Through the Dual-Prism: A Spectral Perspective on Graph Data Augmentation for Graph Classifications

Yutong Xia[1*], **Runpeng Yu**[1*], **Yuxuan Liang**[2†],
Xavier Bresson[1], **Xinchao Wang**[1†], **Roger Zimmermann**[1]

[1]National University of Singapore
[2]The Hong Kong University of Science and Technology (Guangzhou)
{yutong.xia,r.yu}@u.nus.edu; yuxliang@outlook.com; {xaviercs,xinchao,dcsrz}@nus.edu.sg

Abstract

Graph Neural Networks have become the preferred tool to process graph data, with their efficacy being boosted through graph data augmentation techniques. Despite the evolution of augmentation methods, issues like graph property distortions and restricted structural changes persist. This leads to the question: *Is it possible to develop more property-conserving and structure-sensitive augmentation methods?* Through a spectral lens, we investigate the interplay between graph properties, their augmentation, and their spectral behavior, and observe that keeping the low-frequency eigenvalues unchanged can preserve the critical properties at a large scale when generating augmented graphs. These observations inform our introduction of the Dual-Prism (DP) augmentation methods, including DP-Noise and DP-Mask, which retain essential graph properties while diversifying augmented graphs. Extensive experiments validate the efficiency of our approach, providing a new and promising direction for graph data augmentation. Our source code is available at https://github.com/yu-rp/DualPrism.

Introduction

Graph structures, modeling complex systems through nodes and edges, are ubiquitous across various domains, including social networks (Newman, Watts, and Strogatz 2002), bioinformatics (Yi et al. 2022), and transportation systems (Jin et al. 2023a). Graph Neural Networks (GNNs) (Kipf and Welling 2016a) elegantly handle this relational information for tasks like accurate predictions. Their capabilities are further enhanced by graph data augmentation techniques, which artificially diversify the dataset via strategic manipulations, thereby improving the performance and generalization of GNNs. Graph data augmentation has progressed from early random topological modifications, e.g. DropEdge (Rong et al. 2019) and DropNode (Feng et al. 2020), to sophisticated learning-centric approaches like InfoMin (Suresh et al. 2021). Furthermore, techniques inspired by image augmentation's mixup principle (Zhang et al. 2017) have emerged as prominent contenders (Verma et al. 2019; Wang et al. 2021; Guo and Mao 2021).

*Y. Xia and R. Yu contributed equally to this work. † Y. Liang and X. Wang are the corresponding authors.

Though promising, these augmentation methods are challenged by three key issues as follows. (1) *Graph Property Distortion.* Before the era of deep learning, graph properties, e.g., graph connectivity and diameter, served as vital features for classification for decades (Childs et al. 2009). While now they seem to be ignored, many aforementioned contemporary augmentation methods appear to sidestep this tradition and overlook the graph properties. For instance, an example graph from the IMDB-BINARY dataset (Morris et al. 2020) and its augmented graph via DropEdge are illustrated in Figures 1a and 1b, respectively. The polar plot in Figure 1e shows the properties of these graphs, where each axis represents a distinct property. It is evident that DropEdge significantly alters the original graph's properties, as indicated by the stark difference between the shapes of the orange (original) and blue (augmented) pentagons. (2) *Limited Structural Impact.* The majority of existing methods' localized alterations do not capture the broader relationships and structures within the graph, limiting their utility. Consider a social network graph, where removing an edge affects just the immediate node and does little to alter the overall community structure. We thus ask: Can we design more *property-retentive* and *structure-aware* data augmentation techniques for GNNs?

Through the Dual-Prism: A Spectral Lens. Graph data augmentation involves altering components of an original graph. These modifications, in turn, lead to changes in the graph's spectral frequencies (Ortega et al. 2018). Recent research highlighted the importance of the graph spectrum: it can reveal critical graph properties, e.g., connectivity and radius (Chung 1997; Lee, Gharan, and Trevisan 2014). Additionally, it also provides a holistic summary of a graph's intrinsic structure (Chang et al. 2021), providing a global view for graph topology alterations. Building on this foundation, a pivotal question arises: *Could the spectral domain be the stage for structure-aware and property-retentive augmentation efforts?* Drawing inspiration from *dual prisms*—which filter and reconstruct light based on spectral elements—can we design a *polarizer* to shed new light on this challenge? With this in mind, we use spectral graph theory, aiming to answer the following questions: 1) Can a spectral approach to graph data augmentation preserve essential graph properties effectively? 2) How does spectral-based augmentation impact broader graph structures? 3) How does spectral-
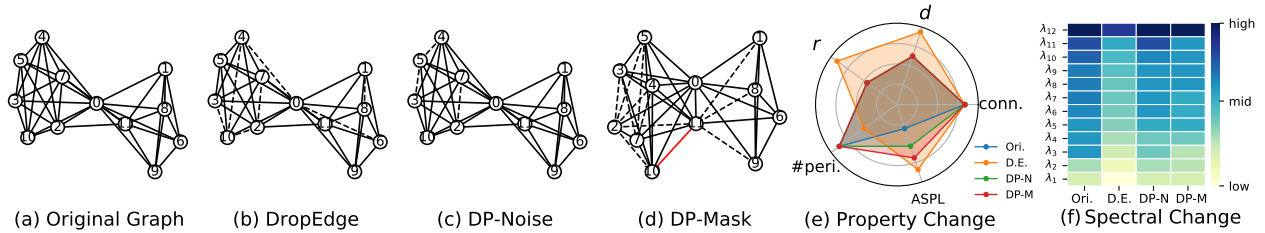
Figure 1: Visualization of (a) a graph from the IMDB-BINARY dataset and its augmented graphs via (b) DropEdge (Rong et al. 2019), (c) DP-Noise (ours), and (d) DP-Mask (ours). Dashed line: Dropped edge. Red line: Added edge. (e) Five properties of these graphs. r: radius. d: diameter. conn.: connectivity. ASPL: average shortest path length. #peri: number of periphery. Ori.: Original. D.E.: DropEdge. DP-N: DP-Noise. DP-M: DP-Mask. (f) The eigenvalues of these four graphs.

based augmentation compare to existing methods in enhancing the efficiency of GNNs for graph classification?

