MORGAN: To Bridge Mixture of Experts and Spectral Graph Neural Network

Lihui Liu[1], **Yuchen Yan**[2*]

[1] Wayne State University
[2] University of Illinois at Urbana-Champaign
hw6926@wayne.edu, yucheny5@illinois.edu

Abstract

Graph Neural Networks (GNNs) have demonstrated strong performance across a wide range of tasks by leveraging the structural properties of graph-structured data. To tackle the challenge of edge heterophily—where connected nodes may possess dissimilar labels or features—two primary families of GNNs have emerged: Mixture-of-Experts (MoE)-based spatial GNNs and frequency filtering-based spectral GNNs. MoE-based spatial GNNs intuitively assign specialized experts to different hops in the graph but often lack a solid theoretical foundation. In contrast, spectral GNNs are grounded in graph signal processing theory, yet they typically rely on handcrafted filters and ad-hoc global operators, which limits their scalability and adaptability. In this work, we uncover an inherent connection between these two paradigms by showing that *eigengraph components in spectral methods can be interpreted as experts within the MoE framework*. Building on this insight, we propose MORGAN, a novel spectral GNN that combines frequency filtering from spectral GNNs with the expert assignment strategy from MoE-based spatial GNNs. MORGAN performs eigen-decomposition of the graph Laplacian, partitions the spectrum into multiple frequency bands, and assigns a dedicated expert network to each band. A learnable gating mechanism dynamically combines the outputs of these experts based on their spectral characteristics. To support scalable and inductive learning, we further introduce MORGAN(L), a localized variant that incorporates subgraph sampling to perform spectral filtering without requiring access to the full graph Laplacian. Extensive experiments on real-world benchmark datasets demonstrate that MORGAN consistently achieves competitive or superior performance compared to state-of-the-art baselines, particularly in inductive node classification tasks under heterophilic settings.

Introduction

With the rapid advancement of big data and artificial intelligence, graph-structured data has become increasingly prevalent across a wide range of domains. In response, Graph Neural Networks (GNNs) (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018) have emerged as powerful tools for learning on graphs, demonstrating strong performance in various tasks such as node classification (Wu et al. 2019; He et al. 2022), link prediction (Liu 2025a; Liu et al. 2022, 2025), node clustering (Bianchi, Grattarola, and Alippi 2020), and knowledge graph reasoning (Yan et al. 2021; Liu, Wang, and Tong 2025; Liu et al. 2021).

Most existing GNNs (Hamilton, Ying, and Leskovec 2017; Kipf and Welling 2016) rely on the homophily assumption—that connected nodes tend to share similar labels or features. Based on this assumption, these models perform message passing primarily along directly connected node pairs, encouraging their embeddings to become similar. However, real-world graphs often exhibit edge heterophily, where directly connected nodes may have dissimilar labels or attributes. This mismatch poses a significant challenge for traditional GNN architectures.

To address the edge heterophily issue, two major categories of approaches have emerged: (1) Mixture-of-Experts (MoE)-based spatial GNNs, and (2) Frequency filtering-based spectral GNNs. MoE-based spatial GNNs (Wang et al. 2023a; Zeng et al. 2024) utilize multiple experts to capture multi-hop, long-range dependencies from the center node in the spatial domain. This design allows the model to align embeddings of distant nodes, thus mitigating the effects of heterophily. In contrast, spectral GNNs operate in the spectral domain, leveraging insights from graph signal processing: low-frequency signals correspond to homophilic patterns, while high-frequency signals capture heterophilic relationships. These models address heterophily by designing adaptive frequency filters that balance low-pass and high-pass filtering. Both categories of methods have shown promising empirical performance on node classification tasks under heterophilic settings.

However, both categories of methods suffer from critical limitations. For MoE-based spatial GNNs, the primary limitation lies in their lack of solid theoretical grounding. These methods heuristically treat neighbors at different hop distances as distinct experts. However, in real-world graphs, neighborhood boundaries are often ambiguous, and distant nodes may not differ significantly in their attributes compared to closer ones—limiting the ability of such models to truly address the edge heterophily issue. Consequently, MoE-based spatial GNNs often struggle with justifying expert assignments in a principled way. On the other hand, frequency filtering-based spectral GNNs are grounded in graph signal

[*]Two authors have equal contribution. Lihui Liu is the corresponding author.

processing theory, but they suffer from a different limitation: their filtering functions are typically manually designed and ad hoc. These handcrafted filters often depend on precomputed global operators on eigenvalues/eigenvectors of the graph Laplacian, which limit their applicability to inductive scenarios—particularly for large-scale or dynamically evolving graphs (Zhu et al. 2021b). This raises a fundamental question: *How can we effectively address the core limitations of these two paradigms?*

In this paper, we begin by analyzing these two major classes of methods and uncover an inherent connection between them. Specifically, we show that *the eigendecomposition used in frequency filtering-based spectral GNNs can be interpreted as a special case of the Mixture-of-Experts (MoE) framework*, which underpins many MoE-based spatial GNNs. Building on this insight, we propose the core idea of our work: *treating eigengraphs (i.e., outer products of eigenvectors) as experts, and learning their corresponding gating functions in the MoE style.* This approach not only combines the ideas of the spectral and spatial GNNs but also addresses the key limitations of both. It offers a principled theoretical foundation while removing the reliance on manually crafted filters by naturally aligning each expert with a distinct spectral component of the graph.

Based on this key idea, we introduce MORGAN, a novel spectral GNN framework rooted in the Mixture-of-Experts paradigm. MORGAN includes two key variants: MORGAN(G), the base model, and MORGAN(L), a localized extension designed for scalable inductive learning. In MORGAN(G), we first perform eigen-decomposition on the graph Laplacian and partition the resulting spectrum—both eigenvalues and eigenvectors—into multiple clusters, each naturally representing a specific frequency band. Then, we assign a dedicated expert to each cluster, enabling the model to learn specialized filters adapted to different spectral regions. A gating function is used to dynamically weight these experts based on the spectral characteristics of the input. To support inductive learning on large-scale graphs, we develop MORGAN(L), which combines spectral filtering with subgraph sampling strategies from spatial GNNs. Each sampled subgraph is processed independently, without requiring access to the full Laplacian of the entire graph. This hybrid approach retains the benefits of spectral signal processing while ensuring scalability and efficiency. We evaluate MORGAN on 8 real-world benchmark datasets with diverse graph structures. Results show that both variants of MORGAN achieve competitive or superior performance compared to state-of-the-art baselines, particularly on inductive node classification tasks under heterophilic conditions.

