Rethinking Spectral Graph Neural Networks with Spatially Adaptive Filtering

Jingwei Guo*
University of Liverpool
Liverpool, UK
jingwei.guo@liverpool.ac.uk

Kaizhu Huang†
Duke Kunshan University
Suzhou, China
kaizhu.huang@dukekunshan.edu.cn

Xinping Yi
Southeast University
Nanjing, China
xyi@seu.edu.cn

Zixian Su*
University of Liverpool
Liverpool, UK
zixian.su@liverpool.ac.uk

Rui Zhang
Xi'an Jiaotong-Liverpool University
Suzhou, China
rui.zhang02@xjtlu.edu.cn

ABSTRACT

Whilst spectral Graph Neural Networks (GNNs) are theoretically well-founded in the spectral domain, their practical reliance on polynomial approximation implies a profound linkage to the spatial domain. As previous studies rarely examine spectral GNNs from the spatial perspective, their spatial-domain interpretability remains elusive, *e.g.*, what information is essentially encoded by spectral GNNs in the spatial domain? In this paper, to answer this question, we establish a theoretical connection between spectral filtering and spatial aggregation, unveiling an intrinsic interaction that spectral filtering implicitly leads the original graph to an adapted new graph, explicitly computed for spatial aggregation. Both theoretical and empirical investigations reveal that the adapted new graph not only exhibits non-locality but also accommodates signed edge weights to reflect label consistency among nodes. These findings thus highlight the interpretable role of spectral GNNs in the spatial domain and inspire us to rethink graph spectral filters beyond the fixed-order polynomials, which neglect global information. Built upon the theoretical findings, we revisit the state-of-the-art spectral GNNs and propose a novel Spatially Adaptive Filtering (SAF) framework, which leverages the adapted new graph by spectral filtering for an auxiliary non-local aggregation. Notably, our proposed SAF comprehensively models both node similarity and dissimilarity from a global perspective, therefore alleviating persistent deficiencies of GNNs related to long-range dependencies and graph heterophily. Extensive experiments over 13 node classification benchmarks demonstrate the superiority of our proposed framework to the state-of-the-art models.

1 INTRODUCTION

Graph Neural Networks (GNNs) have shown remarkable abilities to uncover the intricate dependencies within graph-structured data, and achieved tremendous success in graph machine learning [10, 24, 70]. Spectral GNNs are a class of GNNs rooted in spectral graph theory [15, 60], implementing graph convolutions via spectral filters [17, 39]. Whilst various spectral filtering strategies [14, 27, 29, 32, 33, 36, 42, 64, 67] have been proposed for spectral GNNs, their practical implementations invariably resort to approximating graph filters with fixed-order polynomials for computational efficiency [32, 67]. This truncated approach essentially relies on the direct extraction of spatial features from the local regions of nodes. As such, the spatial domain of a graph, albeit loosely connected to spectral GNNs in theory, still plays a crucial role in effectively learning node representations.

However, there is a notable lack of research examining spectral GNNs from the spatial perspective. Though recent studies analyze both spectral and spatial GNNs to elucidate their similarities in model formulations [4, 13, 28], outcomes [50, 77], and expressiveness [3, 12, 62, 67], they ignore exploring the interpretability that could arise mutually from the other domain. Specifically, whilst most spectral GNNs have well explained their learned graph filters in the spectral domain [27, 32, 33, 67], understandings from the spatial viewpoint are merely limited to fusing multi-scale graph information [45]; this unfortunately lacks a deeper level of interpretability in the vertex domain. Therefore, a natural question arises: *what information is essentially encoded by spectral GNNs in the spatial domain?*

In this work, we attempt to answer this question by exploring the connection between spectral filtering and spatial aggregation. The former is the key component in spectral GNNs, while the latter is closely associated with spatial GNNs utilizing recursive neighborhood aggregation. In existing GNN frameworks, these two approaches rarely interact each other at the risk of domain information trade-offs due to uncertainty principles [1, 23, 34]. Recognizing the spatial significance in spectral filtering, He et al. [33] have recently considered non-negative constraints as part of a generalized graph optimization problem. Notably, however, spatial aggregation meanwhile resembles the optimizing trajectory of the same optimization problem through iterative steps, which may be easily overlooked. Inspired by such observation, we examine, for the first time, the theoretical interaction between spectral filtering and spatial aggregation. This exploration has led us to uncover an intriguing theoretical interplay, *i.e.*, spectral filtering implicitly modifies the original graph, transforming it into a new one that explicitly functions as a computation graph for spatial aggregation. Delving deeper, we discover that the adapted new graph enjoys some desirable properties, enabling a direct link among nodes that originally require multiple hops to do so, thereby exhibiting nice non-locality. Moreover, we find that the new graph edges allow signed weights, which turns out capable of distinguishing between label agreement and disagreement of the connected nodes.

*Also with Xi'an Jiaotong-Liverpool University.
†Corresponding Author

Overall, these findings underscore the interpretable role and significance of spectral GNNs in the spatial domain, inspiring us to rethink graph spectral filters beyond the fixed-order polynomials, which limit the effective propagation range of models and hinder their ability to capture long-range dependencies. Concretely, we propose a novel Spatially Adaptive Filtering (SAF) framework, for fully exploring spectral GNNs in the spatial domain. SAF leverages the adapted new graph by spectral filtering for auxiliary spatial aggregation and allows individual nodes to flexibly balance between spectral and spatial features. By performing non-local aggregation with signed edge weights, our SAF adeptly overcomes the limitations of truncated polynomials, enabling the model to capture both node similarity and dissimilarity at a global scale. As a benefit, it can mitigate persistent deficiencies of GNNs regarding long-range dependencies and graph heterophily. The contributions are summarized as follows:

- Our investigation into spectral GNNs in the spatial domain reveals that graph spectral filtering fundamentally alters the original graph, imbuing it with non-locality and signed edge weights that discern label consistency among nodes.
- We propose Spatially Adaptive Filtering (SAF) framework, a paradigm-shifting approach to spectral GNNs that jointly leverages graph learning in both spatial and spectral domains, making it a powerful tool for capturing long-range dependencies and handling graph heterophily.
- Extensive experiments over 13 node classification benchmarks exhibit notable improvements of up to 15.37%, and show that SAF beats the best-performing spectral GNNs on average.

2 RELATED WORKS

2.1 Graph Neural Networks

GNNs can be broadly divided into spatial-based and spectral-based models. Spatial GNNs leverage the spatial connections among nodes to perform message passing, also known as spatial aggregation [26, 30]. For a thorough review, we direct readers to the works [70, 75]. Spectral GNNs leverage the graph's spectral domain for convolution or, alternatively, spectral filtering [17, 18, 39]. Prevailing approaches focus on developing polynomial graph filters, by either learning polynomial coefficients, such as GPR-GNN [14], BernNet [33], ChebNetII [32], and JacobiConv [67], or concurrently optimizing the polynomial basis for better real-world adaption, as seen in models like LON-GNN [64] and OptBasisGNN [29]. Diverging from this trend, ARMA [6] employs rational filter functions while still approximating them with polynomials. Although these methods are theoretically grounded in the spectral domain, their practical reliance on polynomial approximation hints at a profound linkage to the spatial domain. However, the spatial-domain interpretation of spectral GNNs is rarely examined. To this end, we delve into in this paper the intrinsic information spectral GNNs convey within the spatial context.

2.2 Unified Viewpoints for GNNs

Several works have explored the nuances between spatial and spectral GNNs. Early studies by Balcilar et al. [4] and Chen et al. [13] examined their similarities in model formulations. Chen et al. [12] proved their spatial GNN's anti-oversmoothing ability via spectral analysis. Ma et al. [50] and Zhu et al. [77] utilized the graph signal denoising problem to integrate both GNN types. Balcilar et al. [3] and Wang and Zhang [67] further explored their expressiveness equivalence. Recently, Sun et al. [62] have highlighted the feature space constraints of both spatial and spectral GNNs, while Guo and Wei [28] attempted to combine them via a residual connection module. Though these studies effectively bridge spectral and spatial GNNs, they remain focused on congruencies. Unlike them, our work represents the first endeavor to delve into the interpretability of spectral GNNs in the spatial domain, emphasizing the theoretical synergy between spectral filtering and spatial aggregation. The empirical success of our proposed method (as compared to unified GNNs in Tables 1 and 2), stemming from this in-depth analysis, further underscores our practical contributions to the literature.

2.3 Long-range Dependencies

While substantial efforts have been directed towards capturing long-range dependencies in spatial GNNs [12, 20, 25, 49, 52, 69], the exploration of the same challenge in spectral GNNs remains under-studied. To fill this gap, we propose a SAF framework, which emerges as a valuable consequence of analyzing spectral GNNs in the spatial domain, enhancing their long-range dependency capture. Concurrently, Bo et al. [7] also introduced Specformer to addresses long-range dependencies for spectral GNNs, using a Transformer based set-to-set spectral filter. However, it lacks spatial-domain interpretability and introduce more trainable parameters. In contrast, our approach creates a non-local new graph without learning additional parameters, simultaneously elucidating the interpretive implications of spectral GNNs in the spatial domain.

2.4 Graph Heterophily

Graph heterophily [52, 76], where different labeled nodes connect, challenges GNNs operating under the homophily assumption [51]. Although many GNNs have been crafted to manage heterophilic connections [8, 14, 16, 66, 71, 73], our proposed SAF stands out in addressing graph heterophily. Specifically, SAF innovatively conducts an auxiliary non-local aggregation using signed edge weights, emphasizing both intra-class similarity and inter-class difference on a global scale. One should note that a recent work [44] bear some resemblance to ours, introducing GloGNN and GloGNN++ to capture global homophily beyond immediate neighborhoods by learning signed edge weights. However, their approach, albeit demonstrating a grouping effect [43], restricts the optimization objective into a K-hop neighborhood, focusing on similar local structural information. Conversely, our SAF framework ensures the theoretical properties of non-local learning (as proved in Section 4.2), while effectively modeling label relationships. This capability directly benefits downstream classification tasks, offering a notable superiority on real-world applications.

3 NOTATIONS AND PRELIMINARIES

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$, where the number of nodes is denoted by N. We define the adjacency matrix as $\mathbf{A} \in \mathbb{R}^{N\times N}$ with $A_{i,j}$ representing the edge weight between node pairs $v_i, v_j \in \mathcal{V}$. The degree matrix $\mathbf{D}$ is obtained by summing the rows of $\mathbf{A}$ into a diagonal matrix. We denote the graph Laplacian

matrix as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, which is often normalized into $\hat{\mathbf{L}} = \mathbf{I}_N - \hat{\mathbf{A}}$ with $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{I}_N$ being an identity matrix. Let $\hat{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ be the eigendecomposition of $\hat{\mathbf{L}}$, where the columns of $\mathbf{U}$ refer to eigenvectors and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ consists of eigenvalues with each $\lambda_n \in [0, 2]$. For node classification on graph $\mathcal{G}$, nodes are associated with a feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ where F refers to raw feature dimensions, and each of them is assigned a class c_i with a one-hot vector $\mathbf{y}_i \in \mathbb{R}^C$ where $C \leq N$ is class number.