We begin with an empirical exploration, where we aim to understand the interplay between topological modifications and their spectral responses. Our insights reveal that changes in graph properties mainly manifest in *low-frequency components*. Armed with this, we unveil our Dual-Prism (DP) augmentation strategies, DP-Noise and DP-Mask, by only changing the high-frequency part of the spectrum of graphs. Figures 1c and 1d provide a visualization of the augmented graphs via our proposed methods, i.e., DP-Noise and DP-Mask. As shown in Figure 1e, compared with DropEdge, our approaches skillfully maintain the inherent properties of the original graph, differing only slightly in the ASPL. Note that although we solely present one example underscoring our method's capability, its robustness is consistently evident across all scenarios.

In addition to the properties, we further explore the spectrum comparison, shown in Figure 1f. Compared with DropEdge, the spectrum shifts caused by our methods are noticeably smaller. Interestingly, despite our approaches' relative stability in the spectral domain, they induce substantial changes in the spatial realm (i.e., notable edge modifications). This spectral stability helps retain the core properties, while the spatial variations ensure a rich diversity in augmented graphs. Conversely, DropEdge, despite only causing certain edge changes, disrupts the spectrum and essential graph properties significantly. Simply put, **our methods skillfully maintain graph properties while also diversifying augmented graphs.** In the Experiments section, we evaluate the efficacy of our methods on graph classification, across diverse settings: supervised, semi-supervised, unsupervised, and transfer learning on various real-world datasets. Our concluding thoughts are presented in the last section.

Contributions. Our main contributions are outlined as follows. *(1) Prism – Bridging Spatial and Spectral Domains:* We introduce a spectral lens to shed light on spatial graph data augmentation, aiming to better understand the spectral behavior of graph modifications and their interplay with inherent graph properties. *(2) Polarizer – Innovative Augmentation Method:* We propose the globally-aware and property-retentive augmentation methods, Dual-Prism (DP), including DP-Noise and DP-Mask. Our methods are able to preserve inherent graph properties while simultaneously enhancing the diversity of augmented graphs. *(3) New Light – Extensive Evaluations:* We conduct compre-

hensive experiments spanning supervised, semi-supervised, unsupervised, and transfer learning paradigms on 21 real-world datasets. The experimental results demonstrate that our proposed methods can achieve state-of-art performance on the majority of datasets.

Related work

Data Augmentations for GNNs. Graph data augmentation refers to the process of modifying a graph to enhance or diversify the information contained within, which can be used to bolster the training dataset for better generalization or model variations in real-world networks (Ding et al. 2022; Zhao et al. 2022). Early methods are grounded in random modification to the graph topology. Techniques like DropEdge (Rong et al. 2019), DropNode (Feng et al. 2020), and random subgraph sampling (You et al. 2020) introduce stochastic perturbations in the graph structure. In addition to random modification, there is a wave of methods utilizing more sophisticated, learning-based strategies to generate augmented graphs (Suresh et al. 2021). Another research line is inspired by the efficiency of mixup (Zhang et al. 2017) in image augmentation, blending node features or entire subgraphs to create hybrid graph structures (Verma et al. 2019; Wang et al. 2021; Guo and Mao 2021; Han et al. 2022; Park, Shim, and Yang 2022; Ling et al. 2023). However, while the above techniques have advanced the field of graph data augmentation, challenges remain, especially in preserving broader structural changes and graph semantics.

Spectrum and GNNs. Spectral graph theory (Chung 1997) has significantly influenced GNNs(Ortega et al. 2018; Wu et al. 2019; Dong et al. 2020; Bo et al. 2021; Chang et al. 2021; Yang et al. 2022), evolving from defining early GNN convolutions using the Laplacian spectrum (Hammond, Vandergheynst, and Gribonval 2011; Defferrard, Bresson, and Vandergheynst 2016) to enhancing scalability (Nt and Maehara 2019). This approach is pivotal in various domains, including graph contrastive learning (GCL) (Liu et al. 2022; Lin, Chen, and Wang 2022; Yang et al. 2023; Chen, Lei, and Wei 2024), adversarial attacks (Entezari et al. 2020; Chang et al. 2021), and multivariate time series (Cao et al. 2020; Jin et al. 2023b). Zooming into the GCL domain, where data augmentation plays a pivotal role, (Liu et al. 2022) introduced the general rule of effective augmented graphs in GCL via a spectral perspective; GCL-SPAN (Lin, Chen, and Wang 2022) presented a novel augmentation method for GCL, focusing on the invariance of graph representation in the spectral domain; GASSER (Yang et al. 2023) uses a se-
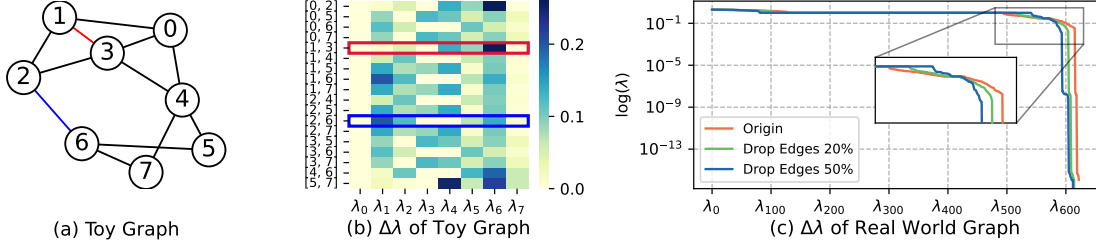
Figure 2: (a) A toy graph $\mathcal{G}$ consisting of eight nodes. (b) Absolute variation in eigenvalues of $\mathcal{G}$ when adding an edge at diverse positions. The red and blue rectangles represent when adding the corresponding edges in $\mathcal{G}$ and the change of the eigenvalues. (c) A real-world case in the REDDIT-BINARY dataset where, when dropping 20% and 50%, the high frequency is more vulnerable.

lective perturbation across various frequency bands to maintain homophily ratios while introducing perturbations to optimize task-relevant features; and PolyGCL (Chen, Lei, and Wei 2024) demonstrates the spectral filters can be effectively utilized for contrastive learning on graphs. Though promising, these methods do not fully address the intricacies of maintaining *inherent graph properties* during augmentation.