In summary, our main contributions are as follows:

- **Conceptual Insight:** We identify a theoretical connection between spatial MoE-based and spectral frequency filtering-based GNNs and propose a bridging key idea that treats eigengraphs as experts, with learnable gating functions.
- **Novel Framework:** We introduce MORGAN, a Mixture-of-Experts-based spectral GNN, and develop two variants: MORGAN(G) for general spectral modeling, and MORGAN(L) for scalable inductive learning.
- **Empirical Validation:** We conduct extensive experiments on 8 benchmark datasets, demonstrating that MORGAN consistently outperforms existing methods in inductive node classification, especially under heterophily.

Problem Definition

Notations. We adopt the following notation throughout the paper: bold uppercase letters (e.g., $\mathbf{A}$) denote matrices, bold lowercase letters (e.g., $\mathbf{u}$) represent vectors, and standard lowercase letters (e.g., α) refer to scalars. The transpose of a matrix or vector is indicated with a superscript $\top$, as in $\mathbf{A}^\top$ or $\mathbf{u}^\top$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. We define $\mathbf{X} \in \mathbb{R}^{n \times d}$ as the node feature matrix, where each node has a feature vector of dimension d. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is defined such that $\mathbf{A}_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, and 0 otherwise. The degree matrix is denoted by $\mathbf{D} = \operatorname{diag}\left(\left\{\sum_j \mathbf{A}_{ij}\right\}_{i=1}^{n}\right)$. To include self-loops, we define the augmented adjacency matrix as $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Its corresponding degree matrix is $\tilde{\mathbf{D}} = \operatorname{diag}\left(\left\{\sum_j \tilde{\mathbf{A}}_{ij}\right\}\right)$.

Graph homophily and heterophily. Homophily and heterophily describe the tendency of nodes to connect to others with the same or different labels, respectively. These notions are studied at multiple levels, including edge-level (Zhu et al. 2020; Luan et al. 2021), node-level (Pei et al. 2020), and graph-level (Lim et al. 2021) perspectives. In this work, we focus on *edge-level heterophily*, defined as the proportion of edges that connect nodes with different labels. Formally, for a graph $\mathcal{G}$:

$$h(\mathcal{G}) = \frac{1}{|\mathcal{E}|} \sum_{(v_i, v_j) \in \mathcal{E}} \mathbf{1}\{y_i \neq y_j\},$$

where y_i denotes the label of node v_i, and $\mathbf{1}\{\cdot\}$ is the indicator function.

Graph convolutional network. The layer-wise propagation rule of Graph Convolutional Network (Kipf and Welling 2016) is defined as:

$$\mathbf{H}^{(t+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(t)}\mathbf{W}^{(t)}\right), \tag{1}$$

where $\mathbf{H}^{(t)}$ and $\mathbf{H}^{(t+1)}$ denote the node embedding matrices at the t-th and $(t+1)$-th layers respectively, with $\mathbf{H}^{(0)} = \mathbf{X}$ as the input feature matrix. $\mathbf{W}^{(t)}$ is a trainable weight matrix, and $\sigma(\cdot)$ is a non-linear activation function. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ represents the adjacency matrix with added self-loops, and $\tilde{\mathbf{D}}$ is the corresponding degree matrix.

By removing the non-linearity $\sigma(\cdot)$ and collapsing all layers into a single transformation, the SGC model (Wu et al. 2019) is as the following:

$$\mathbf{H}^{(d)} = \tilde{\mathbf{S}}^d\mathbf{X}\mathbf{W}, \tag{2}$$

where $\tilde{\mathbf{S}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$ is the normalized adjacency matrix and $\mathbf{W}$ is the merged weight matrix obtained by compressing all layer-wise parameters: $\mathbf{W} = \prod_{i=0}^{t-1}\mathbf{W}^{(i)}$.

Mixture of expert. The original Mixture-of-Experts formulation (Jacobs et al. 1991) combines a set of experts (classifiers) $E_1, \ldots, E_C$ using a mixture (gating) function G that returns a distribution over the experts given the input x:

$$y = \sum_{i=1}^{C} G_i(x) \cdot E_i(x) \tag{3}$$

Here, $G_i(x)$ is the weight assigned to the i-th expert E_i.

Method

In this section, we present the details of our proposed method. We begin with a theoretical analysis that reveals a fundamental connection between Mixture-of-Experts (MoE) architectures and spectral Graph Neural Networks (GNNs), showing how expert specialization can naturally correspond to different regions of the graph spectrum. Motivated by this insight, we propose a novel MoE-based spectral GNN framework named MORGAN, which leverages spectral decomposition to facilitate frequency-aware expert learning. Within this framework, we introduce two model variants. The first, MORGAN(G), is a base model that performs eigen-decomposition of the graph Laplacian and clusters the spectrum—both eigenvalues and eigenvectors—into multiple frequency bands. Each cluster is handled by a dedicated expert, enabling the model to learn specialized spectral filters tailored to distinct frequency components. To extend this approach to large-scale or dynamic graphs, we further propose MORGAN(L), a localized variant that incorporates subgraph sampling strategies. MORGAN(L) applies spectral filtering independently within each sampled subgraph, allowing for efficient inductive learning without requiring access to the full graph Laplacian.

Analysis & Key Idea

In this subsection, we analyze the two main categories of existing methods designed to address the edge heterophily issue: (1) Mixture-of-Experts (MoE)-based spatial GNNs, and (2) Frequency filtering-based spectral GNNs. Based on a detailed examination of their motivations and limitations, we propose a novel idea to design a MoE-based spectral GNN, which integrates their strengths while mitigating their respective weaknesses.

