Revisiting Graph Contrastive Learning from the Perspective of Graph Spectrum

Nian Liu\textsuperscript{1}, Xiao Wang\textsuperscript{1,2*}, Deyu Bo\textsuperscript{1}, Chuan Shi\textsuperscript{1,2*}, Jian Pei\textsuperscript{3}

\textsuperscript{1}Beijing University of Posts and Telecommunications  
\textsuperscript{2}Peng Cheng Laboratory  
\textsuperscript{3}Simon Fraser University  
{nianliu, xiaowang, bodeyu, shichuan}@bupt.edu.cn, jpei@cs.sfu.ca

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\textsuperscript{1}), 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 \cite{11,23,14}. 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 \cite{10}. Therefore, Graph Contrastive Learning (GCL) is developed as a typical technique for self-supervised learning without the explicit usage of labels \cite{28,9,36}.

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 \cite{36}. Typical GCL methods have

\*Corresponding authors.

\textsuperscript{1}Code available at https://github.com/liun-online/SpCo

sought to elaborately design different graph augmentation strategies. For example, the heuristic based
methods including node or edge dropping [32], feature masking [33], and diffusion [9]; and the learn-
ing based methods including InfoMin [26, 30], disentanglement [13], and adversarial training [31].
Although various graph augmentation strategies are proposed, the fundamental augmentation mech-
nism is not well understood. What information should we preserve or discard in an augmented
target GCL and whether there exists a general augmentation rule? 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 [20] 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 can 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 in-
formation 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 spectral graph contrastive
learning (SpCo), a general GCL framework, which can 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 [24] and finally obtain the augmented structure used for the following target GCL model.

Our contributions are summarized as follows. Firstly, we answer the question “what information is
learned by GCL and whether there exists a general augmentation rule”. To the best of our knowledge,
this is 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

---

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.
strategies, but also explain why GCL works by proposing the contrastive invariance theorem. Our work provides deeper understanding on the nature of GCL. Secondly, we answer the question “how to utilize the augmentation rule for GCL”. We show that the augmentation rule provides a novel insight to estimate the current augmentation strategies. We propose a novel concept optimal contrastive pair and theoretically derive a general 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. We validate the effectiveness of SpCo on five datasets. We consistently gain improvements compared with those 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 \( A \in \{0, 1\}^{N \times N} \), where \( A_{ij} \in \{0, 1\} \) denotes the relation between nodes \( i \) and \( j \) in \( \mathcal{V} \). The node degree matrix \( D = diag(d_1, \ldots, d_n) \), where \( d_i = \sum_{j \in \mathcal{V}} A_{ij} \) is the degree of node \( i \) in \( \mathcal{V} \). Graph \( \mathcal{G} \) is often associated with a node feature matrix \( X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{N \times d} \), where \( x_i \) is a \( d \) dimensional feature vector of node \( i \) in \( \mathcal{V} \). Let \( \mathcal{L} = D - A \) be the unnormalized graph Laplacian of \( \mathcal{G} \). If we set symmetric normalized adjacency matrix as \( \mathcal{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \), then \( \tilde{\mathcal{L}} = I_n - \mathcal{A} = D^{-\frac{1}{2}} (D - A) D^{-\frac{1}{2}} \) is the symmetric normalized graph Laplacian.

Since \( \tilde{\mathcal{L}} \) is symmetric normalized, its eigen-decomposition is \( U \Lambda U^\top \), where \( \Lambda = diag(\lambda_1, \ldots, \lambda_N) \) and \( U = [u_1, \ldots, u_N] \in \mathbb{R}^{N \times N} \) are the eigenvalues and eigenvectors of \( \tilde{\mathcal{L}} \), 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 \( \mathcal{F}_\kappa = \{ \lambda_1, \ldots, \lambda_{\lfloor \kappa N \rfloor} \} \) the amplitudes of low-frequency components and by \( \mathcal{F}_\kappa = \{ \lambda_{\lfloor \kappa N \rfloor + 1}, \ldots, \lambda_N \} \) the amplitudes of high-frequency components. The graph spectrum is defined as these amplitudes of different frequency components, denoted as \( \phi(\lambda) \), indicating which parts of frequency are enhanced or weakened [20]. Additionally, we rewrite \( \tilde{\mathcal{L}} = \lambda_1 \cdot u_1 u_1^\top + \cdots + \lambda_N \cdot u_N u_N^\top \) as the eigenspace related to \( \lambda_i \), denoted as \( \mathcal{S}_i \).