Spectral Filtering. Spectral filtering is essential in spectral GNNs. It selectively shrinks or amplifies the Fourier coefficients of node features [17] and usually take the form as

$$\mathbf{Z} = g_\psi(\hat{\mathbf{L}})\mathbf{X} = \mathbf{U}g_\psi(\mathbf{\Lambda})\mathbf{U}^T\mathbf{X}. \tag{1}$$

Here, $g_\psi : [0, 2] \to \mathbb{R}$ defines a graph filter function, which are often approximated by a K-order polynomial in practice. Specifically, we have $g_\psi(\lambda) = \sum_{k=0}^{K} \psi_k P_k(\lambda) = \sum_{k=0}^{K} \omega_k \lambda^k$ where $P_k : [0, 2] \to \mathbb{R}$ refers to a polynomial basis and both ψ_k and ω_k denote the polynomial coefficient.

Spatial Aggregation. Spatial Aggregation is a central component of spatial GNNs, facilitating the propagation of node information along graph edges and its subsequent aggregation within node neighborhood. To provide a more formal illustration of spatial aggregation, let's consider the widely adopted GNN model, APPNP [25]. This model begins by applying a feature transformation, given by $\mathbf{Z}^{(0)} = f(\mathbf{X})$. The propagation then proceeds as:

$$\mathbf{Z}^{(k)} = (1 - \eta)\mathbf{Z}^{(0)} + \eta\hat{\tilde{\mathbf{A}}}\mathbf{Z}^{(k-1)}, \quad k = 1, 2, \cdots, K, \tag{2}$$

where $\hat{\tilde{\mathbf{A}}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, and η refers to the update rate.

4 RETHINKING SPECTRAL GNNS FROM THE SPATIAL PERSPECTIVE

In this section, we provide both theoretical and empirical analyses to examine spectral GNNs from the spatial perspective and answer the question, *i.e.*, what information is essentially encoded by spectral GNNs in the spatial domain?

4.1 Interplay of Spectral and Spatial Domains through the Lens of Graph Optimization

The graph signal denoising problem [60] was initially leveraged in [50, 77] as a means to interpret GNNs with smoothness assumption, which yet does not always hold in certain real-world graph scenarios such as heterophily [76]. Without loss of generality, in this work, we consider a more generalized graph optimization problem[1]

$$\arg\min_{\mathbf{Z}} \ \mathcal{L} = \alpha\|\mathbf{X} - \mathbf{Z}\|_F^2 + (1 - \alpha) \cdot \mathrm{tr}(\mathbf{Z}^T\gamma_\theta(\hat{\mathbf{L}})\mathbf{Z}) \tag{3}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times d}$ refers to node representations, $\gamma_\theta(\hat{\mathbf{L}})$ determines the rate of propagation [61] by operating on the graph spectrum, i.e., $\gamma_\theta(\hat{\mathbf{L}}) = \mathbf{U}\gamma_\theta(\mathbf{\Lambda})\mathbf{U}^T$, and $\alpha \in (0, 1)$ is a trade-off coefficient. In case of setting $\gamma_\theta(\hat{\mathbf{L}}) = \hat{\mathbf{L}}$, Eq. (3) turns into the well-known graph signal denoising problem. To ensure the convexity of the objective in Eq. (3), a positive semi-definite constraint is imposed on $\gamma_\theta(\hat{\mathbf{L}})$, *i.e.*, $\gamma_\theta(\lambda) \geq 0$ for $\lambda \in [0, 2]$. Then, one can address this minimization problem through either closed-form or iterative solutions.

Closed-form Solution. The closed-form solution can be obtained by setting the derivative of the objective function $\mathcal{L}$ to 0, i.e., $\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = 2\alpha(\mathbf{Z} - \mathbf{X}) + 2(1 - \alpha)\gamma_\theta(\hat{\mathbf{L}})\mathbf{Z} = 0$. Let $g_\psi(\lambda) = (1 + \frac{1-\alpha}{\alpha}\gamma_\theta(\lambda))^{-1}$, we can observe that the closed-form solution in Eq. (4) is equivalent to the spectral filtering in Eq. (1).

$$\mathbf{Z}^* = (\mathbf{I} + \frac{1 - \alpha}{\alpha}\gamma_\theta(\hat{\mathbf{L}}))^{-1}\mathbf{X} = g_\psi(\hat{\mathbf{L}})\mathbf{X} = \mathbf{U}g_\psi(\mathbf{\Lambda})\mathbf{U}^T\mathbf{X}. \tag{4}$$

As $\gamma_\theta(\lambda) \geq 0$, this establishes a more stringent constraint for the graph filter in spectral GNNs, i.e., $0 < g_\psi(\lambda) \leq \frac{\alpha}{\alpha + (1-\alpha)\cdot 0} = 1$, which is termed as a non-negative constraint in this paper.

Iterative Solution. Alternatively, we can take an iterative gradient descent method such that $\mathbf{Z}^{(k)} = \mathbf{Z}^{(k-1)} - b\frac{\partial \mathcal{L}}{\partial \mathbf{Z}}|_{\mathbf{Z}=\mathbf{Z}^{(k-1)}}$ with a step size $b = \frac{1}{2}$, which yields a concise iterative solution in Eq. (5) with $\hat{\mathbf{A}}^{\text{new}} = \mathbf{I} - \gamma_\theta(\hat{\mathbf{L}})$. Notably, by taking $\hat{\mathbf{A}}^{\text{new}}$ as a new computation graph, this solution closely mirrors the spatial aggregation in Eq. (2).

$$\mathbf{Z}^{(k)} = \alpha\mathbf{X} + (1 - \alpha)\hat{\mathbf{A}}^{\text{new}}\mathbf{Z}^{(k-1)}, \quad k = 1, 2, \cdots, K \tag{5}$$

Theoretical Interaction — the Adapted New Graph. With the non-negative constraint, it is evident that both spectral filtering and spatial aggregation effectively address the generalized graph optimization problem in Eq. (3), despite their distinctive forms and operation domains. Upon closer examination, we discover a compelling relationship between the graph filter $g_\psi(\lambda)$ in Eq. (4) and the new graph $\hat{\mathbf{A}}^{\text{new}}$ in Eq. (5), given $g_\psi(\lambda) = (1 + \frac{1-\alpha}{\alpha}\gamma_\theta(\lambda))^{-1}$,

$$\hat{\mathbf{A}}^{\text{new}} = \mathbf{I} - \gamma_\theta(\hat{\mathbf{L}}) = \mathbf{I} - \frac{\alpha}{1 - \alpha}(g_\psi(\hat{\mathbf{L}})^{-1} - \mathbf{I}) \tag{6}$$

which unveils an intrinsic inter-play, *i.e.*, spectral filtering implicitly leads the original graph to an adapted new graph, explicitly computed for spatial aggregation.

What are the differences between $\hat{\mathbf{A}}^{\text{new}}$ and $g_\psi(\hat{\mathbf{L}})$? Whereas the former as the uncovered new graph elucidates the inherent spatial node relationships, the latter is a graph operation, primarily processing graph features within spectral domain. It is crucial to understand that $g_\psi(\hat{\mathbf{L}})$ may not result in a dense matrix, especially with fixed-order polynomial approximation. This is because it captures up to only a K-hop neighborhood, i.e., $g_\psi(\hat{\mathbf{L}}) = \sum_{k=0}^{K} \psi_k P_k(\hat{\mathbf{L}}) = \sum_{k=0}^{K} \omega_k \hat{\mathbf{A}}^k$, practically limiting spectral GNNs' effective propagation range. In contrast, our newfound graph $\hat{\mathbf{A}}^{\text{new}}$ intrinsically enjoys a non-local property, as confirmed in the following section. Building upon this discovery, we further devise a framework to break domain barriers, overcoming the limitations of current spectral GNNs due to truncated polynomials (see details in Section 5).

4.2 In-depth Analysis of the adapted new graph

To deepen our understanding of the interpretability produced by spectral GNNs in the spatial domain, we embark upon a blend of theoretical and empirical inquiries into the adapted new graph.

4.2.1 ***Non-locality.*** Our examination of the adapted new graph illuminates its non-local nature, particularly evident in the infinite series expansion of the original graph's adjacency matrix. To elucidate, we first introduce an pivotal mathematical construct, the Neumann series, in the following lemma.

[1]This problem was first introduced in [33] for theoretically grounded graph filters. However, in this study, we repurpose it as a bridge between spectral filtering and spatial aggregation.

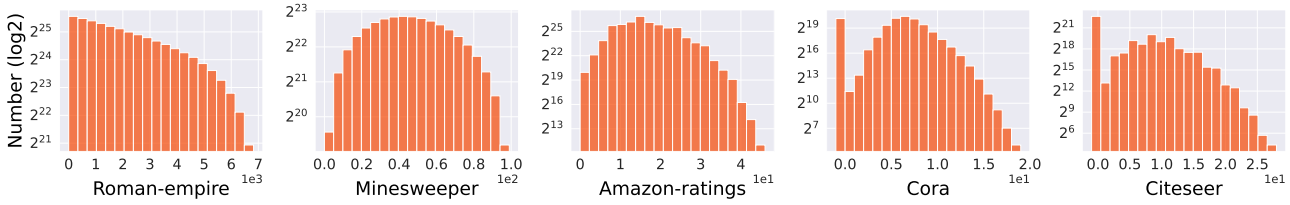


Figure 1: Distributions of connected nodes in the new graph based on their geodesic/shortest-path distance (as $\Delta_{i,j}$) in the original graph. Nodes, distant in the original graph ($\Delta_{i,j} > 1$ in x-axis), can be linked in the new graph (Number > 0 in y-axis).

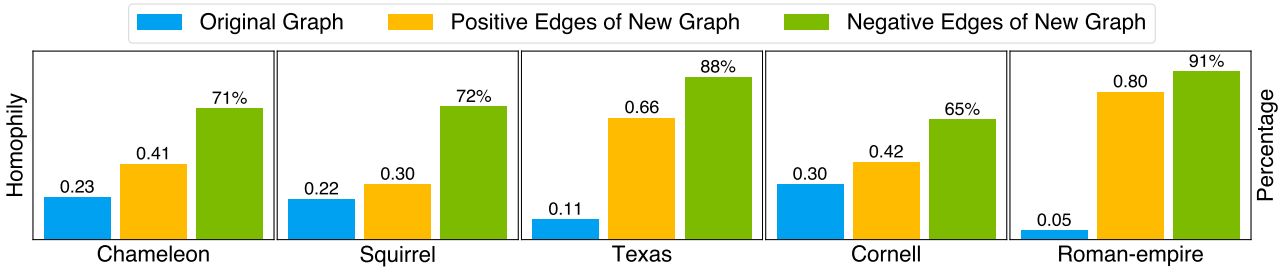


Figure 2: Left y-axis: Homophily comparison between original and new graphs, considering only positive edges (blue and yellow bars). Right y-axis: Percentage of edges connecting nodes from different classes, identified by negative edges (green bar).