Analysis on the motivation and limitation of MoE based spatial GNNs. To address the edge heterophily issue—where connected node pairs often have different labels—methods such as (Gasteiger, Bojchevski, and Günnemann 2022; Chien et al. 2021) intuitively aim to incorporate information from more distant nodes in the message-passing mechanism. Either explicitly or implicitly, these approaches adopt a Mixture-of-Experts (MoE) framework in the spatial domain. The classical MoE formulation is as follows:

$$\text{MoE}(x) = \sum_{j=1}^{k} g_j(x) \cdot f_j(x), \tag{4}$$

where f_j denotes the j-th expert and $g_j(x)$ the gating function.

In these MoE-based spatial GNN methods, $f_j(x)$ represents the node's attributes, while the gating function $g_j(x)$ and its corresponding experts are typically designed based on localized topological patterns such as neighbor hops, community structures, or node degrees. For example, in (Wang et al. 2023a; Zeng et al. 2024), different experts independently handle messages from 1-hop, 2-hop, or 3-hop neighborhoods.

Although this design is intuitive and straightforward, it lacks a strong theoretical foundation. The underlying assumption is that distant neighbor nodes exhibit behaviors distinct from those of closer neighbors, implying that neighbors at different distances can naturally serve as separate experts. However, this assumption does not always hold. In real-world graphs, local neighborhoods often lack clear boundaries, and similar topological structures may correspond to vastly different semantic contexts, rendering expert specialization ambiguous and fragile. Consequently, the absence of a solid theoretical basis for MoE-based spatial GNNs may limit their generalization and applicability to real-world graphs, where expert specialization tends to be opaque.

Analysis on the motivation and limitation of frequency filtering-based spectral GNNs. Frequency filtering-based spectral GNNs represent the other primary category of methods addressing the edge heterophily problem. Unlike MoE-based spatial GNNs, these methods originate from spectral graph theory and are supported by strong theoretical foundations. In spectral graph theory, the unnormalized graph Laplacian is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, while its symmetrically normalized form is given by:

$$\mathbf{L}_{\text{sym}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}. \tag{5}$$

Let $\mathbf{L}_{\text{sym}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the eigen-decomposition of $\mathbf{L}_{\text{sym}}$, where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]$ contains orthonormal eigenvectors and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues. These eigenvalues satisfy $\lambda_i \in [0, 2)$ and are typically ordered such that $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ (Lurie 1999).

These methods observe that small eigenvalues often correspond to smoothing operations over the attributes of connected node pairs, while large eigenvalues typically relate to discriminative operations on those attributes. Consequently, most approaches in this category employ specially designed frequency filters applied to the eigenvalues. The core idea is to use low-pass filters for smoothing and high-pass filters for discrimination. For example, FAGCN trains a combination of low-pass and high-pass filters to adaptively address the edge heterophily problem.

However, a major limitation of this category is that the frequency filters are manually designed and ad-hoc. This ad-hoc design reduces flexibility and scalability, meaning these methods often struggle with inductive tasks or large-scale graphs due to the computational cost of eigen-decomposition on the observed graph.

Key idea of MoE based spectral GNN. After analyzing the motivations and limitations of the above two main categories of methods, a natural question arise: *Could we combine the MoE structure from the spatial GNNs and the spectral graph theory from the spectral GNNs together to avoid their separate limitations?*

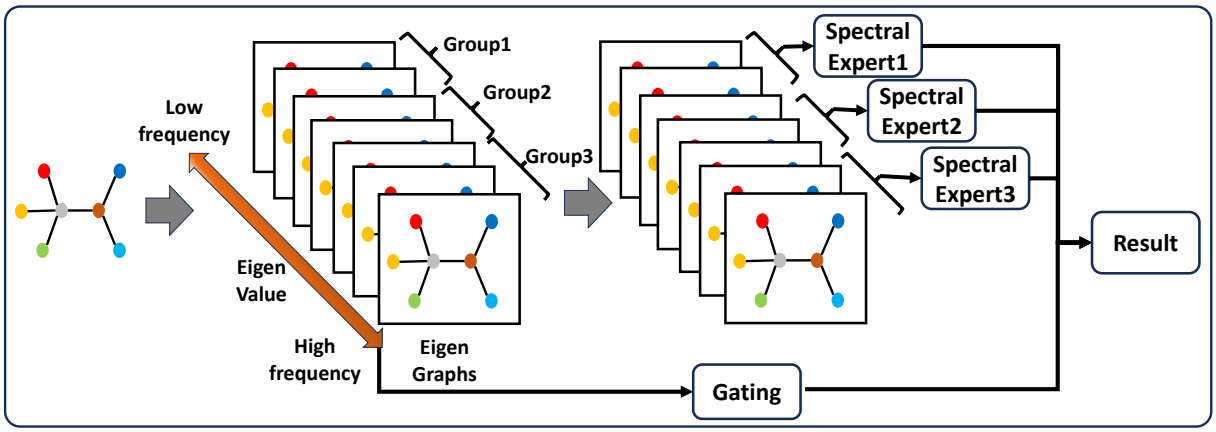

Figure 1: The framework of MORGAN.

To answer this question, we return to analyzing the spectral graph theory. According to Eq. (5) and the eigen-decomposition of the symmetrically normalized graph Laplacian $\mathbf{L}_{\text{sym}}$, we can express $\mathbf{L}_{\text{sym}}$ in the following form:

$$\mathbf{L}_{\text{sym}} = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top, \tag{6}$$

where we refer to $\mathbf{u}_i\mathbf{u}_i^\top \in \mathbb{R}^{n\times n}$ as the i-th eigengraph. Hence, the essence of $\mathbf{L}_{\text{sym}}$ is a weighted sum over eigengraphs. Similarly, for the self-loop-augmented normalized Laplacian utilized in Simplified graph convolutional network (SGC), $\tilde{\mathbf{L}}_{\text{sym}} = \mathbf{I} - \tilde{\mathbf{S}}$, we have:

$$\tilde{\mathbf{S}} = \mathbf{I} - \tilde{\mathbf{L}}_{\text{sym}} = \tilde{\mathbf{U}}(\mathbf{I} - \tilde{\mathbf{\Lambda}})\tilde{\mathbf{U}}^\top = \sum_{i=1}^{n}(1-\tilde{\lambda}_i)\tilde{\mathbf{u}}_i\tilde{\mathbf{u}}_i^\top, \tag{7}$$

where $\tilde{\lambda}_i$s and $\tilde{\mathbf{u}}_i$s are the eigenvalues and eigenvectors of $\tilde{\mathbf{L}}_{\text{sym}}$.