Graph Contrastive Learning (GCL) [25][5][35] aims to learn discriminative embeddings without supervision, whose pipeline is shown in Fig. 1(a). We summarize the representative GCL in Appendix 2. Specifically, two augmentations are randomly extracted from \( A \) 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 [36][32][35] use classical InfoNCE loss [19] as the optimization objective:

\[
\mathcal{L}(h_i^{V_1}, h_i^{V_2}) = \log \frac{\exp(\theta(h_i^{V_1}, h_i^{V_2})/\tau)}{\exp(\theta(h_i^{V_1}, h_i^{V_2})/\tau) + \sum_{k \neq i} \exp(\theta(h_i^{V_1}, h_k^{V_2})/\tau)},
\]

where \( h_i^{V_1} \) and \( h_i^{V_2} \) 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

\[
\mathcal{L}_{\text{InfoNCE}} = \sum_{i} \frac{1}{2} \left( \mathcal{L}(h_i^{V_1}, h_i^{V_2}) + \mathcal{L}(h_i^{V_2}, h_i^{V_1}) \right).
\]

3 Impact of Graph Augmentation: An Experimental Investigation

In this section, we aim to explore what information should be considered in two contrasted augmentations from the perspective of graph spectrum. Specifically, we design a simple GCL framework shown in Fig. 2. Two input augmentations are adjacency matrix \( A \) and generated \( V \). Then, we utilize a shared GCN with one layer as the encoder to encode \( A \) and \( V \) and get their nodes embeddings as \( H_A \) and \( H_V \). We train the GCN by utilizing InfoNCE loss as in Eq. (1). More experimental settings can be found in appendix 2.1.

![Figure 2: The case study model.](image-url)
Generating augmentation $V$ 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. 3. Specifically, we divide the eigenvalues of $\mathcal{L}$ into $\mathcal{F}_L$ and $\mathcal{F}_H$ parts, and conduct augmentations in these two parts, respectively. Taking the augmentation in $\mathcal{F}_L$ for example, we keep the high-frequency part as $u_{N/2}^Tu_{N/2}^T + \cdots + u_Nu_N^T$. Then, we gradually add the eigenspaces in $\mathcal{F}_C$ back with rates $20\%, 40\%, 60\%, 80\%$, starting from the lowest frequency. Therefore, $V$ augmenting $20\%$ in $\mathcal{F}_L$ is $u_1u_1^T + \cdots + u_{N/2}^Tu_{N/2}^T + u_Nu_N^T$. Similarly, $V$ augmenting $20\%$ in $\mathcal{F}_H$ is $u_1u_1^T + \cdots + u_N/2u_{N/2}^T + u_{(N+1)/2}u_{(N+1)/2}^T + \cdots + u_{0.7N}u_{0.7N}^T$. Please note that we set graph spectrum of $V$, $\phi_V(\lambda) = 1$, $\forall \lambda \in [0, 2]$ above, in that we just want to test the effect of different $u_iu_i^T$ and avoid the influence from eigenvalues $\lambda$.

![Figure 3: The generation of $V$.](image)

Results and analyses We conduct the node classification on four datasets: Cora, Citeseer [11], BlogCatalog, and Flickr [16]. The accuracy (ACC) is shown in Fig. 4. In appendix B.2, we also report the results when both high and low frequency components are added back in the high-to-low frequency order. Results. For each dataset, in generated $V$, (1) when the lowest part of frequencies are kept, the best performance is achieved; (2) when more frequencies in $\mathcal{F}_H$ are involved, the performance generally rises. Analyses. From the graph spectra of $A$ and $V$ shown in Fig. 5 we can see that in generated $V$, (1) when the lowest part of frequencies are kept, the difference of amplitude, i.e., the graph spectrum, in $\mathcal{F}_L$ between $A$ and $V$ becomes smaller; (2) when more frequencies in $\mathcal{F}_H$ are involved, the margin of graph spectrum in $\mathcal{F}_H$ between $A$ and $V$ becomes larger. Combining results and observations, we propose the following general Graph Augmentation rule, called GAME rule:

