HeterSEED: Semantics–Structure Decoupling for Heterogeneous Graph Learning under Heterophily

Xinyi Li
School of Computer Science and Technology
Zhejiang Normal University

Ming Li*
Zhejiang Key Laboratory of Intelligent Education Technology and Application
Zhejiang Normal University

Lu Bai
School of Artificial Intelligence
Beijing Normal University

Lixin Cui
Central University of Finance and Economics

Feilong Cao
School of Mathematical Sciences
Zhejiang Normal University

Ke Lv
School of Engineering Science
University of Chinese Academy of Sciences

Yunliang Jiang
School of Computer Science and Technology
Zhejiang Normal University
School of Artificial Intelligence
Huzhou Normal University

Pietro Lio
Department of Computer Science and Technology
University of Cambridge

Abstract

Many real-world heterogeneous graphs exhibit pronounced heterophily, where connected nodes often have dissimilar labels or play different semantic roles. In such settings, standard heterogeneous graph neural networks that aggregate messages along metapaths or meta-relations primarily based on feature similarity can propagate misleading information, since feature similarity may be misaligned with underlying relational semantics. In this paper, we propose **HeterSEED**, a semantics–structure decoupling framework for heterogeneous graph learning under heterophily. HeterSEED decouples representation learning into a heterogeneous semantic channel that captures type- and relation-aware local semantics and a structure-aware heterophily channel that separates homophilic and heterophilic neighborhoods via pseudo-label-guided partitioning and aggregates them using metapath-based structural weights. A node-level adaptive fusion mechanism then combines the two channels to produce context-dependent node representations. We train HeterSEED by jointly minimizing a supervised classification loss and a cross-covariance regularizer that discourages linear dependence between the two channels and encourages complementary semantic and structural representations. Theoretically, we establish an existential expressiveness separation under a correct pseudo-label-induced partition and characterize how reducing heterophilic aggregation mass lowers squared prediction error in a linear smoothing model. Experiments on five real-world heterogeneous graphs, including two large-scale networks at the million-node and hundred-million-edge scale, demonstrate that HeterSEED consistently outperforms representative heterogeneous graph neural networks and recent heterophily-aware baselines, especially in strongly heterophilic regimes.

*Corresponding author: mingli@zjnu.edu.cn

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

1 Introduction

Background. Learning on heterogeneous graphs has become an important paradigm for modeling complex systems with multiple node and relation types, such as academic networks, recommender systems, and knowledge graphs [17, 27]. By encoding type-specific semantics and relation patterns, heterogeneous graph neural networks (HGNNs) [3, 32, 36] have achieved promising results on tasks including node classification, link prediction, and recommendation.

Motivation. A common design in existing HGNNs is to aggregate messages along metapaths or meta-relations, weighting neighbors mainly by feature similarity. This design implicitly assumes that similar features indicate similar labels or semantic roles. However, many real-world heterogeneous graphs exhibit pronounced heterophily, where connected nodes often have dissimilar labels or play different roles [10, 26, 38, 40]. Under such conditions, feature similarity along metapaths becomes an unreliable proxy for relational semantics, and naive feature-similarity-driven aggregation can propagate misleading information and blur decision boundaries. This challenge is further exacerbated by the coexistence of multiple node types and complex inter-type relations, where homophilic and heterophilic interactions are intertwined within local neighborhoods. Figure 1 shows a ***weak*** correlation between metapath-based feature similarity and label homophily, suggesting that feature similarity explains little variance in label homophily. This indicates that feature-based aggregation alone is insufficient to capture relational semantics in heterogeneous graphs under heterophily. Recent heterophily-aware methods for heterogeneous graphs [16, 26], such as metapath-level heterophily metrics and auxiliary prediction tasks, mark important progress. However, they still rely on feature-based message passing over metapath views and do not explicitly decouple heterogeneous semantics from heterophilic structures or learn node-wise structure-aware fusion of homophilic and heterophilic signals.

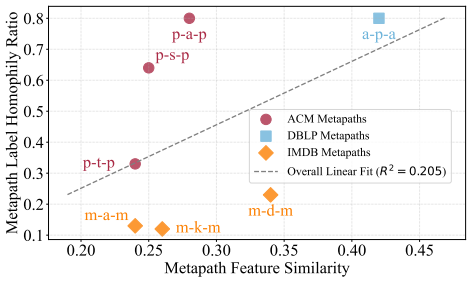


Figure 1: Metapath feature similarity vs. label homophily ratio across datasets. Each point denotes one metapath, and the dashed line shows the global linear fit ($R^2 = 0.205$).

Method. In this paper, we propose HeterSEED, a semantics–structure decoupling framework with explicit heterophily-awareness for heterogeneous graphs. Specifically, HeterSEED decomposes representation learning into two complementary channels: (i) a *Heterogeneous Semantic Aggregation Channel* that captures type-specific semantic information across multiple node and relation types via metapath- or relation-based aggregation, and (ii) a *Structure-Aware Heterophily Channel* that leverages pseudo labels and metapath-based structural weights to separate homophilic and heterophilic neighborhoods and aggregate them through structure-guided operators. A node-level structure-aware fusion mechanism then combines the two channels by learning fusion weights from the concatenated semantic and structural embeddings, enabling adaptive balancing of the two channels and robust node representations under heterophily. From a theoretical perspective, we establish an existential expressiveness separation between HeterSEED and the specified semantic aggregation class under a correct pseudo-label-induced partition (Theorem 1), and analyze bias reduction under an explicit linear smoothing model (Proposition 1).

