Spectral Feature Augmentation for Graph Contrastive Learning and Beyond

Yifei Zhang[1], Hao Zhu[2,3], Zixing Song[1], Piotr Koniusz[3,2,*], Irwin King[1]

[1]The Chinese University of Hong Kong
[2]Australian National University
[3]Data61/CSIRO
{yfzhang, zxsong, king}@cse.cuhk.edu.hk; allenhaozhu@gmail.com; piotr.koniusz@data61.csiro.au

Abstract

Although augmentations (*e.g.*, perturbation of graph edges, image crops) boost the efficiency of Contrastive Learning (CL), feature level augmentation is another plausible, complementary yet not well researched strategy. Thus, we present a novel spectral feature argumentation for contrastive learning on graphs (and images). To this end, for each data view, we estimate a low-rank approximation per feature map and subtract that approximation from the map to obtain its complement. This is achieved by the proposed herein incomplete power iteration, a non-standard power iteration regime which enjoys two valuable byproducts (under mere one or two iterations): (i) it partially balances spectrum of the feature map, and (ii) it injects the noise into rebalanced singular values of the feature map (spectral augmentation). For two views, we align these rebalanced feature maps as such an improved alignment step can focus more on less dominant singular values of matrices of both views, whereas the spectral augmentation does not affect the spectral angle alignment (singular vectors are not perturbed). We derive the analytical form for: (i) the incomplete power iteration to capture its spectrum-balancing effect, and (ii) the variance of singular values augmented implicitly by the noise. We also show that the spectral augmentation improves the generalization bound. Experiments on graph/image datasets show that our spectral feature augmentation outperforms baselines, and is complementary with other augmentation strategies and compatible with various contrastive losses.

Introduction

Semi-supervised and supervised Graph Neural Networks (GNNs) (Velickovic et al. 2018; Hamilton, Ying, and Leskovec 2017; Song, Zhang, and King 2022; Zhang et al. 2022b) require full access to class labels. However, unsupervised GNNs (Klicpera, Bojchevski, and Günnemann 2019; Wu et al. 2019; Zhu and Koniusz 2021; Chen et al. 2023) and recent Self-Supervised Learning (SSL) models do not require labels (Song et al. 2021; Pan et al. 2018) to train embeddings. Among SSL methods, Contrastive Learning (CL) achieves comparable performance with its supervised counterparts on many tasks (Chen et al. 2020; Gao, Yao, and Chen 2021; Zhang and Zhu 2019, 2020). CL has also been applied recently to the graph domain. A typical Graph Contrastive Learning (GCL) method forms multiple graph views via stochastic augmentation of the input to learn representations by contrasting so-called positive samples with negative samples (Zhu et al. 2020; Peng et al. 2020; Zhu, Sun, and Koniusz 2021; Zhu and Koniusz 2022; Zhang et al. 2022c). As an indispensable part of GCL, the significance of graph augmentation has been well studied (Zhu et al. 2021b; Yin et al. 2022).

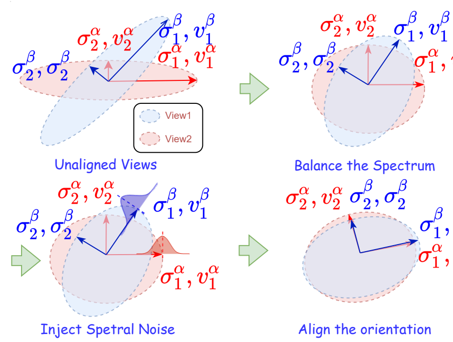


Figure 1: Our GCL with spectral feature augmentation by the incomplete power iteration implicitly performs three steps. Let blue and red ellipses represent spectra of feature maps $\mathbf{H}^\alpha$ and $\mathbf{H}^\beta$ of two views.

Popular random data augmentations are just one strategy to construct views, and their noise may affect adversely downstream tasks (Suresh et al. 2021; Tian et al. 2020). Thus, some works (Yin et al. 2022; Tian et al. 2020; Suresh et al. 2021) learn graph augmentations but they require supervision.

The above issue motivates us to propose a simple/efficient data augmentation model which is complementary with existing augmentation strategies. We target Feature Augmentation (FA) as scarcely any FA works exist in the context of CL and GCL. In the image domain, a simple FA (Upchurch et al. 2017; Bengio et al. 2013) showed that perturbing feature representations of an image results in a representation of another image where both images share some semantics (Wang et al. 2019). However, perturbing features randomly ignores covariance of feature representations, and ignores semantics correlations. Hence, we opt for injecting random noise into

*Corresponding author. PK was primarily concerned with the theoretical analysis (*e.g.*, Prop. 2 & 3).

the singular values of feature maps as such a spectral feature augmentation does not alter the orthogonal bases of feature maps by much, thus helping preserve semantics correlations.

Moreover, as typical GCL aligns two data views (Wang and Isola 2020), unbalanced singular values of two data views may affect the quality of alignment. As several leading singular values (acting as weights on the loss) dominate the alignment process, GCL favors aligning the leading singular vectors of two data views while sacrificing remaining orthogonal directions with small singular values. In other words, the unbalanced spectrum leads to a suboptimal orthonormal bases alignment, which results in a suboptimal GCL model.

To address rebalancing of unbalanced spectrum and augmenting leading singular values, we present a novel and efficient *Spectral Feature Augmentation* (SFA). To this end, we propose the so-called incomplete power iteration which, under just one or two iterations, partially balances singular values of feature maps and implicitly injects the noise into these singular values. We evaluate our method on various datasets for node level tasks (*i.e*., node classification and node clustering). We also show that our method is compatible with other augmentation strategies and contrastive losses.

We summarize our contributions as follows:

i. We propose a simple/efficient spectral feature augmentation for GCL which is independent of different contrastive losses, *i.e*., we employ InfoNCE and Barlow Twin.

ii. We introduce the so-called incomplete power iteration which, under just one or two iterations, partially balances spectra of two data views and injects the augmentation noise into their singular values. The rebalanced spectra help align orthonormal bases of both data views.

iii. As the incomplete power iteration is stochastic in its nature, we derive its analytical form which provably demonstrates its spectrum rebalancing effect in expectation, and captures the variance of the spectral augmentation.

iv. For completeness, we devise other spectral augmentation models, based on the so-called MaxExp and Power Norm. operators and Grassman feature maps, whose rebalancing and noise injection profiles differ with our method.

Related Work

Data Augmentation. Augmentations are usually performed in the input space. In computer vision, image transformations, *i.e*., rotation, flipping, color jitters, translation, noise injection (Shorten and Khoshgoftaar 2019), cutout and random erasure (DeVries and Taylor 2017) are popular. In neural language processing, token-level random augmentations, *e.g*., synonym replacement, word swapping, word insertion, and deletion (Wei and Zou 2019) are used. In transportation, conditional augmentation of road junctions is used (Prabowo et al. 2019). In the graph domain, attribute masking, edge permutation, and node dropout are popular (You et al. 2020a). Sun *et al*. (Sun, Koniusz, and Wang 2019) use adversarial graph perturbations. Zhu *et al*. (Zhu et al. 2021b) use adaptive graph augmentations based on the node/PageRank centrality (Page et al. 1999) to mask edges with varying probability.

