S3GCL: Spectral, Swift, Spatial Graph Contrastive Learning

Guancheng Wan [1] **Yijun Tian** [2] **Wenke Huang** [1] **Nitesh V Chawla** [2] **Mang Ye** [1 3]

Abstract

Graph Contrastive Learning (GCL) has emerged as a highly effective self-supervised approach in graph representation learning. However, prevailing GCL methods confront two primary challenges: 1) They predominantly operate under homophily assumptions, focusing on low-frequency signals in node features while neglecting heterophilic edges that connect nodes with dissimilar features. 2) Their reliance on neighborhood aggregation for inference leads to scalability challenges and hinders deployment in real-time applications. In this paper, we introduce S3GCL, an innovative framework designed to tackle these challenges. Inspired by spectral GNNs, we initially demonstrate the correlation between frequency and homophily levels. Then, we propose a novel cosine-parameterized Chebyshev polynomial as low/high-pass filters to generate biased graph views. To resolve the inference dilemma, we incorporate an MLP encoder and enhance its awareness of graph context by introducing structurally and semantically neighboring nodes as positive pairs in the spatial domain. Finally, we formulate a cross-pass GCL objective between full-pass MLP and biased-pass GNN filtered features, eliminating the need for augmentation. Extensive experiments on real-world tasks validate S3GCL proficiency in generalization to diverse homophily levels and its superior inference efficiency.

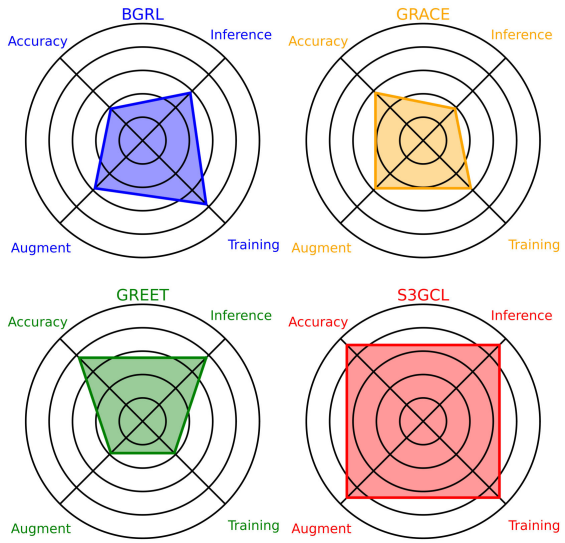


Figure 1. **Performance and efficiency comparison of different methods:** (Testing) Accuracy, (Time of) Inference, (Time of) Training, and (Time of) Augmentation in contrastive learning. Results are generated on homophilic and heterophilic datasets. For all metrics except Accuracy, we employ **reverse** ranking along each axis, where a larger polygonal area indicates superior performance. Our method demonstrates a superior balance between effectiveness and efficiency. On large-scale datasets like Obgn-Arxiv, ours accelerates inference by $\times 174$ while maintaining competitive performance. Further details are shown in Tab. 2 and Sec. 4.3.

1. Introduction

Graph Neural Networks (GNNs) (Hamilton et al., 2017; Kipf & Welling, 2017) play a crucial role in analyzing graph-structured data (Xiong et al., 2024b;a). They excel in identifying complex interactions and distinct node characteristics through message-passing mechanisms (Wu et al., 2020; Dai et al., 2022). Traditionally, GNN research has focused on supervised training, which requires a lot of labeled data. However, data labeling is costly and time-consuming. To overcome this, Graph Contrastive Learning (GCL) has emerged as a self-supervised technique in graph representation learning (Zhu et al., 2021b; Xia et al., 2021), particularly useful when task-specific labels are scarce. GCL works by maximizing agreement between different augmented views of the same example and minimizing it between differently augmented views of separate examples. Its efficacy has been validated in diverse fields, including social network analysis and recommender systems (Xu et al., 2021; Cai et al., 2023; Liu et al., 2023c; Tian et al., 2023a).

Despite the recent developments, two key problems of GCL

[1]National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan, China [2]Department of Computer Science, University of Notre Dame, USA [3]Taikang Center for Life and Medical Sciences, Wuhan University, Wuhan, China. Correspondence to: Mang Ye <yemang@whu.edu.cn>.

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

remain unsolved. Firstly, most existing methods are based on homophily assumptions (Zhu et al., 2020), primarily depending on low-pass filter GNNs (*e.g.*, GCN) to learn representations. This focus on low-frequency components implies similarity between connected nodes. While effective for homophilic graphs, this approach overlooks the prevalence of heterophilic graphs in real-world scenarios, where connected nodes have different labels or dissimilar features (Zheng et al., 2022a; He et al., 2022a). Recent efforts to address this include adaptive augmentation (Liu et al., 2023f) and parameterized sampling strategies (He et al., 2023), without giving special consideration to the graph signal filters. Therefore the encoder inevitably encourages neighbor features to be pooled, irrespective of their actual similarity. This approach limits the ability to generate distinct representations and to generalize across graphs with varying homophily levels. Thus, the following question naturally emerges: **I)** *how can we design filters generalizable to graphs of different homophily levels without labels?*

Secondly, although GNNs exhibit a remarkable ability to capture graph-structured contexts, deploying them in large-scale applications presents challenges due to their time-intensive message-passing mechanisms. However, it is crucial to make efficient inferences for latency-sensitive applications. To tackle this problem, many studies have investigated the distillation of knowledge from a pre-trained GNN teacher to a student MLP, then deploying the MLP for inference acceleration (Tian et al., 2023b; Zhang et al., 2022). However, these methods can only be adopted in the context of supervised scenarios, which require task-specific labels to train an effective teacher GNN, limiting their adaptability in scenarios lacking labels. This raises another intriguing question: **II)** *how can we combine the inference-efficient characteristics of MLP and graph context-awareness of GNN in a self-supervised way?*

To address the aforementioned questions, we revisit existing graph contrastive learning from both spectral and spatial perspectives, introducing our **S**pectral, **S**wift, **S**patial **G**raph **C**ontrastive **L**earning (S3GCL). To go beyond the homophily assumption and address **I)**, we first theoretically and empirically analyze the homophily in the spectral domain. We discover that high homophily levels correspond to low-frequency signals, whereas low homophily correlates with high-frequency signals. In supervised tasks, spectral GNNs with polynomial approximations have demonstrated their superiority in both homophilic and heterophilic graphs. Chebyshev polynomial (Defferrard et al., 2016; He et al., 2022b) among them are widely used to approximate various functions in graph signal filtering. However, these require labels to learn the filter shape suited to the graph, posing challenges in label-scarce environments. To overcome this, we propose a cosine-parameterized Chebyshev polynomial approach. This method utilizes cosine modulation for calculating filter values in Chebyshev interpolation, emphasizing the graph high-frequency and low-frequency components respectively. The cosine parameterization method selectively emphasizes certain frequency ranges, effectively decomposing graphs into two distinct biased-pass views. Moreover, this strategy simplifies the filter learning process by adopting less learnable parameters and enhances adaptability to graphs with diverse homophily levels.

In the second place, to address the inefficient inference problem mentioned in **II)**, we propose utilizing an MLP encoder for representation learning. Since the MLP functions as the feature transformation without any spectral filter, it is capable of creating a comprehensive full-pass view of graphs. Building on this, we formulate our *Cross-Pass* GCL objective, contrasting the full-pass MLP with a biased-pass GNN. To further enhance the graph context awareness of the MLP encoder, we divide the cross-pass GCL objective into two parts and introduce positive pairs in the spatial domain. By incorporating structurally neighboring nodes in the full-low component and semantically neighboring nodes in the full-high component, the MLP is enabled to identify finer granularity features from the graph property, which might be overlooked by relying solely on global spectral filter methods. Consequently, with the help of such a cross-pass and cross-architecture design and adopted spatial positive pairs, our method conducts efficient training of GCL without any manually specified graph augmentation. In practical applications, deploying the trained MLP encoder leads to faster inference and superior performance, generalizable to various graph types. Our principal contributions are summarized as follows.

- We identify that existing GCL methods often overlook generalization across varying graphs and encounter challenges in efficient inference within applications.
- We uncover the correlation between homophily and frequency in the spectral domain and introduce biased cosine-parameterized Chebyshev polynomials to facilitate generalization across various graph types.
- We utilize the MLP encoder and incorporate positive pairs in the spatial domain. By integrating this with our innovative cross-pass optimization objective, swift and efficient inference for obtaining representations is achieved.
- Extensive experiments demonstrate that our method not only achieves higher inference speed but also enhances performance, making it adaptable to different graphs.

2. Motivation

2.1. Preliminaries

Notations. Define the graph data as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ signifying the node set encompassing $|\mathcal{V}| = N$ nodes. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ symbolizes the connections between nodes. The feature matrix $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}^\top$ comprises feature vectors $\mathbf{x}_i$ corresponding to node v_i. The

adjacency matrix of $\mathcal{G}$, denoted as $\mathbf{A} \in \mathbb{R}^{n\times n}$, assigns $\mathbf{A}_{ij} = 1$ for existing edges $e_{i,j} \in \mathcal{E}$ and $\mathbf{A}_{ij} = 0$ otherwise. The normalized adjacency matrix is expressed as $\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. The degree matrix $\mathbf{D}$, a diagonal matrix, is defined with $D_{i,i} = \sum_j A_{i,j}$. The graph Laplacian matrix is given by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and the symmetric normalized Laplacian matrix is $\tilde{\mathbf{L}} = \mathbf{I} - \hat{\mathbf{A}}$, where $\mathbf{I}$ is the identity matrix. $\tilde{\mathbf{L}}$ can be decomposed as $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{\Lambda} = \text{diag}(\lambda_0, \ldots, \lambda_{N-1})$ is a diagonal eigenvalue matrix with $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1} \leq 2$, and $\mathbf{U}$ is a unitary matrix consisting of eigenvectors.

Graph Filtering. Graph filtering on features $\mathbf{X}$ is defined by $\mathbf{Z} = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{X}$, where $g(\mathbf{\Lambda})$ represents the graph filter. Directly learning $g(\mathbf{\Lambda})$ necessitates eigendecomposition (EVD), which has a time complexity of $O(N^3)$. Recent studies (Lei et al., 2022; Wang & Zhang, 2022) advocate for approximating $g(\mathbf{\Lambda})$ using polynomials:

$$\mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{X} \approx \mathbf{U}\left(\sum_{i=0}^{K-1} w_k\lambda^k\right)\mathbf{U}^\top\mathbf{X} = \sum_{i=0}^{K-1} w_k\tilde{\mathbf{\Lambda}}^k\mathbf{X}. \quad (1)$$

The polynomial coefficients are denoted by $\{w_k\}$. Furthermore, a K-order polynomial graph filter can be represented as the filter function $g(\lambda) = \sum_{k=0}^{K} w_k\lambda^k$, mapping each eigenvalue λ in the range $[0, 2]$ to $g(\lambda)$.