When applying d layers of SGC, the propagation involves multiplying $\tilde{\mathbf{S}}$ d times:

$$\tilde{\mathbf{S}}^d = \tilde{\mathbf{U}}(\mathbf{I} - \tilde{\mathbf{\Lambda}})^d\tilde{\mathbf{U}}^\top = \sum_{i=1}^{n}(1-\tilde{\lambda}_i)^d\tilde{\mathbf{u}}_i\tilde{\mathbf{u}}_i^\top, \tag{8}$$

where the term $(1-\tilde{\lambda}_i)^d$ serves as the spectral coefficient for the i-th eigengraph. Multiplying Eq. (8) with the nodes' attribution matrix $\mathbf{X}$, we have

$$\tilde{\mathbf{S}}^d\mathbf{X} = \sum_{i=1}^{n}(1-\tilde{\lambda}_i)^d\tilde{\mathbf{u}}_i\tilde{\mathbf{u}}_i^\top\mathbf{X}, \tag{9}$$

Comparing Eq. (9) with Eq. (4), we can find that SGC can be interpreted as an MoE, where $(1-\tilde{\lambda}_i)^d$ is the gating function $g_i(x)$ and $\tilde{\mathbf{u}}_i\tilde{\mathbf{u}}_i^\top\mathbf{X}$ is actually the i-th expert (i.e., $f_i(x)$).

Based on the similarity of the MoE structure and the eigen-decomposition of $\mathbf{L}_{\text{sym}}$/$\tilde{\mathbf{L}}_{\text{sym}}$, we propose the key idea of this paper: *viewing eigengraphs as experts and learning corresponding gating functions*. This key idea can achieve a solid theoretical grounding and avoid ad-hoc designs of the frequency filtering functions via naturally aligning each expert with a specific eigengraph component. With this key idea, we will present the details of our proposed MoE based spectral GNNs in next two subsections, including two variants, MORGAN-G (Global) and MORGAN-L (Local).

MORGAN-G (Global)

Motivated by the above analysis, we propose a Mixture-of-Experts (MoE) framework that adaptively models heterophilous graphs. Specifically, we treat each eigengraph $\tilde{\mathbf{u}}_i\tilde{\mathbf{u}}_i^\top\mathbf{X}$ as an expert, and replace the original eigenvalue-based operation $(1-\tilde{\lambda}_i)^d$ with a learnable weight determined by a gating function:

$$MOE(\mathbf{X}) = \sum_{i=1}^{n} g_i(\mathbf{X})\tilde{\mathbf{u}}_i\tilde{\mathbf{u}}_i^\top\mathbf{X}, \tag{10}$$

Compared to existing heuristic MoE-based spatial methods, Eq. (10) provides stronger theoretical guarantees. Unlike conventional spectral methods, it also offers greater adaptability. In particular, the learnable weights allow the model to dynamically adjust its response to different spectral bands. For example, on heterophilous graphs, the gating function tends to assign higher weights to experts associated with large eigenvalues, while on homophilous graphs, it favors experts linked to small eigenvalues.

Despite its simplicity, this method has one key limitation. The learning function requires a large number of experts—specifically, one for each of the n nodes—leading to significant computational overhead. To address this issue, rather than applying a uniform filter across all eigencomponents and using n experts, we accelerate the learning process by partitioning the eigen-spectrum into M non-overlapping clusters. The model then learns separate filters for each of these spectral bands. We denote these clusters as

$$\{1,\ldots,n\} = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_M, \quad \mathcal{C}_i \cap \mathcal{C}_j = \emptyset,\ \forall i \neq j. \tag{11}$$

Each cluster $\mathcal{C}_m$ corresponds to a frequency range and is associated with a dedicated expert module $\mathcal{E}_m$.

After partitioning the eigen-spectrum into multiple groups, each expert corresponds to a sub-spectrum and is associated

with a real-valued filter order $d_m \in \mathbb{R}$. More specifically, the spectral operation of the m-th expert can be expressed as:

$$f_m(X) = \mathbf{U}^{(m)}\mathbf{U}^{(m)\top}\mathbf{X}, \tag{12}$$

where $\mathbf{U}^{(m)}$ denotes the eigenvectors within the m-th cluster $\mathcal{C}_m$, and the exponentiation is applied element-wise. This operation enables each expert to specialize its filtering behavior according to its corresponding frequency band.

Similar to the standard MoE framework, we introduce a lightweight gating network $\mathcal{G}$ to combine the outputs of all experts. This network computes softmax-normalized weights $\boldsymbol{\alpha} \in \mathbb{R}^M$:

$$\boldsymbol{\alpha} = \text{softmax}\left(\mathcal{G}(\mu_1, \ldots, \mu_M)\right), \quad \mu_m = \text{mean}(\boldsymbol{\Lambda}^{(m)}). \tag{13}$$

Here, the output $\boldsymbol{\alpha}$ represents the learnable weight assigned to each expert, and $\mu_m = \text{mean}(\boldsymbol{\Lambda}^{(m)})$ denotes the average of the eigenvalues within cluster $\mathcal{C}_m$. The final filtered operator is constructed as a convex combination:

$$\text{MoE}(\mathbf{X}) = \sum_{m=1}^{M} \alpha_m f_m(\mathbf{X}) \tag{14}$$

MORGAN-L (Local)

In the previous section, we introduce the MORGAN-G for global graph settings. However, applying spectral graph neural networks to large-scale or inductive scenarios poses significant challenges. Spectral methods typically rely on eigen-decomposition of the graph Laplacian for the entire graph, which is computationally expensive and impractical for large or dynamically changing graphs. Furthermore, global filtering assumes access to the entire graph structure, which contradicts the inductive setting where only partial graph information is available during inference.