A Spectral Lens on Graph Data Augmentations

Preliminaries. An undirected graph $\mathcal{G}$ is represented as $\mathcal{G} = (V, E)$ where V is the set of nodes with $|V| = N$ and $E \subseteq V \times V$ is the set of edges. Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix of $\mathcal{G}$, with elements $a_{ij} = 1$ if there is an edge between nodes i and j, and $a_{ij} = 0$ otherwise. Let $D \in \mathbb{R}^{N \times N}$ be the degree matrix, which is a diagonal matrix with elements $d_{ii} = \sum_j a_{ij}$, representing the degree of node i. The Laplacian matrix of $\mathcal{G}$ is denoted as $L = D - A \in \mathbb{R}^{N \times N}$. The eigen-decomposition of L is denoted as $U \Lambda U^\top$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_N)$ and $U = [u_1^\top, \ldots, u_N^\top] \in \mathbb{R}^{N \times N}$. For graph $\mathcal{G}$, L has n non-negative real eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$. Specifically, the *low-frequency components* refer to the eigenvalues closer to 0, and the *high-frequency components* refer to the relatively larger eigenvalues.

Spectral Analysis Insights

Adopting a spectral viewpoint, we conduct a thorough empirical study to understand the interplay between graph properties, graph topology alterations in the spatial domain, and their corresponding impacts in the spectral domain. The findings from our analysis include four crucial aspects as detailed below.

Obs 1. The position of the edge flip influences the magnitude of spectral changes. In Figures 2a and 2b, we explore how adding different edges to a toy graph affects its eigenvalues. For instance, the addition of the edge $1 \leftrightarrow 3$ (shown as the red line), which connects two proximate nodes, primarily impacts the high-frequency component λ_6 (the red rectangle). In contrast, when adding edge $2 \leftrightarrow 6$ (the blue line) between two distant nodes, the low-frequency component λ_1 exhibits the most noticeable change (the blue rectangle). These variations in the spectrum underscore the significance of the edge-flipping position within the graph's overall topology, whose insight is consistent with findings

by (Entezari et al. 2020; Chang et al. 2021). Such spectral changes not only affect the graph's inherent structural features but also potentially affect the outcomes of tasks relying on spectral properties, such as graph-based learning.

Obs 2. Low-frequency components display greater resilience to edge alterations. Building on Obs 1, we further investigate the different responses of high- and low-frequency components to topology alterations using a real-world graph. We apply DropEdge (Rong et al. 2019) for augmentation by first randomly dropping 20% and 50% of edges and then computing the corresponding eigenvalues, as depicted in Figure 2c. It can be observed that, under random edge removal, low-frequency components exhibit greater robustness compared to their high-frequency counterparts.

Obs 3. Graph properties are crucial for graph classification. Certain fundamental properties of graphs, e.g., diameter and radius, are critical for a variety of downstream tasks, including graph classification (Feragen et al. 2013). In Figures 4a and 4b, we present the distributions of two key graph properties – diameter d and radius r – across the two classes in the REDD-M12 dataset. The different variations in these distributions emphasize their critical role in graph classification. Nevertheless, arbitrary modifications to the graph's topology, e.g., random-manner-based augmentation techniques, could potentially distort these vital properties, illustrated in Figures 1b and 1e.

Obs 4. Specific low-frequency eigenvalues are closely tied to crucial graph properties. From Obs 3, a question is raised: *Can we retain the integrity of these essential graph properties during augmentation?* To explore this, we turn our attention back to the toy graph in Figure 2a and examine the evolution of its properties and eigenvalues in response to single-edge flips. In Figures 4c and 4d, we chart the graph's average shortest path length (denoted by blue dots), diameter d (denoted by green dots) against the reciprocal of its second smallest eigenvalue $1/\lambda_1$ (denoted by red dots)[*]. Our observations reveal a notable correlation between the alterations in d and $1/\lambda_1$.

Further, we investigate correlations among overall spectral shifts, graph properties, and specific eigenvalues. Consistent with established methodologies (Lin, Chen, and Wang 2022; Wang et al. 2022; Wills and Meyer 2020), we adopt the Frobenius distance to quantify the overall spec-

---

[*]For visual clarity, we scaled d and $1/\lambda_1$ by dividing it by 3 and 5, respectively.

(a) Diverse roles of selected eigenvalues   (b) Relative significance of high- and low-frequency eigenvalues
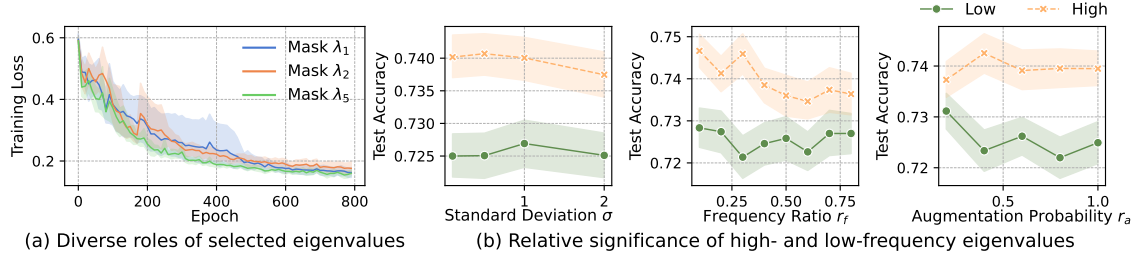
Figure 3: (a) Training loss of GIN model on REDDIT-BINARY when graphs are augmented by masking different eigenvalues. (b) Graph classification performance on IMDB-BINARY under various hyperparameters. The lines represent the average accuracy, while the shaded areas indicate the error margins.
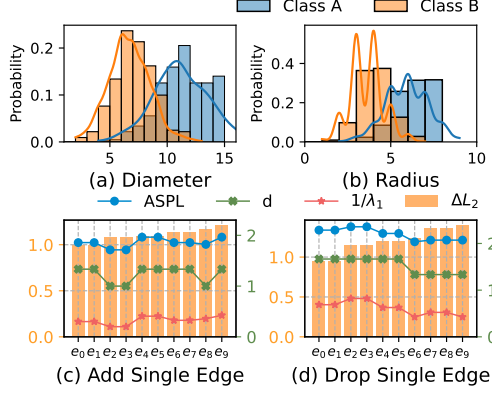

Figure 4: (a) Diameter and (b) radius distributions of different classes in REDD-M12. When (c) adding or (d) removing an edge, variation of the spectral domain ΔL_2, $1/\lambda_1$ of $\mathcal{G}'$, ASPL and diameter d of $\mathcal{G}'$.

tral variations by computing the L_2 distance between the spectrum of $\mathcal{G}$ and augmented graph $\mathcal{G}'$. Notably, this spectral shift does not directly correspond with the changes in properties or eigenvalues. This suggests that by maintaining critical eigenvalues, primarily the low-frequency components, we can inject relatively large spectral changes without affecting essential graph properties. This observation thus leads to our proposition: *preserving key eigenvalues while modifying others enables the generation of augmented graphs that uphold foundational properties*, instead of only focusing on the overall spectral shifts.