Lemma 1. Let $\mathbf{M} \in \mathbb{R}^{N\times N}$ be a matrix with eigenvalues λ_n, if $|\lambda_n| < 1$ for all $n = 1, 2, ..., N$, then $(\mathbf{I} - \mathbf{M})^{-1}$ exists and can be expanded as an infinite series, i.e., $(\mathbf{I} - \mathbf{M})^{-1} = \sum_{t=0}^{\infty} \mathbf{M}^t$, which is known as Neumann series.

With the established non-negative constraint on graph filters, specifically $0 < g_\psi(\lambda) \leq 1$, it becomes evident that the eigenvalues of $\mathbf{I} - g_\psi(\hat{\mathbf{L}})$ falls into the interval permitting Neumann series expansion, as shown in lemma 1 [35]. Building on this observation, we present a non-trivial property of the new graph in the following proposition (see proof in Appendix A.1).

Proposition 1. Given adjacency matrix $\hat{\mathbf{A}}^{\text{new}}$ formulated in Eq. (6), the adapted new graph exhibits non-locality. Specifically, $\hat{\mathbf{A}}^{\text{new}}$ is expressible as an infinite series expansion of the original graph's adjacency matrix $\hat{\mathbf{A}}$. Formally, we have $\hat{\mathbf{A}}^{\text{new}} = \mathbf{I} - \frac{\alpha}{1-\alpha}\sum_{t=1}^{\infty}(\mathbf{I} - \sum_{k=0}^{K}\pi_k\hat{\mathbf{A}}^k)^t = \sum_{t=0}^{\infty}\phi_t\hat{\mathbf{A}}^t$ where π_k and ϕ_t refer to the constant coefficients computed from $\{\psi_0, \psi_1, ..., \psi_K\}$ in distinct ways.

This proposition implies that the new graph engenders immediate links between nodes that originally necessitate multiple hops for connection. To further underpin this theoretical claim, we analyze the general connection status on the new graph by BernNet [33], a spectral GNN adhering to the non-negative constraint. From Figure 1, it is apparent that nodes originally separated by multiple hops achieve direct connections in the new graph.

4.2.2 ***Signed Edge Weights — Discerning Label Consistency.*** Upon further scrutinizing the adapted new graph, we make a notable discovery that it readily accommodates both positive and negative edge weights. A more granular analysis in Figure 2 reveals that a considerable portion of positive edge weights are assigned to the same-class node pairs, enhancing graph homophily (exemplified by edge homophily ratio [76]). Conversely, edges parameterized with negative weights tend to bridge nodes with different labels. These findings demonstrate the newfound graph's adeptness in discerning label consistency among nodes. To theoretically explain this phenomenon, we further present the proposition below (see proof in Appendix A.2):

Proposition 2. Let $\mathbf{Z}^*$ be the node representations optimized by Eq. (3). For $\mathbf{Z}^*$ to be effective in label prediction, it is a necessary condition that $\hat{\mathbf{A}}^{\text{new}}$ accommodates both positive and negative edge weights s.t. for any node pairs $v_i, v_j \in \mathcal{V}$, $\hat{A}^{\text{new}}_{i,j} > 0$ if $\mathbf{y}_i = \mathbf{y}_j$ and $\hat{A}^{\text{new}}_{i,j} < 0$ if $\mathbf{y}_i \neq \mathbf{y}_j$.

Proposition 2 provides a theoretical foundation of our empirical findings on the new graph. The essence lies in the objective in Eq. (3), particularly the trace term $\text{tr}(\mathbf{Z}^T\gamma_\theta(\hat{\mathbf{L}})\mathbf{Z})$. For clarity, let us reinterpret this trace term as $\text{tr}(\bar{\mathbf{Z}}^T(\mathbf{D}^{\text{new}} - \mathbf{A}^{\text{new}})\bar{\mathbf{Z}})$, where $\mathbf{D}^{\text{new}}$ denotes the related degree matrix and $\bar{\mathbf{Z}}$ is derived from rescaling $\mathbf{Z}$. Clearly, this term evaluates label smoothness among adjacent nodes in the new graph, which, given its non-local nature, includes both intra-class ($=$) and inter-class ($\neq$) node connections such that $\mathbf{A}^{\text{new}} = \mathbf{A}^{\text{new}}_{=} + \mathbf{A}^{\text{new}}_{\neq}$. Drawing from proposition 2, we can further dissect the original trace term, splitting it into $\text{tr}(\bar{\mathbf{Z}}^T(\mathbf{D}^{\text{new}}_{=} - \mathbf{A}^{\text{new}}_{=})\bar{\mathbf{Z}}) - \text{tr}(\bar{\mathbf{Z}}^T|(\mathbf{D}^{\text{new}}_{\neq} - \mathbf{A}^{\text{new}}_{\neq})|\bar{\mathbf{Z}})$ where the $|\cdot|$ operation denotes absolute values. As such, it becomes evident that minimizing this trace term not only enhances the representational proximity for same-class node pairs but also strengthens the distinctiveness for different-class nodes pairs. Such nuanced behaviors, inherent to the optimization in Eq. (3), are necessary for GNN models to achieve accurate label predictions.

To summarize, our investigation into spectral GNNs in the spatial domain reveals that graph spectral filtering fundamentally alters

the original graph, imbuing it with non-locality and signed edge weights that capture label consistency among nodes. These findings highlight the interpretable role of spectral GNNs in the spatial domain, and prompt us to rethink current spectral GNNs beyond the truncated polynomial filters.

5 SPATIALLY ADAPTIVE FILTERING FRAMEWORK

Building on our discoveries, we re-evaluate the state-of-the-art spectral GNNs and put forth a paradigm-shifting framework, Spatially Adaptive Filtering (SAF), for joint exploitation of graph-structured data across both spectral and spatial domains (refer to Figure 3). SAF leverages the adapted new graph by spectral filtering for an auxiliary non-local aggregation, addressing enduring challenges in GNNs related to long-range dependencies and graph heterophily.

5.1 Non-negative Spectral Filtering

The proposed SAF requires explicit computation of the newfound graph, as outlined in Eq. (6). This further necessitates the graph filter $g_\psi : [0, 2] \to \mathbb{R}$ to satisfy the non-negative constraint from Eq. (3): $0 \leq g_\psi(\lambda) \leq 1$. However, not all extant graph filters fulfill this prerequisite. For instance, the filter use by GCN [39], $g_\psi(\lambda) = 1 - \lambda$, takes negative values when $\lambda > 1$. In this research, we approximate the graph filter using Bernstein polynomials [21], which are known for their non-negative traits [55] and are essential in a preeminent spectral GNN, BernNet [33]. For $g_\psi(\lambda) \leq 1$ part, we rescale Bernstein polynomials with the following proposition.

Proposition 3. Let $B_{k,K}(x)$ denote the Bernstein polynomial basis of index k and degree K, which is defined as $B_{k,K}(x) = \binom{K}{k}(1-x)^{K-k}x^k$ for $x \in [0, 1]$. Let ψ_k denote the k-th coefficient of a polynomial $p(x)$ of degree K, where $p(x) = \sum_{k=0}^{K} \psi_k B_{k,K}(x)$ with $\psi_k \geq 0$ for all k. Then for all $x \in [0, 1]$, we have $g_\psi(x) \leq \max\{\psi_k\}_{k=0}^{K}$.

Proposition 3 suggests that the Bernstein polynomial function attains its maximum value in $\psi_{\max} = \max\{\psi_k\}_{k=0}^{K}$. Therefore, $g_\psi(\lambda)$ can be rescaled within $[0, 1]$ by $\hat{g}_\psi(\lambda) = \frac{1}{\psi_{\max}} \sum_{k=0}^{K} \psi_k B_{k,K}(\frac{\lambda}{2})$, enabling us to formulate the spectral filtering in SAF as

$$\mathbf{Z}_f = \hat{g}_\psi(\hat{\mathbf{L}}) f_\varphi(\mathbf{X}) = \frac{1}{\psi_{\max}} \sum_{k=0}^{K} \psi_k \frac{1}{2^K} \binom{K}{k} (2\mathbf{I} - \hat{\mathbf{L}})^{K-k} \hat{\mathbf{L}}^k f_\varphi(\mathbf{X})$$

where $f_\varphi(\cdot)$, a two-layer MLP, maps $\mathbf{X}$ from F to C dimensions using 64 hidden units, and $\{\psi_k\}_{k=1}^{K}$ are non-negative learnable parameters. Note that SAF also permits alternative implementations such as using Chebyshev polynomials [17, 31] for graph filter learning, enhancing models like ChebNetII [32] (see details in Appendix E.4).

5.2 Non-local Spatial Aggregation

Once acquiring a suitable spectral filter $\hat{g}_\psi(\lambda)$, we compute the adapted new graph as $\hat{\mathbf{A}}^{\text{new}} = \mathbf{I} - \tau(\mathbf{U}_\mathbf{m} g_\psi(\mathbf{\Lambda}_\mathbf{m})^{-1} \mathbf{U}_\mathbf{m}{}^T - \mathbf{I})$ by Eq. (6) where $\tau = \frac{\alpha}{1-\alpha}$ is a scaling parameter and a partial eigendecomposition can be employed to obtain only m extremal eigenvalues [41], producing a low-rank, robust structure for $\hat{\mathbf{A}}^{\text{new}}$. Equipped with this newfound graph, we proceed to perform non-local aggregation:

$$\mathbf{Z}^{(l)} = (1-\eta)\mathbf{Z}^{(0)} + \eta \hat{\mathbf{A}}^{\text{new}} \mathbf{Z}^{(l-1)}, \quad l = 1, 2, \cdots, L$$

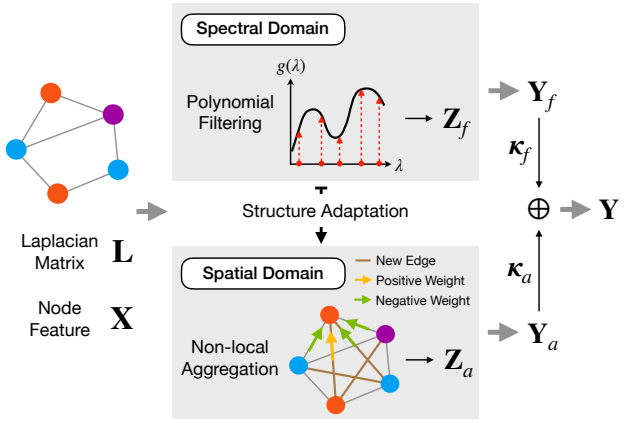


Figure 3: Illustration of the proposed SAF framework, where varying node colors represent different node labels.

where η refers to the update rate and $\mathbf{Z}^{(0)} = f_\varphi(\mathbf{X})$. The iteratively aggregated results are denoted as $\mathbf{Z}_a$. Recognizing the potential noise from the non-local nature of $\hat{\mathbf{A}}^{\text{new}}$, we apply a sparsification technique, leveraging a positive threshold ϵ, and retain only essential elements outside the $[-\epsilon, \epsilon]$ interval. For clarity, this refined model is referred to as SAF-ϵ.