To address the challenges of inductive settings and large-scale graphs, we propose MORGAN-L. Our key idea is to incorporate subgraph sampling techniques from spatial GNNs. Specifically, for a given node v, we apply the eigen-decomposition to a sampled subgraph centered around v, which is significantly smaller than the entire graph. This approach allows our model to operate effectively in inductive scenarios while avoiding the costly eigen-decomposition of the entire graph, thereby enabling scalability to large graphs.

Concretely, for a given node v, we construct a local subgraph $\mathcal{G}v$ using K-hop neighborhood sampling with random node mask. All the sampled nodes and the edges between them form the subgraph G_v. Then, Based on G_v, we compute the Laplacian $\mathbf{L}_{\mathcal{G}_v}$ of the subgraph; Finally, we apply MORGAN and the representation of node in G_v is obtained as follows:

$$\text{MoE}(\mathbf{X}_{\mathbf{L}_{\mathcal{G}_v}}) = \sum_{m=1}^{M} \alpha_m f_m(\mathbf{X}_{\mathbf{L}_{\mathcal{G}_v}}) \tag{15}$$

By decomposing each subgraph into a mixture of localized spectral experts, our model combines the modularity of MoE with the mathematical rigor of spectral graph theory. This formulation ensures scalability, supports inductive inference, and offers a theoretically grounded mechanism for expert specialization via eigengraphs.

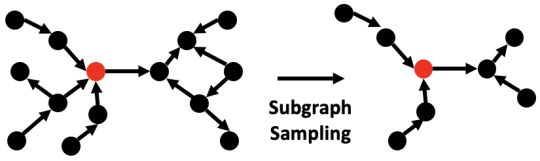

Figure 2: Subgraph sampling.

Complexity Analysis

In this subsection, we present a brief complexity analysis of MORGAN-B and MORGAN-L. Our analysis assumes a one-hop neighbor sampling strategy, where the maximum number of sampled neighbors per node, denoted by k, is empirically limited to 25, leading to a time complexity of $O(k)$ per batch. For MORGAN-B, training with a single node per batch results in N batches, where N is the total number of nodes. The computations per batch mainly include two steps: first, the eigenvalue decomposition of the symmetrically normalized Laplacian L_{sym}, which takes $O(k^3)$ time; Second, the computation of $f_m(X)$ in Eq. (12) is matrix multiplication which involves a cost of $O(k^2 f + kfc)$, where f denotes the feature dimension and c the number of classes. Therefore, the overall time complexity per batch for MORGAN-B is $O(k^3 + k^2 f + kfc)$, scaling to $O(N(k^3 + k^2 f + kfc))$ for the entire graph. For MORGAN-L with M layers, the complexity increases proportionally with the number of layers, resulting in a total time complexity of $O(NM(k^3 + k^2 f + kfc))$.

Experiment

In this section, we evaluate the performance of our method on the semi-supervised node classification task. We first describe the datasets, baseline methods, and experimental settings. Then, we present the results and perform an ablation study to assess the impact of different configurations.

Experiment Setup

Datasets. We adopt 8 datasets for evaluation. The datasets, sourced from Bojchevski and Günnemann (2018), Shchur et al. (2019), Rozemberczki, Allen, and Sarkar (2021), and Platonov et al. (2024), include CHAMELEON, SQUIRREL, SQUIRREL-FILTERED, CHAMELEON-FILTERED, MINESWEEPER, TOLOKERS, AMAZON-RATINGS, and QUESTIONS; These datasets are diverse, varying in scale, domain, and homophily/heterophily ratios. Detailed statistics of the datasets are presented in Table 2.

Baselines. We compare our method against 12 baselines, including (1) general GNN methods including GCN (Kipf and Welling 2016), CHEBNET (Defferrard, Bresson, and Vandergheynst 2016), GRAPHSAGE (Hamilton, Ying, and Leskovec 2017), GAT, APPNP, SGC, GATV2 (Brody, Alon, and Yahav 2021); (2) heterophilic graph-oriented methods including GPRGNN, H2GCN, FAGCN (Bo et al. 2021), BERNNET (He et al. 2021), and JACOBICONV.

Settings. For small-scale datasets, we employ a random split of 60%/20%/20% for train/validation/test sets and con-

Datasets	Squirrel	Chameleon	Squirrel-filt.	Chameleon-filt.	Minesweeper	Tolokers	Amazon-ratings	Questions
GCN	0.374±0.007	0.532±0.012	0.329±0.020	**0.411±0.031**	0.788±0.000	0.784±0.001	0.420±0.002	0.970±0.000
ChebNet	0.350±0.004	0.535±0.005	0.333±0.019	0.372±0.025	0.823±0.001	0.783±0.003	0.393±0.001	0.969±0.001
GraphSAGE	0.387±0.011	0.246±0.043	0.349±0.013	0.360±0.041	0.810±0.002	0.794±0.003	0.436±0.005	0.970±0.000
GAT	0.306±0.006	0.484±0.020	0.329±0.017	0.344±0.024	0.787±0.001	0.776±0.000	0.392±0.001	0.970±0.000
APPNP	0.314±0.008	0.410±0.010	0.312±0.019	0.381±0.020	0.788±0.000	0.778±0.001	0.429±0.002	0.970±0.000
SGC	0.371±0.005	0.486±0.002	0.320±0.016	0.357±0.021	0.786±0.000	0.782±0.000	0.398±0.002	0.970±0.000
GATv2	0.310±0.006	0.468±0.009	0.350±0.013	0.394±0.026	0.788±0.002	0.775±0.001	0.394±0.002	0.970±0.000
GPRGNN	0.343±0.009	0.472±0.020	0.364±0.019	0.394±0.038	0.791±0.000	0.775±0.001	0.414±0.004	0.970±0.000
H_2GCN	0.359±0.005	0.454±0.007	0.335±0.025	0.381±0.026	0.824±0.001	0.788±0.001	**0.442±0.002**	**0.971±0.000**
FAGCN	0.332±0.008	0.412±0.026	0.350±0.030	0.369±0.027	0.789±0.001	0.784±0.002	0.433±0.009	0.970±0.000
BernNet	0.361±0.007	0.578±0.007	0.361±0.020	0.374±0.030	0.788±0.000	0.772±0.007	0.398±0.002	0.969±0.001
JacobiConv	0.221±0.017	0.309±0.015	0.295±0.012	0.348±0.035	0.788±0.000	0.704±0.100	0.355±0.010	0.877±0.176
MORGAN -G	0.470±0.018	**0.612±0.057**	0.358±0.016	0.398±0.037	**0.859±0.008**	0.790±0.008	0.438±0.005	**0.971±0.001**
MORGAN -L	**0.477±0.009**	0.605±0.024	**0.376±0.020**	0.400±0.028	0.857±0.005	**0.795±0.007**	0.441±0.007	**0.971±0.001**