Summary of Contributions. The main contributions of this work are three-fold:

- We propose **HeterSEED**, a heterophily-aware semantics–structure decoupling framework for learning on heterogeneous graphs. Unlike existing HGNNs that primarily rely on feature-similarity-driven aggregation along metapaths, HeterSEED explicitly separates heterogeneous semantic modeling from heterophilic structural modeling.
- We analyze two complementary aspects of HeterSEED: an existential expressiveness separation from the specified semantic aggregation class under an exact pseudo-label-induced partition, and the reduction in squared prediction error obtained by decreasing heterophilic aggregation mass in a linear smoothing model.
- We conduct extensive experiments on five real-world heterogeneous graphs, including two **large-scale** networks at the *million* node and *hundred-million* edge scale, and show that HeterSEED generally outperforms representative HGNNs and recent heterophily-aware baselines on node classification.

2 Preliminaries and Notation

Definition 1 (Heterogeneous Graph [28]). *A heterogeneous graph is a directed graph $\mathcal{G} = (V, E, T, R, \phi, \psi)$, where V and E are the sets of nodes and edges, and T and R are the sets of node types and edge types. Each node $v_i \in V$ is associated with a type $\phi(v_i) = t_i \in T$, and each edge $e_{ij} \in E$ is associated with a type $\psi(e_{ij}) = r_{ij} \in R$. The graph is heterogeneous if $|T| + |R| > 2$. Each node v_i is equipped with a feature vector $x_i \in \mathbb{R}^{d_i}$. Let A^{t_i,t_j} denote the adjacency matrix between node types t_i and t_j, where $A_{uv}^{t_i,t_j} = 1$ indicates that $\phi(u) = t_i$, $\phi(v) = t_j$, and u and v are connected.*

Definition 2 (Metapath [28]). *A metapath p is a sequence of node and edge types of the form $p = t_1 \xrightarrow{r_1} t_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} t_{l+1}$, where $t_1, \ldots, t_{l+1} \in T$ and $r_1, \ldots, r_l \in R$. A metapath encodes a composite semantic relation pattern between node types and corresponds to multiple meta-path instances in $\mathcal{G}$. In this work, we focus on symmetric meta-paths where $t_1 = t_{l+1}$.*

Definition 3 (Metapath-based Graph and Neighbors [8]). *Given a meta-path p in a heterogeneous graph $\mathcal{G}$, the meta-path-based neighbors of a node v are the nodes that can be reached from v through instances of p (including v itself when p is symmetric). The meta-path-based graph $\mathcal{G}_p$ is constructed by linking all pairs of meta-path-based neighbors. The graph $\mathcal{G}_p$ is homogeneous if the head and tail node types of p are identical. We denote the neighbors of v in $\mathcal{G}_p$ as $\mathcal{N}_{\mathcal{G}_p}^v$.*

Definition 4 (Homophily Ratio in Heterogeneous Graphs [26]). *Given a symmetric metapath $p = t_1 \xrightarrow{r_1} t_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} t_{l+1}$, with $t_1 = t_{l+1}$, the metapath–induced graph $\mathcal{G}_p$ is a homogeneous graph constructed based on p. Let E_p denote the edge set of $\mathcal{G}_p$, and let y_i be the label of node v_i when supervision is available. The global homophily ratio of $\mathcal{G}_p$ is defined as*

$$\mathcal{H}(\mathcal{G}_p) = \frac{\left|\{(v_i, v_j) \in E_p : y_i = y_j\}\right|}{|E_p|}. \tag{1}$$

To obtain the overall homophily level of the heterogeneous graph $\mathcal{G}$, we take the average homophily ratio across all symmetric meta-path–induced subgraphs:

$$\mathcal{H} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{H}(\mathcal{G}_p), \tag{2}$$

where $\mathcal{P}$ denotes the set of all symmetric metapaths considered. This formulation reflects the average label consistency of metapath–based homogeneous projections and provides a unified measure of homophily for heterogeneous graphs.

Problem Setup. Given a heterogeneous graph $\mathcal{G}$ and node features $\mathbf{X}$, we consider semi-supervised classification of nodes of a designated target type. Only labels of nodes in $\mathcal{V}_{\text{train}}$ are available during training. The objective is to predict labels for the remaining target nodes by jointly exploiting node attributes and heterogeneous graph structure. For multi-label tasks, each target node may belong to multiple classes.