5.3 Node-wise Prediction Amalgamation

To leverage information from different graph domains, we employ an attention mechanism, allowing nodes to determine the importance of each space. This mechanism produces pairwise weights for a nuanced amalgamation during prediction. Specifically, the weight pair is computed as $\boldsymbol{\kappa}_f = \text{Sigmoid}(\mathcal{P}_f(\mathbf{Z}_f))$, $\boldsymbol{\kappa}_a = \text{Sigmoid}(\mathcal{P}_a(\mathbf{Z}_a))$ where $\boldsymbol{\kappa}_f, \boldsymbol{\kappa}_a \in \mathbb{R}^N$ contain the weights for each node, and $\mathcal{P}_f(\cdot)$ and $\mathcal{P}_a(\cdot)$ are two different mappings from $\mathbb{R}^C$ to $\mathbb{R}$. For simplicity, we implement them using two one-layer MLPs. Given domain predictions $\mathbf{Y}_f, \mathbf{Y}_a \in \mathbb{R}^C$, the final model prediction is attained as

$$\mathbf{Y} = \text{diag}(\boldsymbol{\kappa}_f) \cdot \mathbf{Y}_f + \text{diag}(\boldsymbol{\kappa}_a) \cdot \mathbf{Y}_a \tag{7}$$

where a normalization $[\boldsymbol{\kappa}_f, \boldsymbol{\kappa}_a] \leftarrow \frac{[\boldsymbol{\kappa}_f, \boldsymbol{\kappa}_a]}{\max\{\|\{\boldsymbol{\kappa}_f, \boldsymbol{\kappa}_a\}\|_{1,\delta}\}}$ is performed beforehand to maintain $\boldsymbol{\kappa}_f + \boldsymbol{\kappa}_a = \mathbf{1}$ with small value δ preventing zero division. Similar schemes can be founded in works [58, 68, 78].

5.4 Complexity Analysis

SAF augments spectral GNNs with non-local aggregation and node-wise amalgamation. The first part entails creating a new graph and information propagation. In SAF-ϵ, these two steps are separated by sparsification, culminating in $O(N^3 + N^2 + nnz(\hat{\mathbf{A}}^{\text{new}})d)$ complexity, where nnz denotes non-zero element count. Conversely, SAF, viewing non-local aggregation holistically, can reduce complexity to $O(2dN^2 + dN)$ when $d \ll N$. For node-wise amalgamation, its parallelizable nature ensures computational efficiency. Our method also requires a precomputation of eigendecomposition[2] which, naively

[2]For more on its modern applications and discussions, please refer to Appendix B (including Specformer [7] & FE-GNN [62] featured in our experiments) and E.5.

Table 1: Semi-supervised node classification accuracy (%) ± 95% confidence interval.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
MLP	26.36±2.85	21.42±1.50	32.42±9.91	36.53±7.92	29.75±0.95	57.17±1.34	56.75±1.55	70.52±2.01
GCN	38.15±3.77	31.18±0.93	34.68±9.07	32.36±8.55	22.74±2.37	79.19±1.37	69.71±1.32	78.81±0.84
APPNP	32.73±2.31	24.50±0.89	34.79±10.11	34.85±9.71	29.74±1.04	82.39±0.68	69.79±0.92	79.97±1.58
ARMA	37.42±1.72	24.15±0.93	39.65±8.09	28.90±10.07	27.02±2.31	79.14±1.07	69.35±1.44	78.31±1.33
GPR-GNN	33.03±1.92	24.36±1.52	33.98±11.90	38.95±12.36	28.58±1.01	82.37±0.91	69.22±1.27	79.28±2.25
BernNet	27.32±4.04	22.37±0.98	43.01±7.45	39.42±9.59	29.87±0.78	82.17±0.86	69.44±0.97	79.48±1.47
ChebNetII	**43.42±3.54**	**33.96±1.22**	46.58±7.68	42.19±11.61	30.18±0.81	82.42±0.64	69.89±1.21	79.51±1.03
JacobiConv	36.67±1.63	29.38±0.71	48.50±5.90	43.01±11.92	31.69±0.71	82.93±0.55	70.25±1.02	79.53±1.28
Specformer	36.05±3.47	29.64±0.88	50.00±8.33	43.76±5.84	31.45±0.68	81.44±0.63	66.11±0.98	78.05±1.03
LON-GNN	35.17±1.85	30.25±1.04	45.38±7.92	35.32±8.09	31.51±1.23	81.93±0.74	70.41±1.10	79.57±1.08
OptBasisGNN	35.56±2.86	31.25±1.06	37.11±5.09	32.31±7.11	31.73±0.50	78.69±0.86	63.46±1.30	77.38±0.98
GNN-LF	26.49±2.00	22.01±1.04	39.02±6.24	36.65±9.60	28.28±0.71	81.96±0.92	69.80±1.36	79.50±1.28
GNN-HF	35.57±2.26	22.36±1.26	44.80±5.67	38.79±11.62	29.15±0.78	81.15±0.78	69.68±0.73	79.10±1.19
ADA-UGNN	39.39±2.02	25.65±0.49	47.86±6.65	42.89±8.09	30.78±1.00	82.52±1.04	70.18±1.40	79.78±1.32
FE-GNN	38.23±1.66	31.67±1.60	47.40±5.90	41.21±8.96	26.20±0.76	77.00±0.74	61.24±1.26	75.63±1.33
ClenshawGCN	38.29±2.44	31.24±1.27	49.42±6.01	46.76±12.83	31.84±0.75	82.38±0.77	69.23±1.21	79.76±1.03
SAF	41.82±1.74	31.77±0.69	58.04±3.76	52.49±8.56	33.50±0.55	83.57±0.66	71.07±1.08	79.51±1.12
SAF-ϵ	41.88±2.04	32.05±0.40	**58.38±3.47**	**53.41±5.55**	**33.84±0.58**	**83.79±0.71**	**71.30±0.93**	**80.16±1.25**
Improv.[3]	14.56%	9.68%	15.37%	13.99%	3.97%	1.62%	1.86%	0.68%

complex at $O(N^3)$, can be reduced to $O(m^2 + nnz(\hat{\mathbf{L}})m)$ using Lanczos method [41] for larger graphs with $m \ll N$ denoting iterative steps and is reusable for both training and inference. We present empirical studies on both time and space overheads in Appendix E.5.

6 EXPERIMENTS

6.1 Datasets and Experimental Setup

6.1.1 ***Datasets.*** We evaluate models over 13 real-world datasets from various domains. These include three well-known homophilic graphs: Cora, Citeseer, and Pubmed [59], five commonly used heterophilic graphs: Chameleon, Squirrel [57], Cornell, Texas [52], and Actor [63], as well as five recently introduced benchmarks: Minesweeper (synthetic graph), Tolokers (crowdsourcing platform worker network), Amazon-ratings (product co-purchasing), Roman-empire (word dependency graph) [54], and Penn94 [46] (social network). Detailed statistics are provided in Appendix C.

6.1.2 ***Baselines.*** We compare SAF with 22 models: (1) MLP; (2) Basic GNNs: GCN [39] & APPNP [25]; (3) Spectral GNNs: ARMA [6], GPR-GNN [14], BernNet [33], ChebNetII [32], JacobiConv [67], Specformer [7], LON-GNN [64] & OptBasisGNN [29]; (4) Spatial GNNs: GCNII [12], PDE-GCN [20], GEN [66], NodeFormer [69], GloGNN++ [44] & MGNN [16]; (5) Unified GNNs: GNN-LF [77], GNN-HF [77], ADA-UGNN [50], FE-GNN [62] & ClenshawGCN [28].

6.1.3 ***Setup.*** To follow [32, 33, 67], we fix $K = 10$. For each dataset, we perform a grid search to tune the hyper-parameters of all models. With the best hyper-parameters, we train models with Adam optimizer [38] in 1,000 epochs using early-stopping strategy and a patience of 200 epochs, and report the mean classification accuracies with a 95% confidence interval on 10 random data splits. More experimental details can be found in Appendix D, and the codes will be made available if the paper could be accepted.

6.2 Overall Evaluation

6.2.1 ***Semi-supervised Node Classification.*** In this task, we follow the experimental protocol established by [32] and compare our models with MLP, two basic GNNs, eight popular polynomial spectral GNNs, and five unified GNNs. For data splitting on homophilic graphs (Cora, Citeseer, and Pubmed), we apply the standard division [72] with 20 nodes per class for training, 500 nodes for validation, and 1,000 nodes for testing. On the other five heterophilic graphs, we leverage the sparse splitting [14] with 2.5%/2.5%/95% samples respectively for training/validation/testing. The results are reported in Table 1, where the best results are bold and the underlined letters denote the second highest accuracy. We first observe that both SAF and SAF-ϵ substantially boosts its base model, BernNet, with gains reaching a notable 15.37%. This impressive enhancement is credited to their capacity to effectively exploit the task-beneficial information, which is implicitly encoded by spectral filtering in the spatial domain. This ability is particularly advantageous in contexts with limited supervision, where it allows effective leveraging of extra prior knowledge during training. Generally, our models outperform competitors on all datasets except for Chameleon and Squirrel, where SAF maintains a second-place rank with considerable improvements on BernNet by 14.56% and 9.68%. In these cases, ChebNetII initially surpasses our model, yet, with more training samples, our SAF manages to beats it by margins of 3.93% and 6.28% (see Table 2). Moreover, it can be seen that

[3]Improv. indicates the relative improvement of SAF over its base model, BernNet [33]. For alternative implementation using ChebNetII [32], please refer to Appendix E.4.