<table>
<thead>
<tr>
<th>The General Graph Augmentation Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given two random augmentations $V_1$ and $V_2$, their graph spectra are $\phi_{V_1}(\lambda)$ and $\phi_{V_2}(\lambda)$. Then, $\forall \lambda_m \in [1, 2]$ and $\lambda_n \in [0, 1]$, $V_1$ and $V_2$ are an effective pair of graph augmentations if the following condition is satisfied: $</td>
</tr>
<tr>
<td>We define such pair of augmentations as optimal contrastive pair.</td>
</tr>
</tbody>
</table>

1 Although this rule is derived from contrasting $A$ and $V$, the selection of certain views does not curb the generality of GAME rule. Considering that most of augmentations are obtained from the raw adjacency matrix $A$, it is a natural setting that one view is fixed as $A$ and the other is an augmented one.
4 Analysis of The General Graph Augmentation Rule

In this section, we aim to verify the correctness of GAME rule that whether two contrasted augmentations satisfying GAME rule can perform better in downstream tasks from experimental and theoretical analysis.

Experimental analysis We substitute existing augmentations proposed by MVGRL [9], GCA [36] and GraphCL [32] for augmentation $V$ 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\(^\ref{25}\):

$$\Delta \lambda_i = \lambda'_i - \lambda_i = u_i^\top \Delta A u_i - \lambda_i u_i^\top \Delta D u_i + O(||\Delta A||), \quad (2)$$

where $\lambda'_i$ is the eigenvalue after change, $\Delta A = A' - A$ represent the modification of edges after augmentation, and $\Delta D$ is the respective change in degree matrix. With Eq. (2), we calculate the eigenvalues on Cora after each augmentation, and plot their graph spectra in Fig. 6. Simultaneously, we use the GCL framework in Section 3 to separately contrast adjacency matrix $A$ and these augmentations, and results are shown in Table 1. As shown in Fig. 6, PPR matrix, Heat diffusion matrix and Distance matrix better accord with GAME rule, where they have small difference with $A$ in $F_L$, and have a large difference in $F_H$. Therefore, they outperform other augmentations in Table 1.

![Figure 6: The graph spectra of laplacian, adjacency matrix and nine existing augmentations.](image)

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>GraphCL</th>
<th>GCA</th>
<th>MVGRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Subgraph</td>
<td>Node dropping</td>
<td>Edge perturbation</td>
</tr>
<tr>
<td>Results</td>
<td>32.8±3.3</td>
<td>37.7±1.4</td>
<td>40.2±1.1</td>
</tr>
</tbody>
</table>

We also test the GAME rule in another circumstance, where we contrast among three cases: $A$ and $A^2$ (two-hop of $A$), $A$ and $A^2$, and $A^2$ and $A^2$. 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 adjacency matrix $A$ and the generated augmentation $V$, the amplitudes of $i$-th frequency of $A$ and $V$ are $\lambda_i$ and $\gamma_i$, respectively. With the optimization of InfoNCE loss $\mathcal{L}_{InfoNCE}$, the following upper bound is established:**

$$\mathcal{L}_{InfoNCE} \leq \frac{1+\theta_i}{2} \sum_i \frac{1}{2 - (\lambda_i - \gamma_i)^2},$$

where $\theta_i$ is an adaptive weight of the $i$th term.

\(^1\)Here, we do not use eigenvalue decomposition to obtain $\lambda'$ of $A'$, because the obtained $\lambda'$ are unordered compared with previous $\lambda$ of $A$. That is to say, for certain $\lambda_i$ of $A$, we cannot figure out which eigenvalue of $A'$ matches to it after decomposition, so we cannot calculate the change $\Delta \lambda_i$ for $\lambda_i$ in this case.
The proof is given in the Appendix A.1 where we simplify GCN without the activation function. Theorem 1 indicates an upper bound of GCL loss, implying that maximizing the contrastive loss equals to maximize the upper bound. So, larger $\theta_i$ will be assigned to the smaller $(\lambda_i - \gamma_i)^2$, or $\lambda_i \approx \gamma_i$. Meanwhile, if $\lambda_i \approx \gamma_i$, these two contrasted augmentations are regarded to share the invariance at $\gamma_i$ frequency. Therefore, with contrastive learning, the encoder will emphasize the invariance between two contrasted augmentations from spectrum domain. 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 $F_c$ is smaller. Thus, under the guidance of GAME rule, GCL attempts 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

Based on the GAME Rule, we mainly aim to learn a general and GCL-friendly transformation $\Delta A$ from adjacency matrix $A$ to a new augmentation $A_-$ (or $\Delta A = A_- - A$), where $A$ and $A_-$ are required to be an optimal contrastive pair. Then, they are fed into existing GCL method $\Phi$, i.e. augmenting with the same strategies of $\Phi$ to generate $V_1$ and $V_2$ and training with the corresponding contrastive loss, shown in Fig. 7. The whole pipeline is our proposed spectral graph contrastive learning (SpCo), which can boost existing GCL methods.

