} \boldsymbol{u_i} \qquad (3a) \\ &= < \boldsymbol{S_i}, \Delta \boldsymbol{A} > - \lambda_i \boldsymbol{u_i}^\top \Delta \boldsymbol{D} \boldsymbol{u_i}. \qquad (3b)\end{aligned}$$

In eq. (3a), we use '$*$' to represent element-wise product, and use $< \boldsymbol{P}, \boldsymbol{Q} >$ to represent the sum of all elements of $\boldsymbol{P} * \boldsymbol{Q}$ in eq. (3b). Before further derivation, we give a lemma as following:

**Lemma 1.** $|\lambda_i \boldsymbol{u_i}^\top \Delta\, \boldsymbol{D} \boldsymbol{u_i}| \ \leq N|\lambda_i|$.

This lemma is proved in Appendix A.2. Then with Lemma 1, we gather the changes of all eigenvalues together, and have the following two theorems:

**Theorem 2.** *Given $\Delta_{total} = \sum_i \alpha_i |\Delta\lambda_i|$, we have*

$$\Delta_{total} \iff \sum_i \alpha_i \, |< \boldsymbol{S_i}, \Delta \boldsymbol{A} >| \geq \sum_i \alpha_i \, |< \boldsymbol{S_i}, \Delta_{\boldsymbol{A+}} >| - \sum_j a_j \, |< \boldsymbol{S_j}, \Delta_{\boldsymbol{A-}} >| = \Delta_+ - \Delta_-,$$

*where $\alpha_i \geq 0$ is the weight for the change of $\lambda_i$. Formally, we have $\Delta \boldsymbol{A} = \Delta_{\boldsymbol{A+}} - \Delta_{\boldsymbol{A-}}$, where $\Delta_{\boldsymbol{A+}}$ and $\Delta_{\boldsymbol{A-}}$ indicate which edge is added and deleted, respectively.*

**Theorem 3.** *Given N eigenspaces $[\boldsymbol{S_1}, \ldots, \boldsymbol{S_N}]$, if $| < \boldsymbol{S_i}, \Delta \boldsymbol{A} > |$ is maximized, then $\forall j, \ | < \boldsymbol{S_j}, \ \Delta \boldsymbol{A} > | \to 0$.*

The proof is given in Appendix A.3 and A.4. Theorem 3 implies that $\Delta \boldsymbol{A}$ can only be related to one of these eigenspaces. Please recall that we need $|\Delta\lambda_i|$ in $\mathcal{F}_{\mathcal{L}}$ is smaller than that in $\mathcal{F}_{\mathcal{H}}$. Therefore, we first make $\alpha_i$ is larger for $\Delta\lambda_i$ in $\mathcal{F}_{\mathcal{H}}$ than in $\mathcal{F}_{\mathcal{L}}$. Then, we maximize $\Delta_{total}$, and $\Delta \boldsymbol{A}$ will only be related to $\boldsymbol{S_i}$ in $\mathcal{F}_{\mathcal{H}}$. That means $|\Delta\lambda_i|$ in $\mathcal{F}_{\mathcal{H}}$ will be larger. Particularly, Theorem 2 indicates that we can maximize $\Delta_+$ and minimize $\Delta_-$ simultaneously to maximize $\Delta_{total}$. For $\Delta_+$, we can maximize it by following formula:

$$\begin{aligned}\Delta_+ &= \sum_i \alpha_i \, |< \boldsymbol{S_i}, \Delta_{\boldsymbol{A+}} >| = |\sum_i \alpha_i < \boldsymbol{S_i}, \Delta_{\boldsymbol{A+}} > | \\ &= | < \sum_i \alpha_i \boldsymbol{u_i} \boldsymbol{u_i}^\top, \ \Delta_{\boldsymbol{A+}} > | = \left| < \boldsymbol{U} g(\lambda) \boldsymbol{U}^\top, \ \Delta_{\boldsymbol{A+}} > \right| = |< \boldsymbol{\mathcal{C}}, \ \Delta_{\boldsymbol{A+}} >| \, . \qquad (4a)\end{aligned}$$

In eq. (4a), we define $\boldsymbol{\mathcal{C}} = \boldsymbol{U} g(\lambda) \boldsymbol{U}^\top$ and $\alpha_i = g(\lambda_i)$, where $g(\lambda)$ should be a monotone increasing function. As can be seen, if $\Delta_{\boldsymbol{A+}}$ is learnt to be closed to some predefined $\boldsymbol{\mathcal{C}}$, $\Delta_+$ will increase.

