

# Adaptive Heterogeneous Graph Neural Networks: Bridging Heterophily and Heterogeneity

Qin Chen

State Key Laboratory of General Artificial Intelligence,  
School of Intelligence Science and Technology, Peking  
University  
Beijing, China  
chenqink@pku.edu.cn

## Abstract

Heterogeneous graphs (HGs) are common in real-world scenarios and often exhibit heterophily. However, most existing studies focus on either heterogeneity or heterophily in isolation, overlooking the prevalence of heterophilic HGs in practical applications. Such ignorance leads to their performance degradation. In this work, we first identify two main challenges in modeling heterophily HGs: (i) varying heterophily distributions across hops and meta-paths; (ii) the intricate and often heterophily-driven diversity of semantic information across different meta-paths. Then, we propose the Adaptive Heterogeneous Graph Neural Network (AHGNN) to tackle these challenges. AHGNN employs a heterophily-aware convolution that accounts for heterophily distributions specific to both hops and meta-paths. It then integrates messages from diverse semantic spaces using a coarse-to-fine attention mechanism, which filters out noise and emphasizes informative signals. Experiments on seven real-world graphs and twenty baselines demonstrate the superior performance of AHGNN, particularly in high-heterophily situations.

## CCS Concepts

- Mathematics of computing → Graph algorithms;
- Computing methodologies → Neural networks; Supervised learning.

## Keywords

graph neural networks, heterophily graphs, heterogeneous graphs

## ACM Reference Format:

Qin Chen and Guojie Song. 2025. Adaptive Heterogeneous Graph Neural Networks: Bridging Heterophily and Heterogeneity. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/XXXXXX.XXXXXX>

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CIKM '25, Seoul, Republic of Korea.

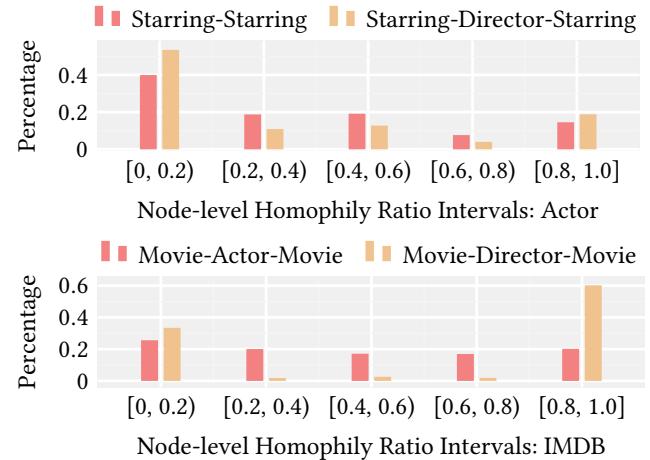
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-2040-6/2025/11  
<https://doi.org/10.1145/XXXXXX.XXXXXX>

Guojie Song\*

State Key Laboratory of General Artificial Intelligence,  
School of Intelligence Science and Technology, Peking  
University  
Beijing, China  
gjsong@pku.edu.cn

## 1 Introduction

Real-world systems often exhibit complex relationships that can be effectively modeled using heterogeneous graphs (HGs) [29], which capture a diverse range of node and edge types. While conventional Graph Neural Networks (GNNs) struggle with this diversity, specialized Heterogeneous Graph Neural Networks (HGNNS) have been developed to extract structural patterns from these networks [5, 11, 25, 37]. In parallel, many real-world graphs exhibit heterophily [26], where connected nodes are frequently dissimilar, often manifesting as label inconsistency [1]. This is in stark contrast to the homophily assumption underlying conventional GNNs [26], leading to diminished performance on heterophily graphs [40]. This challenge has spurred significant research, primarily focused on homogeneous graphs with single node and edge types [2, 3, 16, 36].



**Figure 1: Distributions of local metapath-induced homophily ratios of Actor and IMDB. For clarity, we categorized them into five bins.**

However, despite the widespread presence of real-world graphs that exhibit both heterogeneity and heterophily, this intersection remains underexplored. Our analysis reveals that **significant heterophily exists in heterogeneous graphs**. More importantly, from a meta-path perspective, **different meta-paths can exhibit distinct heterophily distributions**. To illustrate this, we visualize the node-level homophily ratio (see section 2 for formal definitions) for two widely-used heterogeneous datasets: Actor [8] and IMDB [22], as shown in Figure 1. In the Actor dataset, we observe notable

heterophily: *Actors* connected through the same *Movie* or *Director* do not consistently share similar attributes. Interestingly, the choice of meta-path significantly affects the heterophily distribution. For instance, in IMDB, the *Movie-Director-Movie* meta-path yields more extreme homophily ratios—closer to 0 or 1—compared to the *Movie-Actor-Movie* path. This demonstrates that different meta-paths involve varying degrees and patterns of heterophily.

Traditional HGNNS generally omit the heterophily issues, leading to their sub-optimal performance in some real-world HGs with inherent heterophily. Although there have been efforts to extend heterophily-oriented models to account for heterogeneity [8], these adaptations have generally yielded sub-optimal results due to lack of meta-path specific concerns, and sometimes it hurts the model on non-heterophily graphs. In fact, several widely used HGNNS outperform these rewired models, as evidenced in Table 2. This observation raises a key question: *How can we effectively model heterogeneous graphs with inherent heterophily?*

This paper highlights two key challenges in mining heterophily HGs: (i) **The variation in heterophily distribution across different hops and meta-paths.** Heterophily levels can vary significantly across different hops and meta-paths (see Figure 1 for an example). A one-size-fits-all approach to model heterophily across all hops (as induced by meta-paths) and meta-paths may be sub-optimal. Instead, an adaptive strategy tailored to the specific characteristics of each hop and meta-path is necessary to effectively capture heterophily. (ii) **the complex and often heterophily-influenced variation in semantic information across different meta-paths.** Semantic information from meta-paths becomes particularly intricate and multifaceted when these paths traverse heterophily connections and bridges disparate conceptual domains. Consequently, this complexity can render certain meta-paths irrelevant or even noisy for specific nodes or tasks, necessitating models where different nodes can selectively prioritize or filter meta-path information according to their unique context, rather than treating all meta-paths uniformly.

In response to these challenges, we propose the Adaptive Heterogeneous Graph Neural Network (AHGNN), tailored for both homophily and heterophily HGs. AHGNN comprises two components: (i) An Adaptive Heterogeneous Convolution module for hop and meta-path specific heterophily-aware convolution. (ii) A Coarse-to-Fine Semantic Fusion module for selective semantic information integration from different meta-paths. We conduct a comprehensive evaluation on twenty baselines and seven real-world datasets. AHGNN consistently achieves state-of-the-art results. Its performance boost is notably marked in strong heterophily scenarios, showing up to a 4.32% increase in Micro-F1 score. Our few-shot and synthetic experiments further support this finding. Additionally, AHGNN proves to be computationally efficient in our efficiency analysis. We also provide a theoretical analysis of the Adaptive Heterogeneous Convolution.

Generally, our contribution is summarized as follows:

- We identify the two unique challenges in modeling heterogeneous graphs with heterophily.
- We propose a novel model, AHGNN, which handles homophily and heterophily HGs adaptively while maintaining computational efficiency.

- We conducted extensive experiments where AHGNN achieves state-of-the-art performance with up to 4.32% increase in Micro-F1 score. The performance improvement is especially significant on graphs with stronger heterophily. Few-shot and synthetic experiments also corroborate this observation.

## 2 Preliminary

*Definition 1 (Meta-path Induced Sub-graph).* Given a meta-path  $\mathcal{P} = \mathcal{T}_1 \mathcal{T}_2 \cdots \mathcal{T}_L$ , with adjacency matrices  $A^{\mathcal{T}_i \mathcal{T}_j}$  representing connections between types  $\mathcal{T}_i$  and  $\mathcal{T}_j$ , the sub-graph  $\mathcal{G}_{\mathcal{P}}$  induced by  $\mathcal{P}$  is recursively defined as:  $A^{\mathcal{P}} = A^{\mathcal{T}_1} \cdots A^{\mathcal{T}_L} = A^{\mathcal{T}_1} A^{\mathcal{T}_2} \cdots A^{\mathcal{T}_L}$ , where  $A^{\mathcal{P}}$  is its adjacency matrix. The node set of  $\mathcal{G}_{\mathcal{P}}$  includes nodes  $v_i$  and  $v_j$  such that  $\phi(v_i) = \mathcal{T}_1, \phi(v_j) = \mathcal{T}_L$ , and there is a path from  $v_i$  to  $v_j$  conforming to  $\mathcal{P}$ .

*Definition 2 (Homophily Ratio for Homogeneous Graphs).* Given a homogeneous graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , the **global** homophily ratio [7, 40] of  $\mathcal{G}$  is defined via

$$h = \frac{|\{(v_i, v_j) : (v_i, v_j) \in \mathcal{E} \wedge \mathbf{y}_i = \mathbf{y}_j\}|}{|\mathcal{E}|}, \quad (1)$$

where  $\mathbf{y}_i$  denotes the label of node  $v_i \in \mathcal{V}$ .