Feature Augmentation. Samples can be augmented in the feature space instead of the input space (Feng et al. 2021). Wang *et al*. (Wang et al. 2019) augment the hidden space features, resulting in auxiliary samples with the same class identity but different semantics. A so-called channel augmentation perturbs the channels of feature maps (Wang et al. 2019) while GCL approach, COSTA (Zhang et al. 2022c), augments features via random projections. Some few-shot learning approaches augment features (Zhang et al. 2022a) while others estimate the "analogy" transformations between samples of known classes to apply them on samples of novel classes (Hariharan and Girshick 2017; Schwartz et al. 2018) or mix foregrounds and backgrounds (Zhang, Zhang, and Koniusz 2019). However, "analogy" augmentations are not applicable to contrastive learning due to the lack of labels.

Graph Contrastive Learning. CL is popular in computer vision, NLP (He et al. 2020; Chen et al. 2020; Gao, Yao, and Chen 2021), and graph learning. In the vision domain, views are formed by augmentations at the pixel level, whereas in the graph domain, data augmentation may act on node attributes or the graph edges. GCL often explores node-node, node-graph, and graph-graph relations for contrastive loss which is similar to contrastive losses in computer vision. Inspired by SimCLR (Chen et al. 2020), GRACE (Zhu et al. 2020) correlates graph views by pushing closer representations of the same node in different views and separating representations of different nodes, and Barlow Twin (Zbontar et al. 2021) avoids the so-called dimensional collapse (Jing et al. 2022).

In contrast, we study spectral feature augmentations to perturb/rebalance singular values of both views. We outperform feature augmentations such as COSTA (Zhang et al. 2022c).

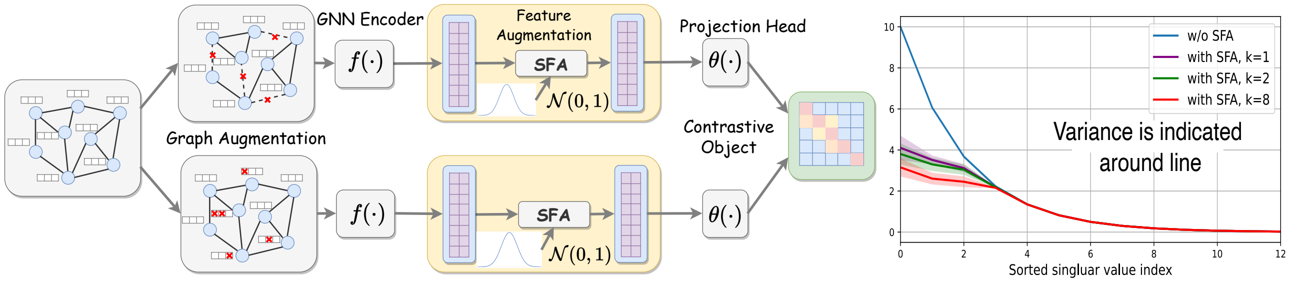
Proposed Method

Inspired by recent advances in augmentation-based GCL, our approach learns node representations by rebalancing spectrum of two data views and performing the spectral feature augmentation via the incomplete power iteration. SFA is complementary to the existing data augmentation approaches. Figure 2a illustrates our framework. The Notation section (supplementary material) explains our notations.

Graph Augmentation ($\mathcal{A}_G$). Augmented graph $(\tilde{\mathbf{A}}, \tilde{\mathbf{X}})$ is generated by $\mathcal{A}_G$ by directly adding random perturbations to the original graph $(\mathbf{A}, \mathbf{X})$. Different augmented graphs are constructed given one input $(\mathbf{A}, \mathbf{X})$, yielding correlated views, *i.e*., $(\widetilde{\mathbf{A}}^{\alpha}, \widetilde{\mathbf{X}}^{\alpha})$ and $(\widetilde{\mathbf{A}}^{\beta}, \widetilde{\mathbf{X}}^{\beta})$. In the common GCL setting (Zhu et al. 2020), the graph structure is augmented by permuting edges, whereas attributes by masking.

Graph Neural Network Encoders. Our framework admits various choices of the graph encoder. We opt for simplicity and adopt the commonly used graph convolution network (GCN) (Kipf and Welling 2017) as our base graph encoder. As shown in Fig. 2a, we use a shared graph encoder for each view, *i.e*., $f : \mathbb{R}^{n\times d_x} \times \mathbb{R}^{n\times n} \longmapsto \mathbb{R}^{n\times d_h}$. We consider two graphs generated from $\mathcal{A}_G$ as two congruent structural views and define the GCN encoder with 2 layers as:

$$\begin{aligned} &f(\mathbf{X}, \mathbf{A}) = \mathrm{GCN}_2\left(\mathrm{GCN}_1(\mathbf{X}, \mathbf{A}), \mathbf{A}\right), \\ &\quad\text{where}\quad \mathrm{GCN}_l(\mathbf{X}, \mathbf{A}) = \sigma\big(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\Theta\big). \end{aligned} \tag{1}$$



(a) Our pipeline. (b) Simulation: spectrum obtained by Alg. 1.

Figure 2: Our GCL model. Two graph views are generated by data augmentation and passed into graph neural network encoders with shared parameters to learn node representations. The proposed spectral feature augmentation rebalances (partially equalizes) the spectrum of each feature map, and implicitly injects the noise into rebalanced singular values. Such representations are fed into the projection head and the contrastive loss. Figure 1 explains the role of our spectral feature augmentation.

Moreover, $\tilde{\mathbf{A}} = \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2} \in \mathbb{R}^{n\times n}$ is the degree-normalized adjacency matrix, $\hat{\mathbf{D}} \in \mathbb{R}^{n\times n}$ is the degree matrix of $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I_N}$ where $\mathbf{I_N}$ is the identity matrix, $\mathbf{X} \in \mathbb{R}^{n\times d_x}$ contains the initial node features, $\mathbf{\Theta} \in \mathbb{R}^{d_x\times d_h}$ contains network parameters, and $\sigma(\cdot)$ is a parametric ReLU (PReLU). The encoder outputs feature maps $\mathbf{H}^\alpha$ and $\mathbf{H}^\beta$ for two views.

Spectral Feature Augmentation (SFA). $\mathbf{H}^\alpha$ and $\mathbf{H}^\beta$ are fed to the feature augmenting function $\mathcal{A}_{F^*}$ where random noises are added to the spectrum via the incomplete power iteration. We explain the proposed SFA in the Spectral Feature Augmentation for GCL section and detail its properties in Propositions 1, 2 and 3. SFA results in the spectrally-augmented feature maps, *i.e.*, $\widetilde{\mathbf{H}}^\alpha$ and $\widetilde{\mathbf{H}}^\beta$. SFA is followed by a shared projection head $\theta : \mathbb{R}^{n\times d_h} \longmapsto \mathbb{R}^{n\times d_z}$ which is an MLP with two hidden layers and PReLU non-linearity. It maps $\widetilde{\mathbf{H}}^\alpha$ and $\widetilde{\mathbf{H}}^\beta$ into two node representations $\mathbf{Z}^\alpha, \mathbf{Z}^\beta \in \mathbb{R}^{n\times d_z}$ (two congruent views of one graph) on which the contrastive loss is applied. As described in (Chen et al. 2020), it is beneficial to define the contrastive loss on $\mathbf{Z}$ rather than $\mathbf{H}$.