**Setting $\boldsymbol{\mathcal{C}}$** We notice that the graph spectrum of Laplacian $\boldsymbol{\mathcal{L}}$ does meet our need about $g(\lambda)$. Therefore, we simply set $\boldsymbol{\mathcal{C}} = \Theta \boldsymbol{\mathcal{L}}$, where $\Theta$ is a parameter. We can increase $\Theta$ in a curricular way:

$$\Theta = \frac{t}{T} \cdot \Theta', \qquad (5)$$

where $\Theta'$ is a constant, and $\Theta$ is used in $t$th epoch. $T$ is the total epoch.

With eq. (4a), we give the following optimization objective:

$$\mathcal{J} = < \boldsymbol{\mathcal{C}}, \ \Delta_{\boldsymbol{A+}} >^2 + \epsilon H(\Delta_{\boldsymbol{A+}}) + < \boldsymbol{f}, \Delta_{\boldsymbol{A+}} \mathbb{1}_n - \boldsymbol{a} > + < \boldsymbol{g}, \Delta_{\boldsymbol{A+}}^\top \mathbb{1}_n - \boldsymbol{b} >, \qquad (6)$$

where $H(\boldsymbol{P})$ is the entropy regularization, defined as $-\sum_{i,j} \boldsymbol{P}_{i,j}(\log(\boldsymbol{P}_{i,j}) - 1)$ [20], and $\epsilon$ is the weight of this term. This term exposes more edges in $\Delta_{\boldsymbol{A+}}$ to optimization. The last two terms are Lagrange constraint conditions, where $\boldsymbol{f} \in \mathbb{R}^N$ and $\boldsymbol{g} \in \mathbb{R}^N$ are Lagrange multipliers, and

$\boldsymbol{a} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{b} \in \mathbb{R}^{N\times 1}$ are distributions[1] that the row and column sums of $\Delta_{\boldsymbol{A}+}$ should meet. Next, the partial of $\mathcal{J}$ with respect to $\Delta_{\boldsymbol{A}+}$ is as following:

$$\partial\mathcal{J} \;/\; \partial(\Delta_{\boldsymbol{A}+})_{ij} = 2 < \boldsymbol{\mathcal{C}},\ \Delta_{\boldsymbol{A}+} > \boldsymbol{\mathcal{C}}_{ij} - \epsilon \log(\Delta_{\boldsymbol{A}+})_{ij} + \boldsymbol{f}_i + \boldsymbol{g}_j \tag{7a}$$

$$= m_{ij} + 2\boldsymbol{\mathcal{C}}_{ij}^2(\Delta_{\boldsymbol{A}+})_{ij} - \epsilon \log(\Delta_{\boldsymbol{A}+})_{ij} + \boldsymbol{f}_i + \boldsymbol{g}_j, \tag{7b}$$

where we separate $2\boldsymbol{\mathcal{C}}_{ij}^2(\Delta_{\boldsymbol{A}+})_{ij}$ from $2 < \boldsymbol{\mathcal{C}},\ \Delta_{\boldsymbol{A}+} > \boldsymbol{\mathcal{C}}_{ij}$, and set the rest part as $m_{ij}$. The next theorem points out when $\mathcal{J}$ can get the maximal value in the domain of definition $(\Delta_{\boldsymbol{A}+})_{ij} \in (0, 1)$:

**Theorem 4.** *Given $(\Delta_{\boldsymbol{A}+})_{ij} \in (0, 1)$, $\mathcal{J}$ exists the maximal value, iff*
*(1) $\boldsymbol{\mathcal{C}}_{ij}^2 < -\frac{\boldsymbol{f}_i+\boldsymbol{g}_j+m_{ij}}{2}$, and $\boldsymbol{f}_i + \boldsymbol{g}_j + m_{ij} < 0$, or*
*(2) $\frac{\epsilon}{2} < \boldsymbol{\mathcal{C}}_{ij}^2 < \frac{\epsilon}{2}\exp(-\frac{\boldsymbol{f}_i+\boldsymbol{g}_j+m_{ij}+\epsilon}{2})$, and $\boldsymbol{f}_i + \boldsymbol{g}_j + m_{ij} + \epsilon < 0$.*

We provide the proof in the Appendix A.5. Normally, we should let eq. (7b) equal to zero and get the analytical solution of $(\Delta_{\boldsymbol{A}+})_{ij}$. However, eq. (7b) is a transcendental equation because of the existence of linear term and logarithm. Thus, we require eq. (7a) to equal to 0. Specifically, in each training epoch, we firstly calculate "$< \boldsymbol{\mathcal{C}}, \Delta_{\boldsymbol{A}+} >$", where $\Delta_{\boldsymbol{A}+}$ is replaced by $\Delta'_{\boldsymbol{A}+}$ got from the last epoch. So, the first term in eq. (7a) is a constant, and the matrix form of solution of current epoch is:

$$\Delta_{\boldsymbol{A}+} = diag(\boldsymbol{u}) \exp\left(2 < \boldsymbol{\mathcal{C}}, \Delta'_{\boldsymbol{A}+} > \boldsymbol{\mathcal{C}} \;/\; \epsilon\right) diag(\boldsymbol{v}) = \boldsymbol{U}_+\boldsymbol{K}_+\boldsymbol{V}_+, \tag{8}$$

where $\boldsymbol{u}_i = \exp\left(\frac{\boldsymbol{f}_i}{\epsilon}\right)$ and $\boldsymbol{v}_j = \exp\left(\frac{\boldsymbol{g}_j}{\epsilon}\right)$. Please note that we require the $\Delta_{\boldsymbol{A}+}$ must meet the Lagrange constraint conditions, mentioned in eq. (6). So, we have the requirements: $\boldsymbol{u} * (\boldsymbol{K}_+\boldsymbol{v}) = \boldsymbol{a}$ and $\boldsymbol{v} * \left(\boldsymbol{K}_+^\top\boldsymbol{u}\right) = \boldsymbol{b}$. We solve this matrix scaling problem [17] by iteratively updating $\boldsymbol{u}$ and $\boldsymbol{v}$:

$$\boldsymbol{u}^{(l+1)} = \boldsymbol{a} \;/\; \left(\boldsymbol{K}_+\boldsymbol{v}^{(l)}\right) \quad \text{and} \quad \boldsymbol{v}^{(l+1)} = \boldsymbol{b} \;/\; \left(\boldsymbol{K}_+^\top\boldsymbol{u}^{(l+1)}\right), \tag{9}$$

where $l$ is the $l$th iteration. This process is Sinkhorn's Iteration [23], which is given in Algorithm B.1, referring to [4]. After this, the new calculated $\Delta_{\boldsymbol{A}+}$ will be used in eq. (8) as $\Delta'_{\boldsymbol{A}+}$ in the next training epoch. There exists a upper bound of the difference between $\Delta_{\boldsymbol{A}+}$ and $\Delta'_{\boldsymbol{A}+}$:

**Theorem 5.** *After Sinkhorn's Iteration, the bound between $\Delta_{\boldsymbol{A}+}$ and $\Delta'_{\boldsymbol{A}+}$ is:*

$$\left|\alpha(\Delta_{\boldsymbol{A}+})_{ij} - (\Delta_{\boldsymbol{A}+})'_{ij}\right| \le \frac{\alpha}{\epsilon^2(1-\gamma)}\{d(r^{(0)}, \boldsymbol{a}) + d(c^{(0)}, \boldsymbol{b})\} + \alpha(1 + \frac{|m_{ij}|}{\epsilon}),$$