Table 1: Evaluation results on heterophilic datasets in the inductive setting.

Dataset	Nodes	Edges	Features	Classes	Heterophily
Chameleon	2,277	62,792	2,325	5	0.765
Squirrel	5,201	396,846	2,089	5	0.776
Chameleon-filtered	890	13,584	2,325	5	0.764
Squirrel-filtered	2,223	65,718	2,089	5	0.793
Minesweeper	10,000	39,402	7	2	0.317
Tolokers	11,758	519,000	10	2	0.405
Amazon-ratings	24,492	93,050	300	5	0.620
Questions	48,921	153,540	301	2	0.160

Table 2: Statistics of the datasets.

duct experiments in the *inductive* setting.[1] For evaluation, we report accuracy (ACC) with standard deviation (std), averaging the results over 5 runs.

Performance

Table 1 presents the performance of various graph neural network models on eight heterophilic datasets in the inductive setting. These datasets exhibit weak feature-label homophily, which poses challenges for traditional message-passing GNNs. Across all datasets, we observe that our proposed models, MORGAN-G and MORGAN-L, consistently outperform existing baselines. Specifically, MORGAN-G achieves the highest accuracy on Chameleon (0.612), and demonstrates strong performance on Squirrel (0.470), Tolokers (0.790), Amazon-ratings (0.438), and Questions (0.971). MORGAN-L further improves results on Squirrel (0.477), Squirrel-filtered (0.376), and maintains top-tier performance on other datasets as well. This indicates the robustness and adaptability of MORGAN in handling varying degrees of heterophily and graph densities. In contrast, classical GNNs such as GCN, GraphSAGE, GAT, and APPNP generally show inferior performance on most heterophilic datasets. For example, GCN achieves only 0.374 on Squirrel and 0.532 on Chameleon. GraphSAGE, while performing moderately well on Amazon-ratings (0.436), significantly underperforms on Chameleon (0.246), likely due to its reliance on neighborhood homogeneity. APPNP and SGC demonstrate stable but unspectacular performance across datasets, reflecting their limited ability to capture long-range dependencies under heterophily. Among recent heterophily-aware methods, models such as H2GCN, GPRGNN, and BernNet show improvements over traditional GNNs. H2GCN, for example, achieves 0.824 on Minesweeper and 0.442 on Amazon-ratings. BernNet also performs competitively on Chameleon (0.578) but fails to generalize well to Tolokers and Questions, suggesting limited inductive capabilities. Notably, JacobiConv performs the worst across almost all datasets, likely due to its instability and sensitivity to graph sparsity and degree variance. Overall, these results underscore the importance of explicitly modeling global structure and long-range interactions when designing GNNs for heterophilic graphs. The superior performance of MORGAN-G and MORGAN-L highlights their ability to overcome the limitations of shallow aggregation and homophily assumptions, demonstrating their effectiveness in real-world inductive tasks with complex graph structures.

Ablation Study

In this section, we conduct ablation studies to evaluate the impact of the number of experts on model performance. We focus on the inductive setting. As shown in Table 3, when the number of experts is small (e.g., 3), the performance is suboptimal. When increasing the number to 5, 7, or 9, the performance improves and becomes more stable. The performance of using 5, 7, or 9 experts are very similar.

# Experts	Chameleon	Squirrel-filt	Chameleon-filt	avg
3	0.532	0.359	0.370	0.420
5	0.612	0.358	0.398	0.456
7	0.609	0.351	0.392	0.451
9	0.603	0.349	0.399	0.450

Table 3: Effect of the Number of Experts on Accuracy

We also analyze the expert weight distribution when using 5 experts. To conserve space, we only show the results for the first four datasets. As illustrated in Figure 3, the weight distributions vary significantly across datasets. For instance, in the Squirrel and Chameleon datasets, the middle experts receive less weight, suggesting they are less informative.

[1] In the inductive setting, validation and test nodes are not seen during training.

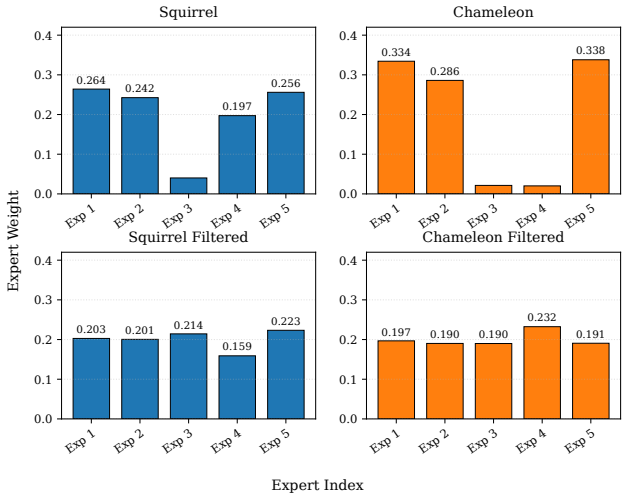

Figure 3: The expert weight distribution across different datasets. We use Squirrel, Chameleon, Squirrel Filtered, Chameleon Filtered.