Firstly, we separate $\Delta A = \Delta A_+ - \Delta A_-$, where $\Delta A_+$ and $\Delta A_-$ indicate which edge is added and deleted, respectively. Next, we indicate how to learn $\Delta A_+$, while the calculation of $\Delta A_-$ is similar. Based on our theoretical derivation in Appendix A.2, the following optimization objective of $\Delta A_+$ should be maximized:

$$J = \frac{<c, \Delta A_+>^2}{\text{Matching Term}} + \epsilon H(\Delta A_+) + \frac{<f, \Delta A_+^T 1_n - a>}{\text{Entropy Reg.}} + \frac{<g, \Delta A_+^T 1_n - b>}{\text{Lagrange Constraint Conditions}}$$

This objective consists of three components: (1) Matching Term. $\forall P, Q \in \mathbb{R}^{N \times N}$, $\forall P, Q = \sum_{ij} P_{ij} Q_{ij}$. To maximize $<c, \Delta A_+>^2$, $\Delta A_+$ should learn to "match" or be similar to $C$. In A.2, we define $C = U g(\lambda)^T$, where $U$ and $g(\lambda)$ are eigenvector matrix and some function about eigenvalues of $A$. According to GAME rule, we set $\phi_\Delta(\lambda) = |\phi_A(\lambda) - \phi_A(\lambda)|$, and we need $\phi_\Delta(\lambda_n) > \phi_\Delta(\lambda_l)$, $\forall \lambda_n \in [1, 2]$ and $\lambda_l \in [0, 1]$. Therefore, we stipulate that $\phi_\Delta(\lambda)$ should be a monotone increasing function. Since $C$ will guide $\Delta A_+$ to capture the change of difference between graph spectra ($\phi_\Delta(\lambda)$), we naturally set $g(\lambda)$ of $C$ also a monotone increasing function. Furthermore, we notice that the graph spectrum of Laplacian $C$ does meet our need about $g(\lambda)$ (shown in Fig. 6), so we simply set $C = \Theta \mathcal{L}$, where $\Theta$ is a parameter updating in training. (2) Entropy Regularization. Here, $H(P) = -\sum_{ij} P_{ij} \log(P_{ij}) - 1$ [21], and $\epsilon$ is the weight of this term. This term aims to increase the uncertainty of the learnt $\Delta A_+$, which encourages more edges (entries in $\Delta A_+$) to join in optimization. (3) Lagrange Constraint Conditions. $f \in \mathbb{R}^{N \times 1}$ and $g \in \mathbb{R}^{N \times 1}$ are Lagrange multipliers, and $\alpha \in \mathbb{R}^{N \times 1}$ and $b \in \mathbb{R}^{N \times 1}$ are distributions. This term restrains the row and column sums of $\Delta A_+$ within some limitation.

Next, we expound how to solve eq. (3). The partial of $J$ with respect to $\Delta A_+$ is as following:

$$\partial J / \partial (\Delta A_+)_ij = 2 <c, \Delta A_+ > C_{ij} - \epsilon \log(\Delta A_+)_ij + f_i + g_j$$

$$= m_{ij} + 2C^2_{ij}(\Delta A_+)_ij - \epsilon \log(\Delta A_+)_ij + f_i + g_j,$$

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

1 We define $\alpha$ and $b$ are both node degree distribution in this paper.
Theorem 2. Given \((\Delta_{A^+})_{ij} \in (0, 1), J\) exists the maximal value, iff 
(1) \(C^2_{ij} < -\frac{\xi_i + g_i + m_{ij}}{2}, \) and \(f_i + g_j + m_{ij} < 0,\) or 
(2) \(\frac{\xi_i}{2} < C^2_{ij} < \frac{\xi}{2} \exp(-\frac{f_i + g_i + m_{ij} + \epsilon}{2}),\) and \(f_i + g_j + m_{ij} + \epsilon < 0.\)

