Spectral Reversal: Counteracting Singular Value Bias for Graph Prompting

Hanxu Yang[*], **Yuhuan Zhao**[*], **Xiaodong He**, **Zhao Kang**[†]
School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, Sichuan, China
yang_hanxu@std.uestc.edu.cn zkang@uestc.edu.cn

Abstract

Pre-training Graph Neural Networks (GNNs) via self-supervised learning has become a dominant paradigm, yet efficiently adapting frozen encoders remains a challenge. Graph prompting offers a parameter-efficient alternative to fine-tuning, but existing methods largely treat pre-trained models as opaque feature extractors, ignoring their internal spectral structure. In this work, we identify a systematic phenomenon in pre-trained GNNs, which we term spectral bias: optimization during pre-training disproportionately aligns representations with directions associated with large singular values, leaving low-energy directions under-explored. We show that these underutilized directions can encode complementary information that is beneficial for downstream adaptation, especially under distribution shift. To leverage this insight, we propose Spectral Reverse Prompt (SRP), a prompting framework that rebalances the spectral contributions of frozen GNN encoders. SRP applies a learnable soft-thresholding mask in the spectral domain to down-weight dominant directions while amplifying weaker ones. In addition, SRP incorporates a null-space augmentation module that captures variation in directions with minimal activation under the frozen encoder. Extensive experiments across multiple benchmarks demonstrate that SRP achieves state-of-the-art performance with minimal additional parameters, highlighting that reweighting spectral components is a principled and effective strategy for parameter-efficient graph adaptation. The code is available at https://github.com/keris-yang/Spectral-Reversal.

1 Introduction

Graph prompt tuning [1, 2] has emerged as a parameter-efficient alternative to fine-tuning for adapting pre-trained Graph Neural Networks [3, 4, 5, 6, 7] (GNNs). Existing approaches typically introduce learnable tokens [8, 9, 10] or modify readout layers [11, 12, 13], while keeping the backbone encoder frozen. However, these methods implicitly treat the pre-trained model as a black-box feature extractor, overlooking its internal structure.

In particular, little attention has been paid to the spectral properties of the learned transformations within pre-trained GNNs. During self-supervised pre-training, optimization tends to align representations with dominant modes of variation, which are often associated with large singular values of the effective transformation. We refer to this phenomenon as spectral bias, where dominant spectral components are emphasized while lower-energy directions remain underutilized. While these dominant directions capture prevalent patterns in pre-training data, they may not align with the requirements of downstream tasks [14], especially under inter-class inequality.

[*]Equal contribution.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

This observation suggests a different perspective on parameter-efficient adaptation: rather than further reinforcing dominant directions, it may be more effective to reweight the spectral contributions of the frozen encoder, allocating capacity to directions that were under-explored during pre-training. To this end, we propose **Spectral Reverse Prompt (SRP)**, a framework that explicitly rebalances spectral contributions through two complementary mechanisms. First, a spectral channel applies a learnable soft-thresholding mask to modulate singular directions, suppressing dominant components while amplifying weaker ones. Second, a null-space channel extracts informative variation from directions that are minimally expressed by the frozen encoder, further expanding the effective representation space.

By focusing adaptation on underutilized spectral components, SRP improves parameter efficiency while enhancing robustness to distribution shift. Empirical results across diverse benchmarks demonstrate that this spectral rebalancing leads to consistent performance gains over existing graph prompting methods across node-level and graph-level benchmarks. Our main contributions can be summarized as follows:

- We identify and quantify a form of spectral bias in pre-trained graph models using a **Downstream–Pretraining Misalignment Score** (DPMS), showing that dominant singular directions are disproportionately aligned with pre-training objectives, while lower-energy directions remain underutilized for downstream tasks.
- We propose SRP, a parameter-efficient graph prompting framework that rebalances spectral contributions of frozen encoders via two complementary components: (i) a learnable soft-thresholding spectral mask that suppresses dominant directions and amplifies weaker ones, and (ii) a null-space augmentation module that captures variation in directions minimally expressed by the encoder.
- We show that emphasizing underutilized spectral components leads to more effective and robust adaptation, particularly in low-resource settings.

2 Related Work

GNNs and pre-training. Numerous studies propose training powerful GNNs via self-supervised learning, which is broadly categorized into two paradigms: contrastive and generative methods. Contrastive methods aim to maximize agreement between augmented instances. Representative works include DGI [15] and InfoGraph [16], which focus on mutual information across different structural scales; GraphCL [17], GRACE [18], and SimGRACE [19] generate contrastive views via structural augmentation, feature masking, or parameter perturbation. Other paradigms avoid negative sampling through bootstrapping [20] or exploit domain-specific motifs [21]. Conversely, generative methods reconstruct specific graph properties: GraphMAE and GraphMAE2 [22, 23] focus on masked node features, while earlier strategies [24] leverage edge prediction. This paradigm of pre-training then fine-tuning has become the core of research in graph domain adaptation [25], graph foundation models [26, 27, 28, 29, 30] and graph reasoning mechanisms [31] in recent years.

Graph tuning methods. Graph prompt tuning and low-rank adaptation, or LoRA, offer parameter-efficient alternatives by optimizing minimal components alongside the backbone. GraphLoRA [32] extends traditional LoRA [33] by incorporating structure-aware contrastive losses. Meanwhile, graph prompting methods typically inject learnable tokens into various stages while preserving fixed message-passing rules. These approaches target different levels: node-feature prompting via GPF [8], edge-level adaptation through EdgePrompt [9], and topology modifications in GraphTOP[34], UniPrompt [35] and CurvPrompt [36]. Further research explores representation-level prompting such as GraphPrompt [37] and GPPT [11], as well as specialized designs: virtual-node prompting [38] and instance-aware schemes [10]. Despite their success, these approaches primarily adapt inputs or structures without explicitly analyzing the consensus between prompting mechanisms and weight metrics. This gap motivates our work: we rebalance spectral contributions and utilize the null space to enable universal parameter efficient, task-driven adaptation.

3 Notations and Preliminaries

General Graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathcal{Y})$ be a graph with N nodes, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ define the node and edge sets, respectively. The node attributes are captured by a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$, with each row $\mathbf{X}_i$ representing the F-dimensional feature of node v_i. Structural connectivity is denoted by the adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$, where $\mathbf{A}_{ij} = 1$ indicates an existing edge between v_i and v_j. Furthermore, each node v_i corresponds to a specific label y_i from the label space $\mathcal{Y}$. Note that $P(\cdot)$ represents a probability distribution, which is utilized conceptually in our context rather than for strict mathematical derivations.

Graph Prompt Tuning. Traditional pretrain-finetuning paradigms jointly update a pretrained graph encoder f_θ and a task-specific head g_ϕ using downstream data. In contrast, graph prompt learning freezes f_θ and optimizes only a newly introduced set of prompt parameters Ψ. The generalized objective for such prompt-based adaptation on a downstream dataset $\mathcal{D}$ is formulated as:

$$\max_{\Psi} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{A},\mathbf{X},y)\in\mathcal{D}} \sum_{i=1}^{N} \log P\left(y_i \mid \text{Predict}_\Psi\left(\mathbf{A}, \mathbf{X}, v_i; f_\theta\right)\right), \tag{1}$$

where $\mathcal{D} = \{(\mathbf{A}, \mathbf{X}, y)\}$. Here, $\text{Predict}_\Psi(\cdot)$ serves as a generic prediction module that leverages the structural topology $\mathbf{A}$, node features $\mathbf{X}$, the target node v_i, and the frozen backbone f_θ to estimate the corresponding label y_i. Depending on the architecture, the prompt Ψ can be injected at various stages: modifying the raw topological and feature inputs (input-level), embedding into intermediate network layers (layer-wise), or adjusting the final output embeddings before the classifier (representation-level).

Singular Value Decomposition. For any given weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$ with rank r, its compact Singular Value Decomposition (SVD) and the induced right null space are given by

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \ker(\mathbf{W}) = \text{span}(\mathbf{V})^\perp = \left\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{V}^\top\mathbf{x} = 0\right\}. \tag{2}$$

In this factorization, $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{n \times r}$ consist of orthonormal columns known as the left and right singular vectors, respectively. The term $\mathbf{\Sigma} \in \mathbb{R}^{r \times r}$ is a diagonal matrix containing the sorted, positive singular values $\{\sigma_i\}_{i=1}^r$ of $\mathbf{W}$ in descending order. Accordingly, the right null space of $\mathbf{W}$ is the orthogonal complement of the subspace spanned by its right singular vectors.

4 The Spectral Bias of Pre-trained GNNs

To establish our theoretical framework, we investigate a frozen GNN encoder with a weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ at an arbitrary layer. For downstream adaptation, we assume $W_{\text{down}} \in \mathbb{R}^{d_{\text{in}} \times C}$ represents the optimal linear classifier trained directly on raw node features.

Origins of Spectral Bias. The spectral bias in pre-trained GNNs fundamentally stems from two compounded phenomena: the nature of gradient descent and the graph message-passing mechanism. During pre-training, W is optimized over T steps of gradient descent. Each step applies a rank-1 update $\Delta W^{(t)} = -\eta_t g^{(t)}(h^{(t)})^\top$ based on the upstream gradient $g^{(t)}$ and layer input $h^{(t)}$. For a GNN, let $h^{(t)} = \phi(\widehat{A}\widetilde{W}^{(t)}\widetilde{h}^{(t)})$ and let the adjacency matrix be eigendecomposed as $\widehat{A} = Q\Lambda Q^\top$, the cumulative update is

$$\Delta W = W^{(T)} - W^{(0)} = -\sum_{t=1}^{T} \eta_t g^{(t)}(h^{(t)})^\top = -\sum_{t=1}^{T} \eta_t g^{(t)} \left[\phi\big(Q\Lambda Q^\top \widetilde{W}^{(t)}\widetilde{h}^{(t)}\big)\right]^\top \tag{3}$$

Since each update deposits energy along $h^{(t)}$, the spectral structure of ΔW is governed by the input covariance $\Sigma_h = \mathbb{E}_t[h^{(t)}(h^{(t)})^\top]$. Let $\Sigma_h = \sum_i \lambda_i e_i e_i^\top$ (with $\lambda_1 \geq \lambda_2 \geq \cdots$) and $P_r = \sum_{i=1}^r e_i e_i^\top$ be the projection onto the top-r principal directions.

Proposition 4.1 (Spectral Alignment and Concentration)**.** *If the training inputs satisfy $\|(I - P_r)h^{(t)}\|_2 \leq \delta\|h^{(t)}\|_2$ for all t, and the cross-step gradient terms vanish, then the cumulative gradient update inherits this concentration:*

$$\frac{\|P_r \Delta W^\top\|_F^2}{\|\Delta W\|_F^2} = 1 - \frac{\|(I - P_r)\Delta W^\top\|_F^2}{\|\Delta W\|_F^2} \geq 1 - \delta^2. \tag{4}$$

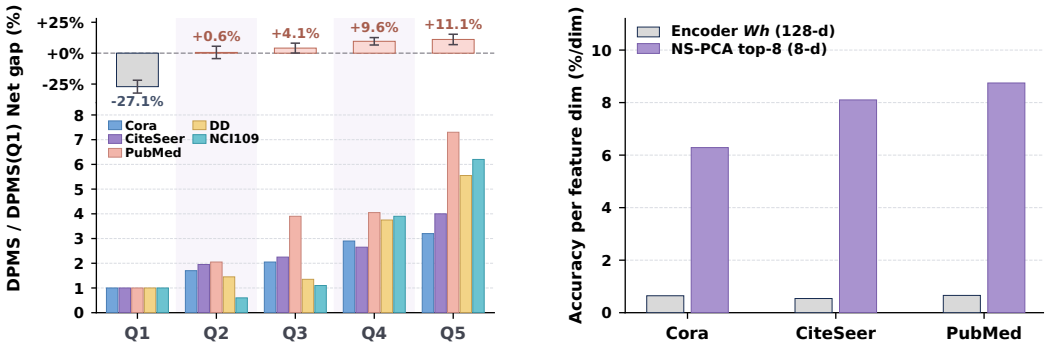

Figure 1: **Left: DPMS and Δ_q in the singular values of the quintices of W_{down}. Right: Information density of encoder and null space.**

The detailed proof is provided in the Appendix B. Proposition 4.1 reveals that when inputs are confined to a low-dimensional subspace, gradient updates restrict their energy to the same principal subspace. Crucially, this gradient-induced bias is compounded by GNN message passing $h^{(\ell)} = \phi(\hat{A}\, W^{(\ell)} h^{(\ell-1)})$, which acts as a graph-spectral filter. Over L layers, scaling frequency components by $\lambda_i^L(\hat{A})$ causes an *anisotropic contraction* of the covariance matrix. This asymptotically forces node representations to collapse toward a dominant vector, severely skewing the spectral distribution of the pre-trained weights.