*Definition 3 (Meta-path Induced Homophily Ratio).* Given a meta-path  $\mathcal{P} = \mathcal{T}_1 \mathcal{T}_2 \cdots \mathcal{T}_L$ , and assume that  $\mathcal{T}_1 = \mathcal{T}_L$ . Therefore, the induced graph  $\mathcal{G}_{\mathcal{P}}$  is a homogeneous graph. The homophily ratio of HG is defined via meta-path-based connection, which is a natural transition from the homophily ratio in homogeneous graphs [40]. The **global** homophily ratio of  $\mathcal{G}_{\mathcal{P}}$  is as

$$h(\mathcal{G}_{\mathcal{P}}) = \frac{|\{(v_i, v_j) : (v_i, v_j) \in \mathcal{E}_{\mathcal{P}} \wedge \mathbf{y}_i = \mathbf{y}_j\}|}{|\mathcal{E}_{\mathcal{P}}|}, \quad (2)$$

where  $\mathbf{y}_i$  denotes the label of node  $v_i$ . We further define the **local** homophily ratio of  $\mathcal{G}_{\mathcal{P}}$  for node  $v_i$  as

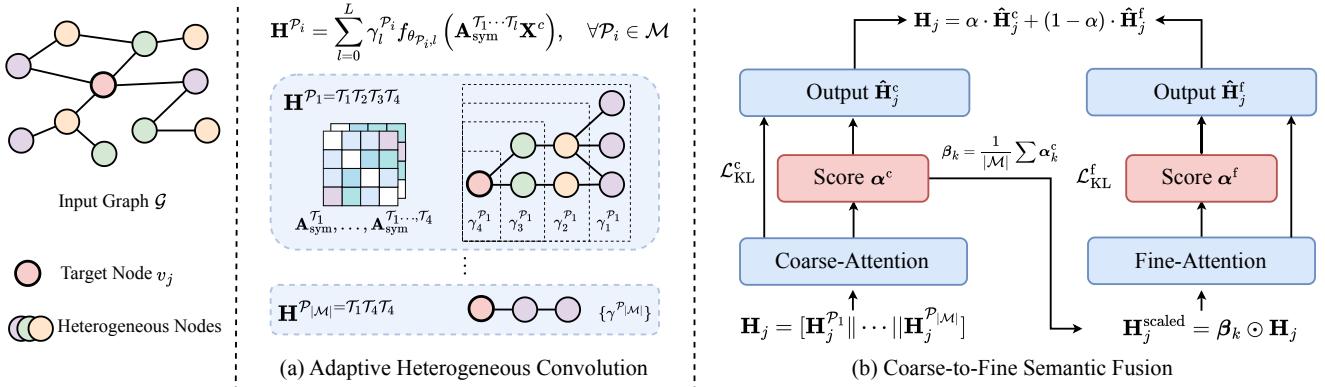
$$h_i(\mathcal{G}_{\mathcal{P}}) = \frac{|\{(v_i, v_j) : v_j \in \mathcal{N}_{\mathcal{P}, i} \wedge \mathbf{y}_i = \mathbf{y}_j\}|}{|\mathcal{N}_{\mathcal{P}, i}|}, \quad (3)$$

where  $\mathcal{N}_{\mathcal{P}, i}$  denotes the neighborhood of  $v_i$  in  $\mathcal{G}_{\mathcal{P}}$ .  $h(\mathcal{G}_{\mathcal{P}})$  describes the general similarity between node linked by a certain meta-path  $\mathcal{P}$ , and  $h_i(\mathcal{G}_{\mathcal{P}})$  is a local version only considering node  $v_i$ 's neighbors. A  $h(\mathcal{G}_{\mathcal{P}})$  up to 1 indicates a strong similarity between nodes with edges, while a  $h(\mathcal{G}_{\mathcal{P}})$  down to 0 means that there are hardly no similarities at all.

*Definition 4 (Homophily Ratio for Heterogeneous Graphs).* Given a set of meta-paths  $\mathcal{M}$ , including all paths up to length  $L$  starting with target node type  $\mathcal{T}_{\text{target}}$ , we define the graph-level homophily ratio as:

$$h = \text{Mean}(h(\mathcal{G}_{\mathcal{P}}) \mid \mathcal{P} \in \mathcal{M}, \mathcal{P} \text{ ends with } \mathcal{T}_{\text{target}}). \quad (4)$$

$h$  represents the overall homophily level of a heterogeneous graph considering all its meta-paths. In this paper, the homophily ratio for all graphs are computed with  $L = 4$ .



**Figure 2: Framework of AHGNN. It first conducts the Adaptive Heterogeneous Convolution per each meta-path for a path-specific embedding. Then for each node, the corresponding embeddings are processed with a Coarse-to-Fine Semantic fusion module, to filter out irrelevant and noisy embeddings, and prioritize informative ones.**

### 3 Model

In this section, we present the proposed Adaptive Heterogeneous Graph Neural Network (AHGNN). The framework is structured into two key stages: (i) Adaptive heterogeneous graph convolution, and (ii) Coarse-to-Fine Semantic fusion. An overview of the AHGNN framework is shown in Figure 2.

#### 3.1 Adaptive Heterogeneous Convolution

In this subsection, we propose the Adaptive Heterogeneous Convolution, which (i) propagates heterogeneous messages across the meta-paths in an efficient manner; (ii) accommodates heterophily data, aligning with the inherent characteristics of graphs where heterophily distributions can differ across hop-neighborhoods and among meta-paths.

In the field of homogeneous heterophily data mining, various studies [3, 40] have introduced methods like (i) hop-separated neighborhood aggregation and (ii) hop-separated transformation. The Adaptive Heterogeneous Convolution draws inspiration from these techniques, adapting the concept from **homogeneous** to **heterogeneous** contexts. It shifts from direct edge-linking to meta-path-based edge-linking. Embeddings along a meta-path of length  $L$  are viewed as separate messages ranging from 0-hop (central) to  $L - 1$  hop. Heterophily distributions can differ across hops [40]. For instance, a node's immediate neighborhood might predominantly exhibit heterophily, whereas its second-order neighborhood could be homophily-dominant. In heterogeneous scenarios, this variance becomes even more pronounced across different meta-paths. For instance, various meta-paths might exhibit distinct homophily distributions, as illustrated in Figure 1. This variability underscores the need for the adaptive modeling of embeddings within each meta-path (for multi-hop) and across multiple meta-paths (for multi-meta-path).

**3.1.1 Heterogeneous Message Propagation.** As outlined in section 2, for a given meta-path  $\mathcal{P}_i$ , the adjacency matrix of the sub-graph it induces is denoted as  $\mathbf{A}^{\mathcal{P}_i}$ . This matrix,  $\mathbf{A}^{\mathcal{P}_i} = \mathbf{A}^{\mathcal{T}_1 \cdots \mathcal{T}_L}$ , can be progressively calculated from shorter meta-paths. Node features,

represented by  $\mathbf{X}$ , are then propagated along the meta-path. Importantly, the message propagation for any meta-path is performed once and only once in the SGC style [35], **before training**. We enumerate all possible meta-paths starting from target node type  $\mathcal{T}_1$  up to length  $L_1$ , forming the set  $\mathcal{M}$ . For each meta-path  $\mathcal{P}_i$  in  $\mathcal{M}$ , we calculate the heterogeneous message propagation:

$$\mathbf{S}^{\mathcal{P}_i} = \left\{ \hat{\mathbf{A}}_{\text{sym}}^{\mathcal{T}_1 \cdots \mathcal{T}_l} \mathbf{X} \mid l = 1, \dots, L \right\}, \quad (5)$$

where  $\hat{\mathbf{A}}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$  is the normalized symmetric version of  $\mathbf{A}$ ,  $\mathbf{D}$  denotes the degree matrix, and  $\mathbf{S}^{\mathcal{P}_i}$  is the ordered set of embeddings, containing messages from  $\mathcal{T}_1 \cdots \mathcal{T}_L$  down to  $\mathcal{T}_1$ , in descending order of meta-path lengths. The pre-calculation enhances the efficiency of the AHGNN. For further details on its efficiency, please refer to subsection 3.4 and subsection 5.8.

**3.1.2 Adaptive Heterogeneous Convolution.** The Adaptive Heterogeneous Convolution is formulated based on the heterogeneous messages previously calculated. For a given meta-path  $\mathcal{P}_i \in \mathcal{M}$  and its associated embedding set  $\mathbf{S}^{\mathcal{P}_i}$ , the convolution is defined as:

$$\mathbf{H}^{\mathcal{P}_i} = \sum_{l=0}^L \gamma_l^{\mathcal{P}_i} f_{\theta_{\mathcal{P}_i,l}} \left( \mathbf{S}_l^{\mathcal{P}_i} \right), \quad (6)$$

where  $\mathbf{S}_l^{\mathcal{P}_i}$  represents the  $l$ -th embedding in the ordered set  $\mathbf{S}^{\mathcal{P}_i}$ . The function  $f_{\theta_{\mathcal{P}_i,l}}$  is a linear layer that projects heterogeneous messages of different dimensions into a unified latent space  $\mathbb{R}^d$ . For meta-paths with common components, the weights of  $f_{\theta_{\mathcal{P}_i,l}}$  are shared, reducing computational load and preserving correlation among shared meta-path components. For example, in meta-path *author-paper-author* and *author-paper-conference*, the projection for *author* and *author-paper* is shared across the model. Drawing inspiration from [6], the learnable parameters  $\{\gamma_l^{\mathcal{P}_i}\}$  are initialized based on the formula  $\gamma_l^{\mathcal{P}_i} = \alpha(1 - \alpha)^l$  for  $l < L - 1$ , and  $\gamma_{L-1}^{\mathcal{P}_i} = (1 - \alpha)^{L-1}$ , where  $\alpha \in (0, 1)$  is a hyper-parameter. Initially, as  $\alpha \in (0, 1)$ , more weight is assigned to longer meta-path components while they remain learnable, allowing the model to adaptively adjust the correlations between different hops.

These parameters adaptively characterize **hop-level** relations within each meta-path. AHGNN employs distinct  $\{\gamma_l\}$  sets for different meta-paths, accommodating the varied homophily distributions **across meta-paths**. The Adaptive Heterogeneous Convolution is initialized to approximate polynomial graph filters while allowing flexibility to accommodate other graph patterns, see section 4 for details.