3 Method: HeterSEED

3.1 Framework Overview

We introduce HeterSEED, a heterophily-aware semantics–structure decoupling framework for learning node representations on heterogeneous graphs. As illustrated in Figure 2, starting from the original heterogeneous graph, a generic HGNN backbone defines a *Heterogeneous Semantic Aggregation Channel* (Figure 2**(a)**) that aggregates messages from first-order neighbors across different node and relation types to produce type-specific semantic embeddings. In parallel, a *Structure-Aware Heterophily Channel* (Figure 2**(c)**) operates on metapath-based views, where a structure-aware weight computation module (Figure 2**(b)**) and iteratively refined pseudo-labels are used to separate homophilic and heterophilic neighborhoods and encode their structural patterns into heterophily-aware embeddings. A node-level gating module then fuses the semantic and structural embeddings, and the fused representations are fed into a classification block trained with both a classification loss and a decoupling loss to obtain semantics–structure decoupled, heterophily-aware node representations.

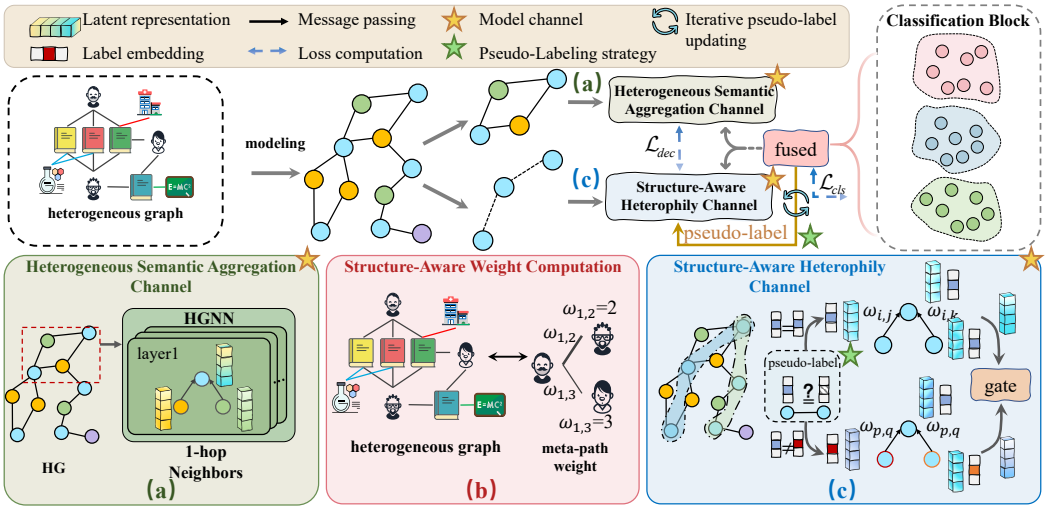


Figure 2: Schematic of the proposed **HeterSEED** framework.

3.2 Heterogeneous Semantic Aggregation Channel

The heterogeneous semantic aggregation channel in HeterSEED is designed to learn type-aware semantic representations $\mathbf{h}_v^{(s)}$ by aggregating one-hop neighbor information across different node and relation types. This channel follows a standard HGNN-style message-passing paradigm and focuses on capturing local semantic consistency induced by the heterogeneous schema, without relying on handcrafted metapaths or explicit heterophily modeling.

Formally, let $\mathcal{N}_r(v)$ denote the set of one-hop neighbors of node v connected via relation type $r \in R$, and let $\mathbf{h}_v^{(l)}$ be the representation of v at layer l (with $\mathbf{h}_v^{(0)} = \mathbf{x}_v$). At the $(l+1)$-th layer, we perform neighborhood-wise convolution over heterogeneous neighbors as

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\mathbf{W}_{\text{self}}^{(l)}\mathbf{h}_v^{(l)} + \sum_{r\in R}\frac{1}{|\mathcal{N}_r(v)|}\sum_{u\in\mathcal{N}_r(v)}\mathbf{W}_r^{(l)}\mathbf{h}_u^{(l)}\right),$$

where $\mathbf{W}_{\text{self}}^{(l)}$ is a learnable weight matrix for self-transformation, $\mathbf{W}_r^{(l)}$ is a relation-specific transformation for messages coming from neighbors connected by relation r, and $\sigma(\cdot)$ is a non-linear activation function (e.g., ReLU). The inner summation aggregates messages from neighbors of type-r in a permutation-invariant manner (we use mean aggregation by default), and the outer summation combines contributions from all relation types. Stacking L such layers yields the heterogeneous semantic embedding $\mathbf{h}_v^{(s)} = \mathbf{h}_v^{(L)}$ for each node v, which encodes type- and relation-aware semantic information and will later be fused with the heterophilic structural embedding produced by the structure-aware heterophily channel.

3.3 Structure-Aware Heterophily Channel

The structure-aware heterophily channel explicitly models relational patterns that are driven by structure rather than feature similarity, producing a heterophily-aware structural embedding $\mathbf{h}_v^{(r)}$ for each node. It leverages pseudo-labels, metapath-induced structural weights, and a gated two-branch aggregation over homophilic and heterophilic neighbors, as detailed in the following subsections.