Contrastive Training. To train the encoders end-to-end and learn rich node representations that are agnostic to downstream tasks, we utilize the InfoNCE loss (Chen et al. 2020):

$$\mathcal{L}_{contrastive}(\tau) = \underbrace{\mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}^+}\left[-\mathbf{z}^\top\mathbf{z}^+/\tau\right]}_{alignment} + \underbrace{\mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}^+}\Big[\log\Big(e^{\mathbf{z}^\top\mathbf{z}^+/\tau} + \sum_{\mathbf{z}^-\in\mathbf{Z}^{\alpha\beta}\setminus\{\mathbf{z},\,\mathbf{z}^+\}} e^{\mathbf{z}^\top\mathbf{z}^-/\tau}\Big)\Big]}_{uniformity}, \tag{2}$$

where $\mathbf{z}$ is the representation of the anchor node in one view (*i.e.*, $\mathbf{z} \in \mathbf{Z}^\alpha$) and $\mathbf{z}^+$ denotes the representation of the anchor node in another view (*i.e.*, $\mathbf{z}^+ \in \mathbf{Z}^\beta$), whereas $\{\mathbf{z}^-\}$ are from the set of node representations other than $\mathbf{z}$ and $\mathbf{z}^+$ (*i.e.*, $\mathbf{Z}^{\alpha\beta} \equiv \mathbf{Z}^\alpha \cup \mathbf{Z}^\beta$ and $\mathbf{z}^- \in \mathbf{Z}^{\alpha\beta} \setminus \{\mathbf{z},\ \mathbf{z}^+\}$). The first part of Eq. (2) maximizes the alignment of two views (representations of the same node become similar). The second part of Eq. (2) minimizes the pairwise similarity via LogSumExp. Pushing node representations away from each other makes them uniformly distributed (Wang and Isola 2020).

Algorithm 1: Spectral Feature Augmentation ($\mathcal{A}_{F^*}$)

Input: feature map $\mathbf{H}$; the number of iterations k;
$\mathbf{r}^{(0)} \sim \mathcal{N}(0, \mathbf{I})$
for $i = 1$ **to** k **do**
  $\mathbf{r}^{(i)} = \mathbf{H}^\top\mathbf{H}\mathbf{r}^{(i-1)}$
end for
$\widetilde{\mathbf{H}} = \mathbf{H} - \frac{\mathbf{H}\mathbf{r}^{(k)}\mathbf{r}^{(k)\top}}{\|\mathbf{r}^{(k)}\|_2^2}$
Return $\widetilde{\mathbf{H}}$

Spectral Feature Augmentation for GCL

Our spectral feature augmentation is inspired by the rank-1 update (Yu, Cai, and Li 2020). Let $\mathbf{H} = f(\mathbf{X}, \mathbf{A})$ be the graph feature map with the singular decomposition $\mathbf{H} = \mathbf{U\Sigma V}^\top$ where $\mathbf{H} \in \mathbb{R}^{n\times d_h}$, $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_{d_h})$ is the diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{d_h}$. Starting from a random point $\mathbf{r}^{(0)} \sim \mathcal{N}(0, \mathbf{I})$ and function $\mathbf{r}^{(k)} = \mathbf{H}^\top\mathbf{H}\mathbf{r}^{(k-1)}$, we generate a set of augmented feature maps[1] $\widetilde{\mathbf{H}}$ by:

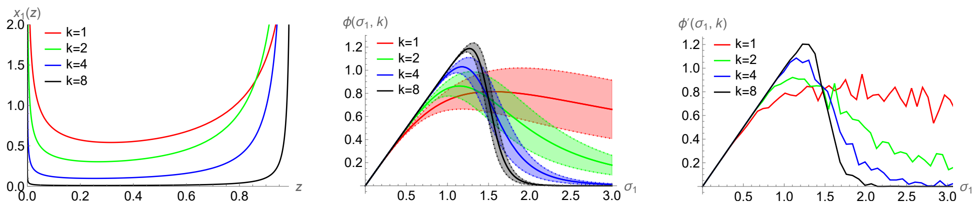
$$\widetilde{\mathbf{H}}(\mathbf{H}; \mathbf{r}^{(0)}) = \mathbf{H} - \mathbf{H}_{\text{LowRank}} = \mathbf{H} - \frac{\mathbf{H}\mathbf{r}^{(k)}\mathbf{r}^{(k)\top}}{\|\mathbf{r}^{(k)}\|_2^2}. \tag{3}$$

We often write $\widetilde{\mathbf{H}}$ rather than $\widetilde{\mathbf{H}}(\mathbf{H}; \mathbf{r}^{(0)})$, and we often think of $\widetilde{\mathbf{H}}$ as a matrix. We summarize the proposed SFA in Alg. 1.

Proposition 1. *Let* $\widetilde{\mathbf{H}}$ *be the augmented feature matrix obtained via Alg. 1 for the k-th iteration starting from a random vector* $\mathbf{r}^{(0)}$ *drawn from* $\mathcal{N}(0, \mathbf{I})$*. Then* $\mathbb{E}_{\mathbf{r}^{(0)}\sim\mathcal{N}(0,\mathbf{I})}(\widetilde{\mathbf{H}}(\mathbf{H}; \mathbf{r}^{(0)})) = \mathbf{U}\widetilde{\mathbf{\Sigma}}\mathbf{V}^\top$ *has rebalanced spectrum*[2] $\widetilde{\mathbf{\Sigma}} = diag\big[(1-\lambda_1(k))\sigma_1, (1-\lambda_2(k))\sigma_2, \cdots, (1-\lambda_{d_h}(k))\sigma_{d_h}\big]$ *where* $\lambda_i(k) = \mathbb{E}_{\mathbf{y}\sim\mathcal{N}(0;\mathbf{I})}\Big(\frac{(y_i\sigma_i^{2k})^2}{\sum_{l=1}^{d_h}(y_l\sigma_l^{2k})^2}\Big)$ *and* $\mathbf{y} = \mathbf{V}^\top\mathbf{r}^{(0)}$*, because* $0 \leq 1-\lambda_1(k) \leq 1-\lambda_2(k) \leq \cdots \leq$

[1] We apply Eq. (3) on both views $\mathbf{H}^\alpha$ and $\mathbf{H}^\beta$ separately to obtain spectrally rebalanced/augmented $\widetilde{\mathbf{H}}^\alpha$ and $\widetilde{\mathbf{H}}^\beta$.

[2] "Rebalanced" means the output spectrum is flatter than the input.



(a) (b) (c)

Figure 3: Toy illustration of Prop. 2 and 3. Let $\sigma_2, \cdots, \sigma_5$ be $1.5, 0.9, 0.2, 0.01$. We investigate the impact of iterations $k \in \{1, 2, 4, 8\}$. Fig. 3a shows distribution $x(z)$ given $\sigma_1 = 2$. Fig. 3b shows the expected value $\phi(\sigma_1, k) = \sigma_1(1 - \lambda_1)$ where $\lambda_1 = \mathbb{E}_{z\sim x_1}(z)$ for $0 \leq \sigma_1 \leq 3$. The deviation is indicated by $\phi_{\pm\omega_1}(\sigma_1, k) = \sigma_1(1 - \lambda_1 \pm \omega_1)$. Finally, Fig. 3c is obtained via Alg. 1 (the incomplete power iteration). To this end, we generated randomly a feature matrix $\mathbf{H}$ and substituted its singular values by $\sigma_1, \cdots, \sigma_5$. Notice that for $k = 1$, $1 \leq \sigma_1 \leq 3$, push-forward $\phi(\sigma_1, 1)$ and $\phi'(\sigma_1, 1)$ in Fig. 3b and 3c are around 0.8 (the balancing of spectrum) and the high deviation indicates the singular value undergoes the spectral augmentation. For $k \geq 2$, both balancing and spectral augmentation effects decline. Note theoretical ϕ in Prop. 2 and real ϕ' from Alg. 1 match.