### 3.2 Coarse-to-Fine Semantic Fusion

In heterogeneous graphs (HG), multiple meta-paths represent diverse semantic spaces. The semantic information within these spaces can vary significantly, and different nodes may prioritize certain spaces over others. This phenomenon is more profound in heterophily HGs, where certain spaces may be helpful for some nodes while being less influential, or even harmful to others. This raises a critical question in modeling heterophily HGs: *Among diverse meta-paths, how to selectively choose the most informative ones that are representative and beneficial for downstream tasks?*

To address this challenge, we propose a two-level attention mechanism. The first level applies **coarse-grained attention** across all available meta-paths to identify informative relationships within the heterogeneous graph. Based on the evaluated importance, the meta-paths are re-weighted, and **fine-grained attention** is then applied. This approach eliminates the need for manual meta-path selection by learning which meta-paths are most relevant to the task. Furthermore, a **KL divergence loss** is employed to encourage specialization among attention heads at each level, ensuring they capture diverse and complementary graph semantics for enhanced expressiveness.

**3.2.1 Coarse-Grained Attention.** The first step applies **coarse-grained attention** to assess the relevance of each meta-path to the task. Formally, for a graph  $G$  with a meta-path set  $\mathcal{M}$  of size  $S = |\mathcal{M}|$ , the Adaptive Heterogeneous Convolution module generates embeddings  $\{\mathbf{H}^{\mathcal{P}_i} \in \mathbb{R}^{N \times d} \mid \mathcal{P}_i \in \mathcal{M}\}$ . These embeddings are vertically stacked to form an input sequence  $\mathbf{H} \in \mathbb{R}^{N \times S \times d}$ , considering all components in  $\mathcal{M}$ . For target node  $v_j$ , the Coarse-Grained Attention is formulated as<sup>1</sup>:

$$\begin{aligned} \mathbf{Q} &= \mathbf{H}_j \mathbf{W}^Q, \quad \mathbf{K} = \mathbf{H}_j \mathbf{W}^K, \quad \mathbf{V} = \mathbf{H}_j \mathbf{W}^V \\ \boldsymbol{\alpha}^c &= \text{softmax} \left( \frac{\mathbf{Q} \mathbf{K}^\top}{\sqrt{d}} \right), \quad \hat{\mathbf{H}}_j^c = (\boldsymbol{\alpha}^c \mathbf{V}) \mathbf{W}^O, \end{aligned} \quad (7)$$

where  $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V, \mathbf{W}^O$  are projection matrices.

**3.2.2 Soft Meta-path Selection with Influence Factors.** We then utilize  $\boldsymbol{\alpha}^c$  to filter out the irrelevant or harmful meta-paths. Instead of directly omitting certain candidates which may potentially discard useful but subtle signals, we introduce a soft meta-path weighting technique through a *influence factor*  $\beta \in \mathbb{R}^S$ , which scales the contribution of each meta-path embedding based on its importance:  $\beta_k = \frac{1}{|\mathcal{M}|} \sum \boldsymbol{\alpha}_k^c$ , where  $\boldsymbol{\alpha}_k^c$  represents the attention weight for the  $k$ -th meta-path based on how informative it is for the current node.  $\beta$  softly modulating the strength of each meta-path by reweighting them as:

$$\mathbf{H}_j^{\text{scaled}}[k] = \beta_k \cdot \mathbf{H}_j[k], \quad (8)$$

<sup>1</sup>For clarity, we describe the procedure using single-head attention, although it is implemented with multi-head attention in practice.

where  $\mathbf{H}_j[k] \in \mathbb{R}^d$  denotes the embedding of node  $v_j$  under meta-path  $\mathcal{P}_k$ .

**3.2.3 Fine-Grained Attention on Modulated Meta-paths.** We then apply **fine-grained attention** to the scaled meta-path embeddings to refine node representations, attending more carefully to interactions between selected meta-paths and task-relevant local semantics. This is formalized as:

$$\begin{aligned} \mathbf{Q}^f &= \mathbf{H}_j^{\text{scaled}} \mathbf{W}^Q, \quad \mathbf{K}^f = \mathbf{H}_j^{\text{scaled}} \mathbf{W}^K, \quad \mathbf{V}^f = \mathbf{H}_j^{\text{scaled}} \mathbf{W}^V, \\ \boldsymbol{\alpha}^f &= \text{softmax} \left( \frac{\mathbf{Q}^f \mathbf{K}^{f\top}}{\sqrt{d}} \right), \quad \hat{\mathbf{H}}_j^f = (\boldsymbol{\alpha}^f \mathbf{V}^f) \mathbf{W}^O, \end{aligned} \quad (9)$$

where  $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V, \mathbf{W}^O$  are projection matrices.

**3.2.4 Final Fusion and KL Divergence Regularization.** The final node representation is obtained by combining coarse and fine-level outputs via a learnable weighted sum:

$$\mathbf{H}_j = \alpha \cdot \hat{\mathbf{H}}_j^c + (1 - \alpha) \cdot \hat{\mathbf{H}}_j^f, \quad (10)$$

where  $\alpha \in [0, 1]$  is a learnable scalar parameter initialized to 0.5.

This framework enables the model to automatically identify, weigh, and refine semantically meaningful meta-paths, making it powerful for challenging heterophily settings in heterogeneous graphs.

### 3.3 Implementation Details

This subsection outlines additional implementation details of AHGNN to enhance its performance.

**Label Propagation.** The label propagation process closely mirrors the Heterogeneous Message Propagation described in subsection 3.1. It is pre-computed before the training for all meta-paths starting from  $\mathcal{T}_1$  up to a maximum length  $L_2$  in the graph:

$$\begin{aligned} \hat{\mathbf{s}}^{\mathcal{P}_i} &= \left\{ \hat{\mathbf{A}}_{\text{sym}}^{\mathcal{T}_1 \dots \mathcal{T}_L} \mathbf{Y}^{\text{train}, c} \mid l = 1, \dots, L \right\} \\ \hat{\mathbf{H}}^{\mathcal{P}_i} &= \sum_{l=0}^L \gamma_l f_{\theta_{\mathcal{P}_i, l}}' \left( \hat{\mathbf{s}}_l^{\mathcal{P}_i} \right), \end{aligned} \quad (11)$$

where  $f_{\theta_{\mathcal{P}_i, l}}$  is a linear layer. The key difference lies in the propagation of one-hot labels  $\mathbf{Y}^{\text{train}, c}$  from the training set rather than node features. These generated embeddings are then appended to the Gated Transformer's input sequence. For instance, if there are  $S$  meta-paths with messages and  $S'$  meta-paths with label-messages, then  $\mathbf{H}_j$  in Equation 7 will have shape  $\mathbb{R}^{(S+S') \times d}$ .

**$L_2$ -Normalization.** In line with the approach in [22], we employ an  $L_2$ -Normalization technique for the final embedding  $\tilde{\mathbf{H}}_j$  in Equation 10:  $\tilde{\mathbf{H}}_j = \frac{\mathbf{H}_j}{\|\mathbf{H}_j\|_2}$ .

### 3.4 Complexity Analysis

The computational complexity of the proposed AHGNN model is near-linear. For **Adaptive Heterogeneous Convolution**, the Heterogeneous Message Propagation calculated before the training has a complexity of  $O(LEF)$  per meta-path, where  $L$  is the meta-path length,  $E$  is the average number of edges in subgraphs, and  $F$  is the feature dimension. During convolution, the complexity is

$\mathcal{O}(LNdF)$  per meta-path, with  $N$  as the number of nodes and  $d$  the hidden dimension. Label propagation adds  $\mathcal{O}(LNdC)$ , where  $C$  is the number of classes. For **Coarse-to-fine Semantic Fusion**, each node processes  $S$  meta-paths, yielding input tokens  $\mathbf{H}_i \in \mathbb{R}^{S \times d}$ . Fusion has a per-node complexity of  $\mathcal{O}(S^2d)$  and a total graph complexity of  $\mathcal{O}(NS^2d)$ . Generally, the complexity of AHGNN is approximately linear to  $N$ .

## 4 Theoretical Analysis

Here, we present a theoretical analysis of Adaptive Heterogeneous Convolution from the perspective of graph filtering. We establish the connection between the proposed Adaptive Heterogeneous Convolution and polynomial graph filters. To simplify our approach, we initially focus on homogeneous graphs.

**Notations.** Consider a homogeneous, connected graph  $\mathcal{G}$  with  $N$  nodes, represented by its adjacency matrix  $\mathbf{A}$  and degree matrix  $\mathbf{D}$ . The normalized adjacency matrix is given by  $\hat{\mathbf{A}}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ , and its eigenvalue decomposition can be expressed as  $\hat{\mathbf{A}}_{\text{sym}} = \mathbf{U} \Lambda \mathbf{U}^\top$ . Let  $\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{N-1}$  denote the eigenvalues of  $\hat{\mathbf{A}}_{\text{sym}}$ . Referring to  $\hat{\mathbf{A}}_{\text{sym}}^{\mathcal{T}_1 \dots \mathcal{T}_l}$  in Equation 5, which can be rewritten as  $\hat{\mathbf{A}}_{\text{sym}}^l$  in homogeneous settings, the corresponding polynomial graph filters for  $l = 0, 1, \dots, L$  are given by  $\sum_{l=0}^L \gamma_l \hat{\mathbf{A}}_{\text{sym}} = \mathbf{U} \beta_{\gamma, L}(\Lambda) \mathbf{U}^\top$ , where  $\beta_{\gamma, L}(\Lambda)$  is a diagonal matrix with  $\beta_{\gamma, L}(\Lambda)_{i,i} = \beta_{\gamma, L}(\lambda_i) = \sum_{l=0}^L \gamma_l \lambda_i^l$ .