*where $\alpha = \frac{\epsilon}{2\boldsymbol{\mathcal{C}}_{ij}^2}$. $\forall(x, x') \in (\mathbb{R}_+^n)^2$, $d(x, x')$ is the Hilbert's projective metric [3] on $\mathbb{R}_+^n$. $\gamma$ is $\kappa(\boldsymbol{K}_+)$, and $\kappa$ is contraction ratio [7]. $r^{(0)}$ and $c^{(0)}$ are the row and column sum vectors of $\boldsymbol{K}_+$.*

The proof is given in Appendix A.6. The calculation of $\Delta_{\boldsymbol{A}-}$ is similar as $\Delta_{\boldsymbol{A}+}$ shown as follows: $\Delta_{\boldsymbol{A}-} = diag(\boldsymbol{u}') \exp\left(-2 < \boldsymbol{\mathcal{C}}, \Delta'_{\boldsymbol{A}-} > \boldsymbol{\mathcal{C}} \;/\; \epsilon\right) diag(\boldsymbol{v}') = \boldsymbol{U}_-\boldsymbol{K}_-\boldsymbol{V}_-$. Now we can obtain $\Delta_{\boldsymbol{A}+}$ and $\Delta_{\boldsymbol{A}-}$, and $\Delta_{\boldsymbol{A}} = \Delta_{\boldsymbol{A}+} - \Delta_{\boldsymbol{A}-}$. Next, we show that the $\Delta_{\boldsymbol{A}}$ is a general component, which can be easily plugged into the current GCL to further improve them.

### 5.2 Instantiation of SpCo with existing GCL

As shown in Fig. 6, we first transform $\boldsymbol{A}$ into an augmented graph $\boldsymbol{A}_$ as follows:

$$\boldsymbol{A}_ = \boldsymbol{A} + \eta \cdot \mathbb{S} * \Delta_{\boldsymbol{A}}, \tag{10}$$

where $\eta$ is the combination coefficient. To make $\Delta_{\boldsymbol{A}}$ sparse, we only keep the values in a limited scope $\mathbb{S}$, for example the one-hop neighbors for each node. Then, we augment $\boldsymbol{A}$ and $\boldsymbol{A}_$ with the same strategies of target GCL model $\Phi$ to generate $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$. Next, $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ are used for GCL. The whole algorithm is given in Algorithm B.2.

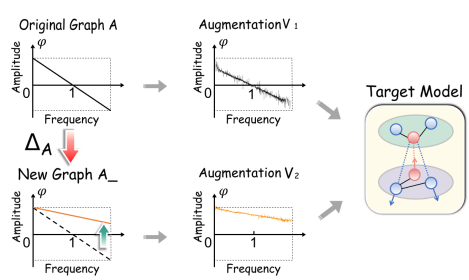



Figure 6: Combine SpCo with existing GCL.

[1] We define $\boldsymbol{a}$ and $\boldsymbol{b}$ are both node degree distribution in this paper.

# 6 Experiments

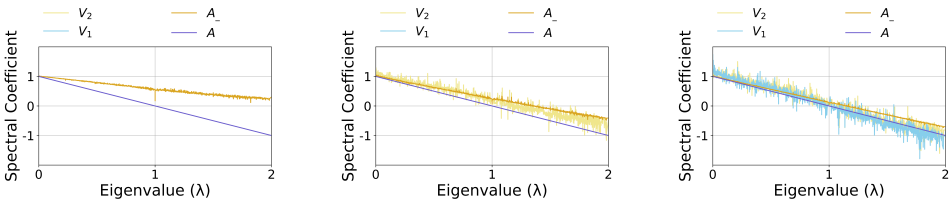
In this section, we mainly evaluate the performance of proposed SpCo on five datasets: Cora, Citeseer, Pubmed [11], BlogCatalog and Flickr [16]. Details of datasets are in Appendix D.2. To verify the applicability of our SpCo, we plug SpCo into three existing GCL methods DGI [27], GRACE [34] and CCA-SSG [32]. The detailed descriptions of DGI, GRACE and CCA-SSG are given in Appendix D.6. We compare with two categories of baselines: semi-supervised GNN models {GCN [11], GAT [26]} and six representative graph contrastive learning methods {DGI [27], MVGRL [9], GRACE [34], GCA [35], GraphCL [31], CCA-SSG [32]}. Please note that we mainly focus on if our SpCo can boost the performance of DGI, GRACE and CCA-SSG. Experimental implementation details are given in Appendix D.1.