Methodology

Drawing inspiration from how prisms decompose and reconstruct light and how a polarizer selectively filters light (Figure 5a), we design our own "polarizer", i.e., the **Dual-Prism** (DP) method for graph data augmentation (Figure 5b).

Proposed Augmentation Methods

The proposed DP methods, including *DP-Noise* and *DP-Mask*, obtain the augmented graphs by directly changing the spectrum of graphs. A step-by-step breakdown is delineated in Algorithm 1. The DP method starts by extracting the Laplacian Matrix L of a graph $\mathcal{G}$ by $L = D - A$ and then computes its eigen-decomposition $L = U\Lambda U^\top$. Based on the frequency ratio r_f, N_a eigenvalues are selected for augmentation, where $N_a = N \times r_f$. Note that since we only target high-frequency eigenvalues, we arrange eigenvalues in increasing order and only focus on the *last N_a eigenvalues*. Then, a binary mask M is formed based on the augmentation ratio r_a to sample the eigenvalues to make a change.

Depending on the chosen augmentation type T, either noise is infused to the sampled eigenvalues, modulated by σ and M (from Line 6 to 8), or the eigenvalues are adjusted using the mask M directly (from Line 9 to 10). Finally, we reconstruct the new Laplacian $\hat{L}$ based on the updated eigenvalues $\hat{\Lambda}$. Given the Laplacian matrix is $L = D - A$, where D is a diagonal matrix, an updated adjacency matrix $\hat{A}$ can be derived by eliminating self-loops. Lastly, we obtain the augmented graph $\hat{\mathcal{G}}$ with its original labels and features retained. **Selection of** L. Note that we adopt the Laplacian matrix L instead of the normalized Laplacian matrix $L_{\text{norm}} = I - D^{-1/2}AD^{-1/2}$. The rationale behind this choice is that reconstructing the adjacency matrix using L_{norm} necessitates solving a system quadratic equation, where the number of unknown parameters equals the number of nodes in the graph. The computational complexity of this solution is more than $O(N^3)$. Even if we approximate it as a quadratic optimization problem, making it solvable with a gradient-based optimizer, the computational overhead introduced by solving such an optimization problem for each graph to be augmented is prohibitively high.

Empirical Evidence

Next, we verify our methods on the improvement of the performance for graph classification via experimental analysis.
Diverse roles of eigenvalues. We begin by masking selected eigenvalues to generate graphs for 20% of REDDIT-BINARY. The training loss when masking the eigenvalues λ_1, λ_2 and λ_5 is shown in Figure 3a. These curves suggest that individual eigenvalues contribute differently to the training process. Specifically, masking λ_1 results in a notably unstable training loss for the initial 500 epochs, evidenced by the expansive blue-shaded region. For λ_2, while the shaded region's extent is smaller, the curve exhibits noticeable fluctuations, particularly around epoch 200. Conversely, when masking λ_5, the training appears more stable, with the green curve showing relative steadiness and a reduced shaded area. These demonstrate the various significance of eigenvalues: λ_1 and λ_2 include more crucial structural and property details of the graph compared to λ_5. As a result, they deserve to be prioritized for preservation during augmentation.
Different importance of high- and low-frequency parts. We then conduct experiments on group-level eigenvalues, i.e., the high- and low-frequency ones, to gain a broader view of the influence exerted by varying frequency eigenvalues. We introduce noise to eigenvalues across various hyperparameter combinations. Concretely, we use the stan-
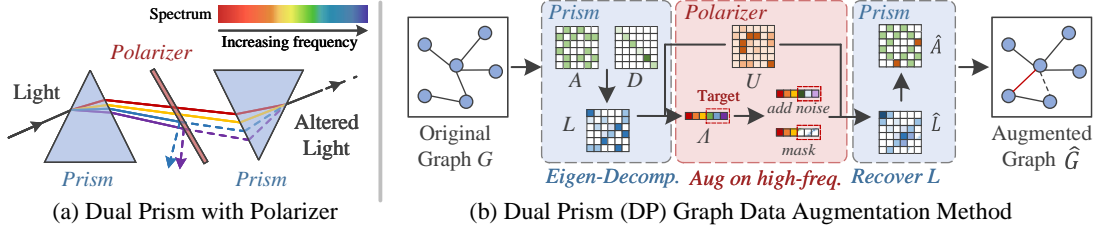
Figure 5: The framework of our Dual-Prism (DP) for graph data augmentation.

dard deviation σ to determine the magnitude of the noise. The frequency ratio r_f dictates the number of eigenvalues N_a we might change, while the augmentation probability r_a specifies the final eigenvalues sampled for modification. The eigenvalues are arranged in ascending order. The setting of 'Low' means that we select candidates from the first N_a eigenvalues, while 'High' denotes selection from the last N_a eigenvalues. As shown in Figure 3b, the orange lines consistently outperform the green lines across all three plots, indicating that the performance associated with perturbing high-frequency eigenvalues consistently exceeds that of their low-frequency counterparts. Moreover, as the frequency ratio r_f increases, the accuracy of the 'Low' scenario remains relatively stable and low. Contrastingly, for the 'High' scenario, a notable decline in accuracy begins once the ratio exceeds around 30%. This suggests that the eigenvalues outside the top 30% of the high-frequency range may start to include more critical information beneficial for graph classification tasks that should not be distorted.