We begin the analysis by establishing the following lemma:

**LEMMA 5.** *For a connected graph, the largest eigenvalue  $\lambda_0$  of the normalized adjacency matrix is equal to 1, and all other eigenvalues  $\lambda_i$  satisfy  $\lambda_i < 1$  for  $i = 1, 2, \dots, N - 1$ .*

**PROOF.** The normalized adjacency matrix  $\hat{\mathbf{A}}_{\text{sym}}$  is symmetric, constructed by normalizing the adjacency matrix with the degrees of the nodes. This normalization ensures that the row and column sums are balanced, leading to the largest eigenvalue being 1. Since  $\lambda_0 \geq \lambda_i$  for  $i = 1, 2, \dots, N - 1$ , if there were another eigenvalue equal to 1, say  $\lambda_1 = 1$ , it would imply the existence of a corresponding eigenvector. However, this would contradict the Perron-Frobenius theorem [27], which asserts that the largest eigenvalue of a non-negative matrix associated with a connected graph is unique. Thus, all other eigenvalues  $\lambda_i$  satisfy  $\lambda_i < 1$  for  $i = 1, 2, \dots, N - 1$ .  $\square$

Then, the following holds:

**THEOREM 6.** *By setting  $\gamma_l > 0$  (via setting  $\alpha > 0$ ) for  $l = 0, 1, \dots, L$  and  $\sum_{l=0}^L \gamma_l = 1$ , if there exists  $l_i \geq 0, i \neq 0$ , then the Adaptive Heterogeneous Convolution  $\beta_{\gamma, L}$  is initialized to a low-pass graph filter with  $|\beta_{\gamma, L}(\lambda_i)| < 1$  strictly for any  $i = 1, 2, \dots, N - 1$ .*

**PROOF.** Given the definition of  $\beta_{\gamma, L}$  and the setting of  $\gamma_l$ , we have:

$$\beta_{\gamma, L}(\lambda_0) = \sum_{l=0}^L \gamma_l \lambda_0^l = \sum_{l=0}^L \gamma_l = \sum_{l=0}^{L-1} \alpha(1-\alpha)^l + (1-\alpha)^L = 1. \quad (12)$$

Since  $\gamma_k > 0$ , and following Lemma Theorem 5, we have:

$$|\beta_{\gamma, L}(\lambda_i)| = \left| \sum_{l=0}^L \gamma_l \lambda_i^l \right| \leq \sum_{l=0}^L \gamma_l |\lambda_i|^l \leq \sum_{l=0}^L \gamma_l 1^l = \sum_{l=0}^L \gamma_l = 1. \quad (13)$$

It is important to note that  $\sum_{l=0}^L \gamma_l |\lambda_i|^l = \sum_{l=0}^L \gamma_l \cdot 1^l$  cannot be achieved, as we assume the existence of at least one  $l_i \geq 0$  with  $i \neq 0$  and  $\lambda_i < 1$ , as established in Lemma 5. Therefore, the following inequality holds:  $\left| \frac{\beta_{\gamma, L}(\lambda_i)}{\beta_{\gamma, L}(\lambda_0)} \right| < 1$ .  $\square$

The Adaptive Heterogeneous Convolution is initially designed to approximate a low-pass filter, which facilitates smooth learning of the graph from the outset. We make the coefficients  $\gamma_l$  learnable, enabling the model to adaptively adjust the relationships between different hops (or meta-path-based hops). This approach effectively addresses the challenge posed by complex real-world heterogeneous graphs, where different meta-paths often exhibit distinct and unique intra-hop distributions and relationships (as illustrated in Figure 1).

Fixing  $\gamma_l$  to strictly follow a low-pass filter can be sub-optimal, whereas allowing meta-path-specific  $\gamma_l$  provides AHGNN with the flexibility to effectively handle multiple informative meta-paths. Additionally, this learnable coefficient design allows the model to accommodate potential high-frequency signals within the graphs that are common in heterophily graphs [40].

## 5 Experiments

### 5.1 Datasets

**Table 1: Statistics of real-world datasets.**  $F_{\text{target}}$  is the feature dimension of target nodes.  $h$  for graph-level homophily ratio defined in Definition 4.

	#Nodes	#Edges	#Class	$F_{\text{target}}$	$h$
DBLP	26108	239566	4	334	0.81
IMDB	11616	34212	3	3066	0.59
ACM	9040	547814	4	1902	0.88
FB-American	9473	495790	3	6386	0.53
FB-MIT	9274	561700	3	6440	0.49
Actor	16255	72425	7	5362	0.29
Ogbn-mag	1939743	42182144	349	128	0.34

We evaluate AHGNN with seven real-world heterogeneous graphs with various fields, scales and homophily ratios. ACM and DBLP [22] are a homophily-based citation network. IMDB [22], represents a heterophily database of online movies and TV shows. FB-American [8] and FB-MIT [31], part of the FB100, detail Facebook users in American universities. They are also heterophily datasets. Actor [30] is a strong heterophily dataset about actors, directors, and writers based on Wikipedia pages. Lastly, Ogbn-mag [10] is a large-scale heterogeneous citation network. For all datasets, the target node type is selected according to the settings in their respective original papers.

### 5.2 Involved Baselines

To comprehensively evaluate the performance of AHGNN, we conduct experiments across four categories of models: (i) **Homogeneous GNNs**: Standard models designed for homogeneous graphs, including GCN [13] and GAT [32]. (ii) **Traditional Heterogeneous GNNs**: Models specifically developed for heterogeneous graphs, such as HetGNN [38], HGT [11], MAGNN [5], SHGN [22],

SeHGNN [37], HINormer [25], LSMPS [14], and Seq-HGNN [4]. (iii) **Heterophily-aware Heterogeneous GNNs:** Methods that handle both heterogeneity and heterophily, including Hetero<sup>2</sup>Net [15], LatGRL [28] and H<sup>2</sup>Gformer [18]. (iv) **GNNs Adapted for Heterogeneity:** We adapt heterophily-aware models using HDHGR techniques [8], denoting the adapted versions with a "-HD" suffix. This group includes H2GCN-HD [40], GPRGNN-HD [3], GloGNN-HD [16], ACMGNN-HD [21], and ALTGNCN-HD [36]. Additionally, we equip two state-of-the-art HGNNs with HDHGR: LSMPS-HD [14] and SeqHGNN-HD [4].

### 5.3 Settings

We evaluate all the models on the node classification task [13]. We adopt a train/validation/test split ratio of 60%/20%/20% for all the datasets. We use the Adam optimizer [12] with a learning rate  $\eta \in \{0.5, 1, 5\} \times 10^{-3}$ , a maximum weight decay of  $5 \times 10^{-6}$ , and a maximum of 200 epochs to train AHGNN. For all models, the hidden dimensions are set to 256 for fairness (except for ogbn-mag we adopt 512). We search the hyper-parameters in the same scope mentioned above for baselines without public training scripts. For ogbn-mag, We adopt a four-stage training strategy with a maximum of 300 epochs for all models. No additional embeddings are adopted. For AHGNN, we choose  $L_1$  in {2, 3, 4},  $L_2$  in {2, 3, 4} and  $\alpha$  in {0.25, 0.4, 0.6, 0.85}. We set  $\lambda_1 = \lambda_2 = 10^{-4}$  in this paper. We run the experiments with five NVIDIA RTX 4090 with 24GB GPU Memory.

### 5.4 Main Evaluation Results

As shown in Table 2 and Table 3<sup>2</sup>, AHGNN achieves state-of-the-art performance, particularly on heterophilous datasets. On heterophily graphs, AHGNN effectively disentangles complex heterophily distributions. For homophilous graphs, it also remains competitive. Notably, on the strongly heterophilous Actor dataset, AHGNN outperforms baselines with up to a 4.32% improvement in Micro-F1.

Key observations are as follows: (i) Conventional HGNNs perform well on homophilous graphs (e.g., DBLP) but show limited effectiveness on heterophilous graphs. (ii) Heterophily-aware models, such as Hetero<sup>2</sup>Net, LatGCL, H<sup>2</sup>Gformer, and rewired variants (denoted by the -HD suffix), remain suboptimal on datasets with mixed homophily and heterophily (e.g., IMDB, FB-MIT, FB-American), often underperforming even compared to standard HGNNs. (iii) While HDHGR techniques generally enhance performance on heterophilous graphs, they may degrade performance on homophilous datasets; for example, LSMPS-HD and SeqHGNN-HD perform worse than their original versions on ACM and DBLP. In general, most baselines fail to address the two challenges outlined in section 1, resulting in suboptimal performance.

### 5.5 AHGNN under Few-shot Scenarios

We conduct a few-shot experiment, as detailed in Table 4, where 20 nodes per class were randomly selected for the training set, and the remaining nodes were split evenly between validation and test sets. AHGNN consistently outperforms other models across all

<sup>2</sup>HDHGR techniques are computationally intensive, leading to Out-Of-Memory (OOM) issues on the Ogbn-mag dataset, which contains 42 million edges.

datasets, with particularly notable improvements on IMDB, FB-MIT, and Actor. On the Actor dataset, AHGNN achieves approximately a 3% increase in Micro-F1 scores. These results further validate AHGNN's effectiveness under few-shot settings.

### 5.6 AHGNN with Varying Heterophily Ratios

We further conduct an experiment using a synthetic dataset, syn-DBLP, to evaluate the performance of AHGNN across different heterophily ratios compared to other models. Following [40], by randomly assigning edges between nodes, we control the graph-level homophily of syn-DBLP to range from 0.8 evenly to 0.1. Figure 3 shows the results, where AHGNN generally achieves more notable performance advancement as the homophily ratio decreases. Such results are consistent with the observation in Table 1, the stronger the heterophily, the larger the performance advancement.

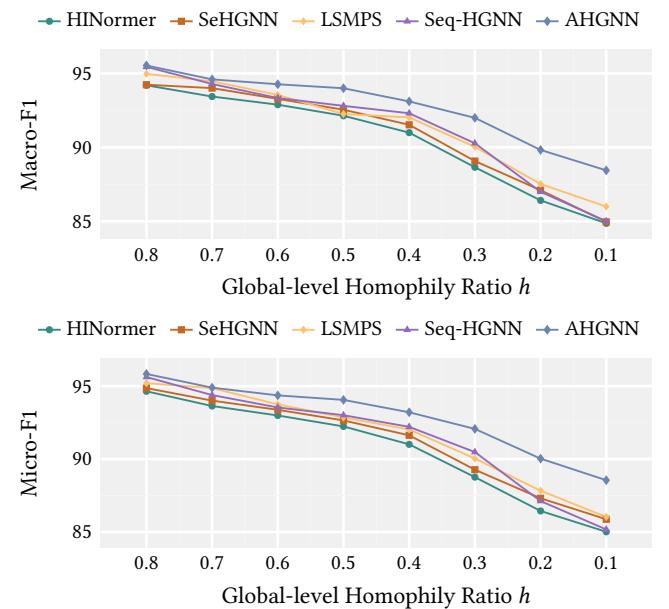


Figure 3: Results on syn-DBLP presented in Macro-F1 and Micro-F1 (scaled up by 100 for clarity).