Table 2: Full-supervised node classification accuracy (%) ± 95% confidence interval.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
MLP	46.59±1.84	31.01±1.18	86.81±2.24	84.15±3.05	40.18±0.55	76.89±0.97	76.52±0.89	86.14±0.25
GCN	60.81±2.95	45.87±0.88	76.97±3.97	65.78±4.16	33.26±1.15	87.18±1.12	79.85±0.78	86.79±0.31
APPNP	52.15±1.79	35.71±0.78	90.64±1.70	91.52±1.81	39.76±0.49	88.16±0.74	80.47±0.73	88.13±0.33
ARMA	60.21±1.00	36.27±0.62	83.97±3.77	85.62±2.13	37.67±0.54	87.13±0.80	80.04±0.55	86.93±0.24
GPR-GNN	67.49±1.38	50.43±1.89	92.91±1.32	91.57±1.96	39.91±0.62	88.54±0.67	80.13±0.84	88.46±0.31
BernNet	68.53±1.68	51.39±0.92	92.62±1.37	92.13±1.64	41.71±1.12	88.51±0.92	80.08±0.75	88.51±0.39
ChebNetII	71.37±1.01	57.72±0.59	93.28±1.47	92.30±1.48	41.75±1.07	88.71±0.93	80.53±0.79	88.93±0.29
JacobiConv	74.20±1.03	57.38±1.25	93.44±2.13	92.95±2.46	41.17±0.64	88.98±0.46	80.78±0.79	89.62±0.41
Specformer	75.06±1.10	**65.05±0.96**	90.33±3.12	90.00±2.79	42.55±0.67	88.85±0.46	80.68±0.90	91.25±0.31
LON-GNN	73.00±2.20	60.61±1.69	87.54±3.45	84.47±3.45	39.10±1.59	89.44±1.12	81.41±1.15	90.98±0.64
OptBasisGNN	74.26±0.74	63.62±0.76	91.15±1.97	89.84±2.46	42.39±0.52	87.96±0.71	80.58±0.82	90.30±0.19
GCNII	63.44±0.85	41.96±1.02	80.46±5.91	84.26±2.13	36.89±0.95	88.46±0.82	79.97±0.65	89.94±0.31
PDE-GCN	66.01±1.56	48.73±1.06	93.24±2.03	89.73±1.35	39.76±0.74	88.62±1.03	79.98±0.97	89.92±0.38
GEN	68.82±0.96	56.05±1.04	92.30±2.30	90.49±1.80	41.08±2.08	89.21±0.54	79.40±0.59	90.40±0.24
NodeFormer	53.02±1.58	34.25±1.96	87.71±2.13	90.00±3.45	41.74±0.61	86.93±1.22	79.58±0.85	91.27±0.39
GloGNN++	72.36±0.85	60.60±1.04	91.48±1.48	89.84±3.62	41.87±1.02	87.21±0.59	79.89±0.61	86.89±0.33
MGNN	72.65±1.16	55.40±1.13	87.05±2.46	85.25±3.28	41.06±0.87	86.59±0.77	78.47±1.53	90.53±0.75
GNN-LF	53.74±1.29	36.15±0.86	76.07±2.62	78.36±2.46	38.39±0.81	88.51±0.89	79.84±0.56	89.86±0.23
GNN-HF	55.97±1.05	35.29±0.72	81.15±2.62	85.41±3.12	38.96±0.77	88.28±0.64	80.04±0.93	90.35±0.30
ADA-UGNN	61.09±1.51	42.02±1.26	84.92±3.12	83.61±3.44	41.10±0.62	88.74±0.85	79.81±1.11	90.61±0.44
FE-GNN	73.00±1.31	63.28±0.81	88.03±1.80	86.07±3.12	41.74±0.67	89.21±0.71	80.26±1.06	90.80±0.30
ClenshawGCN	74.36±0.59	62.94±1.04	91.48±1.97	91.15±2.46	41.98±0.65	88.93±0.85	78.05±0.97	91.10±0.43
SAF	**75.30±0.96**	63.63±0.81	94.10±1.48	92.95±1.97	42.93±0.79	89.80±0.69	80.61±0.81	91.49±0.29
SAF-ϵ	74.84±0.99	64.00±0.83	**94.75±1.64**	**93.28±1.80**	**42.98±0.61**	**89.87±0.51**	**81.45±0.59**	**91.52±0.30**
Improv.[3]	6.77%	12.61%	2.13%	1.15%	1.27%	1.36%	1.37%	3.01%

SAF-ϵ averagely delivers better results than SAF, benefiting from its thresholding sparsity that reduces non-local noise for efficient graph learning. However, this enhancement also incurs higher computational costs, as illustrated in both Section 5 and Appendix E.5.

6.2.2 ***Full-supervised Node Classification.*** [4] To bolster our evaluation, we expand the previously compared baselines to include six cutting-edge spatial GNN models: GCNII & PDE-GCN for capturing long-range dependency, GEN & MGNN for handling graph heterophily, and NodeFormer & GloGNN++ addressing both. For all datasets, we randomly divide them into 60%/20%/20% for training/validation/testing by following [32, 33]. Table 2 summarizes the mean classification accuracies. Our methods demonstrate superior performance across most datasets, with an exception on Squirrel where they achieve comparable results to Specformer. This notable performance is primarily attributed to our SAF's effective non-local aggregation, utilizing signed edge weights to model global label relationships. This enables our methods to outperform GNNs that are specifically tailored for long-range dependency and/or graph heterophily. Besides the eight standard datasets for node classification, our study also extends to five new benchmarks [46, 54] that focus on graph heterophily. Due to space limit, we present detailed results in Appendix E.3.

[4]We borrow this terminology from [12, 32] to denote 60%/20%/20% data splitting.

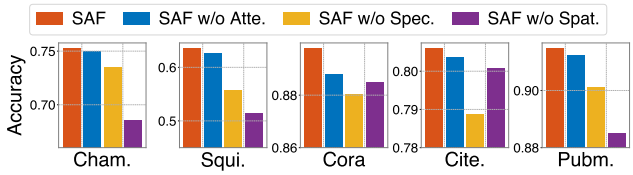


Figure 4: Ablation study of SAF framework on six datasets. Refer to Table 8 in the Appendix for comprehensive results.

6.3 Ablation Study

This section aims to validate our designs by comparing SAF with its three ablated variants – SAF w/o Atte., SAF w/o Spec., and SAF w/o Spat. – in full-supervised node classification. Specifically, Atte., Spec., and Spat. respectively refers to: attention mechanism in "Node-wise Prediction Amalgamation", "Non-negative Spectral Filtering", and "Non-local Spatial Aggregation". For SAF w/o Atte., we remove the attention mechanism and equally blend predictions from different domains. SAF w/o Spec. abandons the spectral filtering phase, practically setting $\boldsymbol{\kappa_f} = \mathbf{0}, \boldsymbol{\kappa_a} = \mathbf{1}$ in Eq. (7). As the SAF w/o Spat. configuration is equivalent to BernNet model, the corresponding results are posted directly. From Figure 4, we can draw several insights: **1)** The impact of Atte. module on our SAF

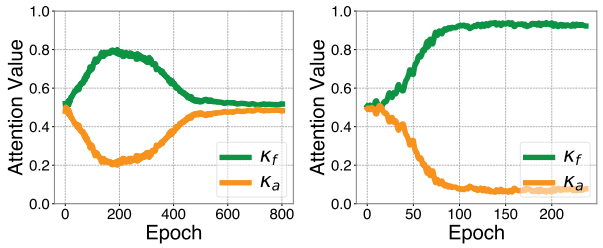
varies by datasets, e.g., on Chameleon and Squirrel, showing a slight performance reduction upon its removal. This observation aligns with our observation that their optimal attention values are close to an even split, as suggested in Figures 5(a) and 9(a) Conversely, Cora dataset exhibits a notable drop, due to its optimal attention weights being far from even, as depicted in Figure 5(b). **2)** Spectral filtering (Spec. module) remains vital for discriminative node representation learning. Specifically, the quality of the adapted new graph fundamentally hinges on the graph spectral filters' training, as underscored by their theoretical interaction in Eq. (6). Practically, the absence of spectral filtering markedly reduces model accuracy, confirming its importance in SAF. **3)** This visualization not only reaffirms the pivotal role of the non-local aggregation (Spat. module), but also underscores its position as the most crucial component in advancing spectral GNNs within the SAF framework.

6.4 Analysis of Attention Trends

We analyze the changing trends of the pair-wise attention weights during training SAF on Squirrel and Cora datasets. From Figure 5, the average weights for filtering and aggregation start similarly but diverge throughout training, showing different trends in heterophilic and homophilic graphs. On the heterophilic graph Squirrel, both weights converge to similar values, demonstrating their mutual importance in modeling complex connectivity. Conversely, $\boldsymbol{\kappa}_f$ becomes dominant on the homophilic graph Cora due to the sufficiency of node proximity information for label prediction, thereby diminishing the relevance of $\boldsymbol{\kappa}_a$ and non-local aggregation.

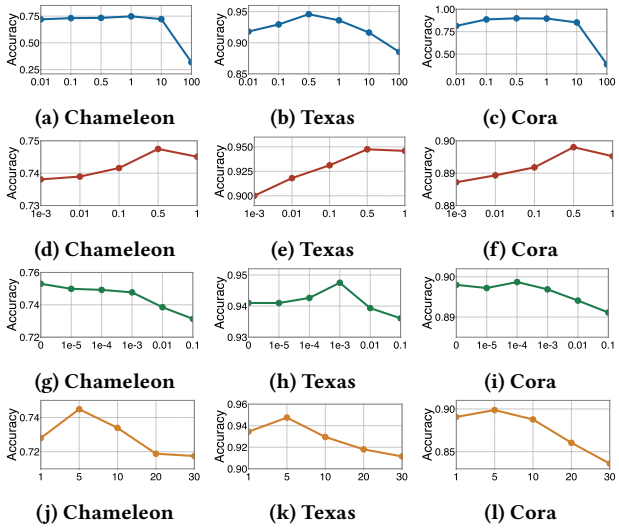
6.5 Parameter Study

This section presents the sensitivity analysis of hyper-parameters including τ, η, ϵ, and L. Figure 6 visualizes how varying these parameters within a broad range influences learning performance, showcasing our model's robust stability over diverse settings. Beyond empirical observation, we also provides deeper insights into parameter understanding and rationalizes the chosen ranges for parameter searching: **1)** The scaling parameter $\tau = \frac{\alpha}{1-\alpha}$, crucial in new graph construction in Eq. (6), stems from the trade-off parameter $\alpha \in (0, 1)$ within the graph optimization problem in Eq.(3). While theoretically we have $0 = \frac{0}{1-0} < \tau < \frac{1}{1-1} = \infty$, practical considerations for extracting structural information suggest a larger penalty on the trace objective term $\mathrm{tr}(\mathbf{Z}^T \gamma_\theta(\hat{\mathbf{L}})\mathbf{Z})$, i.e., keeping $\alpha < 0.5$, thereby limiting $\tau < \frac{0.5}{1-0.5} = 1$. This rationale substantiates our selection of τ within the set $\{0.1, 0.2, ..., 1\}$ as stated in Appendix D.2, aligning with the observed optimal performance in Figures 6(a)-(c). When addressing graphs with noisy structure, we may adjust the upper limit of α to $t \in (0, 1)$, setting τ's maximum possible value to $\frac{t}{1-t}$. For graph benchmarking evaluations in this work, where extracting structural information is important, we practically set $t = 0.5$. **2)** For the non-local aggregation layer number L, a noticeable decline in model performance is observed when L exceeds 10. This is attributed to the non-local nature of our new graph, which facilitates efficient information exchange between nodes. Exceeding a certain number of layers may potentially lead to over-smoothing, where there is an overemphasis on global information, thus degrading model performance. However, choosing the number of layers within a reasonable range generally ensures consistent and impressive model performance, as verified in Figures 6(j)-(l).



(a) Squirrel (b) Cora

Figure 5: Attention changing trends w.r.t. training epochs. Refer to Figure 9 in the Appendix E.2 for more visualizations.