Downstream Capacity and Misalignment. For a frozen encoder W, the downstream classifier is strictly confined to the *spectral reachable set* $\mathcal{H}(W) := \{h \mapsto a^\top (Wh) : a \in \mathbb{R}^{d_{\text{out}}}\}$. Since $\text{col}(W^\top) = \text{col}(V)$, the encoder restricts any downstream head to a k-dimensional subspace (where $k = \text{rank}(W)$). The minimal approximation error against the optimal downstream classifier W_{down} is:

$$\varepsilon^2(W) := \min_A \|W_{\text{down}} - W^\top A\|_F^2 = \|P_{\mathcal{N}}\, W_{\text{down}}\|_F^2, \tag{5}$$

which is achieved at $A^* = U\Sigma^{-1}V^\top W_{\text{down}}$, leaving an irreducible residual defined by the null space projection $P_{\mathcal{N}} = I - VV^\top$. We further find that applying a computationally inexpensive PCA to the null space leads to substantially higher information density than the Encoder. As shown in Figure 1, using only 8 dimensions, the resulting representation achieves classification performance comparable to that of a 128 dimensional Encoder obtained through costly training.

To explicitly quantify the tension between downstream demand and pre-training capacity, we introduce the *Downstream-Pretraining Misalignment Score* for each singular direction v_j. By partitioning the singular values into quintile groups $\{Q_q\}_{q=1}^5$, we further define the aggregate *net gap* (Δ_q):

$$\text{DPMS}_j := \frac{\|W_{\text{down}}^\top v_j\|^2}{\sigma_j^2}, \quad \Delta_q := \frac{\sum_{j\in Q_q} \|W_{\text{down}}^\top v_j\|^2}{\sum_j \|W_{\text{down}}^\top v_j\|^2} - \frac{\sum_{j\in Q_q} \sigma_j^2}{\sum_j \sigma_j^2}. \tag{6}$$

Theorem 4.2 (Spectral Reversal)**.** *The interaction between pre-trained capacity and downstream demand exhibits the following properties:*

- ***Reachability Limits:*** *Target signals within col(V) are fully recoverable via $a_j = \sigma_j^{-1} u_j$, whereas signals in the null space* $\ker(W)$ *yield an irreducible approximation error $\varepsilon^2(W) = \|P_{\mathcal{N}} W_{down}\|_F^2$.*
- ***Capacity Misalignment:*** DPMS_j *grows inversely with the pre-trained capacity σ_j^2. Consequently, severe spectral concentration, where $\sigma_i \gg \sigma_j$, forces the ratio* $\text{DPMS}_j/\text{DPMS}_i$ *to diverge, which is regionally quantified by the net gap Δ_q.*

The theorem above formalizes the fundamental tension between the pre-training and fine-tuning phases. The first property indicates that a frozen backbone inherently introduces *blind spots*: any downstream feature reliance that falls into the null space of the pre-trained weights simply cannot be

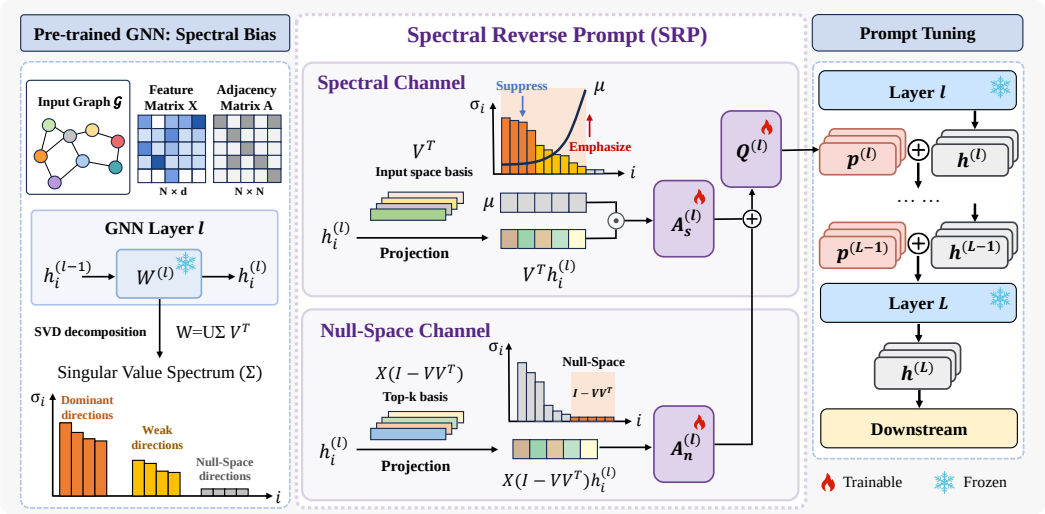

Figure 2: **Overall framework of Spectral Reverse Prompt (SRP).** SRP decomposes each frozen GNN layer weight to expose spectral bias, then uses a spectral channel to suppress dominant directions and amplify weak ones, and a null-space channel to exploit information that is not captured by the frozen encoder. The two channels are fused into layer-wise prompts injected into the frozen GNN for downstream tuning.

recovered, regardless of how the linear head is optimized. The second property further reveals that even within the reachable subspace, there is a severe mismatch in resource allocation. The metric DPMS_j acts as a demand-to-supply ratio. Due to spectral bias, the pre-trained model over-allocates its capacity to dominant directions, leading to a low DPMS and a negative net gap representing redundant supply. Conversely, the tail directions, which are often crucial for specific downstream tasks, are starved of capacity, resulting in a critically high DPMS and a positive net gap. This structural misalignment strongly motivates the need for targeted spectral adaptation.

Theorem 4.3 (PAC-Bayes Generalization Bound)**.** *Consider a learnable adaptor $\hat{\Theta}$ restricted to a bottleneck rank r_s across exactly $|\mathcal{V}_w|$ weak spectral directions, yielding a parameter count of $d_\Theta = \mathcal{O}(|\mathcal{V}_w| r_s)$. Let $\hat{R}(\hat{\Theta})$ and $R(\hat{\Theta})$ be the empirical and expected risks on n samples. Under McAllester's bound with a Gaussian prior bounded by the backbone's condition number $\kappa(W) := \sigma_{max}/\sigma_{min}$, and an analytically optimized posterior, with probability $\geq 1 - \delta$:*

$$R(\hat{\Theta}) \;\leq\; \hat{R}(\hat{\Theta}) + \mathcal{O}\left(\sqrt{\frac{|\mathcal{V}_w| r_s \log \kappa(W) + \log(n/\delta)}{n}}\right). \tag{7}$$

Proof Sketch and Motivation. By Theorem 4.2, the effective signal scale per target direction is restricted by $1/\sigma_{\min}$, which strictly bounds the parameter norm as $\|\hat{\Theta}\|^2 = \mathcal{O}(d_\Theta/\sigma_{\min}^2)$. Optimizing the posterior variance collapses the KL divergence to $\mathcal{O}(d_\Theta \log \kappa(W))$, yielding Equation (7).

Crucially, this bound demonstrates that the generalization gap depends explicitly on the number of activated weak directions $|\mathcal{V}_w|$, rather than the massive original feature dimensions $d_{\text{in}} d_{\text{out}}$. This offers a profound theoretical foundation: by deliberately restricting trainable capacity to only the highly misaligned subspace, one can rectify the spectral bias while inherently securing a significantly tighter generalization guarantee than full fine-tuning. This structural insight directly motivates our algorithmic design in the subsequent section.

5 Method: Spectral Reverse Prompt

Motivated by the spectral reversal principle, we design our method, Spectral Reverse Prompt (SRP). Our approach introduces a layer-wise graph prompt that counteracts singular values to better align

the pretrained models with downstream few-shot tasks. SRP adds prompting vector to node feature:

$$\tilde{z}^{(\ell)} = \hat{A}\, W^{(\ell)} h^{(\ell-1)} + b^{(\ell)} + p^{(\ell)}(h^{(\ell-1)}). \tag{8}$$

Each prompt consists of two complementary channels on orthogonal subspaces of $\mathbb{R}^{d_{\text{in}}}$: a *spectral channel* that reverses the encoder's spectral weighting, and a *null-space channel* that extracts signal from directions invisible to W.

5.1 Spectral Channel: Soft-Threshold Reversal

Motivated by Theorem 4.2, this channel amplifies weak directions while suppressing dominant ones. Node features are projected onto the right-singular basis: $h_{\text{spec}} = V^\top h \in \mathbb{R}^k$ (the spectral coordinates of h). A learnable soft-threshold mask is then applied:

$$\tau = \text{sigmoid}(\tau_0), \qquad \mu_j = \text{sigmoid}\big((\tau\sigma_{\text{max}} - \sigma_j)\cdot\gamma\big), \tag{9}$$

where τ_0 is a scalar threshold parameter, σ_{max} the largest singular value of W, γ is a pre-modulatable temperature coefficient that can be freely adjusted according to specific experiments, and $\mu_j \in (0,1)$ the mask weight for direction j. By construction $\mu_j \approx 0$ for strong directions and $\mu_j \approx 1$ for weak ones. The masked coordinates are linearly projected: $z_{\text{spec}} = (h_{\text{spec}} \odot \mu)\, A_s$, where $A_s \in \mathbb{R}^{r_s \times k}$ is a learnable bottleneck adaptor of rank r_s.

5.2 Null-Space Channel

When $d_{\text{in}} > k$, the null space $\ker(W)$ has dimension $d_{\text{in}} - k$. Features in this subspace are completely invisible to the frozen encoder and inaccessible to the spectral channel. We project the feature matrix $\mathbf{X}$ onto $\ker(W)$ and extract the top-m principal components:

$$\mathbf{X}_\perp = \mathbf{X}(I - VV^\top), \qquad z_{\text{null}} = A_n\,(V_\perp h), \tag{10}$$

where $V_\perp \in \mathbb{R}^{m \times d_{\text{in}}}$ is the orthonormal PCA basis of the null-space-projected data $\mathbf{X}_\perp$, and $A_n \in \mathbb{R}^{r_s \times m}$ is a learnable adaptor. By construction $WV_\perp = 0$ and $V^\top V_\perp = 0$: the two channels operate on orthogonal subspaces. Both $V_\perp$ and V are frozen; only A_n is learned. Figure 1 and proofs given by Appendix B confirm that NS-PCA directions carry higher per-dimension accuracy than the encoder output and that PCA ordering substantially outperforms random null-space selection.