Table 2: Quantitative results (%$\pm\sigma$) on node classification.

| Datasets | Metrics | GCN | GAT | DGI | **DGI+SpCo** | MVGRL | GRACE | **GRACE+SpCo** | GCA | GraphCL | CCA-SSG | **CCA+SpCo** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cora | Ma-F1 | 79.6±0.7 | 81.3±0.3 | 80.4±0.7 | **81.1±0.5** | 81.5±0.5 | 79.2±1.0 | **80.3±0.8** | 79.9±1.1 | 80.7±0.9 | 82.9±0.8 | **83.6±0.4** |
| | Mi-F1 | 80.7±0.6 | 82.3±0.2 | 82.0±0.5 | **82.8±0.7** | 82.8±0.4 | 80.0±1.0 | **81.2±0.9** | 81.1±1.0 | 82.3±0.9 | 83.6±0.9 | **84.3±0.4** |
| Citeseer | Ma-F1 | 68.1±0.5 | 67.5±0.2 | 67.7±0.9 | **68.3±0.5** | 66.8±0.7 | 65.1±1.2 | **65.1±0.8** | 62.8±1.3 | 67.8±1.0 | 67.9±1.0 | **68.5±1.0** |
| | Mi-F1 | 70.9±0.5 | 72.0±0.9 | 71.7±0.8 | **72.4±0.5** | 72.5±0.5 | 68.7±1.1 | **69.4±1.0** | 65.9±1.0 | 71.9±0.9 | 73.1±0.7 | **73.6±1.1** |
| BlogCatalog | Ma-F1 | 71.2±1.2 | 67.6±2.2 | 68.2±1.3 | **71.5±0.8** | 80.3±3.6 | 67.7±1.2 | **68.2±0.4** | 71.7±0.4 | 63.9±2.1 | 72.0±0.5 | **72.8±0.3** |
| | Mi-F1 | 72.1±1.3 | 68.3±2.2 | 68.8±1.4 | **72.3±0.9** | 80.9±3.6 | 68.5±1.3 | **69.4±1.3** | 72.7±0.5 | 64.6±2.1 | 73.0±0.5 | **73.7±0.3** |
| Flickr | Ma-F1 | 48.9±1.6 | 35.0±0.8 | 31.2±1.6 | **33.7±0.7** | 31.2±2.9 | 35.7±1.3 | **36.3±1.4** | 41.2±0.5 | 32.1±1.1 | 37.0±1.1 | **38.7±0.6** |
| | Mi-F1 | 50.2±1.2 | 37.1±0.3 | 33.0±1.6 | **35.2±0.7** | 33.4±3.0 | 37.3±1.0 | **38.1±1.3** | 42.2±0.6 | 34.5±0.9 | 39.3±0.9 | **40.4±0.4** |
| PubMed | Ma-F1 | 78.5±0.3 | 77.4±0.2 | 76.8±0.9 | **77.6±0.6** | 79.8±0.4 | 80.0±0.7 | **80.3±0.3** | 80.8±0.6 | 77.0±0.4 | 80.7±0.6 | **81.3±0.3** |
| | Mi-F1 | 78.9±0.3 | 77.8±0.2 | 76.7±0.9 | **77.4±0.5** | 79.7±0.3 | 79.9±0.7 | **80.7±0.2** | 81.4±0.6 | 76.8±0.5 | 81.0±0.6 | **81.5±0.4** |

## 6.1 Node classification

To more comprehensively evaluate our model, we use two common evaluation metrics, including Macro-F1 and Micro-F1. The results are reported in Table 2, where the training set contains 20 nodes per class. As can be seen, the proposed SpCo can generally improve the performances of the corresponding original models on all datasets, which verifies that our SpCo is widely applicable and effective. We also choose 5 and 10 labeled nodes per class as training set respectively, which are reported in Appendix D.4.1.