### 5.7 Ablation Study

To assess the effectiveness of individual components in AHGNN, we conduct an ablation study by systematically disabling each component in isolation. For the *Adaptive Heterogeneous Convolution*, we fix  $\gamma = 1$  uniformly across all meta-paths. For *Coarse-to-Fine Semantic Fusion*, we replace it with a simple mean of the embeddings. The results, presented in Table 5, yield several insights: (i) Components contribute to performance improvements, with Adaptive Heterogeneous Convolution being the most critical. (ii) The impact of each component varies across datasets, with the largest gains observed on Actor, likely due to its stronger heterophily compared to IMDB and FB-MIT.

**Table 2: Results on real-world datasets presented in Macro-F1 and Micro-F1 (scaled up by 100 for clarity), including mean and standard deviation over all runs and splits. The graph-level homophily ratio of each dataset (as defined in Definition Definition 4 in section 2) is displayed in brackets ( $h$ ). The best results are highlighted in gray.**

	Actor (0.29)		FB-MIT (0.49)		FB-American (0.53)		IMDB (0.59)		DBLP (0.81)		ACM (0.88)	
	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1
GCN	54.18 $\pm$ 0.33	64.99 $\pm$ 0.42	69.03 $\pm$ 2.01	72.05 $\pm$ 1.83	68.38 $\pm$ 1.24	71.83 $\pm$ 0.79	57.88 $\pm$ 1.18	64.82 $\pm$ 1.24	90.01 $\pm$ 0.34	91.29 $\pm$ 0.41	90.77 $\pm$ 0.34	91.90 $\pm$ 0.33
GAT	57.83 $\pm$ 1.28	63.10 $\pm$ 0.66	69.65 $\pm$ 2.35	72.33 $\pm$ 1.94	70.86 $\pm$ 1.75	71.49 $\pm$ 1.05	58.94 $\pm$ 1.35	64.86 $\pm$ 1.12	91.89 $\pm$ 0.83	92.27 $\pm$ 0.20	90.38 $\pm$ 0.24	91.86 $\pm$ 0.31
HetGNN	61.48 $\pm$ 2.56	69.01 $\pm$ 2.68	63.14 $\pm$ 0.43	69.15 $\pm$ 0.80	60.47 $\pm$ 1.38	67.08 $\pm$ 0.83	53.46 $\pm$ 0.87	60.73 $\pm$ 1.49	91.76 $\pm$ 0.48	92.33 $\pm$ 0.41	87.19 $\pm$ 0.35	87.68 $\pm$ 0.22
HGT	63.72 $\pm$ 2.18	69.27 $\pm$ 1.34	63.44 $\pm$ 0.46	61.98 $\pm$ 0.33	59.65 $\pm$ 0.52	62.60 $\pm$ 0.41	63.07 $\pm$ 1.19	67.20 $\pm$ 1.61	93.01 $\pm$ 0.24	93.49 $\pm$ 0.25	90.97 $\pm$ 0.66	91.32 $\pm$ 0.89
MAGNN	66.74 $\pm$ 2.84	73.85 $\pm$ 1.39	72.01 $\pm$ 2.87	73.63 $\pm$ 2.66	71.63 $\pm$ 3.28	72.80 $\pm$ 3.01	67.36 $\pm$ 2.84	68.18 $\pm$ 2.03	93.28 $\pm$ 0.51	93.76 $\pm$ 0.45	91.90 $\pm$ 0.61	91.95 $\pm$ 0.85
SHGN	66.94 $\pm$ 1.37	76.06 $\pm$ 0.86	71.89 $\pm$ 2.45	72.70 $\pm$ 3.01	71.88 $\pm$ 2.71	72.93 $\pm$ 2.92	64.31 $\pm$ 1.35	67.05 $\pm$ 1.31	94.05 $\pm$ 0.31	94.25 $\pm$ 0.32	92.58 $\pm$ 0.61	93.01 $\pm$ 0.51
SeHGNN	67.51 $\pm$ 0.83	76.81 $\pm$ 0.48	72.07 $\pm$ 2.07	74.49 $\pm$ 1.94	70.39 $\pm$ 3.12	73.66 $\pm$ 2.18	67.11 $\pm$ 1.24	67.93 $\pm$ 1.93	94.24 $\pm$ 0.53	94.70 $\pm$ 0.41	93.27 $\pm$ 0.72	93.31 $\pm$ 0.85
HINormer	67.12 $\pm$ 2.18	77.17 $\pm$ 2.85	71.23 $\pm$ 1.53	72.08 $\pm$ 2.06	71.03 $\pm$ 2.64	72.42 $\pm$ 2.66	64.09 $\pm$ 1.56	68.01 $\pm$ 1.96	94.20 $\pm$ 0.45	94.65 $\pm$ 0.34	92.66 $\pm$ 0.73	93.34 $\pm$ 0.85
LSMPS	68.43 $\pm$ 1.09	77.13 $\pm$ 0.65	72.18 $\pm$ 1.97	74.10 $\pm$ 2.31	71.85 $\pm$ 2.10	73.09 $\pm$ 1.94	67.60 $\pm$ 1.38	68.19 $\pm$ 1.85	94.97 $\pm$ 0.48	95.21 $\pm$ 0.63	93.98 $\pm$ 0.83	94.26 $\pm$ 0.53
Seq-HGNN	69.38 $\pm$ 1.42	77.41 $\pm$ 0.96	71.64 $\pm$ 2.18	73.51 $\pm$ 1.92	71.99 $\pm$ 2.07	73.01 $\pm$ 1.55	67.36 $\pm$ 1.42	68.21 $\pm$ 1.45	95.45 $\pm$ 0.48	95.63 $\pm$ 0.67	94.01 $\pm$ 0.53	94.23 $\pm$ 0.38
Hetero <sup>2</sup> Net	70.17 $\pm$ 1.38	77.85 $\pm$ 1.76	71.41 $\pm$ 1.78	72.43 $\pm$ 1.91	71.15 $\pm$ 1.84	72.33 $\pm$ 1.66	65.18 $\pm$ 0.48	68.16 $\pm$ 0.56	94.03 $\pm$ 0.35	94.46 $\pm$ 0.37	92.84 $\pm$ 0.37	93.31 $\pm$ 0.75
LatGRL	70.08 $\pm$ 1.20	77.52 $\pm$ 0.78	71.76 $\pm$ 1.70	74.56 $\pm$ 1.96	71.01 $\pm$ 2.93	72.58 $\pm$ 1.94	67.08 $\pm$ 1.28	68.38 $\pm$ 1.36	92.37 $\pm$ 0.28	94.38 $\pm$ 0.53	92.44 $\pm$ 0.45	93.54 $\pm$ 0.56
H <sup>2</sup> Gformer	69.05 $\pm$ 1.60	78.42 $\pm$ 2.93	71.36 $\pm$ 1.79	72.27 $\pm$ 1.91	70.50 $\pm$ 2.48	72.51 $\pm$ 2.60	65.33 $\pm$ 1.54	67.76 $\pm$ 2.07	92.81 $\pm$ 0.45	93.48 $\pm$ 0.68	93.10 $\pm$ 0.81	93.41 $\pm$ 0.85
H2GCN-HD	64.75 $\pm$ 1.31	74.09 $\pm$ 1.32	71.36 $\pm$ 3.08	73.78 $\pm$ 2.76	70.16 $\pm$ 1.21	72.05 $\pm$ 1.76	58.87 $\pm$ 1.64	59.39 $\pm$ 1.44	92.32 $\pm$ 0.43	92.81 $\pm$ 0.38	87.54 $\pm$ 0.56	90.44 $\pm$ 0.73
GPRGNN-HD	67.12 $\pm$ 1.08	75.53 $\pm$ 1.62	72.17 $\pm$ 1.40	74.17 $\pm$ 1.52	70.01 $\pm$ 1.23	71.57 $\pm$ 1.31	57.72 $\pm$ 0.64	61.03 $\pm$ 0.16	93.56 $\pm$ 0.53	94.17 $\pm$ 0.53	87.40 $\pm$ 0.43	90.29 $\pm$ 0.53
GloGNN-HD	69.33 $\pm$ 2.32	77.63 $\pm$ 1.84	71.35 $\pm$ 1.96	72.36 $\pm$ 1.95	71.01 $\pm$ 1.68	72.31 $\pm$ 2.04	65.14 $\pm$ 1.01	67.75 $\pm$ 1.52	94.22 $\pm$ 0.64	94.35 $\pm$ 0.41	92.05 $\pm$ 0.46	92.75 $\pm$ 0.34
ALTGNN-HD	69.20 $\pm$ 2.28	77.62 $\pm$ 1.81	71.18 $\pm$ 1.93	72.41 $\pm$ 1.90	71.15 $\pm$ 1.72	72.50 $\pm$ 2.08	64.78 $\pm$ 0.98	67.32 $\pm$ 1.47	93.95 $\pm$ 0.68	93.89 $\pm$ 0.43	93.11 $\pm$ 0.66	93.60 $\pm$ 0.31
ACMGCN-HD	69.45 $\pm$ 2.35	77.81 $\pm$ 1.87	71.42 $\pm$ 1.99	72.48 $\pm$ 1.93	71.92 $\pm$ 1.64	72.25 $\pm$ 2.01	65.25 $\pm$ 1.04	67.60 $\pm$ 1.50	94.10 $\pm$ 0.62	94.68 $\pm$ 0.45	92.98 $\pm$ 0.43	93.64 $\pm$ 0.29
LSMPS-HD	70.07 $\pm$ 1.21	78.60 $\pm$ 0.61	71.40 $\pm$ 2.01	72.26 $\pm$ 2.34	72.45 $\pm$ 2.21	72.88 $\pm$ 1.80	67.06 $\pm$ 1.36	68.16 $\pm$ 1.82	94.44 $\pm$ 0.49	94.97 $\pm$ 0.64	93.98 $\pm$ 1.05	93.62 $\pm$ 0.54
SeqHGNN-HD	69.50 $\pm$ 1.31	76.78 $\pm$ 0.78	71.39 $\pm$ 1.86	72.10 $\pm$ 2.06	71.99 $\pm$ 2.53	72.92 $\pm$ 1.21	67.30 $\pm$ 1.26	67.69 $\pm$ 1.35	93.12 $\pm$ 0.03	93.87 $\pm$ 0.86	93.27 $\pm$ 0.14	93.81 $\pm$ 0.20
AHGNN	74.89 $\pm$ 0.96	82.13 $\pm$ 0.08	73.81 $\pm$ 2.07	76.32 $\pm$ 1.59	73.75 $\pm$ 1.42	75.41 $\pm$ 0.80	69.81 $\pm$ 1.34	69.83 $\pm$ 1.39	95.84 $\pm$ 0.37	96.33 $\pm$ 0.45	94.45 $\pm$ 0.48	94.60 $\pm$ 0.47
<i>Improvement</i>	4.72	4.53	1.63	1.76	1.3	1.75	2.21	1.45	0.39	0.70	0.44	0.36