5.3 Channel Fusion and Output

The two channels are summed in a shared bottleneck, then projected to the GNN activation space:

$$p(h) = Q\Big[\underbrace{A_s\,(V^T h \odot \mu)}_{\text{spectral channel}} + \underbrace{A_n\,(V_\perp h)}_{\text{null-space channel}}\Big] + b_p, \tag{11}$$

where Q and b_p are the shared projection and bias. This fusion is lossless since $V^\top V_\perp^\top = 0$. When $d_{\text{in}} \le k + m$, SRP uses a projective fallback $p(h) = (h_{\text{weak}}P)Q + b_p$. Here, $h_{\text{weak}} = h - (hV \odot (1-\mu))V^\top$ directly preserves all weak spectral and null-space signals. Thus, a single adaptor $P \in \mathbb{R}^{d_{\text{in}} \times r_s}$ becomes mathematically isomorphic to the dual-channel architecture (Appendix B.5).

5.4 Complexity Analysis

The proposed SRP exhibits exceptional efficiency, characterized by strictly linear time complexity with respect to the graph size and a trainable parameter footprint drastically smaller than standard LoRA-style adaptations.

Time Complexity. Initial SVD and null-space computations are performed once offline. During training, by pre-computing the heavy input projections, SRP's forward-pass overhead is reduced to $\mathcal{O}(|\mathcal{V}| r_s(k+m+h))$. Since $r_s \ll h$ and $k, m \ll d_{in}$, this linear cost is asymptotically negligible compared to the backbone's heavy message-passing complexity of $\mathcal{O}(|\mathcal{E}| d_{in})$.

Space Complexity. SRP introduces a minimal set of trainable parameters. For a standard n-layer network, combining the initial dual-channel layer and $n-1$ projective fallback layers yields exactly $r_s(k+m+(2n-1)h)+nh+n$ trainable parameters. This scales strictly as $\mathcal{O}(nr_s h)$, avoiding input-dimension (d_{in}) dependencies, making it far smaller than full fine-tuning or LoRA-style baselines. Additionally, frozen buffers $(V, V_\perp^\top)$ require only $\mathcal{O}(d_{in}(k+m) + nhk)$ memory, while activation memory remains $\mathcal{O}(|\mathcal{V}| d_{in})$, identical to the unmodified backbone.

6 Experiments

6.1 Experimental Setup

Datasets. We evaluate the effectiveness of SRP on ten benchmark datasets spanning both node- and graph-level classification tasks. For **node classification** (5-shot), we consider five widely used benchmarks, including the citation networks Cora [39], CiteSeer [39], and PubMed [39], as well as the large-scale datasets Flickr [40] and ogbn-arxiv [41]. For **graph classification** (50-shot), we use five molecular and biological graph datasets from TUDataset [42]: ENZYMES, DD, NCI1, NCI109, and Mutagenicity. These benchmarks exhibit substantial diversity in graph scale, topology, and feature dimensionality. Detailed dataset statistics are provided in Appendix A.

Pre-training strategies and Baselines. To evaluate the compatibility and generality of SRP across different self-supervised paradigms, we conduct experiments under two representative pre-training strategies: GraphCL [17] and SimGRACE [19]. We compare SRP against ten representative baselines, including: Classifier Only (a linear classifier trained on frozen features), GPPT [11] (link-prediction-based prompting), GraphPrompt [37] (readout-level prompting), All-in-one [43] (multi-task prompting), GPF and GPF-plus [8] (feature-level prompting), EdgePrompt and EdgePrompt-plus [9] (edge-level prompting), and GraphLoRA [32] (structure-aware low-rank adaptation). For fair comparison, all methods use the same backbone architecture, pre-trained checkpoints, and evaluation protocol following Prog [44].

Table 1: Performance on node-level benchmarks (5-seed avg.). **Bold** = best, underline = second best.

Pre-training	Tuning Methods	Cora	CiteSeer	PubMed	ogbn-arxiv	Flickr
GraphCL	Classifier Only	$53.05_{\pm 4.76}$	$38.62_{\pm 3.43}$	$64.28_{\pm 4.51}$	$21.15_{\pm 1.64}$	$24.32_{\pm 2.93}$
	GPPT	$50.96_{\pm 6.67}$	$39.50_{\pm 1.67}$	$60.47_{\pm 4.75}$	$17.99_{\pm 1.14}$	$24.35_{\pm 1.84}$
	GraphPrompt	$55.71_{\pm 4.62}$	$40.81_{\pm 2.11}$	$63.47_{\pm 2.23}$	$21.03_{\pm 1.92}$	$\underline{26.08_{\pm 3.44}}$
	ALL-in-one	$38.00_{\pm 4.17}$	$40.27_{\pm 2.09}$	$58.61_{\pm 3.49}$	$16.42_{\pm 2.98}$	$25.08_{\pm 3.44}$
	GPF	$58.52_{\pm 4.07}$	$43.55_{\pm 2.80}$	$67.67_{\pm 3.14}$	$21.73_{\pm 1.75}$	$23.98_{\pm 1.71}$
	GPF-plus	$52.24_{\pm 4.59}$	$38.47_{\pm 3.27}$	$64.30_{\pm 4.58}$	$21.03_{\pm 1.96}$	$25.32_{\pm 2.02}$
	EdgePrompt	$58.60_{\pm 4.46}$	$43.31_{\pm 3.23}$	$\underline{67.76_{\pm 3.01}}$	$21.90_{\pm 1.71}$	$24.83_{\pm 2.78}$
	EdgePrompt+	$62.88_{\pm 6.43}$	$\underline{46.20_{\pm 0.99}}$	$67.41_{\pm 5.25}$	$23.18_{\pm 1.26}$	$25.57_{\pm 3.04}$
	GraphLoRA	$\underline{63.35_{\pm 4.32}}$	$45.98_{\pm 3.97}$	$67.21_{\pm 3.15}$	$\underline{23.26_{\pm 1.89}}$	$25.71_{\pm 3.16}$
	SRP (Ours)	$\mathbf{68.11_{\pm 2.29}}$	$\mathbf{49.02_{\pm 2.26}}$	$\mathbf{68.81_{\pm 3.92}}$	$\mathbf{24.75_{\pm 1.79}}$	$\mathbf{26.39_{\pm 2.11}}$
SimGRACE	Classifier Only	$52.27_{\pm 2.74}$	$40.45_{\pm 3.55}$	$56.72_{\pm 3.80}$	$20.75_{\pm 2.92}$	$25.53_{\pm 3.98}$
	GPPT	$52.07_{\pm 7.65}$	$40.25_{\pm 3.29}$	$58.65_{\pm 5.12}$	$17.76_{\pm 1.80}$	$23.37_{\pm 4.66}$
	GraphPrompt	$51.42_{\pm 2.80}$	$41.74_{\pm 2.22}$	$55.98_{\pm 2.94}$	$20.48_{\pm 2.57}$	$25.88_{\pm 3.81}$
	ALL-in-one	$34.64_{\pm 4.06}$	$38.95_{\pm 2.35}$	$54.18_{\pm 4.70}$	$16.72_{\pm 2.90}$	$27.68_{\pm 4.58}$
	GPF	$58.23_{\pm 4.19}$	$44.87_{\pm 4.35}$	$61.55_{\pm 3.79}$	$21.86_{\pm 2.91}$	$26.51_{\pm 4.69}$
	GPF-plus	$52.27_{\pm 3.34}$	$41.02_{\pm 3.49}$	$56.95_{\pm 3.86}$	$21.44_{\pm 3.77}$	$28.35_{\pm 5.50}$
	EdgePrompt	$58.37_{\pm 4.51}$	$43.94_{\pm 4.15}$	$61.10_{\pm 3.69}$	$21.85_{\pm 2.54}$	$\mathbf{30.12_{\pm 5.04}}$
	EdgePrompt+	$62.40_{\pm 7.97}$	$\underline{46.62_{\pm 2.53}}$	$\underline{64.91_{\pm 5.58}}$	$22.74_{\pm 2.34}$	$\underline{28.50_{\pm 4.08}}$
	GraphLoRA	$\underline{63.06_{\pm 3.64}}$	$46.39_{\pm 6.15}$	$64.33_{\pm 5.23}$	$\underline{23.22_{\pm 2.97}}$	$28.02_{\pm 3.54}$
	SRP (Ours)	$\mathbf{67.24_{\pm 7.31}}$	$\mathbf{51.37_{\pm 5.24}}$	$\mathbf{67.32_{\pm 4.10}}$	$\mathbf{24.01_{\pm 2.91}}$	$28.48_{\pm 3.26}$

6.2 Main Results

We compare the node-level and graph-level classification accuracy of our methods and other baselines. Table 1 and Table 2 report results under GraphCL and SimGRACE pre-training. We make several key observations. First, SRP consistently achieves the best or second best performance across almost all dataset. Second, the gains are most pronounced on node classification benchmarks with high input dimension, where the null-space channel can extract substantial discriminative signal from the frozen encoder. Third, on graph classification benchmarks where the input dimension is small, SRP operates via its projective fallback (Section 5.3), yet remains superior to the strongest baselines, confirming that the spectral reversal mask provides meaningful gains. These results validate the central thesis of this work: inverting the spectral priority of frozen GNN weights is a broadly effective strategy for graph prompt tuning.

Table 2: Performance on graph-level benchmarks (5-seed avg.). **Bold** = best, underline = second best.

Pre-training	Tuning Methods	ENZYMES	DD	NCI1	NCI109	Mutagenicity
GraphCL	Classifier Only	$30.50_{\pm 1.16}$	$62.89_{\pm 2.19}$	$62.49_{\pm 1.95}$	$61.68_{\pm 0.93}$	$66.62_{\pm 1.87}$
	GraphPrompt	$27.83_{\pm 1.61}$	$64.33_{\pm 1.79}$	$63.19_{\pm 1.71}$	$62.18_{\pm 0.48}$	$\underline{67.62_{\pm 0.65}}$
	ALL-in-one	$25.92_{\pm 0.55}$	$66.54_{\pm 1.82}$	$57.52_{\pm 2.61}$	$62.74_{\pm 0.78}$	$63.43_{\pm 2.53}$
	GPF	$30.08_{\pm 1.25}$	$64.54_{\pm 2.22}$	$62.66_{\pm 1.83}$	$62.29_{\pm 0.90}$	$66.54_{\pm 1.85}$
	GPF-plus	$31.00_{\pm 1.50}$	$67.26_{\pm 2.29}$	$64.56_{\pm 1.10}$	$62.84_{\pm 0.22}$	$66.82_{\pm 1.63}$
	EdgePrompt	$29.50_{\pm 1.57}$	$64.16_{\pm 2.13}$	$63.05_{\pm 2.11}$	$62.59_{\pm 0.93}$	$66.87_{\pm 1.88}$
	EdgePrompt+	$34.00_{\pm 1.25}$	$67.98_{\pm 2.05}$	$66.30_{\pm 2.54}$	$\underline{66.52_{\pm 0.91}}$	$67.47_{\pm 2.37}$
	GraphLoRA	$\underline{36.05_{\pm 1.89}}$	$\underline{68.33_{\pm 1.93}}$	$\underline{66.54_{\pm 2.71}}$	$66.40_{\pm 0.96}$	$67.12_{\pm 2.11}$
	SRP (Ours)	$\mathbf{36.32_{\pm 2.65}}$	$\mathbf{69.71_{\pm 1.72}}$	$\mathbf{66.93_{\pm 1.81}}$	$\mathbf{68.18_{\pm 0.55}}$	$\mathbf{69.35_{\pm 1.80}}$
SimGRACE	Classifier Only	$27.07_{\pm 1.04}$	$61.77_{\pm 2.40}$	$61.27_{\pm 3.64}$	$62.12_{\pm 1.10}$	$67.36_{\pm 0.71}$
	GraphPrompt	$26.87_{\pm 1.47}$	$62.58_{\pm 1.84}$	$62.45_{\pm 1.52}$	$62.41_{\pm 0.69}$	$68.03_{\pm 0.78}$
	ALL-in-one	$25.73_{\pm 1.18}$	$65.16_{\pm 1.47}$	$58.52_{\pm 1.59}$	$62.01_{\pm 0.66}$	$64.43_{\pm 1.00}$
	GPF	$28.53_{\pm 1.76}$	$65.64_{\pm 0.70}$	$61.45_{\pm 3.13}$	$61.90_{\pm 1.26}$	$67.19_{\pm 0.74}$
	GPF-plus	$27.33_{\pm 2.01}$	$67.20_{\pm 1.56}$	$61.61_{\pm 2.89}$	$62.84_{\pm 0.23}$	$67.69_{\pm 0.64}$
	EdgePrompt	$29.33_{\pm 2.30}$	$63.97_{\pm 2.14}$	$62.02_{\pm 3.02}$	$62.02_{\pm 1.03}$	$67.55_{\pm 0.85}$
	EdgePrompt+	$32.67_{\pm 2.53}$	$\underline{67.72_{\pm 1.62}}$	$\mathbf{67.07_{\pm 1.96}}$	$\underline{66.53_{\pm 1.30}}$	$\underline{68.31_{\pm 1.36}}$
	GraphLoRA	$\underline{35.17_{\pm 1.42}}$	$67.02_{\pm 1.97}$	$\underline{66.83_{\pm 2.39}}$	$65.33_{\pm 1.04}$	$67.58_{\pm 1.67}$
	SRP (Ours)	$\mathbf{35.38_{\pm 1.88}}$	$\mathbf{68.12_{\pm 1.97}}$	$66.53_{\pm 2.94}$	$\mathbf{67.07_{\pm 0.83}}$	$\mathbf{68.92_{\pm 2.73}}$