(a) Chameleon (b) Texas (c) Cora
(d) Chameleon (e) Texas (f) Cora
(g) Chameleon (h) Texas (i) Cora
(j) Chameleon (k) Texas (l) Cora

Figure 6: Sensitivity analysis for hyper-parameters: τ (blue), η (red), ϵ (green), and L (orange) from top to bottom rows.

7 CONCLUSION AND FUTURE WORK

This paper introduces a fresh spatial perspective on spectral GNNs, shedding light on their interpretability. We reveal that spectral GNNs fundamentally leads the original graph to an adapted new one, which exhibits non-locality and accommodates signed edge weights. This insight leads to our proposed Spatially Adaptive Filtering (SAF) framework, enhancing spectral GNNs for more effective and versatile graph representation learning. While this study focuses on a node-level investigation, it raises intriguing questions about the implications of spectral GNNs at the graph-level in the spatial domain. Future work could expand this examination by exploring the interplay between spatial and spectral domains from a more comprehensive graph-level viewpoint.

REFERENCES

[1] Ameya Agaskar and Yue M Lu. 2013. A spectral graph uncertainty principle. *IEEE Transactions on Information Theory* 59, 7 (2013), 4338–4356.

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2623–2631.

[3] Muhammet Balcilar, Renton Guillaume, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. 2021. Analyzing the expressive power of graph neural networks in a spectral perspective. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[4] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gauzere, Sebastien Adam, and Paul Honeine. 2020. Bridging the gap between spectral and spatial domains in graph neural networks. *arXiv preprint arXiv:2003.11702* (2020).

[5] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15, 6 (2003), 1373–1396.

[6] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. 2021. Graph neural networks with convolutional arma filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[7] Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. 2023. Specformer: Spectral graph neural networks meet transformers. *arXiv preprint arXiv:2303.01028* (2023).

[8] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 3950–3957.

[9] Yunfeng Cai, Guanhua Fang, and Ping Li. 2021. A note on sparse generalized eigenvalue problem. *Advances in Neural Information Processing Systems* 34 (2021), 23036–23048.

[10] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. 2022. Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research* 23, 89 (2022), 1–64.

[11] Heng Chang, Yu Rong, Tingyang Xu, Yatao Bian, Shiji Zhou, Xin Wang, Junzhou Huang, and Wenwu Zhu. 2021. Not all low-pass filters are robust in graph convolutional networks. *Advances in Neural Information Processing Systems* 34 (2021), 25058–25071.

[12] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *International conference on machine learning*. PMLR, 1725–1735.

[13] Zhiqian Chen, Fanglan Chen, Lei Zhang, Taoran Ji, Kaiqun Fu, Liang Zhao, Feng Chen, Lingfei Wu, Charu Aggarwal, and Chang-Tien Lu. 2021. Bridging the gap between spatial and spectral domains: A unified framework for graph neural networks. *arXiv preprint arXiv:2107.10234* (2021).

[14] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive universal generalized PageRank graph neural network. In *International Conference on Learning Representations*. https://openreview.net/forum?id=n6jl7fLxrP

[15] Fan R. K. Chung. 1996. Spectral graph theory.

[16] Guanyu Cui and Zhewei Wei. 2023. MGNN: Graph Neural Networks Inspired by Distance Geometry Problem. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 335–347.

[17] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).

[18] Yushun Dong, Kaize Ding, Brian Jalaian, Shuiwang Ji, and Jundong Li. 2021. AdaGNN: Graph neural networks with adaptive frequency response filter. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 392–401.

[19] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982* (2020).

[20] Moshe Eliasof, Eldad Haber, and Eran Treister. 2021. PDE-GCN: novel architectures for graph neural networks motivated by partial differential equations. *Advances in neural information processing systems* 34 (2021), 3836–3849.

[21] Rida T Farouki. 2012. The Bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design* 29, 6 (2012), 379–419.

[22] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[23] Gerald B Folland and Alladi Sitaram. 1997. The uncertainty principle: A mathematical survey. *Journal of Fourier analysis and applications* 3 (1997), 207–238.

[24] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[25] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations (ICLR)*.

[26] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[27] Jingwei Guo, Kaizhu Huang, Xinping Yi, and Rui Zhang. 2023. Graph neural networks with diverse spectral filtering. In *Proceedings of the ACM Web Conference 2023*. 306–316.

[28] Yuhe Guo and Zhewei Wei. 2023. Clenshaw graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 614–625.

[29] Yuhe Guo and Zhewei Wei. 2023. Graph Neural Networks with Learnable and Optimal Polynomial Bases. *arXiv preprint arXiv:2302.12432* (2023).

[30] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[31] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis* 30, 2 (2011), 129–150.

[32] Mingguo He, Zhewei Wei, and Ji-Rong Wen. 2022. Convolutional neural networks on graphs with chebyshev approximation, revisited. In *NeurIPS*.

[33] Mingguo He, Zhewei Wei, Hongteng Xu, et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems* 34 (2021), 14239–14251.

[34] Werner Heisenberg. 1927. Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. *Zeitschrift für Physik* 43, 3-4 (1927), 172–198.

[35] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

[36] Mingxuan Ju, Shifu Hou, Yujie Fan, Jianan Zhao, Yanfang Ye, and Liang Zhao. 2022. Adaptive kernel graph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7051–7058.

[37] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, and Seunghoon Hong. 2022. Pure transformers are powerful graph learners. *Advances in Neural Information Processing Systems* 35 (2022), 14582–14595.

[38] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

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

[40] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. 2021. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems* 34 (2021), 21618–21629.

[41] Cornelius Lanczos. 1950. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. (1950).

[42] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. 2018. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing* 67, 1 (2018), 97–109.

[43] Xiang Li, Ben Kao, Caihua Shan, Dawei Yin, and Martin Ester. 2020. CAST: A correlation-based adaptive spectral clustering algorithm on multi-scale data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 439–449.

[44] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. 2022. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*. PMLR, 13242–13256.

[45] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard S Zemel. 2019. Lanczosnet: Multi-scale deep graph convolutional networks. *arXiv preprint arXiv:1901.01484* (2019).

[46] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. 2021. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in Neural Information Processing Systems* 34 (2021), 20887–20902.

[47] Derek Lim, Joshua Robinson, Stefanie Jegelka, and Haggai Maron. 2023. Expressive sign equivariant networks for spectral geometric learning. *arXiv preprint arXiv:2312.02339* (2023).

[48] Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. 2022. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013* (2022).

[49] Meng Liu, Zhengyang Wang, and Shuiwang Ji. 2021. Non-local graph neural networks. *IEEE transactions on pattern analysis and machine intelligence* 44, 12 (2021), 10270–10276.

[50] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. 2021. A unified view on graph neural networks as graph signal denoising. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1202–1211.

[51] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[52] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287* (2020).

[53] Oleg Platonov, Denis Kuznedelev, Artem Babenko, and Liudmila Prokhorenkova. 2023. Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond. In *The Second Learning on Graphs Conference*.

[54] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: are we really making progress? *arXiv preprint arXiv:2302.11640* (2023).

[55] Victoria Powers and Bruce Reznick. 2000. Polynomials that are positive on an interval. *Trans. Amer. Math. Soc.* 352, 10 (2000), 4677–4692.

[56] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems* 35 (2022), 14501–14515.

[57] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.

[58] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.

[59] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[60] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* 30, 3 (2013), 83–98.

[61] Daniel Spielman. 2012. Spectral graph theory. *Combinatorial scientific computing* 18 (2012), 18.

[62] Jiaqi Sun, Lin Zhang, Guangyi Chen, Peng Xu, Kun Zhang, and Yujiu Yang. 2023. Feature expansion for graph neural networks. In *International Conference on Machine Learning*. PMLR, 33156–33176.

[63] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 807–816.

[64] Qian Tao, Zhen Wang, Wenyuan Yu, Yaliang Li, and Zhewei Wei. 2023. LONGNN: Spectral GNNs with Learnable Orthonormal Basis. *arXiv preprint arXiv:2303.13750* (2023).

[65] Haorui Wang, Haoteng Yin, Muhan Zhang, and Pan Li. 2022. Equivariant and stable positional encoding for more powerful graph neural networks. In *International Conference on Learning Representations*.

[66] Ruijia Wang, Shuai Mou, Xiao Wang, Wanpeng Xiao, Qi Ju, Chuan Shi, and Xing Xie. 2021. Graph structure estimation neural networks. In *Proceedings of the Web Conference 2021*. 342–353.

[67] Xiyuan Wang and Muhan Zhang. 2022. How powerful are spectral graph neural networks. In *International Conference on Machine Learning*. PMLR, 23341–23362.

[68] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. 2020. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining*. 1243–1253.

[69] Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. 2022. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems* 35 (2022), 27387–27401.

[70] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

[71] Yuchen Yan, Yuzhong Chen, Huiyuan Chen, Minghua Xu, Mahashweta Das, Hao Yang, and Hanghang Tong. 2023. From Trainable Negative Depth to Edge Heterophily in Graphs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[72] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*. PMLR, 40–48.

[73] Jaemin Yoo, Meng-Chieh Lee, Shubhranshu Shekhar, and Christos Faloutsos. 2023. Less is more: Slimg for accurate, robust, and interpretable graph mining. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3128–3139.

[74] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2022. Graph domain adaptation via theory-grounded spectral regularization. In *The Eleventh International Conference on Learning Representations*.

[75] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI open* 1 (2020), 57–81.

[76] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems* 33 (2020), 7793–7804.

[77] Meiqi Zhu, Xiao Wang, Chuan Shi, Houye Ji, and Peng Cui. 2021. Interpreting and unifying graph neural networks with an optimization framework. In *Proceedings of the Web Conference 2021*. 1215–1226.

[78] Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. 2020. Graph geometry interaction learning. *Advances in Neural Information Processing Systems* 33 (2020), 7548–7558.