**Table 3: Results on ogbn-mag are reported Macro-F1 and Micro-F1 (scaled up by 100 for clarity) across all runs and splits. Best results are highlighted, excluding baselines without public code or facing out-of-memory issues.**

	Macro-F1	Micro-F1
GCN	51.91 $\pm$ 0.40	52.10 $\pm$ 0.55
GAT	52.48 $\pm$ 0.81	54.03 $\pm$ 0.45
HGT	54.14 $\pm$ 0.52	54.18 $\pm$ 0.09
SHGN	58.30 $\pm$ 0.49	56.18 $\pm$ 0.85
SeHGNN	58.29 $\pm$ 0.33	58.23 $\pm$ 0.65
HINormer	57.19 $\pm$ 0.39	57.05 $\pm$ 0.84
LSMPS	58.94 $\pm$ 0.28	57.62 $\pm$ 0.45
Seq-HGNN	58.28 $\pm$ 0.45	58.36 $\pm$ 0.38
Hetero <sup>2</sup> Net	58.51 $\pm$ 0.16	58.60 $\pm$ 0.58
LatGCL	57.18 $\pm$ 0.63	58.05 $\pm$ 0.40
AHGNN	60.42 $\pm$ 0.29	60.69 $\pm$ 0.19

## 5.8 Efficiency Analysis

We conduct an efficiency analysis on AHGNN and other baselines<sup>3</sup> on Actor. This analysis estimates the time (in seconds) and peak GPU memory usage (in MB) during pre-computation (if applicable) and training stages. We calculate the average ranking of each model

<sup>3</sup>LSMPS's time costs were recorded for search and train stages.

based on both time and memory considerations. Results in Table 6 reveal that AHGNN is a lightweight model efficient in both time and space. On average, AHGNN ranks among the most efficient models. While consuming slightly more space than SeHGNN, AHGNN offers significant performance improvements, as shown in Table 2. This trade-off between efficiency and performance is considered worthwhile.

## 5.9 Parameter Analysis

We conduct a parameter analysis on the maximum number of hops for heterogeneous message propagation ( $L_1$ ), label propagation ( $L_2$ ), and the parameter  $\alpha$  in Adaptive Heterogeneous Convolution, as shown in Figure 4. The results reveal that: (i) optimal choices of hyper-parameters vary across different datasets due to their inherent characteristics. For instance, the Actor dataset favors shorter meta-paths, while the IMDB dataset benefits from longer ones. Additionally, the best value of  $\alpha$  differs between IMDB and Actor, indicating varying levels of heterophily at different hop levels between these datasets. (ii) The AHGNN generally exhibits robustness to variations in hyper-parameter settings. Despite the presence of dataset-specific optimal settings, sub-optimal configurations of hyper-parameters can yield satisfying results.

**Table 4: Few-shot learning results on real-world datasets presented in Macro-F1 and Micro-F1 (scaled up by 100 for clarity), including mean and standard deviation over all runs and splits. The graph-level homophily ratio of each dataset is displayed in brackets ( $h$ ). The best results are highlighted in gray.**

	Actor (0.29)		FB-MIT (0.49)		FB-American (0.53)		IMDB (0.59)		DBLP (0.81)		ACM (0.88)	
	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1
GCN	45.98 $\pm$ 1.30	48.03 $\pm$ 1.93	38.65 $\pm$ 0.95	41.32 $\pm$ 1.87	35.97 $\pm$ 1.57	38.78 $\pm$ 1.33	35.70 $\pm$ 1.68	36.01 $\pm$ 1.56	88.37 $\pm$ 0.72	89.64 $\pm$ 0.38	87.44 $\pm$ 0.98	88.68 $\pm$ 0.73
GAT	46.75 $\pm$ 1.57	47.75 $\pm$ 2.31	39.41 $\pm$ 1.34	41.74 $\pm$ 1.58	35.76 $\pm$ 1.34	38.09 $\pm$ 1.07	35.88 $\pm$ 1.56	36.21 $\pm$ 1.78	89.96 $\pm$ 0.45	90.18 $\pm$ 0.18	88.01 $\pm$ 0.75	88.64 $\pm$ 0.54
HetGNN	44.91 $\pm$ 1.84	48.85 $\pm$ 1.03	37.23 $\pm$ 1.97	38.56 $\pm$ 1.39	35.46 $\pm$ 1.24	38.66 $\pm$ 1.63	35.12 $\pm$ 2.47	36.80 $\pm$ 3.01	89.31 $\pm$ 0.86	90.85 $\pm$ 1.02	86.36 $\pm$ 0.74	88.12 $\pm$ 1.01
HGT	46.77 $\pm$ 1.31	50.22 $\pm$ 1.95	38.64 $\pm$ 1.65	38.85 $\pm$ 1.09	35.78 $\pm$ 1.60	38.53 $\pm$ 1.32	37.65 $\pm$ 2.53	38.52 $\pm$ 2.80	91.05 $\pm$ 0.76	91.36 $\pm$ 0.55	89.30 $\pm$ 0.63	89.84 $\pm$ 0.70
MAGNN	45.90 $\pm$ 1.63	49.64 $\pm$ 1.88	41.81 $\pm$ 1.83	42.89 $\pm$ 2.04	37.01 $\pm$ 0.86	40.78 $\pm$ 1.51	40.12 $\pm$ 2.97	41.02 $\pm$ 3.01	91.07 $\pm$ 0.48	91.74 $\pm$ 0.67	89.45 $\pm$ 0.47	89.73 $\pm$ 0.74
SHGN	45.86 $\pm$ 1.92	50.63 $\pm$ 2.07	40.07 $\pm$ 1.29	42.73 $\pm$ 1.84	36.31 $\pm$ 1.63	38.96 $\pm$ 1.47	39.97 $\pm$ 3.08	40.32 $\pm$ 3.31	91.45 $\pm$ 0.45	92.01 $\pm$ 0.30	90.01 $\pm$ 0.45	90.47 $\pm$ 0.73
SeHGNN	46.37 $\pm$ 1.04	50.04 $\pm$ 2.53	41.51 $\pm$ 1.57	44.12 $\pm$ 1.78	36.83 $\pm$ 1.45	39.77 $\pm$ 1.57	40.98 $\pm$ 1.94	41.37 $\pm$ 1.53	92.04 $\pm$ 0.56	92.93 $\pm$ 0.42	91.02 $\pm$ 0.45	91.32 $\pm$ 0.47
HINormer	46.83 $\pm$ 1.57	50.97 $\pm$ 3.81	41.86 $\pm$ 1.36	43.90 $\pm$ 1.37	36.86 $\pm$ 1.73	40.01 $\pm$ 1.89	40.83 $\pm$ 2.10	41.14 $\pm$ 1.63	91.85 $\pm$ 0.47	92.77 $\pm$ 0.64	90.01 $\pm$ 0.66	90.34 $\pm$ 0.62
LSMPS	47.44 $\pm$ 1.50	50.65 $\pm$ 1.97	41.73 $\pm$ 1.97	44.42 $\pm$ 0.74	36.97 $\pm$ 1.90	39.64 $\pm$ 1.66	41.03 $\pm$ 2.04	41.18 $\pm$ 1.24	91.86 $\pm$ 0.76	92.61 $\pm$ 0.81	91.85 $\pm$ 0.45	92.41 $\pm$ 0.86
SeqHGNN	47.31 $\pm$ 1.74	50.73 $\pm$ 1.95	42.08 $\pm$ 1.64	44.51 $\pm$ 1.46	37.59 $\pm$ 1.47	41.05 $\pm$ 1.37	40.56 $\pm$ 2.37	40.92 $\pm$ 2.87	92.10 $\pm$ 0.33	93.01 $\pm$ 0.29	91.68 $\pm$ 0.53	92.54 $\pm$ 0.42
Hetero <sup>2</sup> Net	46.87 $\pm$ 2.05	50.56 $\pm$ 1.05	40.65 $\pm$ 1.38	43.31 $\pm$ 1.89	36.83 $\pm$ 1.37	37.83 $\pm$ 1.74	40.37 $\pm$ 2.48	40.44 $\pm$ 2.34	92.07 $\pm$ 0.42	92.32 $\pm$ 0.67	89.75 $\pm$ 0.53	90.11 $\pm$ 0.77
LatGRL	47.32 $\pm$ 1.20	51.06 $\pm$ 0.78	41.41 $\pm$ 1.70	44.84 $\pm$ 1.96	38.26 $\pm$ 2.93	41.31 $\pm$ 1.94	40.96 $\pm$ 1.28	41.66 $\pm$ 1.36	89.58 $\pm$ 0.28	90.64 $\pm$ 0.53	90.01 $\pm$ 0.08	90.32 $\pm$ 0.37
H <sup>2</sup> Gformer	47.81 $\pm$ 1.72	50.69 $\pm$ 3.58	41.94 $\pm$ 1.13	43.82 $\pm$ 1.31	35.17 $\pm$ 1.49	40.68 $\pm$ 1.72	40.84 $\pm$ 1.93	41.05 $\pm$ 1.47	91.43 $\pm$ 0.39	92.58 $\pm$ 0.52	89.76 $\pm$ 0.87	90.16 $\pm$ 0.50
H2GCN-HD	47.02 $\pm$ 1.68	48.51 $\pm$ 2.46	40.04 $\pm$ 1.90	42.66 $\pm$ 1.41	35.75 $\pm$ 2.02	38.40 $\pm$ 1.83	35.62 $\pm$ 2.97	36.24 $\pm$ 2.03	90.04 $\pm$ 0.65	90.75 $\pm$ 0.77	86.36 $\pm$ 0.53	87.10 $\pm$ 0.42
LINKX-HD	46.73 $\pm$ 1.34	47.01 $\pm$ 1.42	38.12 $\pm$ 1.58	41.46 $\pm$ 1.34	34.85 $\pm$ 1.87	38.79 $\pm$ 1.90	36.22 $\pm$ 2.17	36.94 $\pm$ 2.87	90.66 $\pm$ 0.45	91.86 $\pm$ 0.73	86.64 $\pm$ 0.67	86.86 $\pm$ 1.02
GloGNN-HD	47.11 $\pm$ 2.32	50.65 $\pm$ 1.84	40.51 $\pm$ 1.96	40.99 $\pm$ 1.95	37.01 $\pm$ 1.68	39.42 $\pm$ 2.04	40.07 $\pm$ 1.01	40.42 $\pm$ 1.52	91.38 $\pm$ 0.64	91.48 $\pm$ 0.41	89.46 $\pm$ 0.35	91.03 $\pm$ 0.45
ALTGNN-HD	47.48 $\pm$ 2.28	50.67 $\pm$ 1.81	40.74 $\pm$ 1.93	41.42 $\pm$ 1.90	37.35 $\pm$ 1.72	40.53 $\pm$ 2.08	40.64 $\pm$ 0.98	40.75 $\pm$ 1.47	90.93 $\pm$ 0.68	90.74 $\pm$ 0.43	90.74 $\pm$ 0.46	91.75 $\pm$ 0.37
ACMGCN-HD	47.51 $\pm$ 2.35	50.41 $\pm$ 1.87	40.91 $\pm$ 1.99	41.31 $\pm$ 1.93	37.23 $\pm$ 1.64	40.46 $\pm$ 2.01	40.32 $\pm$ 1.04	40.71 $\pm$ 1.50	91.08 $\pm$ 0.62	91.39 $\pm$ 0.45	90.63 $\pm$ 0.63	90.96 $\pm$ 0.46
LSMPS-HD	47.56 $\pm$ 1.44	50.25 $\pm$ 1.86	41.92 $\pm$ 1.84	44.95 $\pm$ 0.86	37.62 $\pm$ 2.08	40.41 $\pm$ 1.87	41.05 $\pm$ 1.82	40.95 $\pm$ 1.37	91.67 $\pm$ 0.66	92.56 $\pm$ 0.67	91.11 $\pm$ 0.50	91.62 $\pm$ 0.95
SeqHGNN-HD	47.45 $\pm$ 1.57	50.78 $\pm$ 1.93	42.40 $\pm$ 1.49	45.68 $\pm$ 1.50	37.94 $\pm$ 1.28	41.72 $\pm$ 1.08	41.00 $\pm$ 2.46	41.42 $\pm$ 2.92	92.14 $\pm$ 0.48	92.40 $\pm$ 0.78	90.96 $\pm$ 0.56	91.45 $\pm$ 0.46
AHGNN	50.58 $\pm$ 2.01	54.52 $\pm$ 1.70	43.89 $\pm$ 0.97	45.57 $\pm$ 1.15	39.58 $\pm$ 0.20	43.81 $\pm$ 1.43	42.05 $\pm$ 2.59	42.57 $\pm$ 2.60	93.12 $\pm$ 0.56	93.48 $\pm$ 0.56	91.87 $\pm$ 0.378	92.99 $\pm$ 0.38
<i>Improvement</i>	3.02	2.99	1.90	1.67	1.32	2.05	1.07	1.30	1.02	0.47	1.24	1.24