Table 3: Ablation Study of Different Variants on GraphCL and SimGRACE.

Variant	GraphCL			SimGRACE		
	Cora	CiteSeer	Avg. Δ	Cora	CiteSeer	Avg. Δ
SRP (full)	$\mathbf{68.1_{\pm 2.3}}$	$\mathbf{49.0_{\pm 2.3}}$	-0.0	$\mathbf{67.2_{\pm 7.3}}$	$\mathbf{51.4_{\pm 5.2}}$	-0.0
SRP-NM (w/o null channel)	$63.4_{\pm 2.8}$	$41.1_{\pm 3.8}$	-6.3	$60.3_{\pm 5.9}$	$43.5_{\pm 2.9}$	-7.4
SRP-NS (w/o spectral channel)	$64.4_{\pm 2.1}$	$45.6_{\pm 3.2}$	-3.6	$63.5_{\pm 6.6}$	$47.7_{\pm 4.3}$	-3.7
SRP-NR (w/o reverse)	$64.7_{\pm 2.7}$	$48.0_{\pm 4.0}$	-2.2	$63.4_{\pm 4.4}$	$49.9_{\pm 4.9}$	-2.7
SRP-Bi (bidirectional)	$64.7_{\pm 3.3}$	$47.9_{\pm 3.3}$	-2.3	$63.0_{\pm 4.5}$	$50.0_{\pm 4.8}$	-2.8

6.3 Ablation Study

To disentangle the contributions of different components in SRP, we evaluate four ablation variants under both GraphCL and SimGRACE pre-training settings: **SRP-NM** removes the null-space channel, **SRP-NS** removes the spectral channel, **SRP-NR** disables spectral reversal by adopting uniform weighting across all singular directions ($\mu_j \equiv 1$), and **SRP-Bi** replaces the reverse weighting strategy with a bidirectional mask that amplifies both dominant and trailing spectral directions. The results in Table 3 reveal a consistent trend across both pre-training frameworks.

Effectiveness of the spectral and null-space channels. Both the spectral channel and the null-space channel play essential roles in extracting discriminative information for downstream adaptation. Removing either component leads to clear performance degradation. In particular, **SRP-NM** and **SRP-NS** consistently underperform the full SRP model across all datasets and pre-training settings, demonstrating that both weak spectral directions and null-space information contribute substantially to effective prompt learning.

Effectiveness of the spectral reversal principle. SRP improves adaptation by reversing the pre-trained spectral bias and emphasizing weak singular directions. When this mechanism is removed (**SRP-NR**) or replaced with bidirectional amplification (**SRP-Bi**), performance consistently declines relative to the full SRP model. These results suggest that selectively emphasizing trailing singular directions is more effective than uniformly weighting or jointly amplifying both spectral extremes for few-shot graph adaptation.

6.4 Additional Evaluation

Label sensitivity We analyzed the impact of the number of labels for each class during prompt tuning in downstream tasks, as shown in Figure 3. We found that the performance of node classification tasks fluctuates significantly with changes in the number of samples when the shot number is

extremely few. In contrast, graph classification tasks are less affected by the number of labels.

Parameter sensitivity. We study the sensitivity of SRP to its two key hyperparameters: the null-space PCA dimension m and the adaptor bottleneck rank r_s and numbers of labels. m controls how many low-amplification singular directions are used to construct the spectral reparameterization subspace, r_s determines the capacity of the lightweight adapter, shot number reflects the ability to fine tune the model with a limited number of labels. Figure 3 reports 5-shot node classification accuracy under GraphCL as we vary $r_s \in \{16, 32, 64, 128\}$ and $m \in \{4, 8, 16, 32\}$ independently. Performance remains stable across a wide range of values, with accuracy plateauing at the default settings (r_s=32, m=16), confirming that SRP is robust to the choice of these hyperparameters and does not require extensive tuning in practice.

Convergence Speed. We also investigate the convergence speed of our method compared with other strong baselines. Figure 4 illustrates the accuracy curves of our method and the baselines under two pre-training strategies. According to Figure 4, we can observe that SRP can generally converge faster than other methods, which means extra efficiency by adopting fewer epoch numbers in tuning the whole pre-trained models.

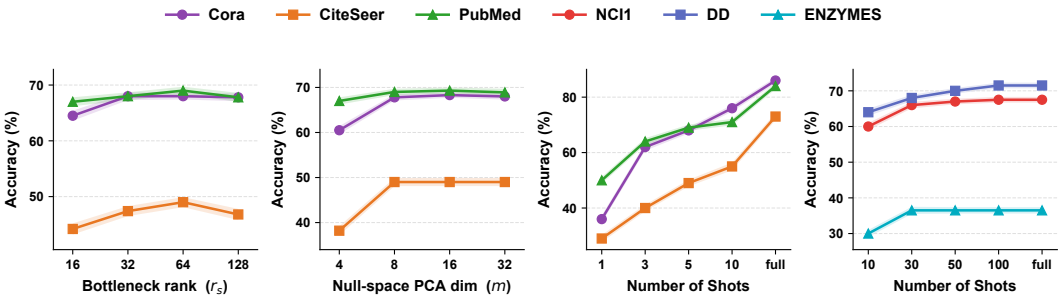

Figure 3: Label Sensitivity and Parameter Sensitivity of NS-PCA and bottleneck rank.

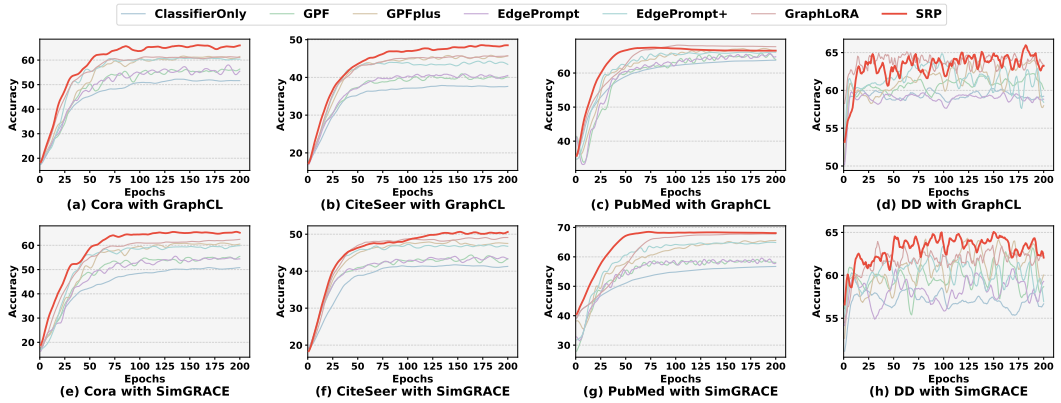

Figure 4: Convergence speed of different methods.

7 Conclusion

In this work, we identified *spectral bias* as a key bottleneck in adapting pre-trained GNNs, where self-supervised objectives overemphasize dominant singular directions while underutilizing lower-energy components. Using the DPMS, we empirically and theoretically showed that downstream utility is often concentrated in these underexplored spectral directions. To address this misalignment, we introduced SRP, a parameter-efficient framework that rebalances spectral contributions of frozen encoders via soft-thresholding and null-space augmentation. By reallocating capacity toward under-utilized components, SRP achieves state-of-the-art performance across diverse benchmarks while

using less than 7% of backbone parameters. Our results highlight a simple but important principle: in parameter-efficient adaptation, *which directions are adapted* can matter more than how many parameters are introduced.

Limitations and Future Works. While SRP remains robustness across different graph homophily levels and backbone architectures according to our additional results C, its spectral reweighting currently relies on a shared singular basis and a global threshold, which may not fully capture region-specific frequency heterogeneity. In addition, the effectiveness of the null-space channel depends on the relationship between input dimensionality and encoder rank, and can be limited when the frozen transformation has little or no null space. Future work includes instance- or structure-adaptive spectral reweighting driven by downstream feedback, as well as extending the spectral perspective to multi-modal graph representation learning.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (No. 62276053); Center for HPC, University of Electronic Science and Technology of China; Sichuan Province Science and Technology Support Program (No. 2025ZDZX0016).

References

[1] Xiangguo Sun, Jiawen Zhang, Xixi Wu, Hong Cheng, Yun Xiong, and Jia Li. Graph prompt learning: A comprehensive survey and beyond. *arXiv preprint arXiv:2311.16534*, 2023.

[2] Xingbo Fu, Zehong Wang, Zihan Chen, Jiazheng Li, Yaochen Zhu, Zhenyu Lei, Cong Shen, Yanfang Ye, Chuxu Zhang, and Jundong Li. Graph prompting for graph learning models: Recent advances and future directions. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6010–6020, New York, NY, USA, 2025. Association for Computing Machinery.

[3] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[4] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[5] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 2017.

[7] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022.

[8] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 2023.

[9] Xingbo Fu, Yinhan He, and Jundong Li. Edge prompt tuning for graph neural networks. In *International Conference on Learning Representations*, 2025.

[10] Jiazheng Li, Jundong Li, and Chuxu Zhang. Instance-aware graph prompt learning. *Transactions on Machine Learning Research*, 2025.

[11] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[12] Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Multigprompt for multi-task pre-training and prompting on graphs. In *Proceedings of the ACM on Web Conference 2024*, 2024.

[13] Xingtong Yu, Yuan Fang, Zemin Liu, and Xinming Zhang. Hgprompt: Bridging homogeneous and heterogeneous graphs for few-shot prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[14] Renhong Huang, Jiarong Xu, Xin Jiang, Chenglu Pan, Zhiming Yang, Chunping Wang, and Yang Yang. Measuring task similarity and its implication in fine-tuning graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[15] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

[16] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations*, 2020.