A PROOFS

A.1 Proof of Proposition 1

Proof. We begin with the assertion that the eigenvalues of $\mathbf{I} - g_\psi(\hat{\mathbf{L}})$ are positive and strictly less than 1, which fulfills the necessary condition for the Neumann series expansion stated in Lemma 1. As such, we can deduce $g_\psi(\hat{\mathbf{L}})^{-1} = (\mathbf{I} - (\mathbf{I} - g_\psi(\hat{\mathbf{L}})))^{-1} = \sum_{t=0}^{\infty}(\mathbf{I} - g_\psi(\hat{\mathbf{L}}))^t$. Owning to the prevalent polynomial approximation, we are eligible to express $g_\psi(\hat{\mathbf{L}})$ w.r.t. adjacency matrix $\hat{\mathbf{A}}$, i.e., $g_\psi(\hat{\mathbf{L}}) = g_\psi(\mathbf{I} - \hat{\mathbf{A}}) = \sum_{k=0}^{K}\pi_k\hat{\mathbf{A}}^k$ where π_k refers to the new coefficients made of up $\{\psi_m\}_{m=0}^{K}$. Substituting this polynomial representation into our Neumann expansion, we obtain $g_\psi(\hat{\mathbf{L}})^{-1} = \sum_{n=0}^{\infty}(\mathbf{I} - \sum_{k=0}^{K}\pi_k\hat{\mathbf{A}}^k)^t$. Now, revisiting $\hat{\mathbf{A}}^{\text{new}}$ in Eq. (6), we have $\hat{\mathbf{A}}^{\text{new}} = \mathbf{I} - \frac{\alpha}{1-\alpha}(g_\psi(\hat{\mathbf{L}})^{-1} - \mathbf{I}) = \mathbf{I} - \frac{\alpha}{1-\alpha}\sum_{t=1}^{\infty}(\mathbf{I} - \sum_{k=0}^{K}\pi_k\hat{\mathbf{A}}^k)^t = \sum_{t=0}^{\infty}\phi_t\hat{\mathbf{A}}^t$ where ϕ_t is a constant coefficient made up of $\{\pi_m\}_{m=0}^{K}$. □

A.2 Proof of Proposition 2

Proof. Let us commence the proof by contradiction. Let C denote the condition described in proposition 2. Assume, for the sake of contradiction, that C is not requisite for the optimal node representations $\mathbf{Z}^*$ to be predictive of node labels. Under this assumption, there are node pairs $v_i, v_j \in \mathcal{V}$ such that: (1) if $\mathbf{y}_i = \mathbf{y}_j$, $\hat{A}^{\text{new}}_{i,j} < 0$; (2) if $\mathbf{y}_i \neq \mathbf{y}_j$, $\hat{A}^{\text{new}}_{i,j} > 0$. Without loss of generality, given the non-locality as proved in proposition 1, we exclude cases where $\hat{A}^{\text{new}}_{i,j} = 0$ from our consideration. Now, consider the second objective term $\text{tr}(\mathbf{Z}^T\gamma_\theta(\hat{\mathbf{L}})\mathbf{Z})$ in Eq. (3). Using the relationship $\gamma_\theta(\hat{\mathbf{L}}) = \mathbf{I} - \hat{\mathbf{A}}^{\text{new}}$, we can expand this term into $\sum_{v_i,v_j\in\mathcal{V}}\hat{A}^{\text{new}}_{i,j}\|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2$. Under (1), for same-class nodes v_i, v_j with $\hat{A}^{\text{new}}_{i,j} < 0$, minimizing the objective term pulls $\mathbf{Z}_i$ and $\mathbf{Z}_j$ apart in the latent space. This behavior violates the canonical understanding that nodes from the same class should exhibit similar representations. Under (2), for different-class nodes v_i, v_j with $\hat{A}^{\text{new}}_{i,j} > 0$, the optimization encourages $\mathbf{Z}_i$ and $\mathbf{Z}_j$ to be more similar. This is in direct opposition to the basic classification principle that nodes from different classes should have distinct representations. Given these contradictions stemming from the mathematical implications in optimization, we must reject assumptions (1) and (2), affirming the necessary condition C for accurate label prediction by $\mathbf{Z}^*$. □

A.3 Proof of Proposition 3

Proof. We denote $p(x) = \sum_{k=0}^{K}\psi_k\binom{K}{k}(1-x)^{K-k}x^k$ as a Bernstein polynomial with $\psi_k \geq 0$ for all k and $\psi_{\max} = \max\{\psi_k\}_{k=0}^{K}$. Given $x \in [0,1]$, we can derive the following inequality as

$$\begin{aligned} p(x) &= \sum_{k=0}^{K}\psi_k\binom{K}{k}(1-x)^{K-k}x^k \leq \psi_{\max}\sum_{k=0}^{K}\binom{K}{k}(1-x)^{K-k}x^k \\ &= \psi_{\max}(1-x+x)^K = \psi_{\max}. \end{aligned}$$

Therefore, we have $p(x) \leq \max\{\psi_k\}_{k=0}^{K}$ for all $x \in [0,1]$. □

Table 3: Statistics of real-world datasets. F and C denotes the number of features and classes. Δ represents graph diameter referring to the longest geodesic distance between nodes on the graph. For Penn94, due to multiple subgraphs, we report Δ of the largest connected component.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	F	C	Δ	$\mathcal{H}$	$\mathcal{H}_{\text{class}}$	$\mathcal{H}_{\text{adjusted}}$
Chameleon	2,277	31,371	2,325	5	11	0.23	0.04	0.03
Squirrel	5,201	198,353	2,089	5	10	0.22	0.03	0.01
Texas	183	279	1,703	5	8	0.11	0.00	-0.23
Cornell	183	277	1,703	5	8	0.30	0.02	-0.08
Actor	7,600	26,659	932	5	12	0.22	0.01	0.00
Cora	2,708	5,278	1,433	7	19	0.81	0.77	0.77
Citeseer	3,327	4,552	3,703	6	28	0.74	0.63	0.67
Pubmed	19,717	44,324	500	5	18	0.80	0.66	0.69
Minesweeper	10,000	39,402	7	2	99	0.68	0.01	0.01
Tolokers	11,758	519,000	10	2	11	0.59	0.18	0.09
Amazon-ratings	24,492	93,050	300	5	46	0.38	0.13	0.14
Roman-empire	22,662	32,927	300	18	6,824	0.05	0.02	-0.05
Penn94	41,554	1,362,229	5	2	8	0.47	0.05	0.02

B EIGENDECOMPOSITION

Eigendecomposition breaks down a matrix into its eigenvalues and eigenvectors, offering insights into matrix properties, especially for the graph Laplacian. Despite computational demands, this technique has attracted surging interest in the graph learning community due to its theoretical richness, and it can be practically expedited for larger graphs using Lanczos [41] and Sparse Generalized Eigenvalue [9] algorithms. Recent advancements also underscore its value in various applications such as graph positional encoding [5, 19], spectral graph convolution [45], graph domain adaptation [74], and graph robustness [11]. For example, Laplacian eigenvectors have been widely used in identifying global position of nodes in the graph [65], particularly in recent popular graph transformers [37, 40, 56], enhancing their expressiveness. Innovations like SignNet, BasisNet [48], and Sign Equivariant [47] have further optimized the processing of these eigenvectors. When exploring the expressive power of GNN models, Specformer [7] employs eigendecomposition for learning set-to-set graph filters, TEDGCN [71] leverages it for adaptive weighting of eigengraphs, and FE-GNN [62] taps into singular value decomposition (SVD) for graph feature expansion. In line with these developments, our method, SAF, also utilizes eigendecomposition to explicitly create a new graph, enabling efficient non-local aggregation with signed weights to tackle long-range dependency and graph heterophily.

C DATASET INFORMATION

We conduct experiments on 13 real-world datasets from various domains. The detailed statistics are summarized in Table 3. Alongside standard data attributes, we also provide the longest geodesic (shortest-path) distance between graph nodes for better illustrating the non-local property that we investigate in Figures 1 and 7. Moreover, we adopt three metrics - edge homophily [76] $\mathcal{H}$, class homophily [46] $\mathcal{H}_{\text{class}}$, and adjusted homophily [53] $\mathcal{H}_{\text{adjusted}}$ - to assess the graph's homophily ratio, which ranges from 0 (high heterophily) to 1 (high homophily). While the first is a commonly used index, the latter two, considering class variability and potential imbalance, have been recently introduced for more accurate estimation. For our main text analysis regarding the adapted new graph, we primarily rely on the edge homophily metric, defined as $\mathcal{H} = |\{(v_i, v_j)|(v_i, v_j) \in \mathcal{E} \wedge \mathbf{y}_i = \mathbf{y}_j\}|/|\mathcal{E}|$, given its simplicity and wide usage. In certain compact sections of this paper, we use four-letter abbreviations for dataset names.

D EXPERIMENTAL DETAILS

In this section, we provide experimental details for reproducibility. As He et al. [32] have made a comprehensive evaluation and share the same experimental protocol with us, we directly leverage their results for models: MLP, GCN, APPNP, ARMA, GPR-GNN, BernNet, ChebNetII, GCNII and PDE-GCN on datasets including Cham., Squi., Texas, Corn., Actor, Cora, Cite., and Pubm.. For JacobiConv, LON-GNN, and OptBasisGNN, we also report their results from corresponding papers [29, 64, 67]. Other experiments are performed on a machine equipped with an NVIDIA GeForce RTX 3090 (24GB) and an Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz (20 cores).

D.1 Baseline Implementations

In our experiments, we leverage the Pytorch Geometric library [22] implementations for GCN and APPNP. For MLP, we include a sequence of linear layers, each of which is followed by batch normalization, ReLU activation, and dropout. The number of MLP layers are tuned from 1 to 5. For the remaining baselines, we resort to their publicably released code. Besides, for the work [44], only the most effective model variant, GloGNN++, is included in our experiments. Models like Geom-GCN [52] and Non-Local GNNs [49], surpassed by these SOTA methods, are excluded from our comparison.

D.2 Hyper-parameters Setting

We perform a grid search on the hyper-parameters of all models (including baselines) for each dataset using the open-source package Optuna [2] To accommodate extensive experiments across diverse datasets in both semi- and full-supervised setting, we define a broad searching space as: learning rate lr $\sim$ {1e-3, 5e-3, 1e-2, 5e-2, 0.1}, weight decay L_2 $\sim$ {0.0, 1e-6, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2}, dropout $\sim$ {0.0, 0.1, ..., 0.8} with step 0.1, non-local aggregation step $L \sim$ {1,2, ...,10} with step 1, scaling parameter $\tau \sim$ {0.1, 0.2, ..., 1.0} with step 0.1, update rate $\eta \sim$ {0.1, 0.2, ..., 1.0} with step 0.1, and threshold $\epsilon \sim$ {0.0, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2}. For other parameters specific to different base models, we strictly follow their instructions in the original papers.

E MORE EXPERIMENTS

E.1 Empirical Evidence of the New Graph

Additional empirical support for the newfound graph's properties is presented in Figures 7 and 8.

E.2 Attention Trends

For the analysis of attention trends, we provide visualizations on more datasets in Figure 9. It is noteworthy that, despite Pubmed being a homophilic graph with $\mathcal{H} = 0.80$, non-local aggregation maintains a pivotal role in our SAF, diverging from the patterns

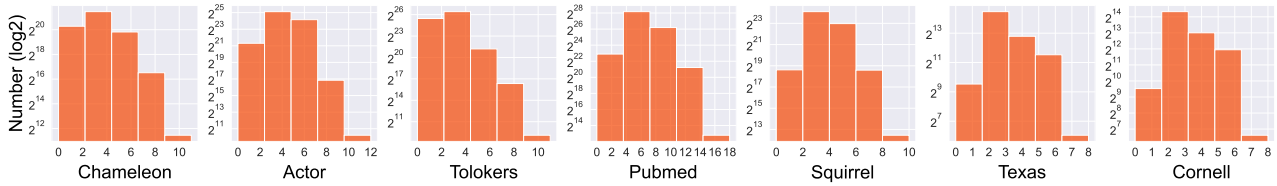


Figure 7: Distributions of original hop distance between adjacent nodes in the adapted new graph on additional seven datasets.