Theoretical Backing and Insights

The eigenvalues of the Laplacian matrix provide significant insights into various graph properties (Chung 1997). Such insights have driven and backed our proposal to modify the high-frequency eigenvalues while preserving their low-frequency counterparts. For example, the second-smallest eigenvalue λ_1, often termed the *Fiedler value*, quantifies the

graph's *algebraic connectivity*. A greater Fiedler value indicates a better-connected graph and it is greater than 0 if and only if the graph is connected. The number of times 0 appears as an eigenvalue is the number of connected components in the graph. In addition to the connectivity, the diameter of a graph is also highly related to the eigenvalues – it can be upper and lower bounded from its spectrum (Chung 1997): $4/n\lambda_1 \leq d \leq 2[\sqrt{2m/\lambda_1}\log_2 n]$, where n and m denotes the number of nodes and the maximum degree of the graph. Besides these widely-used ones, other properties are also highly related to spectrum, e.g., graph diffusion distance (Hammond, Gur, and Johnson 2013). In essence, eigenvalues serve as powerful spectral signatures comprising a myriad of structural and functional aspects of graphs.

Discussion & Comparison

Advantages of the proposed method. At the heart of data augmentation lies a dual objective: to preserve the essence of the data while expanding its diversity. Our approach strikes this balance skillfully. Leveraging spectral view, it safeguards key graph property (Figure 1e), ensuring the augmented data remains authentic and relevant. This not only enhances the training data's quality but also directly boosts the performance, as detailed in the following section.

Comparison with existing works. Our method distinguishes itself from three related spectral view works: (Liu et al. 2022), (Lin, Chen, and Wang 2022), and (Yang et al. 2023). (Liu et al. 2022) focuses on optimal contrastive pair selection instead of a general augmentation method, and GCL-SPAN (Lin, Chen, and Wang 2022) centers on maximizing spectral variance. However, our observations suggest that broad spectral changes do not always correspond with graph properties. In addition, though GCL-SPAN also use a spectral perspective, it essentially alters the spatial domain, instead of direct spectral modifications, while our method's enhanced performance underlines the effectiveness of spectral alterations. GASSER(Yang et al. 2023) perturbs different frequency bands to uphold homophily ratios and fine-tune task-specific features. Whereas our approach concentrates on the essential structural integrity required for graph classification.

Experiments

Experimental Setup. We evaluate our method for graph classification under four different settings, including supervised learning, semi-supervised learning, unsupervised learning, and transfer learning. We conduct our experiments on 21 real-world datasets across three different domains, including bio-informatics, molecule, and social network, from the TUDatasets (Morris et al. 2020), OGB (Hu et al. 2020a) and ZINC chemical molecule dataset (Hu et al. 2020b).

Algorithm 1: Dual-Prism Augmentation

Input: Graph $\mathcal{G}$, Frequency Ratio r_f, Augmentation Ratio r_a, Standard Deviation σ, Augmentation Type T.
1: $N \leftarrow$ the number of nodes in $\mathcal{G}$
2: $L \leftarrow$ Laplacian Matrix of $\mathcal{G}$
      # Λ is arranged in increasing order.
3: U and $\Lambda \leftarrow$ eigenvalue decomposition of L
      # Get the number of eigenvalues to be augmented.
4: $N_a \leftarrow int(N \times r_f)$
5: $M \leftarrow \{m_i \sim Bern(r_a)\}_{i=1}^{N_a}$
      # Add noise to the high-frequency part.
6: **if** $T =$ noise **then**
7:     $\epsilon \leftarrow \{\epsilon_i \sim \mathcal{N}(0,1)\}_{i=1}^{N_a}$
8:     **for** $i \in \{1, \cdots, N_a\}$ **do**$\{\lambda_{N-i} \leftarrow \max(0, \lambda_{N-i} + \sigma M_i \epsilon_i)\}$
      # Mask the high-frequency part.
9: **else if** $T =$ mask **then**
10:     **for** $i \in \{1, \cdots, N_a\}$ **do** $\{\lambda_{N-i} \leftarrow (1 - M_i)\lambda_i\}$
       # Calculate the new Laplacian and new adjacent matrix.
11: $\hat{L} \leftarrow U^\top \hat{\Lambda} U, \hat{A} \leftarrow -\hat{L}$
12: **for** $i \in \{1, \cdots, N\}$ **do** $\{\hat{A}_{ii} \leftarrow 0\}$
Output: Augmented $\hat{\mathcal{G}}$ with edge_index derived from $\hat{A}$, and the label and features unchanged

Table 1: Performance comparisons with GCN and GIN in the *supervised* learning setting. The best and second best results are highlighted with **bold** and <u>underline</u>, respectively. * and ** denote the improvement over the second best baseline is statistically significant at level 0.1 and 0.05 (Newey and West 1987). Baseline results are taken from (Ling et al. 2023; Han et al. 2022).