(a) DGI: Citeseer (b) GRACE: Citeseer (c) CCA-SSG: Citeseer

Figure 7: The visualisation of graph spectrum on Citeseer.

## 6.2 Visualisation of graph spectrum

In this section, we test if the learnt view $\boldsymbol{A}_$ and $\boldsymbol{A}$ meet the GAME rule. We plot the graph spectrum of $\boldsymbol{A}_$, $\boldsymbol{A}$, $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ in one figure for each method on Citeseer, which are shown in Fig. 7. Here, we discard the impact of self-loop operation. For DGI, it does not use topological augmentation. [2] Therefore, we only plot $\boldsymbol{A}_$ and $\boldsymbol{A}$ for it. For GRACE, the augmentation strength of $\boldsymbol{V_1}$ is set to 0.

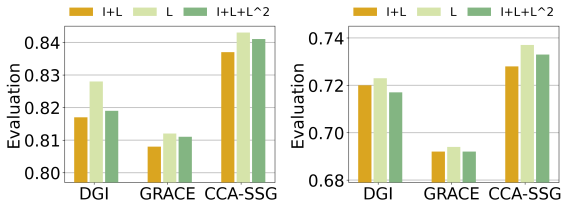
[2] Although in DGI, the authors summary a vector to depict the global view, this summary vector does not reflect any graph structure, thus we think DGI does not have special augmentation strategies on topology.

Thus, the plot of $\boldsymbol{V_1}$ is same with $\boldsymbol{A}$. From the figures, we can see that the difference between $\boldsymbol{A}_$ and $\boldsymbol{A}$ is smaller in $\mathcal{F}_{\mathcal{L}}$ than in $\mathcal{F}_{\mathcal{H}}$, which proves that they are optimal contrastive pair. Meanwhile, they can drive $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ also to obey the GAME rule, and thus boost the final results. More results on Cora are given in Appendix D.4.2.

### 6.3 Hyper-parameter sensitivity

In this subsection, we systematically investigate the sensitivity of two parameters: matrix $\boldsymbol{\mathcal{C}}$ and $\epsilon$. We conduct node classification on Cora and BlogCatalog datasets and report the Micro-F1 values. More experiments of hyper-parameters are given in Appendix D.4.3.

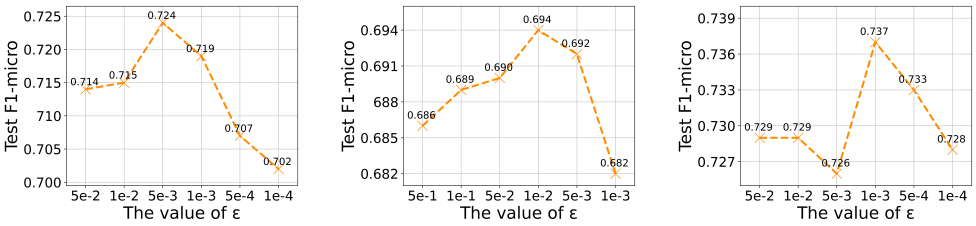
**Analysis of $\boldsymbol{\mathcal{C}}$.** The matrix $\boldsymbol{\mathcal{C}}$ directly affects the final structures of the $\Delta_{\boldsymbol{A}+}$ and $\Delta_{\boldsymbol{A}-}$. Therefore, we give three kinds of $\boldsymbol{\mathcal{C}}$: $\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{L}}$, $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{I}} + \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{L}}^{\boldsymbol{2}}$, and corresponding results are shown in Fig. 8. From the figures, we can see that $\boldsymbol{\mathcal{L}}$ is the best choice compared with two candidates. So, we use $\theta\boldsymbol{\mathcal{L}}$ as $\boldsymbol{\mathcal{C}}$. Other well-designed $\boldsymbol{\mathcal{C}}$ can also replace $\boldsymbol{\mathcal{L}}$ here.