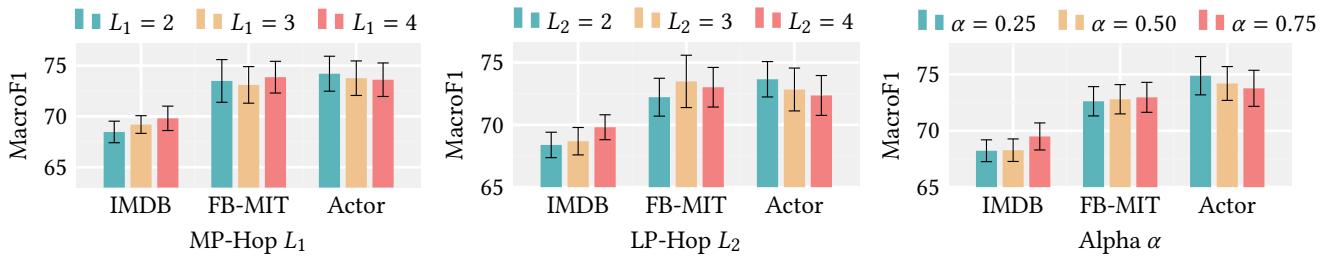


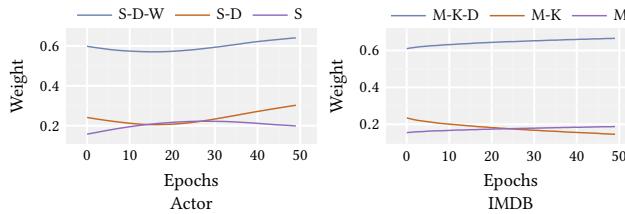
Figure 4: Parameter analysis on IMDB, FB-MIT and Actor.

**Table 5: Ablation Study.** "w/o" denotes without. Scores are scaled up by 100 for clarity. Here AHC stands for Adaptive Heterogeneous Convolution, C2F stands for Coarse-to-Fine Semantic Fusion, and LP for label propagation.

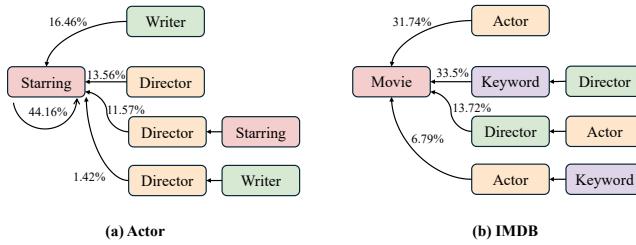
	IMDB (0.59)		FB-MIT (0.49)		Actor (0.29)	
	MacroF1	MicroF1	MacroF1	MicroF1	MacroF1	MicroF1
AHGNN	69.81 $\pm$ 1.34	69.83 $\pm$ 1.39	73.81 $\pm$ 2.07	76.32 $\pm$ 1.59	74.89 $\pm$ 0.96	82.13 $\pm$ 0.08
w/o AHC	65.01 $\pm$ 1.29	66.14 $\pm$ 0.63	71.96 $\pm$ 1.72	72.84 $\pm$ 1.41	69.10 $\pm$ 1.10	76.48 $\pm$ 0.63
w/o C2F	67.80 $\pm$ 1.08	68.17 $\pm$ 0.84	70.01 $\pm$ 2.18	72.18 $\pm$ 1.33	71.49 $\pm$ 1.30	78.35 $\pm$ 0.91
w/o LP	67.15 $\pm$ 2.47	67.38 $\pm$ 2.48	72.41 $\pm$ 1.49	75.16 $\pm$ 1.72	70.31 $\pm$ 0.72	78.11 $\pm$ 0.29
w/o $L_2$ -Norm	68.93 $\pm$ 1.03	69.07 $\pm$ 1.06	73.18 $\pm$ 1.78	75.65 $\pm$ 1.93	71.24 $\pm$ 0.82	79.31 $\pm$ 0.51

**Table 6: Efficiency analysis on Actor, including pre-calculation time (second), training time (second), and peak GPU memory consumption (MB).** - for not-applicable. Avg Rank is the average rank on both time and memory costs.

	Pre-Cal.	Training	Memory	Avg Rank
HGT	-	109.8	9415	#5.0
SeHGNN	7.38	18.36	7317	#2.0
LMSPS*	10.04	79.69	3083	#3.0
Seq-HGNN	-	76.26	8913	#4.5
LatGRL	-	52.57	8547	#3.5
AHGNN	8.81	19.46	7103	#2.0



**Figure 5: Visualization of  $\{\gamma_l\}$  in Adaptive Heterogeneous Convolution**



**Figure 6: Visualization of meta-paths for representing the target nodes in Actor and IMDB.**

## 5.10 Visualization

In this subsection, we first present a visualization of the learnable parameters  $\gamma_k$  associated with each meta-path in the Adaptive Heterogeneous Convolution module (see Figure 5). In Figure 6, we observe that: (i) The parameters  $\{\gamma_l\}$  play their role in modulating the importance of different hops. By adaptively adjusting hop weights during training, the model learns more informative and representative embeddings for each meta-path. (ii) The evolution of  $\{\gamma_l\}$  varies across different hops and graphs, highlighting the need for adaptive modulation. This observation supports our argument in section 1 that a one-size-fits-all approach is suboptimal.