|	Dataset	IMDB-B	IMDB-M	REDD-B	REDD-M5	REDD-M12	PROTEINS	NCI1	ogbg-molhiv
GCN	Vanilla	72.80±4.08	49.47±2.60	84.85±2.42	49.99±1.37	46.90±0.73	71.43±2.60	<u>72.38±1.45</u>	96.80±0.06
	DropEdge	73.20±5.62	49.00±2.94	85.15±2.81	51.19±1.74	47.08±0.55	71.61±4.28	68.32±1.60	96.53±0.07
	DropNode	73.80±5.71	50.00±4.85	83.65±3.63	47.71±1.75	47.93±0.64	72.69±3.55	70.73±2.02	96.52±0.14
	Subgraph	70.90±5.07	49.80±3.43	68.41±2.57	47.31±5.23	47.49±0.93	67.93±3.24	65.05±4.36	96.49±0.17
	M-Mixup	72.00±5.66	49.73±2.67	<u>87.05±2.47</u>	51.49±2.00	46.92±1.05	71.16±2.87	71.58±1.79	-
	SubMix	72.30±4.75	49.73±2.88	85.15±2.37	52.87±2.19	-	72.42±2.43	71.65±1.58	-
	G-Mixup	73.20±5.60	50.33±3.67	86.85±2.30	51.77±1.42	<u>48.06±0.53</u>	70.18±2.44	70.75±1.72	-
	S-Mixup	74.40±5.44	50.73±3.66	**89.30±2.69**	53.29±1.97	-	73.05±2.81	**75.47±1.49**	96.70±0.20
	DP-Noise	**77.90±2.30 ***	**53.60±1.59****	84.60±7.61	**53.42±1.36**	**48.47±0.57**	**75.03±2.66**	69.20±2.57	**97.02±0.19****
	DP-Mask	76.00±3.62	51.20±1.73	76.70±1.34	52.42±2.78	47.25±1.12	73.60±3.10	62.45±3.80	96.90±0.24*
GIN	Vanilla	71.30±4.36	48.80±2.54	89.15±2.47	53.17±2.26	50.23±0.83	68.28±2.47	79.08±2.12	96.54±0.11
	DropEdge	70.50±3.80	48.73±4.08	87.45±3.91	54.11±1.94	49.77±0.76	68.01±3.22	76.47±2.34	96.48±0.09
	DropNode	72.00±6.97	45.67±2.59	88.60±2.52	53.97±2.11	49.95±1.70	69.64±2.98	74.60±2.12	96.51±0.16
	Subgraph	70.40±4.98	43.74±5.74	76.80±3.87	50.09±4.94	49.67±0.90	66.67±3.10	60.17±2.33	96.49±0.13
	M-Mixup	72.00±5.14	48.67±5.32	87.70±2.50	52.85±1.03	49.81±0.80	68.65±3.76	79.85±1.88	-
	SubMix	71.70±6.20	49.80±4.01	90.45±1.93	54.27±2.92	-	69.54±3.15	79.78±1.09	-
	G-Mixup	72.40±5.64	49.93±2.82	90.20±2.84	54.33±1.99	<u>50.50±0.41</u>	64.69±3.60	78.20±1.58	-
	S-Mixup	73.40±6.26	50.13±4.34	90.55±2.11	55.19±1.99	-	69.37±2.86	80.02±2.45	96.84±0.40
	DP-Noise	**78.40±1.82****	**61.67±0.71****	**93.42±1.41****	**57.72±1.87****	**53.70±1.16****	**73.51±4.54****	**90.56±5.78****	**97.43±0.48****
	DP-Mask	76.30±2.56	51.60±1.32	<u>93.25±1.19**</u>	<u>56.50±0.80*</u>	49.11±1.30	<u>72.79±1.95**</u>	<u>80.30±2.01</u>	96.98±0.29
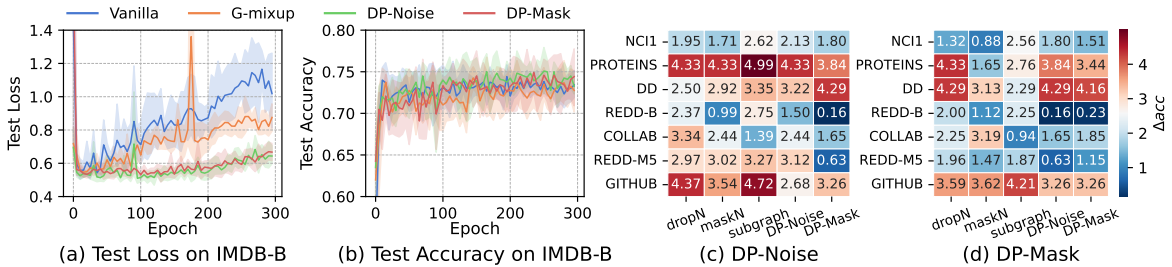
Figure 6: (a) The loss and (b) accuracy curves for supervised learning on test data of IMDB-BINARY using GIN, with four augmentation methods. Curves represent mean values from 5 runs, and shaded areas indicate standard deviation. The accuracy gain (%) in semi-supervised learning when contrasting 5 different augmentation methods with (c) DP-Noise and (b) DP-Mask across 7 datasets. Warmer color means better performance gains.

Table 2: Performance comparisons in the *semi-supervised* learning setting with 1% label ratio. The best and second best results are highlighted with **bold** and <u>underline</u>, respectively. Baseline results are taken from (You et al. 2020).

Dataset	NCI1	COLLAB	GITHUB
Vallina	60.72±0.45	57.46±0.25	54.25±0.22
Aug.	60.49±0.46	58.40±0.97	56.36±0.42
GAE	61.63±0.84	63.20±0.67	59.44±0.44
Infomax	<u>62.72±0.65</u>	61.70±0.77	58.99±0.50
GraphCL	62.55±0.86	<u>64.57±1.15</u>	58.56±0.59
DP-Noise	**63.43±1.39**	**65.94±3.13**	**60.06±2.72**
DP-Mask	62.43±1.08	<u>65.68±1.66*</u>	59.70±0.53

Supervised learning

Performance. We first evaluate our proposed methods in the supervised learning setting. Following the prior works (Han et al. 2022; Ling et al. 2023), we use GIN and GCN as backbones for graph classification on eight different datasets. Table 1 shows the performance of DP methods compared with seven state-of-art (SOTA) baselines, including DropEdge(Rong et al. 2019), DropNode(Feng et al. 2020), Subgraph(You et al. 2020), M-Mixup (Verma et al. 2019), SubMix(Yoo, Shim, and Kang 2022), G-Mixup(Han et al. 2022) and S-Mixup (Ling et al. 2023). According to the results, DP-Noise consistently outperforms other existing methods across the majority of datasets, establishing its dominance in

effectiveness. DP-Mask also shines, often securing a second-place standing. GIN tends to obtain superior outcomes, especially when combined with DP-Noise, exemplified by its 61.67% accuracy on IMDB-M. Note that on REDD-B, GIN achieves more satisfactory performance than GCN, which is a consistent pattern across baselines but becomes particularly pronounced with our methods. This phenomenon may be attributed to the intrinsic characteristics of GIN and GCN: GIN is known for its precision in capturing the complex structures of graphs (Xu et al. 2018), while GCN is characterized by its smoothing effect (Defferrard, Bresson, and Vandergheynst 2016). Our methods' superiority in diversifying the graphs' structures naturally amplifies GIN's strengths. In contrast, GCN may not be as adept at leveraging the enhancements offered by our techniques.

Generalization. Figures 6a and 6b display the test loss and accuracy curves for the IMDB-B dataset, comparing four distinct augmentation strategies: G-mixup, DP-Noise, DP-Mask, and a scenario without any augmentation (i.e., Vanilla). The consistently lower and more stable test loss curves for DP-Noise and DP-Mask is comparison to Vanilla and G-mixup. Concurrently, the accuracy achieved with DP-Noise and DP-Mask is higher, indicating the DP methods' superior generalization and the capacity for enhancing

Table 3: Performance comparisons in the *semi-supervised* learning setting with 10% label ratio. The best and second best results are highlighted with **bold** and <u>underline</u>, respectively. The metric is accuracy (%). * and ** denote the improvement over the second-best baseline is statistically significant at levels 0.1 and 0.05. Baseline results are taken from (You et al. 2020).