In contrast, for Squirrel-filt and Chameleon-filt, all experts contribute more evenly, demonstrating that different experts specialize in capturing different properties of the data. This highlights the benefit of having multiple experts to enhance model expressiveness.

Related Work

Graph Neural Networks (GNNs). Graph neural networks are commonly divided into two main paradigms: spectral and spatial approaches (Zhang, Cui, and Zhu 2020). Spectral methods originate from spectral graph theory, where the idea is to perform convolutions in the frequency domain of a graph. One of the earliest contributions in this line of work was by Bruna et al. (2013), who defined graph convolution operations using the eigen-decomposition of the graph Laplacian. Later, ChebNet (Defferrard, Bresson, and Vandergheynst 2016) proposed using Chebyshev polynomials to avoid costly eigenvalue computations, offering a more scalable formulation. This was further streamlined by GCN (Kipf and Welling 2016), which proposed a localized, first-order approximation to graph convolution. Simplifying things even more, SGC removed the nonlinearities across layers, reducing the model to a linear form. Additional spectral-based advancements include works such as Levie et al. (2018); Li et al. (2018), among others. On the other hand, spatial methods operate directly in the node domain by aggregating information from local neighborhoods. GraphSAGE (Hamilton, Ying, and Leskovec 2017) introduced the idea of sampling and aggregating features from a node's neighbors using various aggregation functions. GAT brought attention mechanisms into GNNs, allowing the model to assign different importance scores to different neighbors. GIN (Xu et al. 2018) took a different approach by using MLPs to design more expressive and theoretically grounded aggregation functions. For readers seeking in-depth surveys on GNN models and their variants, comprehensive overviews can be found in Zhou et al. (2020) and Wu et al. (2020).

Heterophilic Graph Learning. Conventional GNNs typically rely on the assumption of homophily, where linked nodes share similar labels. However, many practical graphs exhibit heterophily, such as knowledge graph (Wang et al. 2022, 2023b, 2024; Liu 2025b; Liu et al. 2024), where this premise fails (McPherson, Smith-Lovin, and Cook 2001), prompting growing interest in specialized models. CayleyNet (Levie et al. 2018) utilizes Cayley filters refined through Jacobi updates, while ARMA (Bianchi et al. 2021) incorporates signal processing techniques via recursive filters to capture long-range dependencies. Geom-GCN enhances message passing by leveraging spatial cues in embedding space. FAGCN (Bo et al. 2021) distinguishes signal components by learning frequency-aware attention. ACM-GCN linearly combines adaptive high- and low-pass filters. CPGNN (Zhu et al. 2021a) introduces a compatibility matrix to balance different relation types. TeDGCN (Yan et al. 2023) generalizes graph convolution depth with continuous, learnable filter orders. Additional advances include methods by (Guo and Wei 2023; Geng et al. 2023; Guo et al. 2023; Yan et al. 2023; Xu et al. 2024). For a comprehensive overview, see Zheng et al. (2022).

Mixture of Expert. Jacobs et al. (Jacobs et al. 1991) introduce the original formulation of Mixture-of-Experts (MoE) models. In this work, they describe a learning procedure for systems composed of many separate neural networks, each devoted to subsets of the training data. Later work (Collobert, Bengio, and Bengio 2001; Jordan and Jacobs 1993) apply the MoE idea to classic machine learning algorithms such as support vector machines. More recently, several studies (Shazeer et al. 2017; Lepikhin et al. 2020; Fedus, Zoph, and Shazeer 2022) have proposed MoE variants for deep learning in language modeling and image recognition domains. Later work (Collobert, Bengio, and Bengio 2001) generalizes the mixture of experts formulation to a non-probabilistic setting where the gating function G outputs arbitrary weights for the experts instead of a normalized probability distribution.

Conclusion

In this work, we propose MORGAN, a novel spectral GNN framework based on the Mixture-of-Experts paradigm, to tackle the challenge of edge heterophily in graph learning. By uncovering a theoretical connection between spatial and spectral GNNs, our method unifies their respective strengths. It uses eigengraphs as learnable experts and incorporates a gating mechanism to adaptively weight their contributions based on the input. This design removes the need for handcrafted spectral filters while maintaining scalability and inductive capability through a localized variant, MORGAN(L). Extensive experiments across multiple datasets demonstrate the effectiveness and generalizability of our approach. Our approach can achieve the state of the art performance compared with different baseline methods. We believe that MORGAN provides a principled and extensible foundation for future research in spectral and heterophilic graph learning.

References

Bianchi, F. M.; Grattarola, D.; and Alippi, C. 2020. Spectral Clustering with Graph Neural Networks for Graph Pooling. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 874–883. PMLR.

Bianchi, F. M.; Grattarola, D.; Livi, L.; and Alippi, C. 2021. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3496–3507.

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

Bojchevski, A.; and Günnemann, S. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. arXiv:1707.03815.

Brody, S.; Alon, U.; and Yahav, E. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*.

Collobert, R.; Bengio, S.; and Bengio, Y. 2001. A Parallel Mixture of SVMs for Very Large Scale Problems. In Dietterich, T.; Becker, S.; and Ghahramani, Z., eds., *Advances in Neural Information Processing Systems*, volume 14. MIT Press.

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

Fedus, W.; Zoph, B.; and Shazeer, N. 2022. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity. arXiv:2101.03961.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2022. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. arXiv:1810.05997.

Geng, H.; Chen, C.; He, Y.; Zeng, G.; Han, Z.; Chai, H.; and Yan, J. 2023. Pyramid graph neural network: A graph sampling and filtering approach for multi-scale disentangled representations. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 518–530.

Guo, K.; Cao, X.; Liu, Z.; and Chang, Y. 2023. Taming over-smoothing representation on heterophilic graphs. *Information Sciences*, 647: 119463.