Homophilic and Heterophilic Neighbor Construction. We first assign a pseudo-label to each node to characterize label consistency among connected nodes. For labeled nodes in the training set, we directly use ground-truth labels, while for unlabeled nodes we infer pseudo-labels from the predictions of a linear classifier applied to the current embeddings (updated iteratively during training). Formally, let $\mathbf{y}$ denote the ground-truth labels and $\hat{\mathbf{y}}$ the predicted labels. The pseudo-label of node v is defined as

$$y_v^{\text{pseudo}} = \begin{cases} y_v, & \text{if } v \in \mathcal{V}_{\text{train}}, \\ \hat{y}_v, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathcal{V}_{\text{train}}$ denotes the set of nodes with available labels. For multi-label tasks, we assign a label set to each node. For a labeled training node, this set contains all its observed positive labels; for an unlabeled node, it contains only the highest-scoring predicted class. Two metapath-connected nodes are treated as homophilic neighbors if their label sets share at least one class, and as heterophilic neighbors otherwise. This rule affects only structural neighborhood partitioning; the supervised classification loss continues to use the original multi-label targets.

Given a symmetric metapath $p \in \mathcal{P}$ and its induced homogeneous graph $\mathcal{G}_p = (V_p, E_p)$, we decompose E_p into homophilic and heterophilic edge sets according to pseudo-label consistency:

$$E^p_{\text{homo}} = \{(u,v) \in E_p \mid y^{\text{pseudo}}_u = y^{\text{pseudo}}_v\}, \qquad E^p_{\text{hetero}} = \{(u,v) \in E_p \mid y^{\text{pseudo}}_u \neq y^{\text{pseudo}}_v\}. \quad (4)$$

The corresponding node sets contain only nodes incident to these edges,

$$V^p_{\text{homo}} = \{v \mid \exists u,\ (u,v) \in E^p_{\text{homo}}\}, \qquad V^p_{\text{hetero}} = \{v \mid \exists u,\ (u,v) \in E^p_{\text{hetero}}\}, \quad (5)$$

yielding two structure-induced subgraphs $\mathcal{G}^p_{\text{homo}} = (V^p_{\text{homo}}, E^p_{\text{homo}})$ and $\mathcal{G}^p_{\text{hetero}} = (V^p_{\text{hetero}}, E^p_{\text{hetero}})$ that capture homophilic and heterophilic relational patterns under metapath p. Aggregating over all symmetric metapaths leads to $\mathcal{G}_{\text{homo}} = \bigcup_{p\in\mathcal{P}} \mathcal{G}^p_{\text{homo}}$ and $\mathcal{G}_{\text{hetero}} = \bigcup_{p\in\mathcal{P}} \mathcal{G}^p_{\text{hetero}}$. We then define the homophilic and heterophilic neighbor sets of node v with respect to these aggregated graphs as

$$\mathcal{N}^v_{\text{homo}} = \{u \in \mathcal{N}_{\mathcal{G}_{\text{homo}}}(v) \mid y^{\text{pseudo}}_u = y^{\text{pseudo}}_v\}, \qquad \mathcal{N}^v_{\text{hetero}} = \{u \in \mathcal{N}_{\mathcal{G}_{\text{hetero}}}(v) \mid y^{\text{pseudo}}_u \neq y^{\text{pseudo}}_v\}, \quad (6)$$

where $\mathcal{N}_{\mathcal{G}_{\text{homo}}}(v)$ and $\mathcal{N}_{\mathcal{G}_{\text{hetero}}}(v)$ denote the neighbors of v in the aggregated homophilic and heterophilic graphs, respectively. These neighbor sets provide a heterophily-aware structural view that is decoupled from the semantic aggregation in the first channel.

Structure-Aware Weight Computation. Following Definition 2, each symmetric metapath $p \in \mathcal{P}$ encodes a type of higher-order semantic and structural correlation. For any node pair (u, v), let $C_p(u, v)$ denote the number of metapath instances connecting them under p. We define the raw structural weight as $\omega_{u,v} = \sum_{p\in\mathcal{P}} C_p(u, v)$, which captures the overall strength of higher-order connectivity between u and v across all considered metapaths. Intuitively, as illustrated in Figure 3, if nodes represent authors and edges encode co-authorship, then node pairs with more co-authored papers have larger $C_p(u, v)$ under the APA metapath, leading to higher $\omega_{u,v}$ and indicating stronger structural relation. To quantify the relative importance of neighbors for node v, we normalize the raw weights via a softmax over its metapath-induced neighbors:

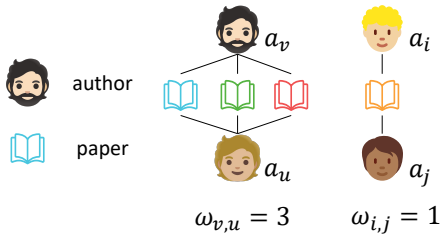


Figure 3: Illustration of metapath-based structural weights.

$$\tilde{\omega}_{u,v} = \frac{\exp(\omega_{u,v})}{\sum_{u'\in\mathcal{N}_{\mathcal{G}}(v)} \exp(\omega_{u',v})}, \quad (7)$$

where $\mathcal{N}_{\mathcal{G}}(v)$ denotes the set of metapath-induced neighbors of v. $\tilde{\omega}_{u,v}$ serves as a structure-aware edge weight, measuring the importance of neighbor u relative to v. This softmax normalization not only ensures numerical stability, but also performs target-wise calibration by converting raw metapath counts into comparable weights, thereby emphasizing relatively stronger structural connections for each node.