Definition 2.1. (Homophily Level h): *Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with its node label vector $\mathbf{y}$. The edge homophily level is quantified as the proportion of edges connecting nodes sharing identical labels. Formally, it is expressed as:*

$$h(\mathcal{G}, \{\mathbf{y}_i : i \in \mathcal{V}\}) = \frac{1}{|\mathcal{E}|}\sum_{(k,l)\in\mathcal{E}} \mathbf{1}(y_k = y_l). \quad (2)$$

Here, $|\mathcal{E}|$ denotes the total number of edges in the graph, and $\mathbf{1}(\cdot)$ represents the indicator function.

Problem Description. Our objective is to advance the field of self-supervised learning for node representation. We aim to develop an encoder $\mathcal{F}$ that generates high-level node representations $\mathbf{Z} = \{z_1, \ldots, z_N\} \in \mathbb{R}^{n\times d}$, where each $z_i \in \mathbb{R}^d$ corresponds to a node v_i. These representations can be utilized in downstream tasks like node classification.

2.2. Motivation

In this subsection, we examine both theoretically and empirically the relationship between graph filters that emphasize different frequencies, and their performance in homophilic or heterophilic graphs. This study contributes to the development of generalizable graph filters for GCL. We employ the widely-used Contextual Stochastic Block Model (CSBM) to generate graphs with varying levels of homophily (refer to Appendix B for a comprehensive description of CSBM).

Initially, we theoretically analyze the performance of spectral GNNs with various filters on different graph types. In addition to homophily levels for categorical node labels, we assess the similarity of numerical node signals in the spectral domain, as described by (Huang & Liò, 2023). This assessment is conducted using the following metric:

Definition 2.2. (Spectral Signal Frequency $\mathbf{f}$): *For a normalized feature signal $x \in \mathbb{R}^n$, the spectral signal frequency $\mathbf{f}(x)$ on graph $\mathcal{G}$ is $\mathbf{f}(x) = \frac{x^\top\mathbf{L}x}{2}$.*

Spectral signal frequency, as related to the Dirichlet Energy (Karhadkar et al., 2022), measures the variance of signal x in the spectral domain across graph $\mathcal{G}$:

Lemma 2.3. *A lower $\mathbf{f}(x)$ suggests reduced distances between connected nodes, indicating smoothness over $\mathcal{G}$, where x_u represents the u-th element of x:*

$$\mathbf{f}(x) = \frac{x^\top\mathbf{L}x}{2} = \sum_{(u,v)\in\mathcal{E}} \frac{(x_u - x_v)^2}{2}. \quad (3)$$

Lemma 2.4. *For a given graph signal x, $\Delta D(x)$ denotes the disparity between intra-class and inter-class distances:*

$$\begin{aligned}\Delta D(x) &= \mathbb{E}\Big[\sum_{\substack{(u,v)\in\mathcal{E}\\ y_u = y_v}} (x_u - x_v)^2 - \sum_{\substack{(u,v)\in\mathcal{E}\\ y_u \neq y_v}} (x_u - x_v)^2\Big] \\ &= 2\mathbb{E}\Big[(p_{intra} - p_{inter})\mathbf{f}(x)\Big]. \end{aligned} \quad (4)$$

A lower $\Delta D(x)$ indicates that the encoder generates node representations that are more similar within the same class and distinct across different classes, demonstating better self-supervised learning performance.

Given graph data $\mathcal{G} \sim \text{CSBM}(\mu_1, \mu_2, p, q)$ generated by the CSBM model, comprising two classes, c_0 and c_1, with intra-class probability p and inter-class probability q for edge formation, we develop a theorem to ascertain the advantageous filter for given graph data:

Theorem 2.5. *Consider graph signals x^l and x^h processed by filters g_l and g_h, respectively. In heterophilic graphs (where $p < q$) and when $\Delta D(x^h) < \Delta D(x^l)$, there exists an integer $M (0 < M \leq N-1)$ such that $\sum_{i=M}^{N-1} g_h^2(\lambda_i) \geq \sum_{i=M}^{N-1} g_l^2(\lambda_i)$, with $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1} \leq 2$.*

The complete proof of this theorem is available in Appendix D. This theorem indicates that in heterophilic graphs, a high-pass filter, which amplifies high frequencies, is more effective for distinct node representations, whereas a low-pass filter is preferable for homophilic graphs. To empirically validate this, we further conduct following experiments: we used ChebNet (Defferrard et al., 2016), known for its ability to approximate various spectral filters in supervised learning. ChebNet redefines the filter in Equation (1) as $\sum_{k=0}^{K} w_k T_k(\tilde{\mathbf{L}})\mathbf{X}$, where $\hat{\mathbf{L}} = 2\mathbf{L}/\lambda_{max} - \mathbf{I}$. The function $T_k(x)$ follows the relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. Expanding on this, (He et al., 2022b) enhanced Chebyshev polynomials

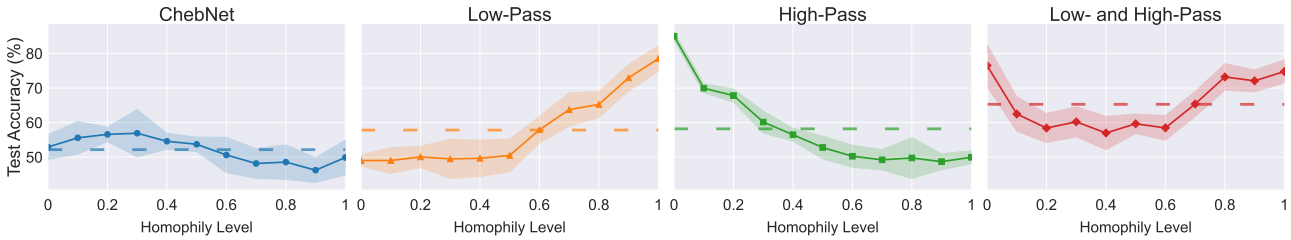


Figure 2. **Case Studies** on graphs with varying homophily levels. The dashed lines represent the mean performance across all homophily levels. We reveal that **adopting both low-pass and high-pass** will bring promising generalization across different homophily levels.

with Chebyshev interpolation, reparameterizing w_k as

$$w_k = \frac{2}{K+1} \sum_{j=0}^{K} \gamma_j T_k(x_j). \tag{5}$$

We generated synthetic graphs with homophily levels ranging from 0 to 1 and applied the GRACE (Zhu et al., 2020) contrastive learning framework, evaluating performance through a node classification task. Due to challenges in learning filter parameters without labels, we initialized $\{\gamma_0^h, \ldots, \gamma_K^h\}$ in Equation (5) as an increasing arithmetic sequence, holding them constant during training as w_k^h for a high-pass filter. Conversely, we set w_k^l for the low-pass filter using a decreasing arithmetic sequence. Specifically, we set γ_0^h and γ_K^h at 0 and 2 for the high-pass filter, and inversely for the low-pass filter. Our experiments involved four filtering methods: ChebNet (with interpolation), low-pass only, high-pass only, and a combination of both. The results are in Figure 2. Our key findings are: 1) Using a biased-pass filter alone can limit generalization across various homophily levels, 2) Spectral GNNs struggle to approximate optimal filters without labels, 3) Employing both low- and high-pass filters separately enhances generalizability. These insights guide the development of our proposed method.

3. Methodology

3.1. Spectral: Cosine-Parameterized Chebyshev Polynomial Encoder

To better handle graphs with different homophily levels and without labels (as discussed in Sec. 2.2), we initially consider using both low- and high-pass filters on the input graph data. However, this method encounters two issues: 1) In the parameterization method, a linear increase in parameters proportional to the order K occurs. This results in uniform growth across all intervals, not emphasizing any specific frequency range. It potentially causes the two graph views to remain coupled and not disentangled. 2) The filter remains static during training and cannot be optimized alongside the contrastive objective. To address these issues, we propose a novel cosine-parameterized Chebyshev polynomial. This approach derives the high-pass filter parameter w_k^h with γ_j^h substituting γ_j in Equation (5). γ_j^h can be calculated as:

$$\gamma_j^h = \sigma(\beta_a^h) + \frac{1}{2}\sigma(\beta_b^h)(1 + \cos((1 + j/K)\pi)). \tag{6}$$

$\sigma = \mathrm{ReLu}(\cdot)$ to ensure the non-negative property of γ_j and guarantee that $\gamma_j^h \leq \gamma_{j+1}^h$. Similarly, we obtain the low-pass filter parameter w_k^l with $\gamma_j^l \geq \gamma_{j+1}^l$:

$$\gamma_j^l = \sigma(\beta_a^l) - \frac{1}{2}\sigma(\beta_b^l)(1 + \cos((1 + j/K)\pi)). \tag{7}$$

We initialize β_a^h and β_a^l as 0 and 2, respectively, and set β_b^h and β_b^l to 2. It is important to note that β is trainable during the contrastive learning process. This approach simplifies learning Chebyshev polynomials without labels by utilizing only two learnable parameters for each filter. Moreover, the cosine-parameterized strategy effectively emphasizes relevant frequencies while diminishing less significant ones, facilitating a smoother frequency distribution. In contrast, linear-increasing parameterization might result in an abrupt distribution, potentially less effective in capturing graph spectral properties. The effectiveness of this method will be further validated in Sec. 4.4. By applying distinct filters with w_k^h and w_k^l, we can decompose the graph into two biased views, each yielding its representation:

$$\mathbf{Z}^h = \sum_{k=0}^{K} w_k^h T_k(\tilde{\mathbf{L}}) f_\theta^h(\mathbf{X}), \quad \mathbf{Z}^l = \sum_{k=0}^{K} w_k^l T_k(\tilde{\mathbf{L}}) f_\theta^l(\mathbf{X}). \tag{8}$$

The notation $f_\theta(\mathbf{X})$ represents the application of the MLP on the node feature matrix $\mathbf{X}$. After processing the features through filters and transformations, we can use these enhanced representations for contrastive learning objectives.