[17] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 2020.

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

[19] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, 2022.

[20] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR Workshop on Geometrical and Topological Representation Learning*, 2021.

[21] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. Motif-based graph self-supervised learning for molecular property prediction. *Advances in Neural Information Processing Systems*, 2021.

[22] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[23] Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM on Web Conference*, pages 737–746, 2023.

[24] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.

[25] Ruiyi Fang, Bingheng Li, Zhao Kang, Qiuhao Zeng, Nima Hosseini Dashtbayaz, Ruizhi Pu, Boyu Wang, and Charles Ling. On the benefits of attribute-driven graph domain adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *Advances in neural information processing systems*, 2024.

[27] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, et al. Multi-Domain Graph Foundation Models: Robust Knowledge Transfer via Topology Alignment. In *Forty-second International Conference on Machine Learning*, 2025.

[28] Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. How Much Can Transfer? BRIDGE: Bounded Multi-Domain Graph Foundation Model with Generalization Guarantees. In *Forty-second International Conference on Machine Learning*, 2025.

[29] Xiaodong He, Haolan He, Ruiyi Fang, Ming Sun, and Zhao Kang. Structure-centric graph foundation model via geometric bases. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

[30] Xiaodong He, Xincheng Wang, and Zhao Kang. Scgfm-art: Amortized relational transport for structure-centric graph foundation models. *arXiv preprint arXiv:2609.20419*, 2026.

[31] Xuanting Xie, Zhaochen Guo, Bingheng Li, Xingtong Yu, Zhifei Liao, Zhao Kang, and Yuan Fang. Clustering as reasoning: A k-means interpretation of chain-of-thought graph learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

[32] Zhe-Rui Yang, Jindong Han, Chang-Dong Wang, and Hao Liu. Graphlora: Structure-aware contrastive low-rank adaptation for cross-graph transfer learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1785–1796, 2025.

[33] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[34] Xingbo Fu, Zhenyu Lei, Zihan Chen, Binchi Zhang, Chuxu Zhang, and Jundong Li. GraphTOP: Graph topology-oriented prompting for graph neural networks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[35] Yongqi Huang, Jitao Zhao, Dongxiao He, Xiaobao Wang, Yawen Li, Yuxiao Huang, Di Jin, and Zhiyong Feng. One prompt fits all: Universal graph adaptation for pretrained models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[36] Quanxin Wang, Xuanting Xie, Bingheng Li, Xingtong Yu, Shuo Wang, Ruiyi Fang, and Zhao Kang. Dynamic graph prompting via topology-routed mixed-curvature experts. *arXiv preprint arXiv:2608.06031*, 2026.

[37] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. In *Proceedings of the ACM Web Conference 2023*, 2023.

[38] Zhen Tan, Ruocheng Guo, Kaize Ding, and Huan Liu. Virtual node tuning for few-shot node classification. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

[39] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 2016.

[40] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020.

[41] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 2020.

[42] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

[43] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

[44] Chenyi Zi, Haihong Zhao, Xiangguo Sun, Yiqing Lin, Hong Cheng, and Jia Li. Prog: A graph prompt learning benchmark. *arXiv preprint arXiv:2406.05346*, 2024.

[45] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

[46] Shuo Wang, Xiangyu Wang, Quanxin Wang, Bailin Wu, Bokui Wang, Shunyang Huang, Boyan Deng, Haonan Liu, Ruiyi Fang, Zhenxiang Xu, Boyu Wang, and Zhao Kang. The post-GCN decade revisited: Curvature-stratified evaluation of relational learning. *arXiv preprint arXiv:2606.06397*, 2026.

[47] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2013.

[48] David A. McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999.

[49] Bingheng Li, Erlin Pan, and Zhao Kang. PC-Conv: Unifying homophily and heterophily with two-fold filtering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

A Dataset Statistics

Tables 4 and 5 summarize the basic statistics of all ten benchmark datasets used in our main experiments and three benchmark in our extension to heterophilic graphs [45].

Table 4: Node-level benchmark datasets. #Feat. is the raw input feature dimension; #Classes is the number of downstream classes.

Dataset	#Nodes	#Edges	#Feat.	#Classes	Task
Cora	2,708	10,556	1,433	7	Node
CiteSeer	3,327	9,104	3,703	6	Node
PubMed	19,717	88,648	500	3	Node
ogbn-arxiv	169,343	1,166,243	128	40	Node
Flickr	89,250	899,756	500	7	Node
Cornell	183	298	1703	5	Node
Squirrel	5201	217073	2089	5	Node
Chameleon	2277	36101	2277	5	Node

Table 5: Graph-level benchmark datasets (TUDataset). Avg. Nodes and Avg. Edges are computed over the graph collection.

Dataset	#Graphs	Avg. Nodes	Avg. Edges	#Feat.	#Classes	Task
ENZYMES	600	32.63	124.27	3	6	Graph
DD	1,178	284.32	1,431.32	89	2	Graph
NCI1	4,110	29.87	64.60	37	2	Graph
NCI109	4,127	29.68	64.26	38	2	Graph
Mutagenicity	4,337	30.32	61.54	14	2	Graph

Our next objective is to investigate how variations in curvature [46], an important structural property in graph representation learning, may affect the performance of our method.

B Proofs of Theoretical Results

In this section, we provide the rigorous mathematical proofs for the theoretical claims presented in Section 4.

B.1 Proof of Proposition 4.1 (Spectral Concentration)

Statement: Given the rank-1 update $\Delta W^{(t)} = -\eta_t\, g^{(t)}(h^{(t)})^\top$ and the spectral alignment condition $\mathbb{E}_t\big[\|(I - P_r)h^{(t)}\|^2/\|h^{(t)}\|^2\big] \le \delta^2$, the cumulative gradient update satisfies:

$$\frac{\mathbb{E}\big[\|P_r\,\Delta W^\top\|_F^2\big]}{\mathbb{E}\big[\|\Delta W\|_F^2\big]} \;\ge\; 1 - \delta^2.$$

Proof. Consider the transpose of the gradient update at a single step t:

$$(\Delta W^{(t)})^\top = -\eta_t\, h^{(t)}(g^{(t)})^\top. \tag{12}$$

We can decompose the input vector $h^{(t)}$ into two orthogonal components using the projection matrix P_r and its orthogonal complement $(I - P_r)$:

$$h^{(t)} = P_r h^{(t)} + (I - P_r)h^{(t)}. \tag{13}$$

The Frobenius norm squared of the update transposed is determined by the outer product of these vectors. Utilizing the property $\|ab^\top\|_F^2 = \|a\|_2^2\|b\|_2^2$, we have:

$$\|(\Delta W^{(t)})^\top\|_F^2 = \eta_t^2\|h^{(t)}\|_2^2\|g^{(t)}\|_2^2. \tag{14}$$

Similarly, the energy of the update residing outside the top-r principal subspace is:

$$\|(I - P_r)(\Delta W^{(t)})^\top\|_F^2 = \eta_t^2\|(I - P_r)h^{(t)}\|_2^2\|g^{(t)}\|_2^2. \tag{15}$$

Assuming that the magnitude of the upstream gradient $\|g^{(t)}\|_2^2$ is independent of the normalized directional projection of $h^{(t)}$ over the training trajectory, we can take the expectation over t:

$$\mathbb{E}\big[\|(I - P_r)\Delta W^\top\|_F^2\big] = \mathbb{E}\left[\frac{\|(I - P_r)h^{(t)}\|_2^2}{\|h^{(t)}\|_2^2} \cdot \eta_t^2 \|h^{(t)}\|_2^2 \|g^{(t)}\|_2^2\right]. \tag{16}$$

Given the spectral alignment condition $\mathbb{E}_t\big[\|(I - P_r)h^{(t)}\|^2 / \|h^{(t)}\|^2\big] \leq \delta^2$, it directly follows that:

$$\mathbb{E}\big[\|(I - P_r)\Delta W^\top\|_F^2\big] \leq \delta^2\, \mathbb{E}\big[\|\Delta W^\top\|_F^2\big] = \delta^2\, \mathbb{E}\big[\|\Delta W\|_F^2\big]. \tag{17}$$

By the Pythagorean theorem for the Frobenius norm, $\|P_r \Delta W^\top\|_F^2 + \|(I - P_r)\Delta W^\top\|_F^2 = \|\Delta W\|_F^2$. Substituting the upper bound yields the final concentration result:

$$\mathbb{E}\big[\|P_r\, \Delta W^\top\|_F^2\big] \geq (1 - \delta^2)\, \mathbb{E}\big[\|\Delta W\|_F^2\big], \tag{18}$$

which concludes the proof.

B.2 Proof of Theorem 4.2 (Reachability and Approximation Limits)

Statement: For a frozen encoder $W = U\Sigma V^\top$ and an optimal downstream classifier W_{down}, the minimum approximation error using any linear head A is $\varepsilon^2(W) = \|P_{\mathcal{N}} W_{\text{down}}\|_F^2$, achieved at $A^* = U\Sigma^{-1}V^\top W_{\text{down}}$.

Proof. The objective is to minimize the approximation error $\mathcal{L}(A) = \|W_{\text{down}} - W^\top A\|_F^2$ over all possible downstream projection heads $A \in \mathbb{R}^{d_{\text{out}} \times C}$. Substituting the compact SVD of the transposed encoder $W^\top = V\Sigma U^\top$ into the objective yields:

$$\mathcal{L}(A) = \|W_{\text{down}} - V\Sigma U^\top A\|_F^2. \tag{19}$$

Since the columns of V are orthonormal, $VV^\top$ forms the orthogonal projection onto $\text{col}(V)$, and $P_{\mathcal{N}} = I - VV^\top$ is the projection onto the null space $\ker(W)$. We can orthogonally decompose the target matrix W_{down} as:

$$W_{\text{down}} = VV^\top W_{\text{down}} + P_{\mathcal{N}} W_{\text{down}}. \tag{20}$$

Because the term $V\Sigma U^\top A$ lies entirely within the column space $\text{col}(V)$, it is orthogonal to $P_{\mathcal{N}} W_{\text{down}}$. By applying the Pythagorean theorem for the Frobenius norm, we can decouple the objective function:

$$\mathcal{L}(A) = \|VV^\top W_{\text{down}} - V\Sigma U^\top A\|_F^2 + \|P_{\mathcal{N}} W_{\text{down}}\|_F^2. \tag{21}$$

The second term, $\|P_{\mathcal{N}} W_{\text{down}}\|_F^2$, is entirely independent of A and represents the irreducible error (the fundamental blind spot). To minimize $\mathcal{L}(A)$, we must force the first term to zero:

$$VV^\top W_{\text{down}} = V\Sigma U^\top A. \tag{22}$$

Left-multiplying both sides by $V^\top$ (noting that $V^\top V = I$) gives:

$$V^\top W_{\text{down}} = \Sigma U^\top A. \tag{23}$$

Since Σ is a diagonal matrix of strictly positive singular values, it is invertible. We can explicitly solve for A by left-multiplying by $U\Sigma^{-1}$:

$$A^* = U\Sigma^{-1}V^\top W_{\text{down}}. \tag{24}$$

Substituting A^* back into $\mathcal{L}(A)$ eliminates the first term, formally proving that the minimum reachable error is exactly:

$$\varepsilon^2(W) = \min_A \mathcal{L}(A) = \|P_{\mathcal{N}} W_{\text{down}}\|_F^2. \tag{25}$$

This establishes both the optimal projection head structure and the irreducible nature of downstream targets residing in the null space.