Homophily–Heterophily Structural Fusion Layer. To explicitly differentiate and integrate homophilic and heterophilic structural information, we design a homophily–heterophily structural fusion layer. This layer operates directly on the input node representations $\mathbf{x}_u$ (e.g., raw features or a shared base embedding), and re-aggregates them over the homophilic and heterophilic neighbor sets using the structure-aware weights. First, two branch-specific structural aggregations are performed:

$$\mathbf{h}^{(v)}_{\text{homo}} = \sum_{u\in\mathcal{N}^v_{\text{homo}}} \tilde{\omega}_{u,v}\, \mathbf{x}_u, \quad \mathbf{h}^{(v)}_{\text{hetero}} = \sum_{u\in\mathcal{N}^v_{\text{hetero}}} \tilde{\omega}_{u,v}\, \mathbf{x}_u.$$

Next, the two structural signals are fused in a dimension-wise adaptive manner. Let $\mathcal{T}_{\text{homo}}(\cdot)$ and $\mathcal{T}_{\text{hetero}}(\cdot)$ denote two learnable transformation operators, and let $\mathbf{g}_v \in \mathbb{R}^d$ be a learnable gating vector for node v. The fused structural embedding is computed as

$$\mathbf{h}_v^{(r)} = \sigma(\mathbf{g}_v) \odot \mathcal{T}_{\text{homo}}\big(\mathbf{h}_{\text{homo}}^{(v)}\big) + \big(1 - \sigma(\mathbf{g}_v)\big) \odot \mathcal{T}_{\text{hetero}}\big(\mathbf{h}_{\text{hetero}}^{(v)}\big),$$

where $\sigma(\cdot)$ is the sigmoid function and $\odot$ denotes element-wise multiplication.

Implementation of Transformation Operators. In our implementation, the transformation operators $\mathcal{T}_{\text{homo}}(\cdot)$ and $\mathcal{T}_{\text{hetero}}(\cdot)$ are implemented by two-layer MLPs with ReLU activations and a shared output dimension. More expressive designs, such as attention-based blocks or other message-passing architectures, are in principle possible, but we use MLPs to keep the structural fusion layer simple and efficient while isolating the effect of homophily–heterophily decoupling. These operators project the homophilic and heterophilic representations into a shared latent space before gated fusion. The resulting $\mathbf{h}_v^{(r)}$ serves as the heterophily-aware structural embedding of node v, which is later fused with the semantic embedding $\mathbf{h}_v^{(s)}$ from the first channel to form the final representation for downstream prediction.

3.4 Label-Guided Masking Mechanism

Let $\mathbf{Z}_{\text{label}} \in \mathbb{R}^{(C+1)\times d}$ be a learnable label embedding matrix, where C is the number of classes, d is the embedding dimension, and the last row $\mathbf{Z}_{\text{label}}[C+1]$ corresponds to a special `[MASK]` token. For each labeled node i with label index $\ell_i \in \{1, \ldots, C\}$, we sample a masked label embedding during training:

$$\tilde{\mathbf{z}}_i^{(\text{label})} = \begin{cases} \mathbf{Z}_{\text{label}}[\ell_i], & \text{with probability } 1-\beta, \\ \mathbf{Z}_{\text{label}}[C+1], & \text{with probability } \beta, \end{cases} \tag{8}$$

where $\beta \in [0, 1]$ is a hyperparameter that controls the masking rate of observed labels. For unlabeled nodes, no label embedding is used and their features remain unchanged.

The masked label embedding is incorporated into the node feature via a learnable projection $g : \mathbb{R}^d \to \mathbb{R}^{d_{\text{feat}}}$: $\tilde{\mathbf{x}}_i = \mathbf{x}_i + g\big(\tilde{\mathbf{z}}_i^{(\text{label})}\big)$, where d_{feat} is the node feature dimension. The resulting $\tilde{\mathbf{x}}_i$ are then fed into the semantic and structural channels of HeterSEED. During inference, masking is disabled and the original label embedding is deterministically used for labeled nodes: $\tilde{\mathbf{x}}_i = \mathbf{x}_i + g\big(\mathbf{Z}_{\text{label}}[\ell_i]\big)$. This design treats label embeddings as feature-level anchors for labeled training nodes, rather than as signals to be propagated across the graph. The stochastic masking mechanism acts as a form of regularization, preventing the model from over-relying on visible labels during training. Disabling masking at inference removes this stochasticity without introducing any additional supervision, ensuring that no extra label information beyond the observed training labels is used at test time.

3.5 Node-Level Semantics–Structure Fusion

Gating-Based Fusion. To determine the node-dependent importance of the structural signal, we first concatenate the two channel outputs to form a joint feature $\mathbf{z}_v = \big[\, \mathbf{h}_v^{(s)} \,\|\, \mathbf{h}_v^{(r)} \,\big]$. This vector is fed into a single-layer feed-forward network to produce a scalar gate $\gamma_v = \sigma(f_{\text{gate}}(\mathbf{z}_v))$, where $f_{\text{gate}}(\cdot)$ is a learnable linear transformation and $\sigma(\cdot)$ is the sigmoid function. The final fused embedding of node v is then obtained as

$$\mathbf{h}_v = (1 - \gamma_v)\, \mathbf{h}_v^{(s)} + \gamma_v\, \mathbf{h}_v^{(r)}. \tag{9}$$

This node-level fusion mechanism endows the model with local adaptivity, allowing it to dynamically adjust the contribution of the semantic and structural channels based on each node's neighborhood context and feature distribution.