3.2. Swift: MLP Encoder and Cross-Pass Objective

Previous methods highlight the challenges of GNNs in terms of inference efficiency and scalability, primarily due to data dependency (Jia et al., 2020). In GCL, generating node representations typically requires message passing to aggregate neighborhood features. However, this aggregation process can be time-consuming, especially in latency-sensitive applications. To mitigate this issue, some research has focused on distillation methods, transferring knowledge from teacher GNNs to student MLPs for more efficient inference in industrial settings. Yet, these methods rely on supervised signals to train a high-quality teacher GNN and guide the student

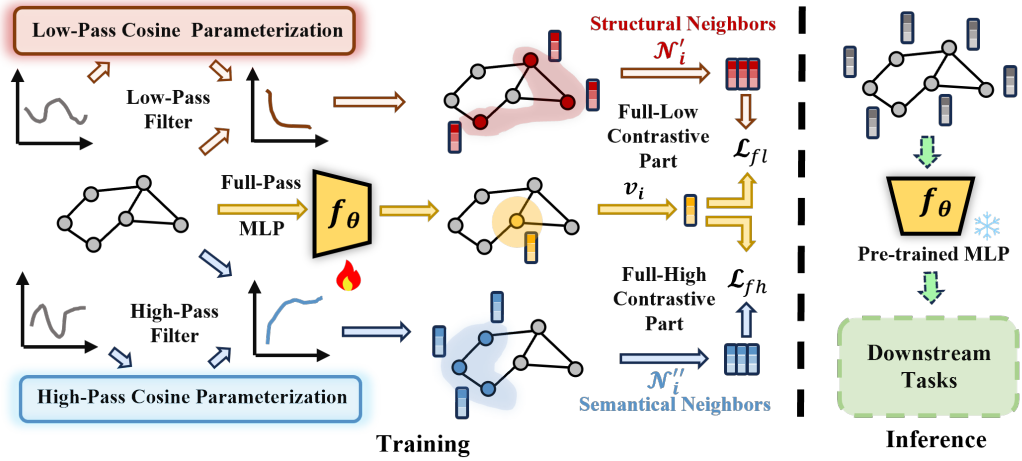


Figure 3. Architecture illustration of S3GCL: Spectral, Swift, Spatial Graph Contrastive Learning. We show the training process (left) and inference process (right) of S3GCL. Best viewed in color. Zoom in for details.

MLP, which is impractical in unsupervised scenarios.

To bridge this gap, we introduce an MLP encoder throughout the entire GCL training process, enabling its direct deployment for efficient inference. Initially, we input the original node feature matrix $\mathbf{X}$ into an L-layer MLP encoder $\mathcal{F}_\theta$, yielding representations $\mathbf{Z}^f = \mathcal{F}_\theta(\mathbf{X})$. Furthermore, the MLP can be regarded as a full-pass filter, focusing solely on feature transformation (Luan et al., 2022). Consequently, this allows us to establish an optimization objective between the full-pass and biased-pass filtered representations. Drawing inspiration from the Info-NCE loss (Chen et al., 2020), we formulate our *Cross-Pass* objective as:

$$\mathcal{L}_{cp} = \frac{-1}{2|\mathcal{V}|}\sum_{v_i\in\mathcal{V}}\left(\log\frac{\mathrm{s}\left(z_p^f, z_p^l\right)}{\sum_{p\neq q}\mathrm{s}\left(z_p^f, z_q^l\right)} + \log\frac{\mathrm{s}\left(z_p^f, z_p^h\right)}{\sum_{p\neq q}\mathrm{s}\left(z_p^f, z_q^h\right)}\right). \tag{9}$$

Here the $\mathrm{s}(z_p^f, z_p^h) = \exp(\omega(z_p^f, z_p^h)/\tau)$, ω is the cosine similarity defined as: $\omega(z_p^f, z_p^h) = z_p^f \cdot z_p^h/(||z_p^f|| \times ||z_p^h||)$. The parameter τ denotes the contrast temperature. This method offers guidance for node representation learning by encouraging the model to produce semantically consistent representations from two distinct graph views. Moreover, from a spectral perspective, this cross-pass process ensures that the MLP encoder captures more invariant information from the task-relevant properties of graph spectral signals. The MLP optimization with different graph frequency ranges also improves the generalizability of graphs.

Upon optimizing the cross-pass GCL objective, we obtain a refined MLP encoder capable of generating high-level and expressive node representations. However, MLPs inherently lack the ability to capture graph structural properties and context. Therefore in our initial studies, we observed that representations learned by MLPs tend to deteriorate and become suboptimal. Thus, enhancing the graph context awareness of MLPs in these scenarios is imperative.

3.3. Spatial: Neighboring Positive Pairs

To address the issue of the MLP encoder limited awareness of graph context, we introduce our concept of positive pairs in the spatial domain. Prior research in spatial GNNs has underscored the significance of the spatial domain in graph learning (Wang et al., 2021; 2022), highlighting its role in enhancing model adaptability to graph structure and effectively managing long-range dependencies and heterogeneity. In our study, knowledge of the spatial domain provides contextual insights that enable the MLP encoder to more effectively comprehend the underlying structure and semantics of the graph. Because the low-pass filter emphasizes neighborhood similarity in the spatial domain, we initially identify structurally neighboring nodes as positive pairs and separate the Cross-Pass objective as delineated in Equation (9) into the Full-Low part, formulated as follows:

$$\mathcal{L}_{fl} = -\frac{1}{2|\mathcal{V}|}\sum_{v_i\in\mathcal{V}}\frac{1}{|\mathcal{N}_i'|}\sum_{v_p\in\mathcal{N}_i'}\log\frac{\mathrm{s}\left(z_i^f, z_p^l\right)}{\sum_{v_q\in\mathcal{V}\setminus v_i}\mathrm{s}\left(z_i^f, z_q^l\right)}. \tag{10}$$

$\mathcal{N}_i'$ denotes the positive sample set of local neighbors for node v_i in node set $\mathcal{V}$. It is important to note that treating structurally neighboring nodes as positive pairs does not necessarily lead to identical representations. As suggested in (Altenburger & Ugander, 2018; Xiao et al., 2023), the phenomenon known as *Monophily* has been observed in both homophilic and heterophilic graphs. It implies that the attributes of a node's friends are likely to resemble those of the node's other friends, indicating two-hop similarities. For instance, nodes v_i and v_k share a common neighbor v_j. The representations z_i^f and z_k^f are derived from the MLP, respectively. Equation (10) encourage these representations to

capture their shared one-hop neighborhood structure pattern, coming from spectral GNN as z_j^l. This *cross-architecture* aligns these two-hop pairs implicitly and extends beyond the homophily assumption. We show one-hop and two-hop representations similarities in Figure 6.

Additionally, the high-pass filter indicates local neighborhood dissimilarity and global long-range similarity within the spatial domain. Therefore, we introduce semantically neighboring nodes as positive pairs, determined by feature-level similarity. Specifically, for each node v_i, we identify its top-k similar nodes based on original features, denoted as $\mathcal{N}_i'' = kNN(v_i, k)$. This leads to the formulation of the Full-High part of our objective, as defined in Equation (9):

$$\mathcal{L}_{fh} = -\frac{1}{2|\mathcal{V}|}\sum_{v_i\in\mathcal{V}}\frac{1}{|\mathcal{N}_i''|}\sum_{v_p\in\mathcal{N}_i''}\log\frac{\mathrm{s}\left(z_i^f, z_p^h\right)}{\sum_{v_q\in\mathcal{V}\setminus v_i}\mathrm{s}\left(z_i^f, z_q^h\right)}. \tag{11}$$

By integrating spatial relationships, the MLP encoder gains enhanced awareness of the graph context. This integration enables it to learn finer granularity features, which may be overlooked by solely relying on global spectral filter methods. Utilizing both components of the loss, we can establish our optimization objective as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{fl} + (1-\alpha)\mathcal{L}_{fh}, \tag{12}$$

where the coefficient α is employed to balance these two loss functions. The deployed MLP encoder is advantageous for inference efficiency, particularly in managing large-scale graphs or real-time applications. The complexity and scalability analysis can be found in Appendix E.

3.4. Discussion and Comprasion

Existing GCL methods, such as GRACE (Zhu et al., 2020) and GBT (Bielak et al., 2021), typically employ a shared primary encoder with the *same architecture*. These methods often apply topology or feature transformations to augment the graph data. In contrast, our S3GCL introduces an innovative *cross architecture* approach. Coupled with our unique spatial positive pairs, S3GCL is **augmentation-free** while still achieving competitive performance. PolyGCL (Chen et al., 2024) also integrates the learnable spectral GNN in GCL, however, they still struggle with the efficient inference issue. Additionally, current research in knowledge distillation, such as FF-G2M (Wu et al., 2023a), employs structurally neighboring samples similar to our approach. However, these studies still rely on labels to guide the student MLP encoder. In contrast, our method attains consistency between the MLP target encoder and the spectral GNN encoder through our unique cross-pass objective at the representation level.

Proposition 3.1. *Given a node representation z_i and its neighboring node representation z_j, optimizing Equation (12) leads to maximizing their mutual information $I(z_j; z_i)$, while inherently minimizing $D_{KL}(z_j||z_i)$:*

$$I(z_j; z_i) \sim \frac{1}{D_{KL}(z_j||z_i)} \tag{13}$$

The proof is detailed in Appendix D. The proposition implies that information gain represented by D_{KL} is negatively correlated to the mutual information. By optimizing the cross-pass objective, we maximize mutual information between structurally or semantically neighboring nodes. This strategy not only facilitates the transfer of knowledge from GNN to MLP without labels, but also enriches the MLP. The trained MLP can be leveraged for more efficient inference (Ding et al., 2021) or saving communication cost (Huang et al., 2023c;b).

One potential limitation of the proposed method is that, in the training phase, we still need polynomial filters to get representations. Although adopting MLP encoder can reduce the training time cost to a certain extent, its advantages can only be fully utilized in the inference phase. We believe this is also one of the directions in this field for the future.

4. Experiment

In this section, we comprehensively evaluate our proposed S3GCL by answering the main questions as follows.

- **Q1: Superiority.** Does S3GCL outperforms the existing state-of-the-art graph contrastive learning methods?
- **Q2: Efficiency.** How about the inference time efficiency of the proposed method?
- **Q3: Effectiveness.** Are proposed cosine-parameterized Chebyshev polynomial, MLP encoder, and spatial positive pairs effective?
- **Q4: Sensitivity.** What is the performance of the proposed method with different hyper-parameters?

The answers of **Q1-Q3** are illustrated in 4.2-4.4, and sensitivity analyses (**Q4**) can be found in the Appendix G. The code is available at https://github.com/GuanchengWan/S3GCL.