B.3 Null-Space PCA Optimality (Proof of Theorem B.1)

Theorem B.1 (Optimality of Null-Space PCA)**.** *Let $\Sigma_X = \mathbb{E}[xx^\top]$ be the input feature covariance and $P_{\mathcal{N}} = I - VV^\top$ the orthogonal projector onto* $\ker(W)$*. Among all orthonormal matrices $N_o \in \mathbb{R}^{d_{in} \times m}$ satisfying $WN_o = 0$, the null-space PCA basis uniquely solves:*

$$N_o^* = \arg\max_{N_o} \left\{ \mathrm{tr}\big(N_o^\top \Sigma_X N_o\big) \;\middle|\; N_o^\top N_o = I_m,\ WN_o = 0 \right\}. \tag{26}$$

Standard PCA (unconstrained) is generically infeasible for (26) *and therefore suboptimal.*

Proof. The constraint $WN_o = 0$ is equivalent to $V^\top N_o = 0$ (since $\ker(W) = \ker(V^\top)$). Writing $N_o = P_{\mathcal{N}} C$ for an arbitrary matrix C, the constraint $V^\top N_o = 0$ is automatically satisfied, and (26) reduces to:

$$\max_{C^\top P_{\mathcal{N}} C = I_m} \mathrm{tr}(C^\top P_{\mathcal{N}} \Sigma_X P_{\mathcal{N}} C).$$

By the Ky Fan maximum theorem [47], the maximum equals $\sum_{j=1}^{m} \lambda_j(P_{\mathcal{N}} \Sigma_X P_{\mathcal{N}})$, attained uniquely by the top-m eigenvectors of $P_{\mathcal{N}} \Sigma_X P_{\mathcal{N}}$—the null-space PCA basis. An unconstrained PCA solution lies generically outside $\ker(W)$ (a measure-zero coincidence would be needed for it to satisfy $V^\top N_o = 0$), so it is infeasible and thus suboptimal. □

B.4 PAC-Bayes Generalization Bound (Proof of Theorem B.2)

Theorem B.2 (PAC-Bayes Generalization Bound)**.** *Let Θ denote SRP's learnable parameters with $d_\Theta = |\mathcal{V}_w| r_s + m r_s + r_s d_{out} + d_{out} + 1$. Let $\hat{R}(\Theta)$ be the empirical risk on n labeled examples, $R(\Theta)$ the expected risk, and $\delta \in (0, 1)$. For any prior π and posterior ρ concentrated at $\hat{\Theta}$, with probability $\geq 1 - \delta$:*

$$R(\hat{\Theta}) \;\leq\; \hat{R}(\hat{\Theta}) + \sqrt{\frac{\mathrm{KL}(\rho \| \pi) + \log(2\sqrt{n}/\delta)}{2n}}. \tag{27}$$

Setting $\pi = \mathcal{N}(0, \kappa^2 I_{d_\Theta})$ and $\rho = \mathcal{N}(\hat{\Theta}, \hat{\sigma}^2 I_{d_\Theta})$ with $\kappa = \kappa(W) := \sigma_{max}/\sigma_{min}$, one obtains:

$$R(\hat{\Theta}) \;\leq\; \hat{R}(\hat{\Theta}) + \sqrt{\frac{|\mathcal{V}_w| r_s \log \kappa(W) + \log(2n/\delta)}{2n}}, \tag{28}$$

where $d_\Theta = O(|\mathcal{V}_w| r_s)$ is the dominant parameter count (spectral adaptor $A \in \mathbb{R}^{|\mathcal{V}_w| \times r_s}$).

Proof. Equation (27) is McAllester's PAC-Bayes bound [48]. For the chosen Gaussian prior and posterior, the KL divergence evaluates to:

$$\mathrm{KL}(\rho \| \pi) = \frac{1}{2} \left[d_\Theta \log \frac{\kappa^2}{\hat{\sigma}^2} - d_\Theta + \frac{d_\Theta \hat{\sigma}^2 + \|\hat{\Theta}\|^2}{\kappa^2} \right]. \tag{29}$$

The effective signal scale per direction satisfies $\mu_j / \sigma_j \leq \mu_j / \sigma_{\min} \leq 1/\sigma_{\min}$, so $\|\hat{\Theta}\|^2 = O(d_\Theta / \sigma_{\min}^2)$. Setting $\kappa = \sigma_{\max}/\sigma_{\min}$ and optimizing $\hat{\sigma}^2$ yields $\mathrm{KL}(\rho \| \pi) = O(d_\Theta \log \kappa(W))$. Substituting into (27) with $d_\Theta \approx |\mathcal{V}_w| r_s$ gives (28).

Complexity interpretation. The bound depends on $|\mathcal{V}_w|$ (number of under-served directions targeted by SRP) rather than on k (total singular directions) or $d_{\text{in}} d_{\text{out}}$ (full-weight dimension). Since DPMS-guided prompting restricts capacity to the weak subspace, the effective model complexity is $|\mathcal{V}_w| r_s \ll k r_s \ll d_{\text{in}} d_{\text{out}}$, directly linking the DPMS-based design philosophy to a tighter generalization bound. □

B.5 Algebraic Equivalence of the Projective Fallback

In our SRP architecture, we introduce a projective fallback path when the input dimension is small, specifically when $d_{\text{in}} \leq k + m$. In this section, we rigorously prove that this fallback is not a heuristic deviation, but rather an exact algebraic simplification of the dual-channel architecture.

Theorem B.3 (Algebraic Isomorphism of the Fallback)**.** *Let the pre-trained weight have rank k, and let $V \in \mathbb{R}^{d_{in} \times k}$ be its right singular vectors. Let $\mu \in \mathbb{R}^k$ denote the soft-threshold mask. When $d_{in} - k \leq m$, the projective fallback parameterized by a single adaptor $P \in \mathbb{R}^{d_{in} \times r_s}$ acting on h_{weak} spans the exact same hypothesis space as the dual-channel prompt parameterized by $A_s \in \mathbb{R}^{k \times r_s}$ and $A_n \in \mathbb{R}^{(d_{in}-k) \times r_s}$.*

Proof. First, we analyze the structure of the dual-channel prompt. The hidden representation h can be orthogonally decomposed into the reachable subspace $h_{\parallel} = hVV^\top$ and the exact null space $h_{\perp} = h(I - VV^\top) = hP_{\mathcal{N}}$. In the dual-channel mode, the target response Z_{dual} is the sum of the masked spectral channel and the null-space channel:

$$Z_{\text{dual}} = (hV\text{diag}(\mu))A_s + (hV_\perp^\top)A_n, \tag{30}$$

where $V_\perp^\top$ acts as the PCA basis matrix of the null space. Crucially, when $d_{\text{in}} - k \leq m$, the required PCA dimension m exceeds or equals the dimension of the entire null space. Thus, the columns of $V_\perp^\top$ simply form a complete orthonormal basis for $P_{\mathcal{N}}$, meaning $V_\perp^\top V_\perp = P_{\mathcal{N}}$.

Next, we analyze the input to the projective fallback, defined as $h_{\text{weak}} = h - (hV\text{diag}(1-\mu))V^\top$. We can expand h using the orthogonal decomposition $h = hVV^\top + hP_{\mathcal{N}}$:

$$\begin{aligned} h_{\text{weak}} &= (hVV^\top + hP_{\mathcal{N}}) - hV\text{diag}(1-\mu)V^\top \\ &= hV\big(I - \text{diag}(1-\mu)\big)V^\top + hP_{\mathcal{N}} \\ &= hV\text{diag}(\mu)V^\top + hP_{\mathcal{N}}. \end{aligned} \tag{31}$$

Equation (31) reveals a critical property: h_{weak} flawlessly preserves both the masked weak spectral signals and the entirety of the null-space signals, seamlessly superimposing them in the original ambient space $\mathbb{R}^{d_{\text{in}}}$.

Applying the single linear adaptor P to h_{weak} yields:

$$Z_{\text{fallback}} = h_{\text{weak}}P = \big(hV\text{diag}(\mu)\big)(V^\top P) + (hP_{\mathcal{N}})P. \tag{32}$$

To prove the isomorphism, we must show that there exists a matrix P such that $Z_{\text{fallback}} = Z_{\text{dual}}$ for any arbitrary A_s and A_n. By comparing Eq. (32) with Eq. (30), and using the identity $hP_{\mathcal{N}}P = hV_\perp^\top(V_\perp P)$, this equivalence requires:

$$V^\top P = A_s \quad \text{and} \quad V_\perp P = A_n. \tag{33}$$

Since V and $V_\perp^\top$ span mutually orthogonal complementary subspaces, the concatenated matrix $[V, V_\perp^\top] \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ is an orthogonal square matrix, and thus strictly invertible. Consequently, we can uniquely solve for P:

$$P = VA_s + V_\perp^\top A_n. \tag{34}$$

Equation (34) demonstrates that optimizing the single matrix P over h_{weak} is mathematically identical to independently optimizing A_s and B_n in their respective sub-manifolds. Therefore, when $d_{\text{in}} \leq k + m$, the projective fallback explicitly circumvents the redundant PCA calculations while strictly maintaining the theoretical capacity and spectral-reversal philosophy of SRP. □

C Additional Experiments and Analysis

C.1 Generalization to GraphGPS and Graphormer

To evaluate whether SRP extends beyond conventional message-passing GNNs, we replace the default backbone with GraphGPS and Graphormer and evaluate on two node-level datasets (Cora and CiteSeer) and two graph-level datasets (NCI1 and ENZYMES). We follow the same pre-training objective, data split, few-shot setting, training budget, and evaluation protocol as in the main experiments. Classifier-only, GraphLoRA, and SRP use the same frozen encoder and downstream splits. We retain the default SRP configuration with adaptor dimension $r = 32$ and PCA dimension $m = 16$. Results are averaged over five random seeds.

SRP improves over Classifier-only in all eight settings, with an average absolute gain of 12.48 percentage points. It achieves the best result in six settings and remains within 0.82 percentage points of GraphLoRA in the other two. These results indicate that spectral rebalancing is not specific to standard message-passing GNNs and can also be applied to hybrid local–global and attention-based graph encoders.

Table 6: Performance of SRP with GraphGPS and Graphormer backbones. Results are reported as mean $\pm$ standard deviation over five runs.

Backbone	Dataset	Classifier-only	GraphLoRA	SRP
GraphGPS	Cora	48.17 ± 6.12	64.33 ± 6.88	$\mathbf{66.62 \pm 5.27}$
GraphGPS	CiteSeer	39.13 ± 4.57	46.80 ± 3.38	$\mathbf{58.74 \pm 5.01}$
GraphGPS	NCI1	63.32 ± 1.51	62.49 ± 1.93	$\mathbf{66.81 \pm 2.13}$
GraphGPS	ENZYMES	35.08 ± 2.82	45.00 ± 3.25	$\mathbf{47.75 \pm 2.71}$
Graphormer	Cora	51.13 ± 4.04	62.88 ± 5.49	$\mathbf{65.69 \pm 5.97}$
Graphormer	CiteSeer	39.01 ± 1.74	47.13 ± 1.72	$\mathbf{54.89 \pm 1.60}$
Graphormer	NCI1	56.75 ± 2.76	$\mathbf{60.54 \pm 1.03}$	59.72 ± 2.75
Graphormer	ENZYMES	26.75 ± 2.41	$\mathbf{39.67 \pm 2.49}$	38.92 ± 2.13

C.2 Role and Sensitivity of the Temperature Coefficient

Recall that the spectral mask is defined as

$$\mu_j = \mathrm{sigmoid}((\tau\sigma_{\max} - \sigma_j)\gamma)\,, \tag{35}$$

where τ determines the spectral cutoff and γ controls the sharpness of the transition around it. The interval $\mu_j \in [0.1, 0.9]$ corresponds to a transition width of approximately $4.39/\gamma$ in singular-value units. Thus, γ controls the smoothness of the partition without changing the cutoff determined by τ.