Decoupling Regularization. To further encourage semantics–structure decoupling, we introduce a regularization term that penalizes linear dependence between the two representation spaces. Let $\mathbf{H}^{(s)}, \mathbf{H}^{(r)} \in \mathbb{R}^{B\times d}$ denote the semantic and structural embeddings of the same $B > 1$ target nodes in the current full batch or sampled mini-batch. Here, d is the shared embedding dimension.

We use the empirical cross-covariance to quantify linear dependence between the semantic and structural channels. We first center the embeddings:

$$\overline{\mathbf{H}}^{(k)} = \mathbf{H}^{(k)} - \frac{1}{B}\mathbf{1}_B\mathbf{1}_B^\top\mathbf{H}^{(k)}, \qquad k \in \{s, r\},$$

where $\mathbf{1}_B$ is the all-ones vector of length B, and $\overline{\mathbf{H}}^{(k)}$ denotes the centered embedding matrix for channel k. The decoupling loss is defined as

$$\mathcal{L}_{\text{dec}} = \left\| \frac{(\overline{\mathbf{H}}^{(s)})^\top \overline{\mathbf{H}}^{(r)}}{B-1} \right\|_F, \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Penalizing this cross-covariance discourages linear dependence between the two channels and encourages them to learn complementary representations.

3.6 Training Objective

For the downstream node classification task, we adopt the standard cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{V}_{\text{train}}|} \sum_{u \in \mathcal{V}_{\text{train}}} \ell_{\text{CE}}(\tilde{y}_u, y_u), \tag{11}$$

where $\mathcal{V}_{\text{train}}$ denotes the set of labeled training nodes and $\ell_{\text{CE}}(\cdot,\cdot)$ is the cross-entropy loss (for multi-label classification we use binary cross-entropy instead). The overall training objective of HeterSEED is

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{dec}}, \tag{12}$$

where α is a hyperparameter that controls the strength of the semantics–structure decoupling regularization.

4 Theoretical Insights

This section provides theoretical insights into the expressive power, prediction bias, and generalization behavior of HeterSEED. We first identify conditions under which structural information enables HeterSEED to distinguish nodes that purely semantic HGNNs cannot. We then examine the effect of homophily–heterophily separation on prediction bias. Finally, we discuss how semantics–structure decoupling relates to representation complexity and generalization.

Setup and Hypothesis Class. Let $\mathcal{G} = (V, E)$ be a heterogeneous graph with node features $\mathbf{X} = (\mathbf{x}_v)_{v \in V}$ and labels $y_v \in \{1, \ldots, C\}$. The heterogeneous semantic aggregation channel of HeterSEED produces embeddings $\mathbf{h}_v^{(s)} = f_s(\mathcal{G}, \mathbf{X}; \theta_s)$, while the structure-aware heterophily channel produces $\mathbf{h}_v^{(r)} = f_r(\mathcal{G}_{\text{homo}}, \mathcal{G}_{\text{hetero}}, \mathbf{X}; \theta_r)$, where $\mathcal{G}_{\text{homo}}$ and $\mathcal{G}_{\text{hetero}}$ are the homophilic and heterophilic graphs constructed in Section 3. The node-level fusion mechanism yields the final embedding

$$\mathbf{h}_v = (1 - \gamma_v)\, \mathbf{h}_v^{(s)} + \gamma_v\, \mathbf{h}_v^{(r)}, \tag{13}$$

where $\gamma_v = \sigma(f_{\text{gate}}([\mathbf{h}_v^{(s)} \| \mathbf{h}_v^{(r)}]))$ is a node-dependent gate and a linear classifier is applied on $\mathbf{h}_v$ to predict labels. We denote by $\mathcal{F}_s$ the hypothesis class of models that only use the semantic channel (standard HGNNs on heterogeneous graphs), and by $\mathcal{F}_{\text{HeterSEED}}$ the class induced by the full HeterSEED architecture.

4.1 Expressive Power of $\mathcal{F}_{\text{HeterSEED}}$

Theorem 1 (Expressiveness Gain)**.** *Consider the semantic message-passing class $\mathcal{F}_s$ specified in Section 3.2, without additional structural or label-feature encodings. There exists a heterophilic heterogeneous graph G with node features X and labels y such that: (i) every $f_s \in \mathcal{F}_s$ of finite depth and width has classification error $R(f_s) \geq \varepsilon_0$ for some $\varepsilon_0 > 0$; (ii) conditional on a correct pseudo-label-induced partition, there exists a HeterSEED model of finite depth and width with $R(f) = 0$, using the same raw features and with label-feature augmentation disabled.*

Remark. Theorem 1 establishes an existential, conditional expressiveness separation for the specified semantic aggregation class. Identical features and a regular typed graph isolate structural information that local semantic message passing cannot distinguish. The positive direction additionally assumes a correct pseudo-label-induced partition; it does not establish that iterative pseudo-labeling recovers this partition, that training reaches zero error, or that HeterSEED outperforms every HGNN on every heterophilic graph (see **Appendix C.2** for more details).