$1 - \lambda_{d_h}(k) \leq 1$ *for* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{d_h}$ *(sorted singular values from the SVD), and so* $(1 - \lambda_i)$ *gets smaller or larger as* σ_i *gets larger or smaller, respectively.*

Push-forward Function (*Partial balancing of spectrum*). Prop. 1 shows that in expectation, our incomplete power iteration rebalances spectrum according to the push-forward function $\phi(\sigma_i; k) = \sigma_i(1-\lambda_i(k))$ where $\lambda_i(k)$ is an expected value of $\lambda'_i(\mathbf{y}, k) = \frac{(y_i\sigma_i^{2k})^2}{\sum_{l=1}^{d_h}(y_l\sigma_l^{2k})^2}$ w.r.t. random variable $\mathbf{y}\sim\mathcal{N}(0, \mathbf{I})$ (see Eq. 15 in suppl. material).

Alg. 1 returns an instance governed by the currently drawn y. The push-forward function in a feed-forward step of network realizes $\phi'(\sigma_i; \mathbf{y}, k) = \sigma_i(1 - \lambda'_i(\mathbf{y}, k))$. Thus, below we study the analytical expression for $\phi(\sigma_i; k)$ and its variance to understand how drawing $\mathbf{y}\sim\mathcal{N}(0, \mathbf{I})$ translates into the variance posed by the implicit spectral augmentation of the singular values.

Proposition 2. Analytical Expectation. *Let* $\beta_i = (\sigma_i^{2k})^2$*, then the expected value* $\mathbb{E}_{\mathbf{y}\sim\mathcal{N}(0;\mathbf{I})}\frac{\beta_i y_i^2}{\beta_i y_i^2+\sum_{l\neq i}\beta_l y_l^2} = \lambda_i(k)$ *can be expressed as* $\mathbb{E}(x_i)$ *over random variable* $x_i = \frac{u}{u+v_i}$ *for* $u \sim \mathcal{G}(\frac{1}{2}, 2)$ *and* $v_i \sim \mathcal{G}(\alpha_i, 2\gamma_i)$ *($\mathcal{G}$ is Gamma distr.) with* $\alpha_i = \frac{1}{2}\frac{(\sum_{l\neq i}\beta_l)^2}{\sum_{l\neq i}\beta_l^2}$ *and* $\gamma_i = \frac{1}{\beta_i}\frac{\sum_{l\neq i}\beta_l^2}{\sum_{l\neq i}\beta_l}$*. As PDF* $x_i^\bullet(z) = \frac{\gamma_i}{(1-(1-z)\gamma_i)^2}\cdot\mathcal{B}\big(\frac{\gamma_i z}{1-(1-z)\gamma_i}; \frac{1}{2}, \alpha_i\big)$ *where* $\mathcal{B}$ *is the Beta distribution and* $x_i^\bullet(z)$ *enjoys the support* $z \in [0; 1]$*, then* $\lambda_i = \mathbb{E}(z) = \int_0^1 z \cdot x_i^\bullet(z)\,\mathrm{d}z = \gamma_i^{\frac{1}{2}}\frac{\Gamma(\frac{3}{2})\Gamma(\frac{1}{2}+\alpha_i)}{\Gamma(\frac{1}{2})\Gamma(\frac{3}{2}+\alpha_i)}\cdot {}_2F_1\big(\frac{3}{2}, \frac{1}{2}+\alpha_i, \frac{3}{2}+\alpha_i, 1-\gamma_i\big)$ *where* ${}_2F_1(\cdot)$ *is the so-called Hypergeometric function.*

Proof. See Proof of Proposition 2 (suppl. material). □

Proposition 3. Analytical Variance. *Following assumptions of Proposition 2, the variance* ω_i^2 *of* $x_i^\bullet(z)$ *can be expressed as* $\omega_i^2 = \mathbb{E}(z^2) - (\mathbb{E}(z))^2 = \int_0^1 z^2 \cdot x_i^\bullet(z)\,\mathrm{d}z - \lambda_i^2 = 0.56419\,\gamma^{\frac{1}{2}}\frac{\Gamma(\frac{1}{2}+\alpha_i)}{\Gamma(\alpha_i)}\big(0.4 \cdot {}_2F_1\big(\frac{5}{2}, 1-\alpha_i, \frac{7}{2}, 1\big)\cdot {}_2F_1\big(\frac{5}{2}, \frac{3}{2}+\alpha_i, \frac{5}{2}+\alpha_i, 1-\gamma_i\big) + 0.28571(\gamma_i - 1)\cdot {}_2F_1\big(\frac{7}{2}, 1-\alpha_i, \frac{9}{2}, 1\big)\cdot {}_2F_1\big(\frac{7}{2}, \frac{3}{2}+\alpha_i, \frac{7}{2}+\alpha_i, 1-\gamma_i\big)\big) - \lambda_i^2.$

Proof. See Proof of Proposition 3 (suppl. material). □

Note on Spectrum Rebalancing. Fig. 3 explains the consequences of Prop. 2 & 3 and connects them with Alg. 1. Notice following: (i) For $k = 1$ (iterations), the analytical form (Fig. 3b) and the simulated incomplete power iteration (Fig. 3c) both indeed enjoy flatten ϕ and ϕ' for $1 \leq \sigma_1 \leq 3$. (ii) The injected variance is clearly visible in that flattened range (we know the quantity of injected noise). (iii) The analytical and simulated variances match.

Choice of Number of Iterations (k). In Fig. 3b, we plot $\phi(\sigma_i; k)$ using our analytical formulation. The best rebalancing effect is achieved for $k = 1$. For example, the red line ($k = 1$) is mostly flat for σ_i . This indicates the singular values $\sigma_i \geq 1$ are mapped to a similar value which promotes flattening of spectrum. When $\sigma_i \geq 2$ the green line eventually reduces to zero. This indicates that only datasets with spectrum falling into range between 1 and 1.2 will benefit from flattening. Important is to notice also that spectrum augmentation (variance) in Fig. 3b is highest for $k = 1$. Thus, in all experiments (including image classification), we set $k = 1$, and the SFA becomes:

$$\widetilde{\mathbf{H}} = \mathbf{H} - \mathbf{H}_{\text{LowRank}} = \mathbf{H}\bigg(\mathbf{I} - \frac{\mathbf{H}^\top\mathbf{H}\mathbf{r}^{(0)}\mathbf{r}^{(0)\top}\mathbf{H}^\top\mathbf{H}}{\|\mathbf{H}^\top\mathbf{H}\mathbf{r}^{(0)}\|_2^2}\bigg). \tag{4}$$

Why Does the Incomplete Power Iteration Work?

Having discussed SFA, below we show how SFA improves the alignment/generalization by flattening large and boosting small singular values due to rebalanced spectrum.