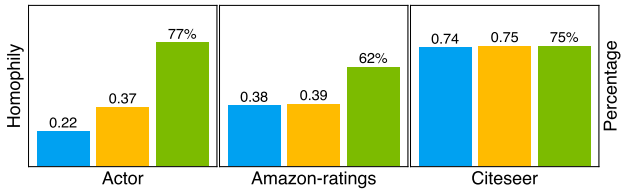
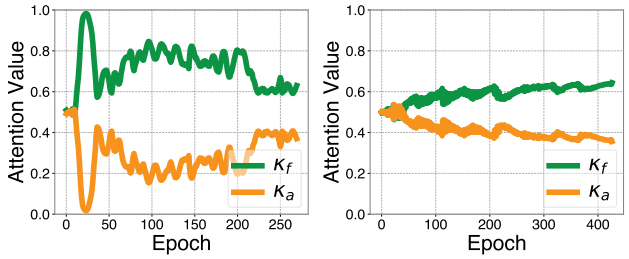


Figure 8: Additional visualizations for illustrating the new-found graph's ability in discerning label consistency.



(a) Chameleon **(b) Pubmed**

Figure 9: Additional changing trends of the averaged attention values with respect to training epochs.

observed on the Cora dataset as shown in Figure 5(b). This discrepancy owes much to the sizable node count within Pubmed, where the capability of non-local aggregation in capturing long-range dependencies proves advantageous for improving model performance.

E.3 New Benchmarks for Graph Heterophily

For a more extensive evaluation across various domains, we also test SAF on five recently introduced datasets, including Mine., Tolo., Amaz., Roma., and Penn94 [46, 54]. In this context, we draw comparisons solely with MLP, GCN, APPNP, along with six GNN models that have previously shown promising results in prior tasks, namely GPR-GNN, BernNet, ChebNetII, JacobiConv, FE-GNN, and ClenshawGCN. Table 4 lists the average classification accuracies, obtained over random splits provided by [46, 54], with a distribution of 50%/25%/25% for training/validation/testing. In summary, SAF achieves significant performance gains of 12.84% and 4.10% on Mine. and Tolo. datasets, respectively, while maintaining competitiveness on the other three.

Table 4: Evaluations on new heterophilic graph datasets

Method	Mine.	Tolo.	Amaz.	Roma.	Penn94
MLP	50.61±0.87	74.58±0.69	45.50±0.38	66.11±0.33	74.58±0.37
GCN	72.25±0.60	76.56±0.85	48.06±0.39	53.49±0.33	82.47±0.27
APPNP	68.48±1.20	74.13±0.62	48.12±0.37	72.99±0.46	75.29±0.27
GPR-GNN	89.76±0.53	75.82±0.50	49.06±0.25	73.19±0.24	81.38±0.16
ChebNetII	83.62±1.51	78.95±0.49	49.76±0.36	74.52±0.54	83.12±0.22
JacobiConv	89.88±0.33	77.24±0.39	43.89±0.28	74.30±0.50	83.35±0.11
FE-GNN	84.68±0.36	79.31±0.59	49.46±0.29	74.50±0.30	82.30±0.54
ClenshawGCN	90.36±0.92	**80.94±0.52**	50.14±0.52	73.14±0.51	**84.71±0.31**
BernNet	77.75±0.61	75.35±0.63	49.84±0.52	74.56±0.74	82.47±0.21
SAF	**90.54±0.30**	79.38±0.58	**50.49±0.28**	**74.87±0.22**	83.86±0.26
Improv.	12.79%	4.03%	0.65%	0.31%	1.39%

E.4 SAF with ChebNetII as Base Model

To expand the versatility of our SAF framework, we introduced ChebNetII [32] as an alternative base model, chosen for its adherence to the non-negative constraint, critical in our model design as stated in Section 5.1. The rationale behind this choice is ChebNetII's use of Chebyshev interpolation for learning Chebyshev polynomials, where the constraint can be ensured by keeping its learnable parameters $\{\gamma_j\}_{j=0}^{K}$ non-negative. Our experiments, as shown in Table 5, confirm that SAF can significantly enhances ChebNetII's performance, underscoring the framework's flexibility with different spectral filters. Interestingly, we observed that SAF, utilizing Bernstein polynomials (SAF-Bern), slightly surpasses its performance with Chebyshev polynomials (SAF-Cheb) in most datasets. The margin of improvement over the base model is also more pronounced with SAF-Bern. This phenomenon could be attributed to the $g_\phi(\lambda) \leq 1$ constraint within SAF (refer to Section 5.1), necessitating the rescaling of filter functions by their maximum values. For Bernstein polynomials, this maximum is readily obtained as the largest polynomial coefficient $\max\{\phi_k\}_{k=0}^{K}$, as per Proposition 3. However, for Chebyshev polynomials, the best theoretical upper bound is the sum of absolute coefficients, $\sum_{k=0}^{K}|\phi_k|$, which is comparatively less precise. This difference may impact the quality of graph construction and, subsequently, the model's performance. Exploring these nuances will be a focal point of our future research.

E.5 Time and Space Overheads

Eigendecomposition. Our SAF framework pre-computes eigendecomposition once per graph and reuses it in Eq. (6). This aspect is crucial, as the forward-pass cost in model training often exceeds the preprocessing expense of eigendecomposition. To empirically validate this, we compare the time overheads of eigendecomposition with the training times of various models in Table 6. It is

Table 5: Full-supervised node classification accuracy (%). SAF-Cheb refers to SAF implementation using Chebyshev polynomials.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
ChebNetII	71.37±1.01	57.72±0.59	93.28±1.47	92.30±1.48	41.75±1.07	88.71±0.93	80.53±0.79	88.93±0.29
SAF-Cheb	74.97±0.66	64.06±0.59	94.43±1.81	92.62±2.13	42.65±1.01	89.56±0.64	80.68±0.68	91.27±0.34
SAF-Cheb-ϵ	75.25±0.96	64.42±0.82	94.26±1.64	93.12±1.64	42.79±1.04	89.61±0.71	81.08±0.68	91.73±0.18
Improv.	3.88%	6.70%	1.15%	0.82%	1.04%	0.90%	0.15%	2.80%

Table 6: Time overheads (s) / Space overheads (MB). For large-scale graph Penn94 (with 41,554 nodes and 1,362,229 edges), we only employ a partial eigendecomposition with 100 extremal eigenvalues to balance between model effectiveness and efficiency.

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.	Penn94
BernNet	8.36/72	13.74/232	3.92/5	4.16/5	4.88/292	5.24/64	5.52/152	6.06/1546	24.05/1902
ChebNetII	22.82/72	30.73/231	11.47/5	9.64/5	14.88/291	19.96/63	16.14/152	36.91/1584	41.67/1850
ClenshawGCN	15.58/98	26.21/398	2.98/5	4.13/5	4.43/313	3.72/68	12.48/152	24.97/1833	191.30/3017
SAF	11.55/112	18.78/440	4.38/5	4.70/5	5.36/733	6.04/120	6.12/237	18.43/4515	23.49/8491
Decomposition	0.58/141	1.59/540	0.02/1	0.02/1	3.93 /1206	1.00/140	0.77 /239	21.34/7641	4.76/4

Table 7: Average running time per epoch (ms) / average total running time (s).

Method	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.
BernNet	14.60/8.36	17.40/13.74	16.10/3.92	14.30/4.16	14.40/4.88	14.90/5.24	16.20/5.52	15.10/6.06
ChebNetII	39.30/22.82	40.70/30.73	42.30/11.47	39.60/9.64	41.80/14.88	39.20/19.96	40.70/16.14	40.60/36.91
JacobiConv	11.10/6.54	11.20/10.01	11.60/3.86	11.20/2.93	11.70/4.22	11.50/5.54	11.30/6.02	11.40/8.47
NodeFormer	135.00/58.96	135.10/79.66	49.10/14.29	67.60/18.89	158.20/66.20	56.70/19.25	73.10/32.00	150.60/68.57
GloGNN++	53.60/35.63	123.70/68.31	14.80/4.47	9.80/3.00	204.80/73.13	89.60/32.68	49.60/12.35	6369.20/5266.53
ClenshawGCN	20.6/15.58	39.7/26.21	11.0/2.98	14.5/4.13	13.7/4.43	11.0/3.72	29.7/12.48	68.1/24.97
SAF-ϵ	30.70/19.11	59.50/54.71	27.30/6.95	29.70/7.86	121.20/33.14	30.20/10.32	32.00/17.19	2054.80/837.10
SAF	19.30/11.55	20.90/18.78	17.70/4.38	18.60/4.70	18.80/5.36	18.90/6.04	18.10/6.12	43.30/18.43

Table 8: Ablation study of SAF framework.

Our Variant	Cham.	Squi.	Texas	Corn.	Actor	Cora	Cite.	Pubm.	Mine.	Tolo.	Amaz.	Roma.
SAF-ϵ	74.84	64.00	94.75	93.28	42.98	89.87	81.45	91.52	90.59	79.45	50.36	74.89
SAF	75.30	63.63	94.10	92.95	42.93	89.80	80.61	91.49	90.54	79.38	50.49	74.87
SAF w/o Atte.	75.01	62.62	89.18	86.07	41.53	88.80	80.37	91.24	89.45	76.18	49.98	74.25
SAF w/o Spec.	73.55	55.70	90.49	88.20	41.06	88.03	78.87	90.12	89.45	78.23	49.13	71.85
SAF w/o Spat.	68.53	51.39	92.62	92.13	41.71	88.51	80.08	88.51	77.75	75.35	49.84	74.56

evident that for most datasets, the time consumed by decomposition is significantly less than the time required for model training. For medium-sized graphs such as Pubmed, while the full decomposition time exceeds that of BernNet, it still maintains efficiency against more advanced GNNs (ChebNetII, ClenshawGCN, SAF). Moving to the large-scale graph, Penn94, where only partial eigendecomposition with 100 extremal eigenvalues is considered, the computation time is markedly reduced compared to all models mentioned.

Model Comparison. To evaluate the efficiency of our model, we compare the running times of our proposed variants, SAF and SAF-ϵ, against four notable spectral GNNs (BernNet, ChebNetII, JacobiConv), two non-local GNNs (NodeFormer, GloGNN++), and one unified GNN model (ClenshawGCN), as detailed in Table 7. One can observe that SAF, while slightly slower than its base model, BernNet, due to the integration of non-local spatial aggregation, remains more efficient than or comparable to other SOTA methods. On the other hand, SAF-ϵ costs more time (but still faster than the non-local GNN, GloGNN++), a consequence of its quadratic complexity from non-local sparsification. However, this trade-off allows SAF-ϵ to produce an high-quality new graph, better capturing long-range interconnections among nodes and addressing graph heterophily.