4.2 Bias Reduction Under Heterophily

We next study how separating homophilic and heterophilic neighbors reduces the bias introduced by heterophilic edges. For clarity, we work in a squared-loss regression setting with scalar labels $y_v \in \{-1, +1\}$ and linear aggregation (the classification case is analogous; see **Appendix C.3**). Let q_v denote the fraction of heterophilic neighbors of node v in the metapath-induced graph, and let $\tilde{q}_v$ denote the effective fraction after applying the structure-aware heterophily channel and node-level fusion in HeterSEED, which down-weights heterophilic neighbors via structure-aware weights and gating.

Proposition 1 (Bias Reduction via Homophily/Heterophily Separation)**.** *Assume that (i) the pseudo-labels used to construct $\mathcal{G}_{homo}$ and $\mathcal{G}_{hetero}$ are δ-accurate, i.e., $\mathbb{P}(y_v^{pseudo} \neq y_v) \leq \delta$, and (ii) the gating function f_{gate} is L-Lipschitz. Let $R(\cdot)$ denote the squared-loss risk. Then there exist universal constants $c, C > 0$ such that, for any purely semantic HGNN $f_s \in \mathcal{F}_s$ and any HeterSEED model $f \in \mathcal{F}_{HeterSEED}$,*

$$\mathbb{E}[R(f_s)] - \mathbb{E}[R(f)] \;\geq\; c\,\mathbb{E}\big[(q_v - \tilde{q}_v)^2\big] - C\delta.$$

Remark. Proposition 1 formalizes that heterophilic neighbors act as structured noise for feature-similarity-driven HGNNs, so a larger heterophily ratio q_v implies larger potential bias. By separating homophilic and heterophilic neighbors and reweighting them via structure-aware weights and pseudo-labels, HeterSEED reduces the effective heterophilic mass from q_v to $\tilde{q}_v$, thereby lowering bias when pseudo-labels are reasonably accurate; see **Appendix C.3**.

5 Experiments

We conduct extensive experiments to evaluate the effectiveness of HeterSEED and address the following research questions:

- **RQ1**: How does HeterSEED perform on node classification compared with state-of-the-art HGNNs and heterophily-aware graph models?
- **RQ2**: How do the key components of HeterSEED contribute to its overall performance?
- **RQ3**: How does HeterSEED perform across different levels of homophily and heterophily?

5.1 Datasets

We evaluate HeterSEED on five real-world heterogeneous graphs, including three widely used benchmarks: DBLP [22], IMDB [22], and ACM [32], and two **large-scale** networks: MAG [31] and RCDD [20]. ACM and DBLP are academic citation networks, IMDB is a heterogeneous movie network, while MAG is a large-scale academic graph with 1,939,743 nodes and 21,111,007 edges, and RCDD is a real-world risk commodity detection network with 13,806,619 nodes and 157,814,864 edges. These two **large-scale datasets** allow us to assess the scalability of HeterSEED on heterogeneous graphs with millions of nodes and hundreds of millions of edges. Further details on these datasets are provided in **Appendix F**.

5.2 Baselines and Experimental Settings

We compare HeterSEED with four groups of baselines: (i) vanilla models, including MLP, GCN [14], and GAT [30]; (ii) state-of-the-art models for heterophilic graphs in the homogeneous setting, including LINKX [19], FAGCN [4], ACM-GCN [21], and GRAIN [37]; (iii) representative HGNNs, including RGCN [25], RGAT [12], HAN [32], HGT [11], SHGN [22], HINormer [23], and DiffGraph [18]; and (iv) the heterophily-aware heterogeneous model HETERO^{2}NET [16]. These baselines cover standard HGNN architectures as well as recent methods tailored to heterophilic graphs. We evaluate all methods on node classification using Macro-F1 and Micro-F1, and additionally report Average Precision (AP) on RCDD in Appendix H.5 following [20]. All experiments use the standard train/validation/test splits from prior work; each setting is repeated five times with different random seeds, and we report the mean and standard deviation. Additional baseline descriptions, experimental settings, and hyperparameter configurations are provided in **Appendix G**.

5.3 Overall Performance Comparison (RQ1)

To address **RQ1**, we compare HeterSEED with 15 baselines on five benchmark datasets, and report Macro-F1 and Micro-F1 in Table 1. Across all datasets and metrics, HeterSEED consistently achieves the best performance, outperforming vanilla GNNs, heterophily-oriented homogeneous models, heterogeneous GNNs, and the heterophily-aware heterogeneous baseline. Relative to heterophily-oriented homogeneous models, HeterSEED shows clear improvements, suggesting that modeling heterophily purely on a collapsed homogeneous view cannot fully capture the rich type-specific information present in heterogeneous graphs. When compared with representative HGNNs, HeterSEED consistently produces higher Macro-F1 and Micro-F1 scores on all five datasets. These HGNNs are effective on relatively homophilic settings but their performance degrades on low-homophily datasets such as IMDB, MAG, and RCDD, where feature-similarity-driven aggregation tends to propagate noisy signals. In contrast, HeterSEED maintains strong performance, which is consistent with our theoretical result in **Theorem 1** that semantics–structure decoupling yields a strictly more expressive hypothesis class than standard HGNNs on heterophilic heterogeneous graphs.