We provide the proof in the Appendix A.3. Normally, we should let eq. (4b) equal to zero and get the analytical solution of \((\Delta_{A^+})_{ij}:\) However, eq. (4b) is a transcendental equation because of the coexistence of linear term and logarithm. Thus, we require eq. (4a) to equal to 0. As the training goes on, \(\Delta_{A^+}\) does not change sharply. So, we firstly rewrite eq. (4a) as follows:

\[
\frac{\partial J}{\partial (\Delta_{A^+})_{ij}} \approx 2 < C, \quad \Delta_{A^+} > C / \epsilon \quad \text{diag}(v) = U_{\epsilon} K_{\epsilon} V_{\epsilon},
\]  

(6)

where \(U_{\epsilon} = \text{diag}(u_i) = \text{diag}(\exp(\frac{\xi_i}{2}))\) and \(V_{\epsilon} = \text{diag}(v_j) = \text{diag}(\exp(\frac{\xi_j}{2}))\). To further calculate \(U_{\epsilon}\) and \(V_{\epsilon}\), we restrain the row and column sums of \(\Delta_{A^+}\) according to Lagrange Constraint Conditions: \(u * (K_{\epsilon} v) = a\) and \(v * (K_{\epsilon}^T u) = b\). We solve this matrix scaling problem \([17]\) by Sinkhorn’s Iteration \([23]\), which is shown in Algorithm 1 \([4]\). There exists a upper bound of the difference between \(\Delta_{A^+}\) and \(\Delta_{A^+}^:\)

Theorem 3. After Sinkhorn’s Iteration, the bound between \(\Delta_{A^+}\) and \(\Delta_{A^+}^:\)

\[
|\alpha(\Delta_{A^+})_{ij} - (\Delta_{A^+})^{ij}_i| \leq \frac{1}{1-\gamma} \{d(r(0), a) + d(c(0), b)\} + \alpha(1 + \frac{m_{ij}}{\epsilon}),
\]

where \(\alpha = \frac{\epsilon}{1-\gamma}, \forall (x, x') \in (\mathbb{R}^N)^2, d(x, x')\) is the Hilbert’s projective metric \([3]\) on \(\mathbb{R}^N_+.\) \(\gamma\) is the contraction ratio \([7]\), \(r(0)\) and \(c(0)\) are the row and column sum vectors of \(K_{\epsilon}\).

The proof is given in Appendix A.4. The calculation of \(\Delta_{A^-}\) is similar as \(\Delta_{A^+}\) shown as follows:

\[
\Delta_{A^-} = \text{diag}(u') \exp(-2 < C, \Delta_{A^-} > C / \epsilon) \quad \text{diag}(v') = U_{\epsilon} K_{\epsilon} V_{\epsilon},
\]  

(7)

where \(\text{diag}(u'), \text{diag}(v')\) and \(\Delta_{A^-}^:\) have the similar meanings as in eq. (6).

Finally, we get the solution \(\Delta = \Delta_{A^+} + \Delta_{A^-}\), utilizing eq. (6) and eq. (7). With learnt transformation \(\Delta\), we can obtain the new augmentation \(\Delta_{\epsilon}\) as:

\[
A_{\epsilon} = A + \eta \cdot S * \Delta_A,
\]  

(8)

where \(*\) means element-wise product, and \(\eta\) is the combination coefficient. To make \(\Delta \) sparse, we use scope matrix \(S\) to limit our focus, e.g. one-hop neighbors for each node. The whole algorithm is given in Algorithm 2.

---

**Algorithm 1:** Sinkhorn’s Iteration

**Input:** Matrix \(K\), distribution \(a \in \mathbb{R}^{N \times 1}\) and \(b \in \mathbb{R}^{N \times 1}\)

**Params:** Iteration number \(\text{Iter}\)

**Output:** \(\Delta_{A^+}\) (or \(\Delta_{A^-}\))

1. Initialize \(u = [1/N, 1/N, \ldots, 1/N]_{1 \times N};\)
2. \(K = \text{diag}(1/a) K;\)
3. for \(i = 1 \text{ to } \text{Iter} \) do
4. \(u = 1/K (b/K^T u);\)
5. end
6. \(v = b/K^T u;\)
7. \(\Delta_{A^+}/\Delta_{A^-} = \text{diag}(u) K \text{diag}(v);\)
8. return \(\Delta_{A^+}/\Delta_{A^-};\)