We evaluate γ from an extremely soft mask to an approximately discrete partition. Results are reported as mean accuracy over five seeds.

Table 7: Sensitivity to the temperature coefficient γ.

Mask setting	Cora	CiteSeer	NCI1	ENZYMES
$\gamma = 0.001$	64.55	47.64	66.63	36.42
$\gamma = 0.1$	64.88	48.03	66.52	**36.92**
$\gamma = 1$	67.80	48.81	**66.96**	36.50
$\gamma = 10$	**68.10**	**48.99**	66.80	36.67
$\gamma = 100$	68.06	48.87	66.84	36.25
Hard threshold (STE)	67.95	48.90	66.89	35.67

At $\gamma = 0.001$, the mask is nearly constant around 0.5, with entropy close to one bit, and therefore provides little distinction between strong and weak directions. Increasing γ to 10 improves accuracy by 3.55 points on Cora and 1.35 points on CiteSeer. For $\gamma \geq 1$, performance becomes stable. At $\gamma = 10$, the mask entropy decreases to 0.004–0.064 bits, and only 0.3%–3.6% of directions remain in the transition region. Further sharpening therefore affects only a small number of boundary directions. We use $\gamma = 10$ for all datasets without dataset-specific tuning.

To separate spectral selection from generic dimensionality reduction, we further construct masks with exactly the same cardinality as the learned $\gamma = 10$ mask, while selecting the strongest, weakest, or random singular directions.

Table 8: Matched-cardinality comparison of different spectral selections.

Dataset	Strong-k	Random-k	Weak-k	Hard (STE)	Soft $\gamma = 10$
Cora	64.25	64.44	67.80	67.95	**68.10**
CiteSeer	47.10	47.46	48.78	48.90	**48.99**
NCI1	66.42	66.45	66.87	**66.89**	66.80

Weak-k consistently outperforms Strong-k and Random-k, particularly on Cora and CiteSeer, showing that the gain is not explained by dimensionality reduction alone. The similar performance of the soft and hard masks is also expected when few singular values lie near the cutoff. The sigmoid mask provides a differentiable relaxation during optimization while becoming nearly discrete after convergence.

C.3 Extension to Heterophilic Graphs

Although the main benchmarks are predominantly homophilic, the SRP architecture is not restricted to homophilic graphs. The connection can be understood from the spectral filtering induced by message passing. Consider

$$Z = \widehat{A}W^\top H, \qquad W = U\Sigma V^\top, \tag{36}$$

and let the normalized graph Laplacian be

$$L = I - \widehat{A}_{\mathrm{sym}} = \Phi\Lambda\Phi^\top. \tag{37}$$

For a graph signal s, its graph frequency can be measured by its normalized Dirichlet energy,

$$\mathrm{freq}(s) = \frac{s^\top L s}{s^\top s}. \tag{38}$$

Since $\widehat{A}_{\mathrm{sym}}\phi_i = (1-\lambda_i)\phi_i$, repeated propagation scales the i-th graph Fourier component by $(1-\lambda_i)^K$ after K propagation steps. Standard message passing therefore favors low-frequency components while attenuating high-frequency variation.

During pre-training, the features used to update the weights have already passed through this propagation operator. Consequently, the feature covariance observed by optimization is biased toward graph-frequency components preserved by message passing. This provides a graph-specific connection between the graph spectrum and the singular spectrum analyzed in our method.

SRP introduces its prompt after neighborhood aggregation:

$$H^{(\ell+1)} = \phi\Big(\widehat{A}W^{(\ell)\top}H^{(\ell)} + p^{(\ell)}(H^{(\ell)})\Big). \tag{39}$$

The prompt $p^{(\ell)}$ is therefore not filtered again by the same $\widehat{A}$ in the current layer, providing a complementary pathway for node-local and high-frequency information suppressed by the frozen message-passing branch.

We further evaluate SRP on three heterophilic datasets using GraphCL pre-training and the same hyperparameter configuration as in the main experiments. Across both homophilic and heterophilic graph data, prior approaches typically rely on carefully designed filters to balance the learned graph representations [49], whereas our method demonstrates robustness to varying levels of heterophily without requiring such specialized designs.

Table 9: Performance on heterophilic graph benchmarks.

Dataset	Classifier-only	GraphLoRA	SRP	SRP vs. GraphLoRA
Cornell	32.22 ± 8.35	45.56 ± 14.34	$\mathbf{51.67 \pm 5.72}$	$+6.11$
Squirrel	23.08 ± 1.98	23.06 ± 1.40	$\mathbf{25.12 \pm 1.47}$	$+2.06$
Chameleon	31.74 ± 1.88	31.43 ± 1.44	$\mathbf{33.58 \pm 2.76}$	$+2.15$

SRP outperforms GraphLoRA on all three datasets. The improvement is particularly clear on Squirrel and Chameleon, where GraphLoRA falls below the Classifier-only baseline while SRP remains above it. These results support the applicability of SRP beyond homophilic settings.

C.4 Scalability and Runtime

The number of trainable SRP parameters is independent of the number of graph nodes and edges, while the dominant computation remains the frozen GNN backbone. Thus, SRP remains applicable to large graphs. In our experiments, including ogbn-arxiv with nearly $170{,}000$ nodes, all runs were conducted on a single NVIDIA RTX 3090 GPU with 24 GB memory.

We additionally compare runtime against GPF, which has the lowest empirical per-epoch runtime among the prompt-learning baselines in our main experiments.

SRP is faster than GPF on five of the six evaluated datasets, indicating that its spectral operations do not introduce substantial training-time overhead.

Table 10: Per-epoch runtime in seconds. Lower is better.

Dataset	GPF	SRP
Cora	0.121	**0.077**
CiteSeer	0.131	**0.084**
PubMed	0.678	**0.508**
ENZYMES	**0.221**	0.292
DD	0.191	**0.136**
NCI1	0.342	**0.297**

C.5 Difference from GraphLoRA

GraphLoRA and SRP both introduce low-dimensional trainable components around a frozen GNN, but they address different adaptation objectives and construct different adaptation subspaces. GraphLoRA focuses on feature and structural distribution shifts between source and target graphs through a structure-aware low-rank adaptation branch. Its low-rank correction remains within the message-passing pathway:

$$\Delta Z^{(\ell)}_{\mathrm{GraphLoRA}} = \widehat{A}\,\Delta W^{(\ell)\top} H^{(\ell)}. \tag{40}$$

SRP instead targets the spectral capacity allocation of the frozen transformation. For

$$W^{(\ell)} = U^{(\ell)}\Sigma^{(\ell)}V^{(\ell)\top}, \tag{41}$$

it explicitly constructs prompts from two complementary subspaces. The spectral channel can be written as

$$P_{\mathrm{spec}} = \big(HV \odot \mu(\Sigma)\big)A_s Q, \qquad \mu_j = \mathrm{sigmoid}\big((\tau\sigma_{\max} - \sigma_j)\gamma\big), \tag{42}$$

while the null-space channel is

$$P_{\mathrm{null}} = HNA_n Q, \qquad \mathrm{col}(N) \subseteq \ker(W). \tag{43}$$

The resulting layer is

$$H^{(\ell+1)} = \phi\Big(\widehat{A}W^{(\ell)\top}H^{(\ell)} + P_{\mathrm{spec}} + P_{\mathrm{null}}\Big). \tag{44}$$

Hence, SRP does not merely impose a low-rank parameter budget: it explicitly selects where adaptation is concentrated according to the singular geometry of the frozen transformation. GraphLoRA learns a flexible low-rank correction from graph-structural objectives, whereas SRP directs adaptation toward weak singular and null-space directions.

The position of the adaptation branch also differs. If

$$H\Delta W^{\top} = \sum_i \phi_i c_i^{\top}, \tag{45}$$

then a GraphLoRA-style correction inside message passing becomes

$$\widehat{A}\Delta W^{\top}H = \sum_i (1-\lambda_i)\phi_i c_i^{\top}, \tag{46}$$

and is therefore filtered by the propagation operator. In contrast,

$$Z_{\mathrm{SRP}} = \widehat{A}W^{\top}H + p(H), \tag{47}$$

injects $p(H)$ after aggregation, so its graph Fourier components are not multiplied again by $(1-\lambda_i)$ within the same layer. This gives SRP a complementary pathway for information suppressed by the frozen message-passing branch.

C.6 Difference from Standard LoRA

Standard LoRA and SRP are both parameter-efficient adaptation methods, but differ in their adaptation target, trainable-subspace geometry, and interaction with graph propagation. LoRA freezes W and learns an additive low-rank update

$$W_{\mathrm{eff}} = W + \Delta W, \qquad \Delta W = BA, \qquad \mathrm{rank}(\Delta W) \le r. \tag{48}$$

For a message-passing GNN layer,

$$Z_{\mathrm{LoRA}} = \widehat{A}(W + BA)^\top H = \widehat{A}W^\top H + \widehat{A}A^\top B^\top H. \tag{49}$$

Thus, although the pretrained parameters remain frozen, LoRA changes the effective linear operator and parameterizes a low-rank subspace in weight-update space.

SRP instead leaves the original message-passing operator unchanged and learns a representation-space prompt:

$$Z_{\mathrm{SRP}} = \widehat{A}W^\top H + p(H), \tag{50}$$

where

$$p(H) = \big[(HV \odot \mu(\Sigma))A_s + (HN)A_n\big]Q + b, \qquad \mathrm{col}(N) \subseteq \ker(W). \tag{51}$$

The two channels therefore do not factorize a weight update; rather, they determine which information from the hidden representation is injected through the prompt.

Another distinction lies in the adaptation subspace. The low-rank constraint in LoRA controls the dimensionality of the update but does not explicitly determine how its energy is distributed across the singular directions of the frozen weight matrix. SRP instead derives its prompt-reading directions from the singular geometry of W and redirects prompt capacity toward weak singular directions and $\ker(W)$.

This distinction is particularly relevant for GNNs. Let

$$L = I - \widehat{A} = \Phi\Lambda\Phi^\top. \tag{52}$$

If $A^\top B^\top H = \Phi C$, the LoRA correction becomes

$$\Delta Z_{\mathrm{LoRA}} = \Phi(I - \Lambda)C, \tag{53}$$

and is therefore filtered by the same graph propagation operator. By contrast, the SRP prompt is added after aggregation and is not multiplied again by $I - \Lambda$ in the current layer. In summary, LoRA provides generic low-rank operator adaptation, whereas SRP preserves the frozen operator and explicitly targets weak singular and null-space directions through a post-aggregation prompt.

C.7 Clarification of the Singular-Value Groups in Figure 1

Let

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k > 0 \tag{54}$$

denote the nonzero singular values of a frozen layer W, sorted in descending order. We define the five singular-value groups by

$$b_q = \left\lfloor \frac{qk}{5} \right\rfloor, \qquad Q_q = \{b_{q-1} + 1, \ldots, b_q\}, \qquad q = 1, \ldots, 5, \tag{55}$$

with $b_0 = 0$. Thus, Q_1 contains the directions with the largest singular values and Q_5 contains the weakest nonzero singular directions; group sizes differ by at most one. Exact zero-singular-value directions are treated separately by the null-space channel.

The quintile partition depends only on the spectrum of the frozen weight W. The downstream classifier W_{down} enters only through the downstream-demand term in DPMS and does not affect the group construction.

C.8 Why SRP Shows Larger Gains on Node-Level Tasks

The difference between node- and graph-level performance can be attributed mainly to input dimensionality, null-space capacity, and graph-level readout.