(a) Cora (b) BlogCatalog

Figure 8: The comparison between three candidates for $\boldsymbol{\mathcal{C}}$.

**Analysis of $\epsilon$.** The $\epsilon$ in eq. (6) controls the strength of entropy regularization, and in eq. (8) also controls the smoothness of exponential. We vary the value of it and plot the results on BlogCatalog in Fig. 9. From the results, we know that $\epsilon$ is a sensitive parameter for SpCo. If $\epsilon$ is too small, the effect of entropy term will diminish. And if $\epsilon$ is too large, the entropy term will interfere the molding of new structure. More results on Cora are given in Appendix D.4.3.




(a) DGI: BlogCatalog (b) GRACE: BlogCatalog (c) CCA-SSG: BlogCatalog

Figure 9: Analysis of the hyper-parameter $\epsilon$ on BlogCatalog.

## 7 Conclusion

In this paper, we fundamentally explore the topological augmentation of GCL from spectral domain. We propose contrastive invariance theorem, and discover a general augmentation (GAME) rule, which deepen our understanding of the essence of GCL. Then, we propose a general augmentation plug-in based on GAME rule, SpCo, to boost existing GCL methods. Extensive experiments verify the effectiveness of SpCo.

**Limitations and broader impact.** On potential limitation is that this work mainly focuses on the homophily graph, rather than the heterophily graphs [1], where high-frequency information is more useful. Despite the great development of GCL, some theoretical foundations are still lacking. Our work points out the great potential of graph spectrum in GCL, and may open a new path to understand and design GCL. Other than that, we do not foresee any direct negative impacts on the society.

## References


[1] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. *arXiv preprint arXiv:2101.00797*, 2021.

[2] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *2nd International Conference on Learning Representations*, 2014.

[3] Peter J Bushell. Hilbert's metric and positive contraction mappings in a banach space. *Archive for Rational Mechanics and Analysis*, 52(4):330–338, 1973.

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 3837–3845, 2016.

[6] Negin Entezari, Saba A Al-Sayouri, Amirali Darvishzadeh, and Evangelos E Papalexakis. All you need is low (rank) defending against adversarial attacks on graphs. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 169–177, 2020.

[7] Joel Franklin and Jens Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its applications*, 114:717–735, 1989.

[8] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[9] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *ICML*, pages 4116–4126, 2020.

[10] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

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

[12] Namkyeong Lee, Junseok Lee, and Chanyoung Park. Augmentation-free self-supervised learning on graphs. *arXiv preprint arXiv:2112.02472*, 2021.

[13] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems*, 34, 2021.

[14] Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. Heterogeneous graph attention networks for semi-supervised short text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4821–4830, 2019.

[15] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

[16] Zaiqiao Meng, Shangsong Liang, Hongyan Bao, and Xiangliang Zhang. Co-embedding attributed networks. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 393–401, 2019.

[17] Arkadi Nemirovski and Uriel Rothblum. On complexity of matrix scaling. *Linear Algebra and its Applications*, 302:435–460, 1999.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[19] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[20] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, (2017-86), 2017.

[21] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[22] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020.

[23] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

[24] Gilbert W Stewart. Matrix perturbation theory. 1990.

[25] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[27] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[28] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, pages 4–24, 2021.

[29] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[30] Longqi Yang, Liangliang Zhang, and Wenjing Yang. Graph adversarial self-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021.

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

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

[33] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2778–2786, 2018.

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

[35] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *WWW*, pages 2069–2080, 2021.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 7.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the code, data, and instructions in the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 6 and Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 6.1 and Appendix D.4.1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix D.5.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] We were unable to find the license for the assets we used.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The datasets are public benchmarks.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The datasets are public benchmarks.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]