Improved Alignment. SFA rebalances the weight penalty (by rebalancing singular values) on orthonormal bases, thus improving the alignment of two correlated views. Consider the alignment part of Eq. (2) and ignore the projection head θ for brevity. The contrastive loss (temperature $\tau > 0$, n nodes) on $\mathbf{H}^\alpha$ and $\mathbf{H}^\beta$ maximizes the alignment of two views:

$$\begin{aligned}\mathcal{L}_a &= \mathop{\mathbb{E}}_{\mathbf{h}^\alpha,\mathbf{h}^\beta}(\mathbf{h}^{\alpha\top}\mathbf{h}^\beta/\tau) = \frac{1}{n\tau}\mathrm{Tr}(\mathbf{H}^{\alpha\top}\mathbf{H}^\beta)\\ &= \frac{1}{n\tau}\sum_{i=1}^{d_h}(\sigma_i^\alpha\mathbf{v}_i^{\alpha\top}\mathbf{v}_i^\beta)(\sigma_i^\beta\mathbf{u}_i^{\alpha\top}\mathbf{u}_i^\beta).\end{aligned} \tag{5}$$

The above equation indicates that for $\sigma_i^\alpha \geq 0$ and $\sigma_i^\beta \geq 0$, the maximum is reached if the right and left singular value matrices are perfectly aligned, *i.e.*, $\mathbf{U}^\alpha = \mathbf{U}^\beta$ and $\mathbf{V}^\alpha = \mathbf{V}^\beta$. Notice the singular values σ_i^α and σ_i^α serve as weighs for the alignment of singular vectors. As the singular value gap $\Delta\sigma_{12} = \sigma_1 - \sigma_2$ is usually significant (spectrum of feature maps usually adheres to the power law $ai^{-\kappa}$ (i is the index of sorted singular values, a and κ control the magnitude/shape), the large singular values tend to dominate the optimization. Such an issue makes Eq. (5) focus only on aligning the direction of dominant singular vectors, while neglecting remaining singular vectors, leading to a poor alignment of the orthonormal bases. In contrast, SFA alleviates this issue. According to Prop. 1, Eq. (5) with SFA becomes:

$$\begin{aligned}\mathcal{L}_a^* &= \mathop{\mathbb{E}}_{\mathbf{h}^\alpha,\mathbf{h}^\beta/\tau\mathbf{r}^\alpha,\mathbf{r}^\beta\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\mathop{\mathbb{E}}(\tilde{\mathbf{h}}^{\alpha\top}\tilde{\mathbf{h}}^\beta)\\ &= \frac{1}{n\tau}\sum_{i=1}^{d_h}(1-\lambda_i^\alpha)\sigma_i^\alpha(\mathbf{v}_i^{\alpha\top}\mathbf{v}_i^\beta)(1-\lambda_i^\beta)\sigma_i^\beta(\mathbf{u}_i^{\alpha\top}\mathbf{u}_i^\beta).\end{aligned} \tag{6}$$

Eq. (6) shows that SFA limits the impact of leading singular vectors on the alignment step if SFA can rebalance the spectra. Indeed, Figure 3b shows the spectrum balancing effect and one can see that $\phi(\sigma_i; k) = \sigma_i(1-\lambda_i(k)) \leq \sigma_i$. The same may be concluded from $0 \leq \lambda_i \leq 1$ in Prop. 2. See the Upper bound of ϕ section (supplementary material) for the estimated upper bound of ϕ. Finally, the Experiments section also shows empirically that SFA leads to a superior alignment.

Improved Alignment Yields Better Generalization Bound. To show SFA achieves the improved generalization bound we quote the following theorem (Huang, Yi, and Zhao 2023).

Theorem 1. *Given a Nearest Neighbor classifier G_f, the downstream error rate of G_f is $\mathrm{Err}\,(G_f) \leq (1-\sigma) + R_\varepsilon$, where $R_\varepsilon = P_{\mathbf{x}_1,\mathbf{x}_2\in\mathcal{A}(\mathbf{x})}\{\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\| \geq \varepsilon\} \leq \frac{\sqrt{2-2\mathcal{L}_a}}{\varepsilon}$, σ is the parameter of the so-called (σ,δ)-augmentation (for each latent class, the proportion of samples located in a ball with diameter δ is larger than σ, $\mathcal{A}(\cdot)$ is the set of augmented samples, $f(\cdot)$ is the encoder, and $\{\|\cdot\| \geq \varepsilon\}$ is the set of samples with ε-close representations among augmented data.*

Proof. See Proof of Theorem 1 (supp. material). □

Theorem 1 says the key to better generalization of contrastive learning is better alignment $\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|$ of positive samples. SFA improves alignment by design. SeeWhy Does the Incomplete Power Iteration Work? See empirical result in the Experiments section. Good alignment (*e.g.*, Fig. 6) due to spectrum rebalancing (*e.g.*, Fig. 5) enjoys $\mathcal{L}_a \leq \mathcal{L}_a^*$ (Eq. (5) and (6)) so one gets $R_\varepsilon^* \leq R_\varepsilon$ and the lower generalization bound $\mathrm{Err}\left(G_f^*\right) \leq \mathrm{Err}\left(G_f\right)$. Asterisk * means SFA is used ($\mathcal{L}_a^*$ replaces $\mathcal{L}_a$).

Experiments

Below we conduct experiments on the node classification, node clustering, graph classification and image classification. For fair comparisons, we use the same experimental setup as the representative Graph SSL (GSSL) methods (*i.e.*, GCA (Zhu et al. 2020))

Method	WikiCS	Comput.	Photo	Cora	CiteSeer	PubMed
RAW	71.9±0.0	73.8±0.1	78.5±0.0	64.6±0.2	65.7±0.1	82.2±0.3
DeepW.	74.3±0.1	85.7±0.1	89.4±0.1	74.6±0.2	50.8±0.1	80.1±0.2
DGI	75.3±0.1	83.9±0.4	91.6±0.2	82.1±0.6	69.5±0.5	86.0±0.2
MVGRL	77.5±0.1	87.5±0.1	91.7±0.1	83.1±0.1	73.3±0.1	84.2±0.1
GRACE	78.1±0.4	87.2±0.2	92.1±0.2	83.5±0.2	73.6±0.2	85.5±0.3
GCA	78.3±0.1	87.8±0.3	92.4±0.1	82.8±0.2	72.8±0.1	85.1±0.2
SUGRL	77.7±0.2	88.8±0.2	93.2±0.4	83.4±0.5	73.0±0.4	84.9±0.3
MERIT	77.9±0.4	87.5±0.2	93.1±0.4	84.1±0.6	74.3±0.4	84.1±0.2
BGRL	79.1±0.6	87.3±0.4	91.5±0.4	83.7±0.5	73.0±0.1	84.6±0.3
G-BT	76.8±0.6	86.8±0.3	92.6±0.5	83.6±0.4	72.9±0.1	84.5±0.1
COSTA	79.1±0.1	88.3±0.1	92.5±0.4	84.3±0.2	72.9±0.3	86.0±0.2
$\mathrm{SFA}_{\mathrm{BT}}$	**80.2**±0.1	88.1±0.1	92.8±0.1	84.1±0.1	73.7±0.2	85.6±0.1
$\mathrm{SFA}_{\mathrm{NCE}}$	79.9±0.1	**89.2**±0.2	**93.5**±0.1	**85.8**±0.1	**75.3**±0.1	**86.2**±0.1

Table 1: Node classification on graph datasets. Note that $\mathrm{SFA}_{\mathrm{InfoNEC}}$ and $\mathrm{SFA}_{\mathrm{BT}}$ can be directly compared to GRACE and G-BT. (Accuracy is reported.)

Method	CIFAR10	CIFAR100	ImageNet-100
SimCLR	90.5	65.5	76.8
$\mathrm{SFA}_{\mathrm{SimCLR}}$	**91.6**	**66.7**	**77.7**
BalowTw	92.0	69.7	80.0
$\mathrm{SFA}_{\mathrm{BT}}$	**92.5**	**70.4**	**80.9**
Siamese	90.51	66.04	74.5
$\mathrm{SFA}_{\mathrm{Siamese}}$	**91.23**	**66.99**	**75.6**
SwAV	89.17	64.88	74.0
$\mathrm{SFA}_{\mathrm{SwAV}}$	**90.12**	**65.82**	**74.8**

Table 2: Image classification on CIFAR10, CIFAR100 and ImageNet-100.