4.1. Experimental Setup

4.1.1. Real-world Datasets

To effectively evaluate our approach in practical scenarios, we employed 14 benchmark graph datasets of various sizes and features, including both homophilic and heterophilic graphs. Please see Appendix A for details about datasets.

4.1.2. Evaluation Protocol

To evaluate the proposed method, we follow (Velickovic et al., 2019) to adopt a standard linear evaluation protocol. In this approach, the representations generated by our MLP encoder are fixed and subsequently employed for training, validation, and testing using a straightforward linear classifier. We repeat this experiment five times for each dataset to

Table 1. **Comparison with the state-of-the-art methods** on homophilic (upper) and heterophilic (lower) real-world datasets. We report node classification accuracies (%) (± standard deviation) for downstream task performance. The best and second results are highlighted with **bold** and underline, respectively.

Methods	Cora	CiteSeer	PubMed	Amz-Comp	Amz-Photo	Coauthor-CS	Obgn-Arxiv	Rank
DGI	83.69±1.20	72.54±1.64	80.41±0.33	86.11±1.21	90.38±0.90	91.57±0.88	70.19±0.73	7.86
GMI	83.11±1.53	73.69±1.89	83.44±1.64	86.91±0.45	91.27±1.23	90.11±0.20	69.23±0.79	8.14
MVGRL	87.67±0.86	74.31±0.82	86.98±1.22	87.91±0.34	92.77±0.57	90.95±0.28	70.88±0.51	4.29
GRACE	86.13±0.56	71.84±1.79	82.33±0.91	81.33±1.74	92.11±1.41	91.33±0.65	70.96±0.31	7.43
GBT	84.53±0.38	73.88±1.91	79.96±1.32	83.11±0.78	92.87±0.48	92.49±0.42	70.32±0.22	6.0
BGRL	83.17±1.77	70.11±0.54	85.57±0.90	88.35±0.32	93.10±0.44	91.72±0.21	71.24±0.35	5.43
DSSL	86.23±1.06	73.99±0.90	86.31±1.34	83.11±0.73	92.78±0.91	92.03±0.30	70.13±0.25	5.43
GREET	86.78±1.11	74.56±1.82	86.38±0.87	87.79±1.18	93.24±0.78	92.33±0.65	71.09±0.43	3.14
SP-GCL	87.45±1.42	73.19±2.10	85.66±0.44	**89.45±1.95**	92.32±0.17	90.91±0.93	69.11±0.36	6.0
S3GCL	**88.47±1.39**	**76.31±1.67**	**87.89±1.23**	88.45±1.98	**94.31±0.83**	**92.55±0.89**	**71.36±0.60**	**1.14**

Methods	Cornell	Texas	Wisconsin	Actor	Roman-empire	Minesweeper	Tolokers	Rank
DGI	65.02±4.87	79.13±2.77	73.33±2.12	28.47±1.29	43.16±0.78	78.91±0.76	78.50±0.81	6.29
GMI	60.99±5.89	75.44±3.84	70.13±3.84	27.51±0.55	45.22±0.83	76.91±0.43	77.23±0.35	8.43
MVGRL	67.90±3.75	76.22±3.90	74.72±0.99	30.11±1.41	47.93±0.21	**79.78±1.27**	79.13±0.77	4.71
GRACE	50.82±4.78	75.32±2.11	74.91±2.78	29.23±0.21	51.58±0.98	78.99±1.45	78.06±0.90	7.14
GBT	54.66±3.10	72.30±2.14	65.32±1.34	23.42±0.55	38.77±0.54	77.32±0.64	**79.43±1.00**	8.14
BGRL	60.93±5.22	71.42±3.19	62.81±4.61	29.26±1.59	52.16±0.09	79.03±0.39	78.08±0.62	7.14
DSSL	76.78±3.68	75.92±2.11	81.33±3.66	33.31±0.86	61.29±0.44	78.76±0.70	78.42±0.85	4.43
GREET	75.23±4.96	78.33±4.78	83.11±3.91	36.42±0.65	63.37±1.91	79.05±1.21	79.21±0.79	2.86
SP-GCL	78.33±3.22	80.21±3.94	79.13±2.91	30.91±0.91	52.16±0.25	78.72±0.42	78.21±0.67	4.43
S3GCL	**81.27±3.67**	**86.12±3.91**	**84.56±2.71**	**36.88±0.34**	**66.27±1.33**	79.33±1.48	79.39±0.44	**1.29**

Table 2. **Statistics** during the training process with different methods in the Cora and Actor dataset. Please see details in Sec. 4.3.

Method	Cora				Actor			
	Augment ms ↓	Training s ↓	Inference ms ↓	Acc ↑	Augment ms ↓	Training s ↓	Inference ms ↓	Acc ↑
BGRL	129.1	4.32	8.33	83.17	132.8	**9.87**	9.69	29.26
GRACE	98.21	6.12	4.38	86.13	98.12	16.85	5.13	29.23
GREET	19272	43.21	1.32	86.78	54272	401.3	2.31	36.42
S3GCL	**0**	**3.59**	**0.32**	**88.47**	**0**	10.91	**0.23**	**36.88**

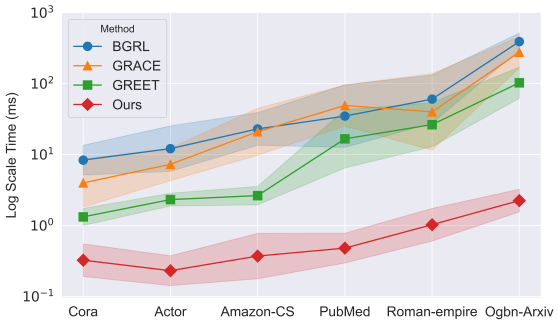


Figure 4. **Inference Time** (log scale) with datasets of different sizes. Please refer to Sec. 4.3 for details.

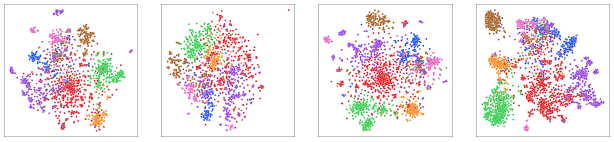
ensure the robustness and reliability of the results. Please see more details about implementation in Appendix C.

4.1.3. Compared Methods

We compare ours against several popular graph contrastive learning methods: DGI (Velickovic et al., 2019), GMI (Peng et al., 2020), MVGRL (Hassani & Khasahmadi, 2020), GRACE (Zhu et al., 2020), GBT (Bielak et al., 2021) and BGRL(Thakoor et al., 2021). We also include more recent GCL methods considering the heterophilic graphs for convincing comparison: GREET (Liu et al., 2023f), DSSL (Xiao et al., 2022) and SP-GCL (Wang et al., 2023).

4.2. Superiority

In this section, we address **Q1** by analyzing the superior performance of S3GCL. We conduct comprehensive experiments on both homophilic and heterophilic graphs, as detailed in Tab. 1. Our findings include the average test accuracy and standard deviation from five runs, along with a comparative ranking across all datasets. Three key observations emerge: 1) S3GCL exhibits competitive performance on typical homophilic graphs, such as social networks. For instance, in the Cora dataset, our method achieves a 0.8% accuracy gain over the next best approach, indicating that



(a) BGRL (b) GRACE (c) GREET (d) S3GCL

Figure 5. **t-SNE Visualization of representations with different methods** on the Cora dataset. Please see Sec. 4.2 for details.

Table 3. **Ablation Study of different variants** on four dataset.

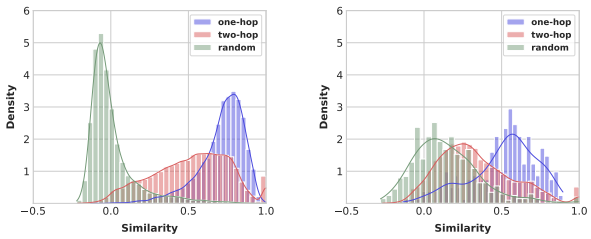
Variants	Cora	PubMed	Actor	Roman-empire
GCN-MLP	85.32±1.11	86.97±0.45	34.32±0.57	57.32±1.22
w/o Param.	84.33±0.78	83.14±0.65	35.11±0.74	58.32±1.06
Linear Param.	87.81±1.44	86.42±0.78	36.23±0.13	65.77±1.54
InfoNCE Loss	84.97±1.46	85.62±0.98	35.87±0.35	63.89±0.91
w/o Struct. Pairs	86.87±1.35	86.67±0.68	36.21±0.47	64.48±1.03
w/o Seman. Pairs	87.54±0.58	87.11±0.93	35.42±0.66	63.11±0.55
S3GCL	88.47±1.39	87.89±1.23	36.88±0.34	66.27±0.1.33

incorporating spatial information enables the MLP to effectively capture graph context. 2) S3GCL demonstrates a consistent ability to generalize across graphs with varying homophily levels. Among 14 benchmark datasets, our method ranks the highest and attains state-of-the-art performance on 11 of them. This underscores the efficacy of different filters in handling diverse graph types. 3) Notably, the performance gains with S3GCL are more pronounced in heterophilic graphs than in homophilic ones, compared to other baseline methods. This suggests that the encoder plays a pivotal role in GCL, an aspect previously overlooked in other approaches. Furthermore, to intuitively underscore the effectiveness of S3GCL, we employ the t-SNE algorithm to visualize the learned node representations. The results, depicted in Figure 5, reveal that our method yields more distinct representations in the latent space compared to competing methods, further highlighting its superiority.

4.3. Efficiency

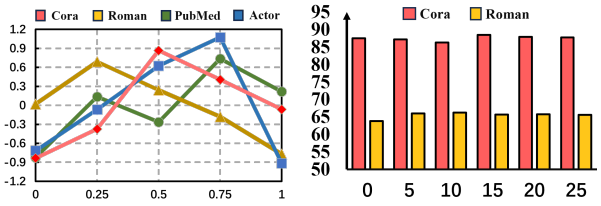
This section addresses **Q2** by examining the inference time across datasets of varying sizes, as illustrated in Figure 4. The inference time for the self-supervised methods discussed in this paper is comparable to that of SAGE (Hamilton et al., 2017). Our analysis reveals that S3GCL consistently outperforms other GCL methods in terms of inference efficiency on almost all datasets, particularly with larger graphs. For instance, in the Ogbn-Arxiv dataset, a 2-layer BGRL requires 389.6ms to obtain representations, whereas S3GCL necessitates only 2.23ms, translating to an impressive **×174** acceleration. This disparity in inference speed becomes increasingly pronounced with larger datasets.