Next, we visualize the impact of individual meta-paths by examining the scaling factor  $\beta$  in the Coarse-to-Fine Semantic Fusion module. The results show that meta-paths with distinct semantic contributions are assigned varying levels of influence. For instance, highly informative meta-paths such as *Starring-Starring* receive greater emphasis, whereas less relevant ones, such as *Starring-Director-Writer*, are assigned lower importance.

## 6 Related Works

### 6.1 Heterogeneous Graph Neural Networks

**Meta-path-based HGNNs.** Meta-path-based approaches utilize predefined or learned meta-paths for feature propagation and semantic fusion. For instance, HAN [34] and MAGNN [5] incorporate graph attention mechanisms, while SeHGNN [37] pre-computes message passing prior to training. Seq-HGNN [4] leverages sequential node embeddings. Additionally, LMSPS [14] introduces a shrinkable sampling strategy for efficient meta-path searching. Alongside model design, there have been efforts to address heterophily through objective function modifications. Hetero<sup>2</sup>Net [15] proposes a masked meta-path strategy and label prediction tasks,

while HGMS [33] and LatGRL [28] adopt contrastive learning objectives. H<sup>2</sup>SGNN takes a spectral perspective [19], while H<sup>2</sup>Gformer involves graph transformers [18] to handle heterophily. However, they fail to handle diverse heterophily distributions across varying hops and meta-paths.

**Meta-path-free HGNNs.** Meta-path-free approaches aggregate neighbor messages similar to homogeneous GNNs, while incorporating additional features such as attention mechanisms or positional encodings for enhanced semantic representation [9, 41]. HetGNN [38] uses random walks to sample semantically consistent neighbors, while SHGN [22] combines node features with learnable edge embeddings for heterogeneous attention. HGT [11] and HINormer [25] further introduce Transformer-style heterogeneous mutual attention. Additionally, LSPI [39] proposes dividing meta-paths into large and small neighbor paths for improved representation.

## 6.2 Heterophily Graph Learning

Traditional GNNs assume homophily (similarity between connected nodes) [20, 26] and are less effective on heterophily graphs where connected nodes differ significantly. Recent works have extended GNNs to heterophily graphs, focusing primarily on homogeneous graphs. Models like H2GCN [40] and GPR-GNN [3] enhance message passing with high-order re-weighting techniques for better heterophily handling. Other approaches, such as LINKX [17], GloGNN [16], MWGNN [23], ACMGCN [21], ALT-GNN [36], and more [24], refine graph convolution for heterophily, excelling on homogeneous data but struggling with heterogeneous graphs due to lack of type-awareness. HDHGR [8] adapts GNNs to heterophily HGs via graph rewiring but underperforms compared to other HGNNs, as shown in Table 2. The key difference lies in handling heterophily: HDHGR rewrites the graph, while AHGNN directly models varying heterophily and semantic spaces.

## 7 Conclusion

In this paper, we aim to improve the performance of Heterogeneous Graph Neural Networks (HGNNs) on heterophilous data. We identify two key challenges: (i) the variation of heterophily distributions across hops and meta-paths, and (ii) the complex, heterophily-influenced semantic variation among different meta-paths. To address these issues, we propose the **Adaptive Heterogeneous Graph Neural Network (AHGNN)**. AHGNN conducts hop-specific and meta-path-specific graph convolution in a heterophily-aware fashion. It further refines node representations through a **Coarse-to-Fine Semantic Fusion** mechanism, which amplifies informative semantics while suppressing noisy signals. Extensive experiments on seven real-world datasets demonstrate that AHGNN achieves superior performance and efficiency, particularly on graphs characterized by strong heterophily.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62276006).

## 8 Declaration on Generative AI Tools

The generative AI tools employed in this work are solely used for language polishing and refinement. These tools do not contribute to the development of the core ideas, methodologies, or results presented in the work.

## References

- [1] Sanghamitra Bandyopadhyay, Ujjwal Maulik, Lawrence B Holder, Diane J Cook, and Lise Getoor. 2005. Link-based classification. *Advanced methods for knowledge discovery from complex data* (2005), 189–207.
- [2] Qin Chen, Liang Wang, Bo Zheng, and Guojie Song. 2025. Dagprompt: Pushing the limits of graph prompting with a distribution-aware graph prompt tuning approach. In *Proceedings of the ACM on Web Conference 2025*. 4346–4358.
- [3] Eli Chien, Jianhua Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive universal generalized pagerank graph neural network. *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (2021).
- [4] Chenguang Du, Kaichun Yao, Hengshu Zhu, Deqing Wang, Fuzhen Zhuang, and Hui Xiong. 2023. Seq-HGNN: Learning Sequential Node Representation on Heterogeneous Graph. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023* (2023).
- [5] Xinyu Fu, Jianqi Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*. 2331–2341.
- [6] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).
- [7] Chenghua Gong, Yao Cheng, Xiang Li, Caihua Shan, Siqiang Luo, and Chuan Shi. 2024. Towards learning from graphs with heterophily: Progress and future. *arXiv preprint arXiv:2401.09769* (2024).
- [8] Jiayan Guo, Lun Du, Wendong Bi, Qiang Fu, Xiaojun Ma, Xu Chen, Shi Han, Dongmei Zhang, and Yan Zhang. 2023. Homophily-oriented Heterogeneous Graph Rewiring. In *Proceedings of the ACM Web Conference 2023*. 511–522.
- [9] Huiting Hong, Hantao Guo, Yucheng Lin, Xiaoqing Yang, Zang Li, and Jieping Ye. 2020. An attention-based graph neural network for heterogeneous structural learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4132–4139.
- [10] Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. 2021. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430* (2021).
- [11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*. 2704–2710.
- [12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2014).
- [13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [14] Chao Li, Zijie Guo, Qiuting He, Hao Xu, and Kun He. 2023. Long-range Dependency based Multi-Layer Perceptron for Heterogeneous Information Networks. *arXiv preprint arXiv:2307.08430* (2023).
- [15] Jintang Li, Zheng Wei, Jiawang Dan, Jing Zhou, Yuchang Zhu, Ruofan Wu, Baokun Wang, Zhang Zhen, Changhua Meng, Hong Jin, et al. 2023. Hetero<sup>2</sup> Net: Heterophily-aware Representation Learning on Heterogeneous Graphs. *arXiv preprint arXiv:2310.11664* (2023).
- [16] Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weinig Qian. 2022. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*. PMLR, 13242–13256.
- [17] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhale Rao, 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.
- [18] Junhong Lin, Xiaojie Guo, Shuaicheng Zhang, Dawei Zhou, Yada Zhu, and Julian Shun. 2024. When Heterophily Meets Heterogeneity: New Graph Benchmarks and Effective Methods. *arXiv preprint arXiv:2407.10916* (2024).
- [19] Kangkang Lu, Yanhua Yu, Zhiyong Huang, Jia Li, Yuling Wang, Meiyu Liang, Xiting Qin, Yimeng Ren, Tat-Seng Chua, and Xidian Wang. 2024. Addressing Heterogeneity and Heterophily in Graphs: A Heterogeneous Heterophilic Spectral Graph Neural Network. *arXiv preprint arXiv:2410.13373* (2024).
- [20] Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, et al. 2024. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618* (2024).
- [21] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2022. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems* 35 (2022), 1362–1375.
- [22] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 1150–1160.
- [23] Xiaojun Ma, Qin Chen, Yuanyi Ren, Guojie Song, and Liang Wang. 2022. Meta-weight graph neural network: Push the limits beyond global homophily. In *Proceedings of the ACM Web Conference 2022*. 1270–1280.
- [24] Xiaojun Ma, Qin Chen, Yi Wu, Guojie Song, Liang Wang, and Bo Zheng. 2023. Rethinking structural encodings: Adaptive graph transformer for node classification task. In *Proceedings of the ACM web conference 2023*. 533–544.
- [25] Qiheng Mao, Zemin Liu, Chenghao Liu, and Jianling Sun. 2023. Hinormer: Representation learning on heterogeneous information networks with graph transformer. In *Proceedings of the ACM Web Conference 2023*. 599–610.
- [26] 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.
- [27] S Unnikrishna Pillai, Torsten Suel, and Seunghun Cha. 2005. The Perron-Frobenius theorem: some of its applications. *IEEE Signal Processing Magazine* 22, 2 (2005), 62–75.
- [28] Zhixiang Shen and Zhao Kang. 2025. When heterophily meets heterogeneous graphs: Latent graphs guided unsupervised representation learning. *IEEE Transactions on Neural Networks and Learning Systems* (2025).
- [29] Yizhou Sun and Jiawei Han. 2012. *Mining heterogeneous information networks: principles and methodologies*. Morgan & Claypool Publishers.
- [30] 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.
- [31] Amanda L Traud, Peter J Mucha, and Mason A Porter. 2012. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications* 391, 16 (2012), 4165–4180.
- [32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [33] Haosen Wang, Chenglong Shi, Can Xu, Surong Yan, and Pan Tang. 2025. Homophily-aware Heterogeneous Graph Contrastive Learning. *arXiv preprint arXiv:2501.08538* (2025).
- [34] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.
- [35] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.
- [36] Zhe Xu, Yuzhong Chen, Qinghai Zhou, Yuhang Wu, Menghai Pan, Hao Yang, and Hanghang Tong. 2023. Node classification beyond homophily: Towards a general solution. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2862–2873.
- [37] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. 2023. Simple and efficient heterogeneous graph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10816–10824.
- [38] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 793–803.
- [39] Yufei Zhao, Shiduo Wang, and Hua Duan. 2025. LSPI: Heterogeneous graph neural network classification aggregation algorithm based on size neighbor path identification. *Applied Soft Computing* (2025), 112656.
- [40] 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.
- [41] Shichao Zhu, Chuan Zhou, Shirui Pan, Xingquan Zhu, and Bin Wang. 2019. Relation structure-aware heterogeneous graph neural network. In *2019 IEEE international conference on data mining (ICDM)*. IEEE, 1534–1539.