Datasets. We use five popular datasets (Zhu et al. 2020, 2021b; Velickovic et al. 2019), including citation networks (Cora, CiteSeer) and social networks (Wiki-CS, Amazon-Computers, Amazon-Photo) (Kipf and Welling 2017; Sinha et al. 2015; McAuley et al. 2015; Mernyei and Cangea 2020). For graph classification, we use NCI1, PROTEIN and DD (Dobson and Doig 2003; Riesen and Bunke 2008). For image classification we use CIFAR10/100 and ImageNet-100 (Deng et al. 2009). See the Baseline Setting section for details (supplementary material).

Baselines. We focus on three groups of SSL models. The first group includes traditional GSSL, *i.e.*, Deepwalk (Perozzi, Al-Rfou, and Skiena 2014), node2vec (Grover and Leskovec 2016), and GAE (Kipf and Welling 2016). The second group is contrastive-based GSSL, *i.e.*, Deep Graph Infomax (DGI) (Velickovic et al. 2019), Multi-View Graph Representation Learning (MVGRL) (Hassani and Ahmadi 2020), GRACE (Zhu et al. 2020), GCA (Zhu et al. 2021b), (Jin et al. 2021) SUGRL (Mo et al. 2022). The last group does not require explicit negative samples, *i.e.*, Graph Barlow Twins(G-BT) (Bielak, Kajdanowicz, and Chawla 2022) and BGRL (Thakoor et al. 2021). We also compare SFA with COSTA (Zhang et al. 2022c) (GCL with feat. augmentation).

Method	NCI1	PROTEIN	DD
GraphCL	77.8±0.4	74.3±0.4	77.8±0.4
LP-Info	75.8±1.2	74.6±0.2	72.5±1.9
JOAO	78.0±0.4	74.5±0.4	77.5±0.5
SimGRACE	77.4±1.0	73.9±0.1	77.3±1.1
SFA_{NCE}	**78.7**±0.4	**75.4**±0.4	**78.6**±0.4

Table 3: Graph classification.(SFA_{NEC} can be directly compared with GraphCL.)

$\mathcal{A}_G$	$\mathcal{A}_F$	$\mathcal{A}_{F^*}$	Am-Comput.	Cora	CiteSeer
×	×	×	85.01	80.04	71.35
✓	×	×	87.25	82.23	74.56
×	×	✓	86.74	84.19	73.15
✓	×	✓	88.74	85.90	75.05
✓	✓	×	87.55	83.35	74.45
✓	✓	✓	88.69	84.50	74.70

Table 4: Ablation study on different augmentation strategies: $\mathcal{A}_G$, $\mathcal{A}_F$ and $\mathcal{A}_{F^*}$.

Ogb-arxiv	Validation	Test
MLP	57.6±0.2	55.5±0.3
Node2vec	71.2±0.3	70.0±0.3
MVGLR	69.3±0.3	68.2±0.2
DGI	71.2±0.1	70.3±0.1
SUGRL	70.2±0.1	69.3±0.2
MERIT	67.2±0.1	65.3±0.2
GRACE	71.4±0.5	70.8±0.1
G-BT	71.1±0.3	70.0±0.2
COSTA	71.6±0.4	71.0±0.4
SFA_{NCE}	**72.3**±0.1	**71.6**±0.4

Table 5: Node classification.

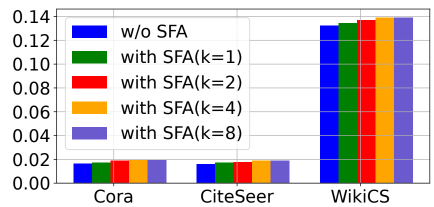


Figure 4: Running time per epoch in seconds.

Evaluation Protocol. We adopt the evaluation from (Velickovic et al. 2019; Zhu et al. 2020, 2021b). Each model is trained in an unsupervised manner on the whole graph with node features. Then, we pass the raw features into the trained encoder to obtain embeddings and train an ℓ_2-regularized logistic regression classifier. Graph Datasets are randomly divided into 10%, 10%, 80% for training, validation, and testing. We report the accuracy with mean/standard deviation over 20 random data splits.

Implementation Details. We use Xavier initialization for the GNN parameters and train the model with Adam optimizer. For node/graph classification, we use 2 GCN layers. The logistic regression classifier is trained with $5,000$ (guaranteed converge). We also use early stopping with a patience of 20 to avoid overfitting. We set the size of the hidden dimension of nodes to from 128 to 512. In clustering, we train a k-means clustering model. For the chosen hyper-parameters see Section D.2. We implement the major baselines using PyGCL (Zhu et al. 2021a). The detailed settings of augmentation and contrastive objectives are in Table 12 of Section D.3

Node Classification[3]**.** We employ node classification as a downstream task to showcase SFA. The default contrastive objective is InfoNCE or the BT loss. Table 1 shows that SFA consistently achieves the best results on all datasets. Notice that the graph-augmented GSSL methods, including GSSL with SFA, significantly outperform the traditional methods, illustrating the importance of data augmentation in GSSL. Moreover, we find that the performance of GCL methods (*i.e.*, GRACE, GCA, DGI, MVGRL) improves by a large margin when integrating with SFA (*e.g.*, major baseline, GRACE, yields 3% and 4% gain on Cora and Citeseer), which shows that SFA is complementary to graph augmentations. SFA also works with the BT loss and improves its performance.

Graph Classification. By adopting the graph-level GNN encoder, SFA can be used for the graph-level pre-training. Thus, we compare SFA with graph augmentation-based models (*i.e.*, GarphCL (You et al. 2020b), JOAO (You et al. 2021)) and augmentation-free models (*i.e.*, SimGrace (Xia et al. 2022), LP-Info (You et al. 2022)). Table 3 shows that SFA outperforms all baselines on the three datasets.

Image Classification[4]**.** As SFA perturbs the spectrum of feature maps, it is also applicable to the image domain. Table 2 presents the top-1 accuracy on CIFAR10/100 and ImageNet-100. See also the Implementation Details (supp. material).

Runtimes. Fig. 4 show SFA incurs a negligible runtime overhead (few matrix-matrix(vector) multiplications).

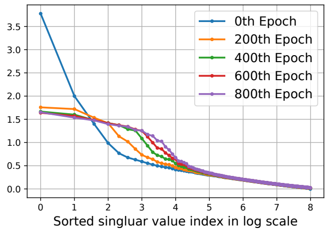
Improved Alignment. We show that SFA improves the alignment of two views during training. Let $\mathbf{H}^\alpha$ and $\mathbf{H}^\beta$ ($\widetilde{\mathbf{H}}^\alpha$ and $\widetilde{\mathbf{H}}^\beta$) denote the feature maps of two views without (with) applying SFA. The alignment is computed by $\|\mathbf{H}^\alpha - \mathbf{H}^\beta\|_F^2$ (or $\|\widetilde{\mathbf{H}}^\alpha - \widetilde{\mathbf{H}}^\beta\|_F^2$). Fig. 6 shows that SFA achieves better alignment. See also the Additional Empirical Results section.