Dataset	NCI1	PROTEINS	DD	COLLAB	REDD-B	REDD-M5	GITHUB
Vallina	73.72±0.24	70.40±1.54	73.56±0.41	73.71±0.27	86.63±0.27	51.33±0.44	60.87±0.17
Aug.	73.59±0.32	70.29±0.64	74.30±0.81	74.19±0.13	87.74±0.39	52.01±0.20	60.91±0.32
GAE	74.36±0.24	70.51±0.17	74.54±0.68	75.09±0.19	87.69±0.40	<u>53.58±0.13</u>	63.89±0.52
Infomax	74.86±0.26	72.27±0.40	75.78±0.34	73.76±0.29	88.66±0.95	53.61±0.31	65.21±0.88
GraphCL	74.63±0.25	74.17±0.34	76.17±1.37	74.23±0.21	89.11±0.19	52.55±0.45	**65.81±0.79**
DP-Noise	**75.30±0.58****	**74.73±1.01**	**76.91±0.81**	**77.05±0.82****	**89.38±0.95**	**54.45±0.64****	<u>65.59±0.88</u>
DP-Mask	74.88±1.84	71.37±4.18	75.64±0.81	76.90±0.62**	88.62±0.63	52.80±0.59	64.95±1.03

Table 4: Performance comparisons in the *unsupervised* learning results. The best and second best results are highlighted with **bold** and <u>underline</u>. The metric is accuracy (%). * and ** denote the improvement over the second-best baseline is statistically significant at levels 0.1 and 0.05, respectively. Baseline results are taken from (You et al. 2020; Lin, Chen, and Wang 2022).

Dataset	NCI1	PROTEINS	DD	MUTAG	REDD-B	REDD-M5	IMDB-B
sub2vec	52.84±1.47	53.03±5.55	-	61.05±15.80	71.48±0.41	36.68±0.42	55.26±1.54
graph2vec	73.22±1.81	73.30±2.05	-	83.15±9.25	75.78±1.03	47.86±0.26	71.10±0.54
InfoGraph	76.20±1.06	74.44±0.31	75.23±0.39	89.01±1.13	82.50±1.42	53.46±1.03	73.03±0.87
GraphCL	77.87±0.41	74.39±0.45	<u>78.62±0.40</u>	86.80±1.34	89.53±0.84	**55.99±0.28**	71.14±0.44
MVGRL	68.68±0.42	74.02±0.32	75.20±0.55	89.24±1.31	81.20±0.69	51.87±0.65	71.84±0.78
AD-GCL	69.67±0.51	73.59±0.65	74.49±0.52	<u>89.25±1.45</u>	85.52±0.79	53.00±0.82	71.57±1.01
JOAO	72.99±0.75	71.25±0.85	66.91±1.75	85.20±1.64	78.35±1.38	45.57±2.86	71.60±0.86
GCL-SPAN	71.43±0.49	**75.78±0.41**	75.78±0.52	89.12±0.76	83.62±0.64	54.10±0.49	**73.65±0.69**
DP-Noise	**79.69±0.70**	74.60±0.43	78.59±0.23	87.63±1.98	90.90±0.32**	55.54±0.15	71.42±0.41
DP-Mask	<u>79.47±0.22</u>	<u>74.70±0.29</u>	**79.97±1.09****	**89.98±1.36**	**91.21±0.24****	<u>55.92±0.49</u>	71.78±0.37

model stability.

Semi-supervised learning

Performance. We then evaluate our methods in a semi-supervised setting comparing with five baselines, including training from scratch without and with augmentations (denoted as Vanilla and Aug.), GAE (Kipf and Welling 2016b), Informax (Veličković et al. 2018) and GraphCL(You et al. 2020). Tables 2 and 3 provide the performance comparison when utilizing 1% and 10% label ratios. At the more challenging 1% label ratio, DP-Noise achieves SOTA results across all three datasets, and DP-Mask secures the second-best performance on two out of the three datasets. As the label ratio increases to 10%, DP-Noise maintains its efficacy, showing SOTA performance on six out of seven datasets.

Augmentation Pairing Efficiency. To investigate optimal combinations that could potentially enhance performance, we evaluate the synergistic effects on accuracy gain (%) when pairing DP-Noise and DP-Mask with different augmentation methods using the same setting in (You et al. 2020). From Figures 6c and 6d, overall, the diverse accuracy gains across datasets indicate that there is no one-size-fits-all "partner" for DP-Noise and DP-Mask; the efficacy of each combination varies depending on the dataset. However, generally, both DP-Noise and DP-Mask exhibit enhanced performance when paired with dropN at a large degree.

Unsupervised representation learning

We next evaluate DP methods in unsupervised learning and compare them with eight baselines, including three representation learning methods (sub2vec(Adhikari et al. 2018), graph2vec(Narayanan et al. 2017) and InfoGraph(Sun et al. 2019)) and five GCL-based methods (GraphCL(You et al. 2020), MVGRL(Hassani and Khasahmadi 2020), AD-GCL(Suresh et al. 2021), JOAO(You et al. 2021) and GCL-SPAN(Lin, Chen, and Wang 2022)). Table 4 shows the per-

formance of our methods in an unsupervised setting. From the results, DP methods surpass other baselines on five out of seven datasets. Notably, compared with another spectral-based method GCL-SPAN (Lin, Chen, and Wang 2022), our methods outperform it on most datasets, especially on the molecules dataset NCI1 (an increase of around 11.5% accuracy). This superiority can be attributed to the fact that while GCL-SPAN is also spectral-based, it operates on graph modification in the spatial domain rather than directly manipulating the spectral domain like DP methods. Given the critical role that structural patterns in molecular data play on classification tasks, the enhanced performance underscores the efficiency of our direct spectral modifications in creating more effective and insightful augmented graphs.

Transfer learning

We conduct transfer learning experiments on molecular property prediction in the manner of (Hu et al. 2020b). Specifically, we initially pre-train models on the extensive chemical molecule dataset ZINC (Sterling and Irwin 2015), then fine-tune the models on eight distinct datasets within a similar domain. Due to space constraints, the table containing performance comparisons and analysis is provided in the Appendix.