Guo, Y.; and Wei, Z. 2023. Graph neural networks with learnable and optimal polynomial bases. In *International Conference on Machine Learning*, 12077–12097. PMLR.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 1024–1034.

He, M.; Wei, Z.; Huang, Z.; and Xu, H. 2022. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. arXiv:2106.10994.

He, M.; Wei, Z.; Xu, H.; et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34: 14239–14251.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1): 79–87.

Jordan, M.; and Jacobs, R. 1993. Hierarchical mixtures of experts and the EM algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, 1339–1344 vol.2.

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

Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; and Chen, Z. 2020. GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding. arXiv:2006.16668.

Levie, R.; Monti, F.; Bresson, X.; and Bronstein, M. M. 2018. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1): 97–109.

Li, R.; Wang, S.; Zhu, F.; and Huang, J. 2018. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Lim, D.; Hohne, F.; Li, X.; Huang, S. L.; Gupta, V.; Bhalerao, O.; and Lim, S.-N. 2021. Large scale learning on non-homophilous graphs: new benchmarks and strong simple methods. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713845393.

Liu, L. 2025a. HyperKGR: Knowledge Graph Reasoning in Hyperbolic Space with Graph Neural Network Encoding Symbolic Path. In Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; and Peng, V., eds., *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics.

Liu, L. 2025b. Monte Carlo Tree Search for Graph Reasoning in Large Language Model Agents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, 4966–4970. New York, NY, USA: Association for Computing Machinery. ISBN 9798400720406.

Liu, L.; Du, B.; Ji, H.; Zhai, C.; and Tong, H. 2021. Neural-answering logical queries on knowledge graphs. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 1087–1097.

Liu, L.; Du, B.; Xu, J.; Xia, Y.; and Tong, H. 2022. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1098–1108.

Liu, L.; Hill, B.; Du, B.; Wang, F.; and Tong, H. 2024. Conversational Question Answering with Language Models Generated Reformulations over Knowledge Graph. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics.

Liu, L.; Wang, Z.; and Tong, H. 2025. Neural-Symbolic Reasoning over Knowledge Graphs: A Survey from a Query Perspective. *SIGKDD Explor. Newsl.*, 27(1): 124–136.

Liu, L.; Wang, Z.; Zhou, D.; Wang, R.; Yan, Y.; Xiong, B.; He, S.; and Tong, H. 2025. TransNet: Transfer Knowledge for Few-shot Knowledge Graph Completion. arXiv:2504.03720.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2021. Is Heterophily A Real Nightmare For Graph Neural Networks To Do Node Classification? arXiv:2109.05641.

Lurie, J. 1999. Review of Spectral Graph Theory: by Fan R. K. Chung. *SIGACT News*, 30(2): 14–16.

McPherson, M.; Smith-Lovin, L.; and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1): 415–444.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. arXiv:2002.05287.

Platonov, O.; Kuznedelev, D.; Diskin, M.; Babenko, A.; and Prokhorenkova, L. 2024. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? arXiv:2302.11640.

Rozemberczki, B.; Allen, C.; and Sarkar, R. 2021. Multi-scale Attributed Node Embedding. arXiv:1909.13021.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. arXiv:1701.06538.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2019. Pitfalls of Graph Neural Network Evaluation. arXiv:1811.05868.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. arXiv:1710.10903.

Wang, H.; Jiang, Z.; You, Y.; Han, Y.; Liu, G.; Srinivasa, J.; Kompella, R. R.; and Wang, Z. 2023a. Graph Mixture of Experts: Learning on Large-Scale Graphs with Explicit Diversity Modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wang, R.; Li, Z.; Sun, D.; Liu, S.; Li, J.; Yin, B.; and Abdelzaher, T. 2022. Learning to sample and aggregate: few-shot reasoning over temporal knowledge graphs. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NeurIPS '22.

Wang, R.; Li, Z.; Yang, J.; Cao, T.; Zhang, C.; Yin, B.; and Abdelzaher, T. 2023b. Mutually-paced Knowledge Distillation for Cross-lingual Temporal Knowledge Graph Reasoning. In *Proceedings of the ACM Web Conference 2023*, WWW '23.

Wang, R.; Zhang, Y.; Li, J.; Liu, S.; Sun, D.; Wang, T.; Wang, T.; Chen, Y.; Kara, D.; and Abdelzaher, T. 2024. MetaHKG: Meta Hyperbolic Learning for Few-shot Temporal Reasoning. In *Proceedings of the 47th International Conference on Research and Development in Information Retrieval*, SIGIR '24.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying Graph Convolutional Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 6861–6871. PMLR.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24.

Xu, H.; Yan, Y.; Wang, D.; Xu, Z.; Zeng, Z.; Abdelzaher, T. F.; Han, J.; and Tong, H. 2024. Slog: An inductive spectral graph neural network beyond polynomial filter. In *Forty-first International Conference on Machine Learning*.

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

Yan, Y.; Chen, Y.; Chen, H.; Xu, M.; Das, M.; Yang, H.; and Tong, H. 2023. From Trainable Negative Depth to Edge Heterophily in Graphs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yan, Y.; Liu, L.; Ban, Y.; Jing, B.; and Tong, H. 2021. Dynamic knowledge graph alignment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4564–4572.

Zeng, H.; Lyu, H.; Hu, D.; Xia, Y.; and Luo, J. 2024. Mixture of Weak and Strong Experts on Graphs. In *International Conference on Learning Representations*.

Zhang, Z.; Cui, P.; and Zhu, W. 2020. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1): 249–270.

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

Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2020. Graph neural networks: A review of methods and applications. *AI open*, 1: 57–81.

Zhu, J.; Rossi, R. A.; Rao, A.; Mai, T.; Lipka, N.; Ahmed, N. K.; and Koutra, D. 2021a. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11168–11176.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 7793–7804. Curran Associates, Inc.

Zhu, M.; Wang, X.; Shi, C.; Ji, H.; and Cui, P. 2021b. Interpreting and Unifying Graph Neural Networks with An Optimization Framework. In *Proceedings of the Web Conference 2021*, WWW '21, 1215–1226. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383127.