Additionally, for smaller datasets, training time tends to outweigh inference time (Zheng et al., 2022b). Consequently,



(a) Cora (b) Cornell

Figure 6. **Cosine Similarity** assessed through pairs comprising one-hop neighbors, two-hop neighbors, and randomly sampled nodes on the Cora and Cornell dataset.



(a) Coefficient α (b) Semantic Neighbors k

Figure 7. **Analysis on hyper-parameter.** Performance with hyper-parameter α and k, where red and green represent the Cora and PubMed respectively, while yellow and blue represent the Roman-empire and Actor respectively.

we also perform experiments focusing on the training process with smaller graphs, detailed in Tab. 2. The results demonstrate that S3GCL not only achieves a balance between accuracy and inference time but also requires similar or less training time compared to other baseline methods.

4.4. Effectiveness

In this section, we undertake ablation studies to address **Q3**. As detailed in Tab. 3, we first assess the effectiveness of our proposed cosine-parameterized Chebyshev polynomial. Our initial baseline is a basic cross-architecture GCL model, namely GCN-MLP contrastive learning (GCN-MLP). The results indicate that relying solely on a low-pass filter, based on the homophily assumption, leads to adaptability issues in graphs with lower homophily levels. Subsequently, we evaluate the original ChebNet with interpolation but without parameterization (w/o Param.) and a linear parameterization strategy (Linear Param.). ChebNet exhibits poor performance across all datasets. This is likely due to the absence of labels, a spectral GNN struggles to learn the optimal filter shape. The linear parameterization approach results in suboptimal performance, as it fails to allocate unbalanced attention across different frequency intervals, potentially causing the two graph views to remain not disentangled.

Further, we investigate the efficacy of spatial pairs through three experiments: removing structurally neighboring pairs

(w/o Struct. Pairs), removing semantically neighboring pairs (w/o Seman. Pairs), and removing both (InfoNCE Loss in Equation (9)). The findings affirm the significance of both structurally and semantically neighboring nodes in the spatial domain, which substantially enhances the MLP awareness of graph context. This enhancement is pivotal for the MLP encoder to learn more expressive representations.

5. Conclusion

In this paper, we address a significant gap in existing graph contrastive learning methods: their limited generalization across graphs with varying levels of homophily and challenges in efficient inference for applications. We empirically and theoretically analyze the relationship between frequency and homophily levels. We then introduce a cosine-parameterized Chebyshev polynomial, designed to adapt to different graph types and facilitate training spectral GNNs in label-scarce scenarios. To tackle the issue of latency in inference, we leverage an MLP encoder and incorporate spatial positive pairs to enhance its awareness of the graph context. Based on these insights, we propose the S3GCL framework, which is strategically designed to achieve both competitive generalizable performance and efficient inference.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant (62361166629, 62176188, 623B2080).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Altenburger, K. M. and Ugander, J. Monophily in social networks introduces similarity among friends-of-friends. *Nature human behaviour*, 2(4):284–290, 2018.

Azabou, M., Ganesh, V., Thakoor, S., Lin, C.-H., Sathidevi, L., Liu, R., Valko, M., Veličković, P., and Dyer, E. L. Half-hop: A graph upsampling approach for slowing down message passing. In *International Conference on Machine Learning*, pp. 1341–1360. PMLR, 2023.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Bielak, P., Kajdanowicz, T., and Chawla, N. V. Graph barlow twins: A self-supervised representation learning framework for graphs. *arXiv preprint arXiv:2106.02466*, 2021.

Cai, X., Huang, C., Xia, L., and Ren, X. Lightgcl: Simple yet effective graph contrastive learning for recommendation. *arXiv preprint arXiv:2302.08191*, 2023.

Cao, J., Ku, D., Du, J., Ng, V., Wang, Y., and Dong, W. A structurally enhanced, ergonomically and human–computer interaction improved intelligent seat's system. *Designs*, 1(2):11, 2017. doi: 10.3390/designs1020011.

Chen, J., Lei, R., and Wei, Z. Polygcl: Graph contrastive learning via learnable spectral polynomial filters. In *ICLR*, 2024.

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

Dai, E., Zhao, T., Zhu, H., Xu, J., Guo, Z., Liu, H., Tang, J., and Wang, S. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. *arXiv preprint arXiv:2204.08570*, 2022.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *NeurIPS*, 29, 2016.

Deshpande, Y., Sen, S., Montanari, A., and Mossel, E. Contextual stochastic block models. *NeurIPS*, 31, 2018.

Ding, M., Kong, K., Li, J., Zhu, C., Dickerson, J., Huang, F., and Goldstein, T. Vq-gnn: A universal framework to scale up graph neural networks using vector quantization. *NeurIPS*, 34:6733–6746, 2021.

Fatemi, B., El Asri, L., and Kazemi, S. M. Slaps: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, 34:22667–22681, 2021.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *IJCV*, pp. 1789–1819, 2021.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

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

He, D., Liang, C., Liu, H., Wen, M., Jiao, P., and Feng, Z. Block modeling-guided graph convolutional neural networks. In *AAAI*, volume 36, pp. 4022–4029, 2022a.

He, D., Zhao, J., Guo, R., Feng, Z., Jin, D., Huang, Y., Wang, Z., and Zhang, W. Contrastive learning meets homophily: two birds with one stone. In *ICML*, pp. 12775–12789. PMLR, 2023.

He, M., Wei, Z., and Wen, J.-R. Convolutional neural networks on graphs with chebyshev approximation, revisited. *NeurIPS*, 35:7264–7276, 2022b.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *NeurIPS*, 33: 22118–22133, 2020.

Huang, K. and Liò, P. An effective universal polynomial basis for spectral graph neural networks. *arXiv preprint arXiv:2311.18177*, 2023.

Huang, W., Wan, G., Ye, M., and Du, B. Federated graph semantic and structural learning. 2023a.

Huang, W., Ye, M., Shi, Z., Li, H., and Du, B. Rethinking federated learning with domain shift: A prototype view. In *CVPR*, 2023b.

Huang, W., Ye, M., Shi, Z., Wan, G., Li, H., Du, B., and Yang, Q. A federated learning for generalization, robustness, fairness: A survey and benchmark. *arXiv*, 2023c.

Jia, Z., Lin, S., Ying, R., You, J., Leskovec, J., and Aiken, A. Redundancy-free computation for graph neural networks. In *ACM SIGKDD*, pp. 997–1005, 2020.

Jiang, H., Qin, F., Cao, J., Peng, Y., and Shao, Y. Recurrent neural network from adder's perspective: Carry-lookahead rnn. *Neural Networks*, 144:297–306, December 2021.

Karhadkar, K., Banerjee, P. K., and Montúfar, G. Fosr: First-order spectral rewiring for addressing oversquashing in gnns. *arXiv preprint arXiv:2210.11790*, 2022.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Kong, K., Chen, J., Kirchenbauer, J., Ni, R., Bruss, C. B., and Goldstein, T. Goat: A global transformer on large-scale graphs. In *International Conference on Machine Learning*, pp. 17375–17390. PMLR, 2023.

Lei, R., Wang, Z., Li, Y., Ding, B., and Wei, Z. Evennet: Ignoring odd-hop neighbors improves robustness of graph neural networks. *NeurIPS*, 35:4694–4706, 2022.

Lin, T. and Cao, J. Touch interactive system design with intelligent vase of psychotherapy for alzheimer's disease. *Designs*, 4(3):28, 2020. doi: 10.3390/designs4030028.

Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., and Yu, P. Graph self-supervised learning: A survey. *IEEE TKDE*, 2022a.

Liu, Y., Tu, W., Zhou, S., Liu, X., Song, L., Yang, X., and Zhu, E. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7603–7611, 2022b.

Liu, Y., Ding, K., Lu, Q., Li, F., Zhang, L. Y., and Pan, S. Towards self-interpretable graph-level anomaly detection. *arXiv preprint arXiv:2310.16520*, 2023a.

Liu, Y., Ding, K., Wang, J., Lee, V., Liu, H., and Pan, S. Learning strong graph neural networks with weak information. *arXiv preprint arXiv:2305.18457*, 2023b.

Liu, Y., Liang, K., Xia, J., Zhou, S., Yang, X., Liu, X., and Li, S. Z. Dink-net: Neural clustering on large graphs. In *ICML*, 2023c.

Liu, Y., Yang, X., Zhou, S., Liu, X., Wang, S., Liang, K., Tu, W., and Li, L. Simple contrastive graph clustering. *IEEE TNNLS*, 2023d.

Liu, Y., Yang, X., Zhou, S., Liu, X., Wang, Z., Liang, K., Tu, W., Li, L., Duan, J., and Chen, C. Hard sample aware network for contrastive deep graph clustering. In *AAAI*, volume 37, pp. 8914–8922, 2023e.

Liu, Y., Zheng, Y., Zhang, D., Lee, V., and Pan, S. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *AAAI*, 2023f.

Liu, Z., Wan, G., Prakash, B. A., Lau, M. S., and Jin, W. A review of graph neural networks in epidemic modeling. *arXiv preprint arXiv:2403.19852*, 2024.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.-W., and Precup, D. Revisiting heterophily for graph neural networks. *NeurIPS*, 35:1362–1375, 2022.

Pan, E. and Kang, Z. Beyond homophily: Reconstructing structure for graph-agnostic clustering. *arXiv preprint arXiv:2305.02931*, 2023.

Pan, S., Zheng, Y., and Liu, Y. Integrating graphs with large language models: Methods and prospects. *IEEE Intelligent Systems*, 39(1):64–68, 2024.

Pang, J., Wang, Z., Tang, J., Xiao, M., and Yin, N. Sa-gda: Spectral augmentation for graph domain adaptation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 309–318, 2023.

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

Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., and Prokhorenkova, L. A critical look at the evaluation of gnns under heterophily: are we really making progress? *arXiv preprint arXiv:2302.11640*, 2023.

Robbins, H. and Monro, S. A stochastic approximation method. *AoMS*, pp. 400–407, 1951.

Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

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

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. In *NeurIPS Workshop*, 2018.

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

Tian, Y., Dong, K., Zhang, C., Zhang, C., and Chawla, N. V. Heterogeneous graph masked autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023a.

Tian, Y., Zhang, C., Guo, Z., Zhang, X., and Chawla, N. Learning mlps on graphs: A unified view of effectiveness, robustness, and efficiency. In *ICLR*, 2023b.