Empirical Evolution of Spectrum. Fig. 5 (Cora) shows how the singular values of features maps $\mathbf{H}$ (without SFA) and $\widetilde{\mathbf{H}}$ (with SFA) evolve. As training progresses, the gap between consecutive singular values gradually decreases due to SFA. The leading components of spectrum (where the signal is) become more balanced: empirical results match Prop. 1 & 2.

Ablations on Augmentations. Below, we compare graph augmentation $\mathcal{A}_G$, channel feature augmentation $\mathcal{A}_F$ (random noise added to embeddings directly) and the spectral feature augmentation $\mathcal{A}_{F^*}$. Table 4 shows that using $\mathcal{A}_G$ or $\mathcal{A}_{F^*}$ alone improves performance. For example, $\mathcal{A}_G$ yields 2.2%, 2.2% and 3.2% gain on Am-Computer, Cora, and CiteSeer. $\mathcal{A}_{F^*}$ yields 1.7%, 4.1% and 1.8% gain on Am-Computer,

[3]Implementation and evaluation are based on PyGCL (Zhu et al. 2021a): https://github.com/PyGCL/PyGCL.

[4]Implementation/evaluation are based on Solo-learn (da Costa et al. 2022): https://github.com/vturrisi/solo-learn.



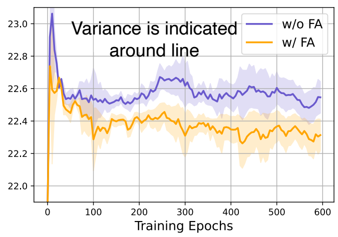
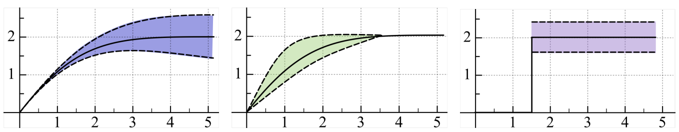


Figure 5: Spectrum of $\tilde{\mathbf{H}}$ when training on Cora with SFA.

Figure 6: Alignment of f. maps of two views (lower is better).



(a) SFA (b) MaxExp(F) (c) Grassman

Figure 7: Spectrum augmentation variants (dashed lines: injected noise var.) Power Norm. & Power Norm.* resemble (b) and (a).

Spectrum Aug.	**Am-Comput.**	**Cora**	**CiteSeer**
SFA (ours)	88.83	85.90	75.3
MaxExp(F)	87.13	84.19	72.85
MaxExp(F) (w/o noise)	86.83	83.52	72.35
Power Norm.*	86.7	82.9	70.4
Power Norm.	86.6	82.7	70.2
Power Norm. (w/o noise)	86.3	82.5	69.9
Grassman	86.23	82.57	70.15
Grassman (w/o noise)	85.83	81.17	69.85
Grassman (rand. SVD)	85.33	81.01	69.95
Matrix Precond.	82.52	78.14	67.73

Table 6: Results of various spectral augmentations.

Cora, and CiteSeer. Importantly, when both $\mathcal{A}_G$ and $\mathcal{A}_{F^*}$ are applied, the gain is **3.7**%, **6.0**% and **4.8**% on Am-Computer, Cora, and CiteSeer over "no augmentations". Thus, SFA is complementary to existing graph augmentations. We also notice that $\mathcal{A}_{F^*}$ with $\mathcal{A}_G$ outperforms $\mathcal{A}_F$ with $\mathcal{A}_G$ by 1.1%, 2.4% and 1.7%, which highlights the benefit of SFA.

Effect of Number of Iterations (k). In Eq. (3), we set $k \in \{0, 1, 2, 4, 8\}$ for our model with the InfoNCE loss on Cora, Citeseer and Am-Computer. The case of $k = 0$ means that no power iteration is used, *i.e.*, the solution simplifies to the feature augmentation by subtracting from $\mathbf{H}$ perturbation $\mathbf{H}\mathbf{r}^{(0)}\mathbf{r}^{(0)\top}/\|\mathbf{r}^{(0)}\|_2^2$ with random $\mathbf{r}^{(0)}$. Table 8 shows that without power iteration, the performance of the model drops. The best gain in Table 8 is achieved for $1 \le k \le 2$. This is consistent with Prop. 2, Fig. 3b and 3c, which show that the spectrum balancing effect (approximately flat region of ϕ) and significant spectrum augmentation (indicated by the large deviation in Fig. 3b) are achieved only when the power iteration is incomplete (low k, *i.e.*, $1 \le k \le 2$).

Robustness to Noisy Features. Below we check on Cora and CiteSeer if GCL with SFA is robust to noisy node features in node classification setting. We draw noise $\Delta\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$

ϵ		10^{-4}	10^{-3}	10^{-2}	10^{-1}
Cora	w/o SFA	80.55	70.67	62.85	59.23
	with SFA	83.14	76.56	69.59	62.55
CiteSeer	w/o SFA	67.48	62.32	52.69	42.81
	with SFA	72.12	70.36	60.40	46.44

Table 7: Results on noisy features.

Power Iteration	**Am-Computer**	**Cora**	**CiteSeer**
$k = 0$	87.25	83.04	71.35
$k = 1$	88.83	85.90	73.56
$k = 2$	88.72	85.59	75.32
$k = 4$	88.64	85.29	74.82
$k = 8$	88.27	85.23	74.91

Table 8: Results on different k of SFA.

and inject it into the original node features as $\mathbf{x} + \epsilon\Delta\mathbf{x}$, where $\epsilon \ge 0$ controls the noise intensity. Table 7 shows that SFA is robust to the noisy features. SFA partially balances the spectrum (Fig 3b and 2b) of leading singular components where the signal lies, while non-leading components where the noise resides are left mostly unchanged by SFA.

Other Variants of Spectral Augmentation. Below, we experiment with other ways of performing spectral augmentation. Figure 7 shows three different push-forward functions: our SFA, MaxExp(F) (Koniusz and Zhang 2020) and Grassman feature maps (Harandi et al. 2015). As MaxExp(F) and Grassman do not inject any spectral noise, we equip them with explicit noise injectors. To determine what is the best form of such an operator, we vary (i) where the noise is injected, (ii) profile of balancing curve. As the push-forward profiles of SFA and MaxExp(F) are similar (Figure 7), we implicitly inject the noise into MaxExp(F) along the non-leading singular values (*c.f*. leading singular values of SFA). We detail the formulation of these methods in supp. material[5].

Table 6 shows that SFA outperforms MaxExp(F), Power Norm., Power Norm.* and Grassman. As SFA augments parts of spectrum where signal resides (leading singular values) which is better than augmenting non-leading singular values (some noise may reside there) as in MaxExp(F). As Grassman binarizes spectrum, it may reject some useful signal at the boundary between leading and non-leading singular values. Finally, Matrix Preconditioning model reduces the spectral gap, thus rebalances the spectrum (also non-leading part).

Conclusions

We have shown that GCL is not restricted to only link perturbations or feature augmentation. By introducing a simple and efficient spectral feature augmentation layer we achieve significant performance gains. Our incomplete power iteration is very fast. Our theoretical analysis has demonstrated that SFA rebalances the useful part of spectrum, and also augments the useful part of spectrum by implicitly injecting the noise into singular values of both data views. SFA leads to a better alignment with a lower generalization bound.

[5]Our supplementary mat. is at: https://arxiv.org/abs/2212.01026.

Acknowledgments

We thank anonymous reviewers for their valuable comments. The work described here was partially supported by grants from the National Key Research and Development Program of China (No. 2018AAA0100204) and from the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 2410021, Research Impact Fund, No. R5034-18). PK was in part funded by CSIRO's Machine Learning and Artificial Intelligence Future Science Platform (MLAI FSP) Spatiotemporal Activity.