**Algorithm 2:** The proposed SpCo

**Input:** \(\Phi\), augmentation \(A_u\), \(A, \mathcal{L}, X\)

**Params:** Total epochs \(\text{Iter}\), \(\Theta, \epsilon, a\) and \(b\)

1. for \(i = 1 \text{ to } T\) do
2. \(C = \Theta \mathcal{L};\)
3. Calculate \(K_{\epsilon} / K_{\epsilon} \text{ in eq. (6), (7)};\)
4. Get \(\Delta_{A^+} / \Delta_{A^-}\) through Algorithm 1 \([4]\);
5. \(A_{\epsilon} = A + \eta(\Delta_{A^+} - \Delta_{A^-}) S\) with eq. (8);
6. Update \(\Theta;\)
7. for \(j = 1 \text{ to } \text{Iter} \) do
8. \(V_1, V_2 = A_u \Phi(A), A_u \Phi(A_{\epsilon});\)
9. Train \(\Phi(V_1, V_2, X);\)
10. end
11. end
6 Experiments

In this section, we mainly evaluate the performance of proposed SpCo on five datasets: Cora, Citeseer, Pubmed, BlogCatalog and Flickr. Details of datasets are in Appendix D.2. We select two categories of baselines: semi-supervised GNN models {GCN, GAT} and six representative graph contrastive learning methods {DGI, MVGRL, GRACE, GCA, GraphCL, CCA-SSG}. These GCL methods can be divided into three categories based on their contrastive losses: BCE loss (DGI, MVGRL), InfoNCE loss (GRACE, GCA, GraphCL) and CCA loss (CCA-SSG). To verify the applicability of our SpCo, we select one baseline from each category (DGI, GRACE and CCA-SSG) to integrate with SpCo. The detailed descriptions of DGI, GRACE and CCA-SSG are given in Appendix D.6. Experimental implementation details are given in Appendix D.1.

Table 2: Quantitative results (%±σ) on node classification.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Metrics</th>
<th>GCN</th>
<th>GAT</th>
<th>DGI</th>
<th>DGI+SpCo</th>
<th>MVGRL</th>
<th>GRACE</th>
<th>GRACE+SpCo</th>
<th>GCA</th>
<th>GraphCL</th>
<th>CCA-SSG</th>
<th>CCA+SpCo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cora</td>
<td>Ma-F1</td>
<td>79.6±0.7</td>
<td>81.3±0.3</td>
<td>80.4±0.7</td>
<td>81.1±0.5</td>
<td>81.5±0.5</td>
<td>79.2±1.0</td>
<td>80.3±0.8</td>
<td>79.9±1.1</td>
<td>80.7±0.9</td>
<td>82.9±0.8</td>
<td>83.6±0.4</td>
</tr>
<tr>
<td></td>
<td>Mi-F1</td>
<td>80.7±0.6</td>
<td>82.3±0.2</td>
<td>82.0±0.5</td>
<td>82.8±0.7</td>
<td>82.8±0.4</td>
<td>80.0±1.0</td>
<td>81.2±0.9</td>
<td>81.1±0.8</td>
<td>82.3±0.9</td>
<td>83.6±0.9</td>
<td>84.3±0.4</td>
</tr>
<tr>
<td>Citeseer</td>
<td>Ma-F1</td>
<td>68.1±0.5</td>
<td>67.4±0.2</td>
<td>67.7±0.9</td>
<td>68.3±0.5</td>
<td>66.8±0.7</td>
<td>65.1±1.2</td>
<td>65.1±0.8</td>
<td>62.8±1.3</td>
<td>67.8±1.0</td>
<td>69.4±1.0</td>
<td>68.5±1.0</td>
</tr>
<tr>
<td></td>
<td>Mi-F1</td>
<td>70.9±0.5</td>
<td>72.0±0.9</td>
<td>71.7±0.8</td>
<td>72.4±0.5</td>
<td>72.5±0.5</td>
<td>68.7±1.1</td>
<td>69.4±1.0</td>
<td>65.9±1.0</td>
<td>71.9±0.9</td>
<td>73.1±0.7</td>
<td>73.6±1.1</td>
</tr>
<tr>
<td>BlogCatalog</td>
<td>Ma-F1</td>
<td>71.2±1.2</td>
<td>67.6±2.2</td>
<td>67.7±1.2</td>
<td>68.3±0.9</td>
<td>67.7±1.2</td>
<td>68.2±0.4</td>
<td>71.7±0.4</td>
<td>63.9±2.1</td>
<td>71.0±0.5</td>
<td>72.8±0.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mi-F1</td>
<td>72.1±1.3</td>
<td>68.3±2.2</td>
<td>68.8±1.4</td>
<td>72.4±0.9</td>
<td>68.9±3.0</td>
<td>68.5±1.3</td>
<td>69.4±1.3</td>
<td>72.7±0.5</td>
<td>64.6±2.1</td>
<td>73.0±0.5</td>
<td>73.7±0.3</td>
</tr>
<tr>
<td>Flickr</td>
<td>Ma-F1</td>
<td>48.9±1.6</td>
<td>35.0±0.8</td>
<td>31.2±1.6</td>
<td>33.7±0.7</td>
<td>31.2±0.9</td>
<td>35.7±0.3</td>
<td>36.3±1.4</td>
<td>41.2±0.5</td>
<td>32.1±1.1</td>
<td>37.0±1.1</td>
<td>38.7±0.6</td>
</tr>
<tr>
<td></td>
<td>Mi-F1</td>
<td>50.2±1.2</td>
<td>37.1±1.3</td>
<td>33.0±1.6</td>
<td>35.2±0.7</td>
<td>33.4±0.5</td>
<td>37.4±1.0</td>
<td>38.1±1.3</td>
<td>42.2±0.6</td>
<td>34.5±0.9</td>
<td>39.3±0.9</td>
<td>40.4±0.4</td>
</tr>
<tr>
<td>PubMed</td>
<td>Ma-F1</td>
<td>78.5±0.3</td>
<td>77.4±0.2</td>
<td>76.8±0.9</td>
<td>77.6±0.6</td>
<td>79.8±0.4</td>
<td>80.0±0.7</td>
<td>80.3±0.3</td>
<td>80.4±0.6</td>
<td>77.0±0.4</td>
<td>80.7±0.6</td>
<td>81.3±0.3</td>
</tr>
<tr>
<td></td>
<td>Mi-F1</td>
<td>78.9±0.3</td>
<td>77.8±0.2</td>
<td>76.7±0.9</td>
<td>77.4±0.5</td>
<td>79.7±0.3</td>
<td>79.9±0.7</td>
<td>80.7±0.2</td>
<td>81.4±0.6</td>
<td>76.8±0.5</td>
<td>81.0±0.6</td>
<td>81.5±0.4</td>
</tr>
</tbody>
</table>