Larger null-space capacity. For a first-layer transformation

$$W \in \mathbb{R}^{h \times d_{\mathrm{in}}}, \qquad W = U\Sigma V^\top, \tag{56}$$

if $\mathrm{rank}(W) = k$, then

$$\dim \ker(W) = d_{\mathrm{in}} - k \geq d_{\mathrm{in}} - h. \tag{57}$$

Hence, when $d_{\text{in}} \gg h$, the first frozen layer necessarily contains a large right null space. The node-classification datasets have input dimensions of 1433 (Cora), 3703 (CiteSeer), 500 (PubMed), 500 (Flickr), and 128 (ogbn-arxiv), whereas the graph-classification datasets contain only 3–89 input dimensions. With $h = 128$, the null-space dimensions of the first-layer weights on Cora and CiteSeer are at least 1305 and 3575, respectively; on Cora we indeed observe a 1305-dimensional null space. In contrast, when $d_{\text{in}} \leq h$, the first layer may have full column rank and

$$\dim \ker(W) = 0. \tag{58}$$

Graph-level datasets therefore provide substantially less capacity for the null-space channel, and SRP relies more heavily on spectral reversal or the projective fallback.

Graph-level readout. Node classification predicts directly from each node representation:

$$\hat{y}_i = C(z_i + p_i), \tag{59}$$

so a node-local prompt directly affects the prediction of node i. Graph classification instead aggregates node representations. With mean pooling,

$$g = \frac{1}{N_g}\mathbf{1}^\top H, \qquad \Delta g = \frac{1}{N_g}\mathbf{1}^\top P. \tag{60}$$

If $P = \Phi C$, then

$$\Delta g = \frac{1}{N_g}\sum_j (\mathbf{1}^\top \phi_j) c_j^\top. \tag{61}$$

For the combinatorial Laplacian of a connected graph,

$$\mathbf{1}^\top \phi_j = 0, \qquad j > 1. \tag{62}$$

Mean readout therefore preserves the constant component while cancelling many nonstationary node-local variations. Sum pooling has the same directional effect up to scaling, which can reduce the contribution of local information recovered by SRP.

Cross-graph consistency. For node classification on a single graph, define the graph frequency of the j-th feature direction by

$$q_j = \frac{(Hv_j)^\top L(Hv_j)}{\|Hv_j\|_2^2}. \tag{63}$$

This quantity has a consistent meaning on the same graph. For graph classification, however,

$$q_{g,j} = \frac{(H_g v_j)^\top L_g(H_g v_j)}{\|H_g v_j\|_2^2}, \tag{64}$$

and the same v_j can correspond to different graph frequencies across different graphs. Large variation in $q_{g,j}$ weakens the correspondence between a shared singular direction and complementary graph-frequency information. These factors help explain why the empirical gains are generally larger on node-level benchmarks.

C.9 Why the Gains Are Larger on Some Smaller Datasets

The observed trend is better explained by spectral complement capacity than by dataset size alone, as input dimensionality and graph size are not independently varied in the current benchmarks. Cora and CiteSeer contain 2,708 and 3,327 nodes but have input dimensions of 1,433 and 3,703, respectively. In contrast, ogbn-arxiv contains 169,343 nodes with only 128 input dimensions, while Flickr contains 89,250 nodes with 500 input dimensions.

The exact right-null-space dimension is

$$\dim \ker(W) = d_{\text{in}} - \text{rank}(W), \tag{65}$$

which depends on the input dimension and weight rank rather than directly on the number of nodes. Cora and CiteSeer therefore provide substantially greater capacity for the null-space channel.

A second factor is local frequency heterogeneity. SRP currently uses a shared right-singular basis V and a global spectral threshold,

$$\mu_j = \mathrm{sigmoid}((\tau\sigma_{\max} - \sigma_j)\gamma)\,. \tag{66}$$

For a graph partitioned into regions $\mathcal{R}_1, \ldots, \mathcal{R}_M$, define

$$x_{r,j} = H_{\mathcal{R}_r} v_j, \qquad q_{r,j} = \frac{x_{r,j}^\top L_{\mathcal{R}_r} x_{r,j}}{x_{r,j}^\top x_{r,j}}. \tag{67}$$

When $\mathrm{Var}_r(q_{r,j})$ is large, the same singular direction can represent different graph-frequency characteristics across regions. A single global threshold can therefore become less precise as local frequency heterogeneity increases. Accordingly, the observed performance trend is more closely associated with input dimensionality, null-space capacity, and local frequency heterogeneity than with graph size itself.

D Computational Complexity

D.1 Compute Resources Usage.

All experiments are conducted on a workstation running the Ubuntu 22.04 LTS operating system. The hardware configuration includes a 13th Gen Intel(R) Core(TM) i9-13900K CPU and an NVIDIA GeForce RTX 3090 GPU equipped with 24GB of memory. Furthermore, the software environment is built using CUDA 12.1 and Python 3.10.19, utilizing PyTorch 2.5.1 and PyTorch Geometric 2.7.0 for the model implementations.

D.2 Complexity Analysis.

SRP's per-layer forward pass: (i) V-projection $O(|\mathcal{V}|kd_{\text{in}})$ (pre-computable); (ii) mask $O(|\mathcal{V}|k)$; (iii) A-projection $O(|\mathcal{V}|kr_s)$; (iv) N-projection $O(|\mathcal{V}|md_{\text{in}})$ (pre-computable); (v) B-projection $O(|\mathcal{V}|mr_s)$; (vi) Q-projection $O(|\mathcal{V}|r_s d_{\text{out}})$. Trainable-path cost: $O(|\mathcal{V}|(k+m)r_s + |\mathcal{V}|r_s d_{\text{out}})$, negligible vs. backbone's $O(|\mathcal{E}|d_{\text{in}})$.

We analyze the time and space complexity of SRP under a two-layer GCN backbone with $L = 2$ frozen layers. Let N denote the number of nodes, $|\mathcal{E}|$ the number of edges, d the raw input feature dimension, h the hidden (and output) dimension, $k = \mathrm{rank}(W) \le h$ the numerical rank of each frozen weight $W \in \mathbb{R}^{h \times d_{\text{in}}}$, m the null-space PCA dimension (`null_pca_dim`, default m=16), and r the shared bottleneck rank (`r_shared`, default r=32).

Initialisation. The initialisation of SRP involves two SVD computations and one null-space projection, all performed once before training. For each frozen weight W_i, the reduced SVD $W_i = U_i \Sigma_i V_i^\top$ costs $\mathcal{O}(k^2 d_{\text{in},i})$ time (thin SVD of an $h \times d_{\text{in},i}$ matrix). For the first layer, when $d > k + m$ holds and the dual-channel path is activated, SRP additionally projects the node-feature matrix $X \in \mathbb{R}^{N \times d}$ onto the null space via $X_{\text{null}} = X - XV_0V_0^\top$, costing $\mathcal{O}(Ndk)$, followed by a truncated SVD of the centred X_{null} to obtain the top-m null-space PCA basis $N_{\text{mat}} \in \mathbb{R}^{d \times m}$, which costs $\mathcal{O}(\min(N,d)^2 \max(N,d))$ but is performed on CPU once and cached as a frozen buffer. Summing across layers, the total initialisation time is $\mathcal{O}(k^2 d + Ndk + \min(N,d)^2 \max(N,d))$, which for typical datasets satisfies $Nd \ll \min(N,d)^2 \max(N,d)$, so the dominant term is the one-time null-space PCA. This cost is paid once before any gradient step and does not affect training throughput.

Per-forward-pass time complexity. During training and inference, `get_prompt` is invoked once per GNN layer per forward pass. For the first layer operating in the dual-channel mode ($d > k + m$), the three dominant operations are $h_v = hW_0^- V_0 \in \mathbb{R}^{N \times k}$ at cost $\mathcal{O}(Ndk)$, $h_{\text{null}} = hW_0^- N_{\text{mat}} \in \mathbb{R}^{N \times m}$ at cost $\mathcal{O}(Ndm)$, and the two bottleneck projections $z_{\text{spec}} = (h_v \odot \mathbf{s})A \in \mathbb{R}^{N \times r}$ and $z_{\text{null}} = h_{\text{null}} B \in \mathbb{R}^{N \times r}$ at cost $\mathcal{O}(N(kr + mr))$, followed by the output projection $zQ + \mathbf{b}$ at cost $\mathcal{O}(Nrh)$. Since $k, m \ll d$ and $r \ll k, m$, the total for layer 0 is $\mathcal{O}(Nd(k+m))$. For the second layer operating under the projective fallback ($d_{\text{in}} = h \le k + m$), the cost is $\mathcal{O}(Nh^2 + Nhr)$, which collapses to $\mathcal{O}(Nh^2)$. The backbone GCN message-passing itself costs $\mathcal{O}(|\mathcal{E}|h + Ndh)$ across the two layers. Combining prompt and backbone, the overall per-forward-pass complexity is

$$\mathcal{O}\big(Nd(k+m) + Nh^2 + |\mathcal{E}|h\big) \;=\; \mathcal{O}\big(Ndh + |\mathcal{E}|h\big),$$

where the equality uses $k \leq h$ and $m \ll h$. Thus SRP adds no asymptotic overhead beyond the backbone GNN's own message-passing and projection costs.

Parameter count and space complexity. SRP introduces only a small set of trainable parameters. For layer 0 (dual-channel): $A_0 \in \mathbb{R}^{k\times r}$, $B_0 \in \mathbb{R}^{m\times r}$, $Q_0 \in \mathbb{R}^{r\times h}$, $\mathbf{b}_0 \in \mathbb{R}^h$, and the scalar threshold τ_0, contributing $kr + mr + rh + h + 1 = r(k + m + h) + h + 1$ parameters. For layer 1 (projective fallback): $P_1 \in \mathbb{R}^{h\times r}$, $Q_1 \in \mathbb{R}^{r\times h}$, $\mathbf{b}_1 \in \mathbb{R}^h$, and τ_1, contributing $2rh + h + 1$ parameters. The total trainable parameter count is therefore $r(k + m + 3h) + 2h + 2$. The additional memory consists of frozen buffers: $V_0 \in \mathbb{R}^{d\times k}$ and $N_{\mathrm{mat}} \in \mathbb{R}^{d\times m}$ for layer 0, and $V_1 \in \mathbb{R}^{h\times k}$ for layer 1, totalling $d(k + m) + hk$ floating-point values. The activation memory during a forward pass is $\mathcal{O}(Nd)$ for the null-space projections of the full feature matrix, which is at most $\mathcal{O}(Nd)$—identical to the GCN input tensor. Overall, SRP's space overhead is $\mathcal{O}(d(k + m) + hk)$ in parameters and buffers, with the number of *trainable* parameters scaling as $\mathcal{O}(r(k + m + h)) = \mathcal{O}(rh)$, far smaller than any full fine-tuning or LoRA baseline of comparable expressiveness.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claimed theoretical results are formalized in Section 4 with additional proofs in Appendix B, and the claimed empirical results are supported in Section 6.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We clearly state the assumptions used in Section 4 and complete proofs are provided in an Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 6 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code and implementation details are publicly available at `https://github.com/keris-yang/Spectral-Reversal`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 6 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experimental results in Section 6 include averages and standard deviations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We show our compute resources usage in AppendixD

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research adhered to every aspect of the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: All datasets used in this work are publicly available benchmark datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets (e.g., Cora, Citeseer), all of which are properly cited in the paper. We adhere to the licenses and terms of use as specified by their original creators.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The proposed SRP implementation is publicly released on GitHub. The repository includes source code, dependency specifications, installation instructions, and documentation for running the main node-level and graph-level experiments.

    Guidelines: Code and details

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not involved in the core method development.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.