Conclusion

In this study, we adopt a spectral perspective, bridging graph properties and spectral insights for property-retentive and globally-aware graph data augmentation. Stemming from this point, we propose a novel augmentation method DP, including DP-Noise and DP-Mask. By focusing on different frequency components in the spectrum, our method skillfully preserves graph properties while ensuring diversity in augmented graphs. Our extensive experiments validate the efficacy of our methods across various learning paradigms on the graph classification task.

Acknowledgement

References

Adhikari, B.; Zhang, Y.; Ramakrishnan, N.; and Prakash, B. A. 2018. Sub2vec: Feature learning for subgraphs. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part II 22*, 170–182. Springer.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3950–3957.

Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33: 17766–17778.

Chang, H.; Rong, Y.; Xu, T.; Bian, Y.; Zhou, S.; Wang, X.; Huang, J.; and Zhu, W. 2021. Not all low-pass filters are robust in graph convolutional networks. *Advances in Neural Information Processing Systems*, 34: 25058–25071.

Chen, J.; Lei, R.; and Wei, Z. 2024. PolyGCL: GRAPH CONTRASTIVE LEARNING via Learnable Spectral Polynomial Filters. In *The Twelfth International Conference on Learning Representations*.

Childs, L.; Nikoloski, Z.; May, P.; and Walther, D. 2009. Identification and classification of ncRNA molecules using graph properties. *Nucleic acids research*, 37(9): e66–e66.

Chung, F. R. 1997. *Spectral graph theory*, volume 92. American Mathematical Soc.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Ding, K.; Xu, Z.; Tong, H.; and Liu, H. 2022. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter*, 24(2): 61–77.

Dong, X.; Thanou, D.; Toni, L.; Bronstein, M.; and Frossard, P. 2020. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal processing magazine*, 37(6): 117–127.

Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020. 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*, 169–177.

Feng, W.; Zhang, J.; Dong, Y.; Han, Y.; Luan, H.; Xu, Q.; Yang, Q.; Kharlamov, E.; and Tang, J. 2020. Graph random neural networks for semi-supervised learning on graphs. *Advances in neural information processing systems*, 33: 22092–22103.

Feragen, A.; Kasenburg, N.; Petersen, J.; de Bruijne, M.; and Borgwardt, K. 2013. Scalable kernels for graphs with continuous attributes. *Advances in neural information processing systems*, 26.

Guo, H.; and Mao, Y. 2021. Intrusion-free graph mixup.

Hammond, D. K.; Gur, Y.; and Johnson, C. R. 2013. Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel. In *2013 IEEE global conference on signal and information processing*, 419–422. IEEE.

Hammond, D. K.; Vandergheynst, P.; and Gribonval, R. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2): 129–150.

Han, X.; Jiang, Z.; Liu, N.; and Hu, X. 2022. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, 8230–8248. PMLR.

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

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020a. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.

Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; and Leskovec, J. 2020b. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*.

Jin, G.; Liang, Y.; Fang, Y.; Huang, J.; Zhang, J.; and Zheng, Y. 2023a. Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. *arXiv preprint arXiv:2303.14483*.

Jin, M.; Shi, G.; Li, Y.-F.; Wen, Q.; Xiong, B.; Zhou, T.; and Pan, S. 2023b. How Expressive are Spectral-Temporal Graph Neural Networks for Time Series Forecasting? *arXiv preprint arXiv:2305.06587*.

Kipf, T. N.; and Welling, M. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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

Lee, J. R.; Gharan, S. O.; and Trevisan, L. 2014. Multiway spectral partitioning and higher-order cheeger inequalities. *Journal of the ACM (JACM)*, 61(6): 1–30.

Lin, L.; Chen, J.; and Wang, H. 2022. Spectral Augmentation for Self-Supervised Learning on Graphs. In *The Eleventh International Conference on Learning Representations*.

Ling, H.; Jiang, Z.; Liu, M.; Ji, S.; and Zou, N. 2023. Graph Mixup with Soft Alignments. *arXiv preprint arXiv:2306.06788*.

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

Morris, C.; Kriege, N. M.; Bause, F.; Kersting, K.; Mutzel, P.; and Neumann, M. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*.

Narayanan, A.; Chandramohan, M.; Venkatesan, R.; Chen, L.; Liu, Y.; and Jaiswal, S. 2017. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*.

Newey, W. K.; and West, K. D. 1987. Hypothesis testing with efficient method of moments estimation. *International Economic Review*, 777–787.

Newman, M. E.; Watts, D. J.; and Strogatz, S. H. 2002. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl_1): 2566–2572.

Nt, H.; and Maehara, T. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.

Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J. M.; and Vandergheynst, P. 2018. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5): 808–828.

Park, J.; Shim, H.; and Yang, E. 2022. Graph transplant: Node saliency-guided graph mixup with local structure preservation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7966–7974.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2019. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations*.

Sterling, T.; and Irwin, J. J. 2015. ZINC 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11): 2324–2337.

Sun, F.-Y.; Hoffman, J.; Verma, V.; and Tang, J. 2019. InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization. In *International Conference on Learning Representations*.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 15920–15933.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2018. Deep Graph Infomax. In *International Conference on Learning Representations*.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, 6438–6447. PMLR.

Wang, H.; Zhang, J.; Zhu, Q.; and Huang, W. 2022. Augmentation-free graph contrastive learning with performance guarantee. *arXiv preprint arXiv:2204.04874*.

Wang, Y.; Wang, W.; Liang, Y.; Cai, Y.; and Hooi, B. 2021. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, 3663–3674.

Wills, P.; and Meyer, F. G. 2020. Metrics for graph comparison: a practitioner's guide. *Plos one*, 15(2): e0228728.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, 6861–6871. PMLR.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yang, K.; Han, H.; Jin, W.; and Liu, H. 2023. Augment with Care: Enhancing Graph Contrastive Learning with Selective Spectrum Perturbation. *arXiv preprint arXiv:2310.13845*.

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

Yi, H.-C.; You, Z.-H.; Huang, D.-S.; and Kwoh, C. K. 2022. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1): bbab340.

Yoo, J.; Shim, S.; and Kang, U. 2022. Model-agnostic augmentation for accurate graph classification. In *Proceedings of the ACM Web Conference 2022*, 1281–1291.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph contrastive learning automated. In *International Conference on Machine Learning*, 12121–12132. PMLR.

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

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

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