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.

6.2 Visualisation of graph spectrum

In this section, we test if the learnt view $A_-$ and $A$ meet the GAME rule. We plot the graph spectrum of $A_-$, $A$, $V_1$ and $V_2$ in one figure for each method on Citeseer, which are shown in Fig. 8. Here, we discard the impact of self-loop operation. For DGI, it does not use topological augmentation. Therefore, we only plot $A_-$ and $A$ for it. For GRACE, the augmentation strength of $V_1$ is set to 0. Thus, the plot of $V_1$ is same with $A$. From the figures, we can see that the difference between $A_-$
and $A$ is smaller in $F_{\mathcal{C}}$ than in $F_{\mathcal{H}}$, which proves that they are optimal contrastive pair. Meanwhile, they can drive $V_1$ and $V_2$ 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 $\mathcal{C}$ 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 $\mathcal{C}$.** The matrix $\mathcal{C}$ directly affects the final structures of the $\Delta_{A^{+}}$ and $\Delta_{A^{-}}$. Therefore, we give three kinds of $\mathcal{C}$: $\mathcal{I} + \mathcal{L}$, $\mathcal{L}$ and $\mathcal{I} + \mathcal{L} + \mathcal{L}^2$, and corresponding results are shown in Fig. 9. From the figures, we can see that $\mathcal{L}$ is the best choice compared with two candidates. So, we use $\theta \mathcal{L}$ as $\mathcal{C}$. Other well-designed $\mathcal{C}$ can also replace $\mathcal{L}$ here.

**Analysis of $\epsilon$.** The $\epsilon$ in eq. (3) controls the strength of entropy regularization, and in eq. (6) also controls the smoothness of exponential. We vary the value of it and plot the results on BlogCatalog in Fig. 10. 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.

![Figure 9: The comparison between three candidates for $\mathcal{C}$.](image)

![Figure 10: Analysis of the hyper-parameter $\epsilon$ on BlogCatalog.](image)

### 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.

**Acknowledgments and Disclosure of Funding**

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62192784, 62172052, 62002029, U1936014).
References


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]
   (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]