Tian, Y., Song, H., Wang, Z., Wang, H., Hu, Z., Wang, F., Chawla, N. V., and Xu, P. Graph neural prompting with large language models. In *AAAI*, 2024a.

Tian, Y., Zhang, C., Kou, Z., Liu, Z., Zhang, X., and Chawla, N. V. Ugmae: A unified framework for graph masked autoencoders. *arXiv preprint arXiv:2402.08023*, 2024b.

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

Wan, G., Huang, W., and Ye, M. Federated graph learning under domain shift with generalizable prototypes. In *AAAI*, 2024.

Wang, H., Yin, H., Zhang, M., and Li, P. Equivariant and stable positional encoding for more powerful graph neural networks. *arXiv preprint arXiv:2203.00199*, 2022.

Wang, H., Zhang, J., Zhu, Q., Huang, W., Kawaguchi, K., and Xiao, X. Single-pass contrastive learning can work for both homophilic and heterophilic graph. *Transactions on Machine Learning Research*, 2023.

Wang, R., Mou, S., Wang, X., Xiao, W., Ju, Q., Shi, C., and Xie, X. Graph structure estimation neural networks. In *WWW*, pp. 342–353, 2021.

Wang, X. and Zhang, M. How powerful are spectral graph neural networks. In *ICML*, pp. 23341–23362. PMLR, 2022.

Wu, L., Lin, H., Huang, Y., Fan, T., and Li, S. Z. Extracting low-/high-frequency knowledge from graph neural networks and injecting it into mlps: An effective gnn-to-mlp distillation framework. *arXiv preprint arXiv:2305.10758*, 2023a.

Wu, L., Lin, H., Huang, Y., and Li, S. Z. Quantifying the knowledge in gnns for reliable distillation into mlps. *arXiv preprint arXiv:2306.05628*, 2023b.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE TNNLS*, pp. 4–24, 2020.

Xia, J., Wu, L., Wang, G., Chen, J., and Li, S. Z. Progcl: Rethinking hard negative mining in graph contrastive learning. *arXiv preprint arXiv:2110.02027*, 2021.

Xia, J., Wu, L., Chen, J., Hu, B., and Li, S. Z. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *WWW*, pp. 1070–1079, 2022.

Xiao, T., Chen, Z., Guo, Z., Zhuang, Z., and Wang, S. Decoupled self-supervised learning for non-homophilous graphs. *arXiv preprint arXiv:2206.03601*, 2022.

Xiao, T., Zhu, H., Chen, Z., and Wang, S. Simple and asymmetric graph contrastive learning without augmentations. *arXiv preprint arXiv:2310.18884*, 2023.

Xiong, S., Yang, Y., Fekri, F., and Kerce, J. C. Tilp: Differentiable learning of temporal logical rules on knowledge graphs. *arXiv preprint arXiv:2402.12309*, 2024a.

Xiong, S., Yang, Y., Payani, A., Kerce, J. C., and Fekri, F. Teilp: Time prediction over knowledge graphs via logical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16112–16119, 2024b.

Xu, D., Cheng, W., Luo, D., Chen, H., and Zhang, X. Infogcl: Information-aware graph contrastive learning. In *NeurIPS*, pp. 30414–30425, 2021.

Xu, J., Dai, E., Luo, D., Zhang, X., and Wang, S. Learning graph filters for spectral gnns via newton interpolation. *arXiv preprint arXiv:2310.10064*, 2023.

Yang, X., Liu, Y., Zhou, S., Wang, S., Tu, W., Zheng, Q., Liu, X., Fang, L., and Zhu, E. Cluster-guided contrastive graph clustering network. *arXiv preprint arXiv:2301.01098*, 2023.

Yin, N., Wang, M., Chen, Z., De Masi, G., Xiong, H., and Gu, B. Dynamic spiking graph neural networks. In *AAAI*.

Yin, N., Feng, F., Luo, Z., Zhang, X., Wang, W., Luo, X., Chen, C., and Hua, X.-S. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 1621–1634. IEEE, 2022a.

Yin, N., Shen, L., Li, B., Wang, M., Luo, X., Chen, C., Luo, Z., and Hua, X.-S. Deal: An unsupervised domain adaptive framework for graph-level classification. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3470–3479, 2022b.

Yin, N., Shen, L., Wang, M., Lan, L., Ma, Z., Chen, C., Hua, X.-S., and Luo, X. Coco: A coupled contrastive framework for unsupervised domain adaptive graph classification. In *International Conference on Machine Learning*, pp. 40040–40053. PMLR, 2023a.

Yin, N., Wang, M., Chen, Z., Shen, L., Xiong, H., Gu, B., and Luo, X. Dream: Dual structured exploration with mixup for open-set graph domain adaption. In *The Twelfth International Conference on Learning Representations*, 2023b.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. *NeurIPS*, 33:5812–5823, 2020a.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In *NeurIPS*, pp. 5812–5823, 2020b.

Zhang, G., Zhang, S., and Yuan, G. Bayesian graph local extrema convolution with long-tail strategy for misinformation detection. *ACM Transactions on Knowledge Discovery from Data*, 2024.

Zhang, S., Liu, Y., Sun, Y., and Shah, N. Graph-less neural networks: Teaching old mlps new tricks via distillation. In *ICLR*, 2022.

Zheng, X., Liu, Y., Pan, S., Zhang, M., Jin, D., and Yu, P. S. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022a.

Zheng, X., Liu, Y., Bao, Z., Fang, M., Hu, X., Liew, A. W.-C., and Pan, S. Towards data-centric graph machine learning: Review and outlook. *arXiv preprint arXiv:2309.10979*, 2023a.

Zheng, Y., Pan, S., Lee, V., Zheng, Y., and Yu, P. S. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *NeurIPS*, 35:10809–10820, 2022b.

Zheng, Y., Zhang, H., Lee, V., Zheng, Y., Wang, X., and Pan, S. Finding the missing-half: Graph complementary learning for homophily-prone and heterophily-prone graphs. *arXiv preprint arXiv:2306.07608*, 2023b.

Zhou, H., Srivastava, A., Zeng, H., Kannan, R., and Prasanna, V. Accelerating large scale real-time gnn inference using channel pruning. *arXiv preprint arXiv:2105.04528*, 2021.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. Deep graph contrastive representation learning. In *ICML*, 2020.

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

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. Graph contrastive learning with adaptive augmentation. In *WWW*, pp. 2069–2080, 2021b.

A. Datasets Details

We evaluate S3GCL on seven homophilic graphs: Cora, CiteSeer, PubMed, Wiki-CS, Amazon-Computer, Amazon-Photo, CoAuthor-CS, and Ogbn-Arxiv (Sen et al., 2008; Shchur et al., 2018; Hu et al., 2020) and seven heterophilic graph data: Cornell, Texas, Wisconsin, Actor, Roman-empire, Minesweeper, and Tolokers (Rozemberczki et al., 2021; Platonov et al., 2023). Since other widely used datasets (*i.e.* Squirrel and Chameleon) are faced with drawbacks of the presenting duplicate nodes (Platonov et al., 2023), so we abandon them in our experiments. We utilize a commonly used split of 60 %/20%/20% for train/validation/test sets. For Obgn-Arxiv, we process the dataset in PyG using OGB public interfaces with the standard public split setting. The statistics of datasets are provided in Tab. 4, which also include homophily level h. The details are introduced as follows:

- **Cora, CiteSeer**, and **PubMed**. These datasets are recognized as classic examples of homophilic citation networks. In these graphs, nodes are indicative of academic papers, while edges signify the citations connecting them. The papers are represented through bag-of-word features, and the research themes of the papers are reflected in the labels.
- **Amazon Computers** and **Amazon Photo**. In these Amazon-based co-purchase networks, each node corresponds to a product, with edges indicating frequent co-purchases between products. Product reviews are converted into bag-of-word feature representations, and the product categories serve as labels.
- **CoAuthor CS**. Extracted from the Microsoft Academic Graph, the network map co-authorship connections. Nodes represent authors, and co-authorship is marked by edges. The features are bag-of-words embeddings from paper keywords, and labels denote the authors' research areas.
- **Ogbn-Arxiv**. A large-scale directed graph from the arXiv's Computer Science section. Nodes represent scientific papers, with edges indicating citations. Features are derived from bag-of-words representations of titles and abstracts, and labels denote the paper's arXiv category. Primarily used for node classification tasks in graph machine learning.
- **Cornell, Texas**, and **Wisconsin**. Originating from the WebKB project, these heterophilic networks represent web pages from various university computer science departments. Nodes are these web pages, linked by edges that denote hyperlinks. The content of each page is encapsulated in bag-of-words features, with labels identifying the web page types.
- **Actor**. This unique heterophilic network maps the co-occurrence of actors within movies. Nodes symbolize actors, with edges revealing co-occurrences. Wikipedia pages provide keyword-based features, and the labels are constituted of words linked to the respective actors.
- **Roman-empire**. Derived from the English Wikipedia article on the Roman Empire, this dataset represents words as nodes, connected based on their adjacency or syntactic dependency in the text. Nodes are classified based on their syntactic roles, with features sourced from FastText word embeddings.
- **Minesweeper**. A synthetic dataset inspired by the Minesweeper game. Nodes represent cells, connected to adjacent cells, with the task to identify nodes that are mines. Node features indicate the count of neighboring mines.
- **Tolokers**. Based on data from the Toloka crowdsourcing platform, this dataset includes nodes representing workers who participated in various projects. Edges connect workers who work on the same task. The objective is to predict banned workers, with features derived from their profile information and task performance.

Table 4. **Statistics** of datasets used in experiments.