References

Bengio, Y.; Mesnil, G.; Dauphin, Y. N.; and Rifai, S. 2013. Better Mixing via Deep Representations. In *ICML*.

Bielak, P.; Kajdanowicz, T.; and Chawla, N. V. 2022. Graph Barlow Twins: A self-supervised representation learning framework for graphs. *Knowledge-Based Systems*, 256: 109631.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*.

Chen, Y.; Fang, Y.; Zhang, Y.; and King, I. 2023. Bipartite Graph Convolutional Hashing for Effective and Efficient Top-N Search in Hamming Space. *arXiv preprint arXiv:2304.00241*.

da Costa, V. G. T.; Fini, E.; Nabi, M.; Sebe, N.; and Ricci, E. 2022. solo-learn: A Library of Self-supervised Methods for Visual Representation Learning. *JMLR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

DeVries, T.; and Taylor, G. W. 2017. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*.

Dobson, P. D.; and Doig, A. J. 2003. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*.

Feng, S. Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; and Hovy, E. 2021. A Survey of Data Augmentation Approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

Grover, A.; and Leskovec, J. 2016. Node2vec: Scalable Feature Learning for Networks. In *KDD*.

Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.

Harandi, M.; Hartley, R.; Shen, C.; Lovell, B.; and Sanderson, C. 2015. Extrinsic Methods for Coding and Dictionary Learning on Grassmann Manifolds. *IJCV*.

Hariharan, B.; and Girshick, R. B. 2017. Low-Shot Visual Recognition by Shrinking and Hallucinating Features. In *ICCV*.

Hassani, K.; and Ahmadi, A. H. K. 2020. Contrastive Multi-View Representation Learning on Graphs. In *ICML*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. B. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

Huang, W.; Yi, M.; and Zhao, X. 2023. Towards the Generalization of Contrastive Self-Supervised Learning. In *ICLR*.

Jin, M.; Zheng, Y.; Li, Y.-F.; Gong, C.; Zhou, C.; and Pan, S. 2021. Multi-scale contrastive siamese networks for self-supervised graph representation learning. *ArXiv preprint*.

Jing, L.; Vincent, P.; LeCun, Y.; and Tian, Y. 2022. Understanding dimensional collapse in contrastive self-supervised learning. In *ICLR*.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR*.

Koniusz, P.; and Zhang, H. 2020. Power Normalizations in Fine-grained Image, Few-shot Image and Graph Classification. *TPAMI*.

McAuley, J. J.; Targett, C.; Shi, Q.; and van den Hengel, A. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*.

Mernyei, P.; and Cangea, C. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Mo, Y.; Peng, L.; Xu, J.; Shi, X.; and Zhu, X. 2022. Simple unsupervised graph representation learning. In *AAAI*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; and Zhang, C. 2018. Adversarially Regularized Graph Autoencoder for Graph Embedding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In *WWW*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: online learning of social representations. In *KDD*.

Prabowo, A.; Koniusz, P.; Shao, W.; and Salim, F. D. 2019. COLTRANE: ConvolutiOnaL TRAjectory NEtwork for Deep Map Inference. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys 2019, New York, NY, USA, November 13-14, 2019*, 21–30. ACM.

Riesen, K.; and Bunke, H. 2008. IAM graph database repository for graph based pattern recognition and machine learning. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R. S.; Giryes, R.; and Bronstein, A. M.

2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. In *NeurIPS*.

Shorten, C.; and Khoshgoftaar, T. M. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*.

Sinha, A.; Shen, Z.; Song, Y.; Ma, H.; Eide, D.; Hsu, B.-J.; and Wang, K. 2015. An overview of microsoft academic service (mas) and applications. In *WWW*.

Song, Z.; Meng, Z.; Zhang, Y.; and King, I. 2021. Semi-supervised Multi-label Learning for Graph-structured Data. In *CIKM*, 1723–1733.

Song, Z.; Zhang, Y.; and King, I. 2022. Towards an optimal asymmetric graph structure for robust semi-supervised node classification. In *KDD*.

Sun, K.; Koniusz, P.; and Wang, Z. 2019. Fisher-Bures Adversary Graph Convolutional Networks. *Conference on Uncertainty in Artificial Intelligence*, 115: 465–475.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*.

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

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What Makes for Good Views for Contrastive Learning? In *NeurIPS*.

Upchurch, P.; Gardner, J. R.; Pleiss, G.; Pless, R.; Snavely, N.; Bala, K.; and Weinberger, K. Q. 2017. Deep Feature Interpolation for Image Content Changes. In *CVPR*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *ICLR*.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *ICML*.

Wang, Y.; Pan, X.; Song, S.; Zhang, H.; Huang, G.; and Wu, C. 2019. Implicit Semantic Data Augmentation for Deep Networks. In *NeurIPS*.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In *EMNLP*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying Graph Convolutional Networks. In *ICML*, 6861–6871.

Xia, J.; Wu, L.; Chen, J.; Hu, B.; and Li, S. Z. 2022. SimGRACE: A Simple Framework for Graph Contrastive Learning without Data Augmentation. In *WWW*.

Yin, Y.; Wang, Q.; Huang, S.; Xiong, H.; and Zhang, X. 2022. Autogcl: Automated graph contrastive learning via learnable view generators. In *AAAI*.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph Contrastive Learning Automated. In *ICML*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020a. Graph Contrastive Learning with Augmentations. In *NeurIPS*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020b. Graph Contrastive Learning with Augmentations. In *NeurIPS*.

You, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2022. Bringing Your Own View: Graph Contrastive Learning without Prefabricated Data Augmentations. In *WSDM*.

Yu, T.; Cai, Y.; and Li, P. 2020. Toward faster and simpler matrix normalization via rank-1 update. In *European Conference on Computer Vision*, 203–219. Springer.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In *ICML*.

Zhang, H.; Zhang, J.; and Koniusz, P. 2019. Few-shot Learning via Saliency-guided Hallucination of Samples. In *CVPR*, 2770–2779.

Zhang, S.; Wang, L.; Murray, N.; and Koniusz, P. 2022a. Kernelized Few-Shot Object Detection With Efficient Integral Aggregation. In *CVPR*, 19207–19216.

Zhang, Y.; and Zhu, H. 2019. Doc2hash: Learning discrete latent variables for documents retrieval. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2235–2240.

Zhang, Y.; and Zhu, H. 2020. Discrete wasserstein autoencoders for document retrieval. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Zhang, Y.; Zhu, H.; Meng, Z.; Koniusz, P.; and King, I. 2022b. Graph-adaptive Rectified Linear Unit for Graph Neural Networks. In *Proceedings of the ACM Web Conference 2022*, 1331–1339.

Zhang, Y.; Zhu, H.; Song, Z.; Koniusz, P.; and King, I. 2022c. COSTA: Covariance-Preserving Feature Augmentation for Graph Contrastive Learning. In *KDD*.

Zhu, H.; and Koniusz, P. 2021. Simple Spectral Graph Convolution. In *ICLR*.

Zhu, H.; and Koniusz, P. 2022. Generalized Laplacian Eigenmaps. *NeurIPS*.

Zhu, H.; Sun, K.; and Koniusz, P. 2021. Contrastive Laplacian Eigenmaps. *NeurIPS*.

Zhu, Y.; Xu, Y.; Liu, Q.; and Wu, S. 2021a. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116*.

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

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021b. WWW. In *Proceedings of the Web Conference 2021*.