Dataset	#Nodes	#Edges	#Classes	#Features	Homophily Level
Cora	2,708	5,278	7	1,433	0.810
Citeseer	3,327	4,552	6	3,703	0.736
Pubmed	19,717	44,324	3	500	0.802
Amz-Comp	13,752	574,418	10	767	0.777
Amz-Photo	7,650	287,326	8	745	0.827
Coauthor-CS	18,333	327,576	15	6,805	0.808
Obgn-Arxiv	169,343	1,166,243	40	128	0.655
Cornell	183	298	5	1,703	0.305
Texas	183	325	5	1,703	0.108
Wisconsin	251	515	5	1,703	0.196
Actor	7,600	30,019	5	932	0.219
Roman-empire	22,622	65,854	18	300	0.047
Minesweeper	10,000	78,804	2	7	0.683
Tolokers	11,758	1,038,000	2	10	0.595

B. Synthetic Datasets

In this study, we employ the Contextual Stochastic Block Model (CSBM) (Deshpande et al., 2018; Xu et al., 2023) to create synthetic graphs. These graphs are characterized by adjustable edge probabilities within and between different classes. The fundamental concept is that nodes within the same class exhibit a consistent feature distribution. The generated graph is denoted as $\mathcal{G} \sim \text{CSBM}(n, f, \sigma, \mu)$, where n represents the total node count, f is the feature dimension, and σ and μ are hyperparameters. The hyperparameters σ and μ respectively influence the contributions from the graph's structure and the node features. We consider two classes of equal size, c_1 and c_0, each comprising $n/2$ nodes. The CSBM generates features as follows:

$$\mathbf{x}_i = \sqrt{\frac{\mu}{n}} y_i u + \frac{w_i}{\sqrt{f}}, \tag{14}$$

where $y_i \in \{-1, +1\}$ indicates the label of node v_i, μ is the mean value of the Gaussian distribution, $u \sim \mathcal{N}(0, I/f)$, and the elements of w_i follow independent standard normal distributions. We define the average degree of the generated graph as d, and the adjacency matrix $\mathbf{A}$ for the CSBM graph is described by:

$$P(\mathbf{A}_{ij} = 1) = \begin{cases} \frac{1}{n}(d + \sigma\sqrt{d}) & \text{when } y_i = y_j \\ \frac{1}{n}(d - \sigma\sqrt{d}) & \text{when } y_i \neq y_j. \end{cases} \tag{15}$$

The homophily level h is adjustable by modifying $\sigma = \sqrt{d(2h-1)}$, with a range of $-\sqrt{d} \leq \sigma \leq \sqrt{d}$. A completely heterophilic graph corresponds to $\sigma = -\sqrt{d}$, while a completely homophilic graph occurs when $\sigma = \sqrt{d}$. In accordance with (Xu et al., 2023), we set $d = 5$, $\mu = 1$, $n = 3000$, and $f = 4000$ to generate our synthetic dataset. By varying σ, we can produce graphs with different levels of homophily for our case studies.

C. Implementation Details

The experiments are conducted using NVIDIA GeForce RTX 3090 GPUs as the hardware platform, coupled with Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz. The deep learning framework employed was Pytorch, version 1.11.0, alongside CUDA version 11.3. Regarding the network architecture, consistent with prevalent practices in GCL, we used GraphSage (Hamilton et al., 2017) as a 2-layer encoder for all baseline models. We utilize the MLP projection for aligning the outputs of two models. The hidden layer size was set to 1024 for each dataset. For optimization, Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951) was chosen, featuring a momentum of 0.9 and a weight decay of $1e-5$. The learning rate was configured to $5e-4$ during the training process and $1e-2$ for the linear evaluation phase.

D. Detailed Proofs

D.1. Proof of Theorem 2.5

Theorem 4.4. *Consider graph signals x^l and x^h processed by filters g_l and g_h, respectively. In heterophilic graphs (where $p < q$) and when $\Delta D(x^h) < \Delta D(x^l)$, there exists an integer $M (0 < M \leq N-1)$ such that $\sum_{i=M}^{N-1} g_h^2(\lambda_i) \geq \sum_{i=M}^{N-1} g_l^2(\lambda_i)$, with $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1} \leq 2$.*

Proof. Given a graph $\mathcal{G}$ generated by CSBM with intra-class probability p and inter-class probability q, and referring to Lemma 2.4, we have:

$$\Delta D(x^h) - \Delta D(x^l) = 2\mathbb{E}[(\frac{p}{p+q} - \frac{q}{p+q})\mathbf{f}(x^h) - (\frac{p}{p+q} - \frac{q}{p+q})\mathbf{f}(x^l)]. \tag{16}$$

With $\Delta D(x^h) < \Delta D(x^l)$ and $p < q$, it follows that:

$$\Delta D(x^h) - \Delta D(x^l) = 2\mathbb{E}\left[\frac{p-q}{p+q}[\mathbf{f}(x^h) - \mathbf{f}(x^l)]\right] < 0 \tag{17}$$

Therefore, $\mathbf{f}(x^h) > \mathbf{f}(x^l)$. By the definition of spectral signal frequency in Definition 2.2, we compute $\mathbf{f}(x)$ as follows:

$$\begin{aligned}\mathbf{f}(x) &= \frac{x^\top \mathbf{L} x}{2} = \sum_{(u,v)\in\mathcal{E}} \frac{(x_u - x_v)^2}{2},\\ \mathbf{f}(x^l) &= \frac{x^{l\top} \mathbf{L} x^l}{2} = \sum_{(u,v)\in\mathcal{E}} \frac{(x^l_u - x^l_v)^2}{2},\\ \mathbf{f}(x^h) &= \frac{x^{h\top} \mathbf{L} x^h}{2} = \sum_{(u,v)\in\mathcal{E}} \frac{(x^h_u - x^h_v)^2}{2},\end{aligned} \tag{18}$$

With $\mathbf{L}$ being real and symmetric, a set of orthonormal eigenvectors v_i forms a complete basis in $\mathbb{R}^N$. Any graph signal $x \in \mathbb{R}^N$ can be decomposed into to a weighted sum of v_i, represented as $x = \sum_{i=0}^{N-1} \alpha_i v_i$, where $\alpha_i = s^\top v_i$. α_i as the coefficient of x at frequency component i of graph $\mathcal{G}$. Hence, the filtered graph signals are:

$$\begin{aligned}x^l = g_l(\mathbf{L})x = \mathbf{U} g_l(\mathbf{\Lambda})\mathbf{U}^\top x = \left(\sum_{i=0}^{N-1} g_l(\lambda_i) v_i v_i^\top\right)\left(\sum_{i=0}^{N-1} \alpha_i v_i\right) = \sum_{i=0}^{N-1} g_1(\lambda_i)\alpha_i v_i,\\ x^h = g_h(\mathbf{L})x = \mathbf{U} g_h(\mathbf{\Lambda})\mathbf{U}^\top x = \left(\sum_{i=0}^{N-1} g_h(\lambda_i) v_i v_i^\top\right)\left(\sum_{i=0}^{N-1} \alpha_i v_i\right) = \sum_{i=0}^{N-1} g_2(\lambda_i)\alpha_i v_i.\end{aligned} \tag{19}$$

The spectral signal frequency is decomposed into:

$$\begin{aligned}\mathbf{f}(x^l) = \left(\sum_{i=0}^{N-1} g_l(\lambda_i)\alpha_i v_i\right)^\top \left(\sum_{i=0}^{N-1} \lambda_i v_i v_i^\top\right)\left(\sum_{i=0}^{N-1} g_l(\lambda_i)\alpha_i v_i\right) = \sum_{i=0}^{N-1} \lambda_i \alpha_i^2 g_l^2(\lambda_i).\\ \mathbf{f}(x^h) = \left(\sum_{i=0}^{N-1} g_h(\lambda_i)\alpha_i v_i\right)^\top \left(\sum_{i=0}^{N-1} \lambda_i v_i v_i^\top\right)\left(\sum_{i=0}^{N-1} g_h(\lambda_i)\alpha_i v_i\right) = \sum_{i=0}^{N-1} \lambda_i \alpha_i^2 g_h^2(\lambda_i).\end{aligned} \tag{20}$$

Given $\mathbf{f}(x^h) > \mathbf{f}(x^l)$, we deduce $\sum_{i=0}^{N-1} \lambda_i g_h^2(\lambda_i) > \sum_{i=0}^{N-1} \lambda_i g_l^2(\lambda_i)$, leading to:

$$\mathbf{f}(x^h) - \mathbf{f}(x^l) = \sum_{i=0}^{N-1} \alpha_i^2 \Big[\lambda_i g_h^2(\lambda_i) - \lambda_i g_l^2(\lambda_i)\Big] > 0 \tag{21}$$

Then we move to prove that there exist an interger $M (0 < M \leq N-1)$ such that $\sum_{i=M}^{N-1} g_h^2(\lambda_i) \geq \sum_{i=M}^{N-1} g_l^2(\lambda_i)$. We prove this statement by contradiction. Suppose there does not exist an integer $M (0 < M \leq N-1)$ that satisfies this condition. We have $\sum_{i=M}^{N-1} g_h^2(\lambda_i) < \sum_{i=M}^{N-1} g_l^2(\lambda_i)$ with $M (0 < M \leq N-1)$, then following inequalities exist:

$$(\lambda_1 - \lambda_0)\Big[g_h^2(\lambda_1) + g_h^2(\lambda_2) + \ldots + g_h^2(\lambda_{N-1})\Big] \leq (\lambda_1 - \lambda_0)\Big[g_l^2(\lambda_1) + g_l^2(\lambda_2) + \ldots + g_l^2(\lambda_{N-1})\Big] \tag{22}$$

$$(\lambda_2 - \lambda_1)\Big[g_h^2(\lambda_2) + \ldots + g_h^2(\lambda_{N-1})\Big] \leq (\lambda_2 - \lambda_1)\Big[g_l^2(\lambda_2) + \ldots + g_l^2(\lambda_{N-1})\Big] \tag{23}$$

$$\vdots$$

$$(\lambda_{N-1} - \lambda_{N-2})\Big[g_h^2(\lambda_{N-1})\Big] \leq (\lambda_{N-1} - \lambda_{N-2})\Big[g_l^2(\lambda_{N-1})\Big] \tag{24}$$

We assume that both filters g_l and g_h possess the same ℓ_2-norms. This assumption allows us to concentrate on the role of different frequency ranges and avoid trivial solutions:

$$\lambda_0\Big[g_h^2(\lambda_0) + g_h^2(\lambda_1) + \ldots + g_h^2(\lambda_{N-1})\Big] = \lambda_0\Big[g_l^2(\lambda_0) + g_l^2(\lambda_1) + \ldots + g_l^2(\lambda_{N-1})\Big]. \tag{25}$$

Summing both sides of the above inequalities and considering Equation (25), we deduce:

$$\sum_{i=0}^{N-1} \lambda_i g_h^2(\lambda_i) \leq \sum_{i=0}^{N-1} \lambda_i g_l^2(\lambda_i),$$

which contradicts the earlier results that $\sum_{i=0}^{N-1} \lambda_i g_h^2(\lambda_i) > \sum_{i=0}^{N-1} \lambda_i g_l^2(\lambda_i)$.Therefore, the initial assumption is invalid, and there must exist an integer $M(0 < M \leq n-1)$ such that $\sum_{i=M}^{N-1} g_h^2(\lambda_i) \geq \sum_{i=M}^{N-1} g_l^2(\lambda_i)$, which completes the proof.

D.2. Proof of Prop. 3.1

Proposition 4.5. *Given a node representation z_i and its neighboring node representation z_j, optimizing Equation (12) leads to maximizing their mutual information $I(z_j; z_i)$, while inherently minimizing $D_{KL}(z_j||z_i)$:*

$$I(z_j; z_i) \sim \frac{1}{D_{KL}(z_j||z_i)} \tag{26}$$

Proof. Mutual Information Neural Estimation (MINE)(Belghazi et al., 2018) converts mutual information maximization into minimizing the InfoNCE loss Equation (12), where the positive pair is z_i and its neighboring node representation z_j. Utilizing the relationship between mutual information and information entropy, we derive:

$$I(z_j; z_i) = H(z_j) + H(z_i) - H(z_j, z_i), \tag{27}$$

where $H(\cdot)$ denotes information entropy, and $H(\cdot, \cdot)$ represents joint entropy. The Kullback-Leibler divergence, or information gain, in relation to information entropy is defined as:

$$D_{KL}(z_j \parallel z_i) = H(z_j, z_i) - H(z_j). \tag{28}$$

By integrating Equation (27) and Equation (28), we obtain:

$$I(z_j; z_i) = H(z_i) - D_{KL}(z_j \parallel z_i). \tag{29}$$

As per Equation (29), there is a negative correlation between $I(z_j; z_i)$ and $D_{KL}(z_j \parallel z_i)$. This implies that maximizing $I(z_j; z_i)$ inherently minimizes $D_{KL}(z_j \parallel z_i)$, thereby fulfilling the objective of transferring graph property knowledge from the GNN to the MLP:

$$I(z_j; z_i) \sim \frac{1}{D_{KL}(z_j||z_i)}. \tag{30}$$

E. Complexity Analysis

In this subsection, we delve into the time complexity of S3GCL. The process of acquiring the semantically neighboring node set $\mathcal{N}_i''$ involves executing the k-nearest neighbors (kNN) algorithm on the entire graph. This step, a preprocessing phase external to formal training, is performed only once per dataset. Typically, computing a kNN graph requires a time complexity of $\mathcal{O}(N^2 D_f)$, where N is the number of nodes and D_f represents the feature dimension. This computation becomes particularly intensive when N is large. To mitigate this, following the approach in (Liu et al., 2023b), we initially apply the locality-sensitive approximation algorithm (Fatemi et al., 2021) in two separate batches. Subsequently, we merge the resulting local kNN graphs to form a comprehensive global kNN graph, ensuring potential connectivity between all nodes rather than just those within the same batch. This strategy effectively reduces the computational cost from $\mathcal{O}(N^2 D_f)$ to $\mathcal{O}(NBD_f)$, with B denoting the batch size.

During the learning process, the computation for propagating K-order polynomial filters is $\mathcal{O}(KE)$, where E represents the number of edges. The MLP encoder requires $\mathcal{O}(ND_f D)$ computations, with D being the representation dimension. The projection step involves $\mathcal{O}(NDD_p)$ computations, where D_p is the projection dimension. The complexity of our final loss Equation (12) is $\mathcal{O}(ND_p B')$, with B' as the batch size in contrastive learning. Overall, our method scales linearly with the number of nodes N, demonstrating its scalability.

F. Related Work

Graph Contrastive Learning. Within the domain of GCL (Liu et al., 2022a; Zhu et al., 2021a; Liu et al., 2023e;d; Yang et al., 2023; Liu et al., 2022b), contemporary studies have primarily focused on two distinct strategies: augmentation-based and augmentation-free techniques. Augmentation-based techniques, illustrated by GRACE (Zhu et al., 2020; Huang et al., 2023a; Wan et al., 2024), GCA(You et al., 2020a), and MVGRL(Hassani & Khasahmadi, 2020), concentrate on improving graph representation learning through diverse data augmentations such as edge dropping and attribute masking (You et al., 2020b). These techniques aspire to optimize mutual information across various graph perspectives. On the other hand, augmentation-free strategies, as demonstrated in studies such as SimGRACE (Xia et al., 2022) and BGRL (Thakoor et al., 2021), deviate from intricate data augmentation tactics. Alternatively, they depend on various graph encoders to create unique views, promoting the convergence of analogous node or class representations from these diverse perspectives (Yin et al., 2022b; 2023a; Tian et al., 2024b;a). In this study, we present a new, augmentation-free, yet effective cross-architecture GCL technique. It attains cross-pass uniformity between the node and its spatial neighbors, originating from full-pass MLP and biased-pass spectral GNN respectively.

Heterophilic Graphs. Despite significant advancements in GNNs (Zhang et al., 2024; Liu et al., 2024; Jiang et al., 2021; Yin et al., 2023b), most adhere to the homophily principle. However, real-world graphs often diverge from the homophily principle and exhibit heterophily, where connected nodes possess dissimilar features and distinct class labels (Zheng et al., 2023b; Pan & Kang, 2023). To tackle such graphs, numerous methods aimed at excelling in either heterophily or both homophily and heterophily contexts have emerged recently (Kong et al., 2023; Azabou et al., 2023; Pang et al., 2023). Among these, spectral GNNs have garnered notable success for their proficiency in learning filters of diverse shapes with labels, suitable for a variety of graphs. Contemporary works like GREET(Liu et al., 2023f) and NeCo (He et al., 2023) address non-homophily scenarios through edge discriminating augmentation and parameterized neighbor sampling, respectively. However, these approaches overlook the special consideration of the filter role, leading to suboptimal outcomes. In contrast, we propose a cosine-parameterized Chebyshev polynomial, enhancing generalization across both homophilic and heterophilic graphs without relying on labels.

Inference Acceleration on graphs. Recently, more and more research focuses on applying machine learning methods to industrial tasks (Cao et al., 2017; Lin & Cao, 2020). GNNs typically utilize message passing for feature aggregation from neighbors (Liu et al., 2023a; Zheng et al., 2023a; Pan et al., 2024), yet this process during inference can hinder latency-sensitive applications (Yin et al., 2022a; Yin et al.). Initial efforts in acceleration, such as quantization (Ding et al., 2021) and pruning (Zhou et al., 2021), failed to eliminate neighborhood dependencies. In response, GLNN (Zhang et al., 2022) introduces knowledge distillation (KD) (Hinton et al., 2015; Gou et al., 2021) to transfer knowledge from a teacher GNN to a student MLP, using the latter for inference acceleration. Subsequent studies (Tian et al., 2023b; Wu et al., 2023b) have enhanced this distillation process. Yet, these approaches primarily concentrate on well-trained teacher GNNs and label-supervised signals. Our work, in contrast, adopts a self-supervised approach, employing cross-pass GCL to enable MLP to acquire task-relevant knowledge and utilizing spatial positive nodes to foster graph-context awareness in MLP. The deployed MLP enjoyed both swift inference and competitive performance.

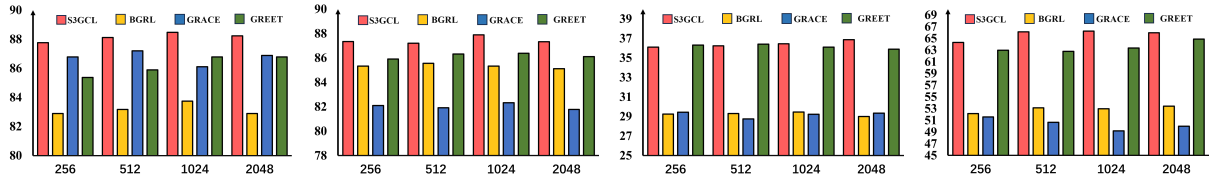
G. Sensitivity

In this section, we analyze the method sensitivity to various hyper-parameters. Our findings, as depicted in Figure 7, indicate that our approach demonstrates low sensitivity to the number of semantic neighbors k. Appropriate k will enhance performance, but too much k will introduce noise and affect the learning effect. Furthermore, we perform ablation studies on the coefficient α to highlight the respective roles of $\mathcal{L}_{fl}$ and $\mathcal{L}_{fh}$ in the learning process. The results, shown in Figure 6, indicate that S3GCL encourages greater similarity between one-hop and two-hop neighbors compared to randomly selected pairs, across both homophilic and heterophilic graphs. This suggests that S3GCL effectively makes nodes cognizant of their one-hop neighbor context, irrespective of homophily levels.

H. Additional Experimental Results

H.1. Representation Dimension

This section delves into the impact of varying dimensions on different datasets, with the findings presented in Figure 9. The results suggest that larger dimensions generally lead to improved outcomes for all graphs. However, it is observed that excessively small dimensions can hinder the learning process, while overly large dimensions might cause overfitting issues.

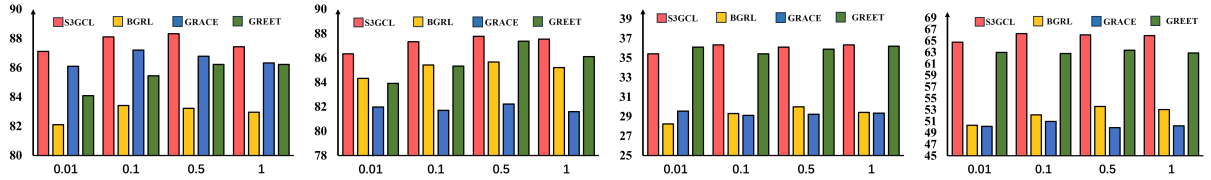


(a) Cora (b) PubMed (c) Actor (d) Roman-empire

Figure 8. **Impact** of varying representation dimensions on four datasets, where red and yellow represent the S3GCL and BGRL respectively, while blue and green represent the GRACE and GREET respectively.

H.2. Contrastive Learning Temperature

The diagnostic analysis regarding the contrastive learning temperature is presented in Figure 8. Overall, the performance remains stable across various contrastive learning temperatures, with the exception of some extreme cases (e.g., when τ is set to 0.01). Setting the temperature too high in contrastive learning can result in inadequate differentiation between similar and dissimilar pairs, leading to the loss of valuable information and reduced model sensitivity to subtle yet crucial distinctions. On the other hand, an excessively low temperature may cause the model to overemphasize minor differences between samples, potentially leading to overfitting and diminished generalization capabilities on novel, unseen data.



(a) Cora (b) PubMed (c) Actor (d) Roman-empire

Figure 9. **Impact** of varying contrastive learning temperature τ on four datasets, where red and yellow represent the S3GCL and BGRL respectively, while blue and green represent the GRACE and GREET respectively.