Among all baselines, HETERO^{2}NET is the most related to our work, as it is explicitly designed for heterogeneous graphs with heterophily. It frequently achieves the second-best results across datasets, yet HeterSEED consistently surpasses it on all metrics, including challenging cases such as IMDB, MAG, and RCDD. Moreover, on the two large-scale datasets MAG and RCDD, where several baselines run out of memory or suffer notable performance drops, HeterSEED remains trainable and achieves the best scores. The fact that HeterSEED suffers much less degradation as heterophily increases, while other HGNNs and HETERO^{2}NET become biased toward noisy neighbors, empirically echoes **Proposition 1**, which shows that separating homophilic and heterophilic neighborhoods reduces the prediction bias induced by heterophilic edges.

Overall, these results indicate that the semantics–structure decoupling with structure-aware fusion not only improves accuracy under heterophily at scale, but also behaves in line with the expressiveness and bias-reduction properties predicted by our theoretical analysis in Section 4.

Table 1: Performance comparison among HeterSEED and 15 baselines on node classification. The **bold** and underline records indicate the best and the second best results. "OOM": out of memory.

Method \ Dataset	DBLP		IMDB		ACM		MAG		RCDD	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
Vanilla Model										
MLP	$80.36_{\pm 0.31}$	$80.92_{\pm 0.25}$	$57.01_{\pm 0.67}$	$61.65_{\pm 0.93}$	$86.50_{\pm 0.32}$	$86.53_{\pm 0.31}$	$6.35_{\pm 0.38}$	$23.60_{\pm 0.17}$	$86.08_{\pm 0.34}$	$97.27_{\pm 0.43}$
GCN	$90.84_{\pm 0.32}$	$91.47_{\pm 0.34}$	$57.88_{\pm 1.18}$	$64.82_{\pm 0.64}$	$92.17_{\pm 0.24}$	$92.12_{\pm 0.23}$	$25.14_{\pm 0.33}$	$47.26_{\pm 0.36}$	$91.30_{\pm 0.38}$	$98.29_{\pm 0.12}$
GAT	$91.05_{\pm 0.76}$	$91.73_{\pm 0.50}$	$58.94_{\pm 1.35}$	$64.86_{\pm 0.43}$	$92.26_{\pm 0.94}$	$92.19_{\pm 0.93}$	$22.94_{\pm 0.49}$	$43.79_{\pm 0.24}$	$89.75_{\pm 0.21}$	$98.02_{\pm 0.04}$
Homogeneous Model with Heterophily										
LINKX	$75.05_{\pm 1.45}$	$77.78_{\pm 1.59}$	$58.98_{\pm 0.45}$	$62.03_{\pm 0.41}$	$89.91_{\pm 1.14}$	$89.69_{\pm 1.09}$	$14.63_{\pm 0.56}$	$33.43_{\pm 0.22}$	OOM	OOM
FAGCN	$82.40_{\pm 0.28}$	$83.08_{\pm 0.28}$	$63.68_{\pm 0.56}$	$67.49_{\pm 0.25}$	$89.27_{\pm 0.62}$	$89.34_{\pm 0.78}$	$16.87_{\pm 0.41}$	$37.99_{\pm 0.95}$	$90.62_{\pm 0.32}$	$98.09_{\pm 0.07}$
ACM-GCN	$81.80_{\pm 0.32}$	$82.91_{\pm 0.14}$	$65.44_{\pm 0.49}$	$68.71_{\pm 0.40}$	$88.97_{\pm 0.85}$	$89.21_{\pm 0.53}$	$13.42_{\pm 0.79}$	$34.47_{\pm 0.70}$	$84.89_{\pm 0.47}$	$94.12_{\pm 0.11}$
GRAIN	$78.47_{\pm 0.07}$	$79.48_{\pm 0.07}$	$59.93_{\pm 0.18}$	$63.28_{\pm 0.16}$	$92.78_{\pm 0.20}$	$92.77_{\pm 0.20}$	OOM	OOM	OOM	OOM
Heterogeneous Model under Homophily										
RGCN	$91.52_{\pm 0.50}$	$92.07_{\pm 0.50}$	$61.26_{\pm 0.33}$	$65.21_{\pm 0.73}$	$91.95_{\pm 0.44}$	$91.75_{\pm 0.35}$	$27.01_{\pm 0.21}$	$48.80_{\pm 0.24}$	$92.25_{\pm 0.34}$	$98.30_{\pm 0.07}$
HAN	$91.67_{\pm 0.69}$	$92.05_{\pm 0.62}$	$57.74_{\pm 0.96}$	$64.63_{\pm 0.58}$	$90.89_{\pm 0.43}$	$90.79_{\pm 0.43}$	$8.94_{\pm 0.16}$	$26.76_{\pm 0.36}$	$87.32_{\pm 0.34}$	$97.46_{\pm 0.07}$
RGAT	$92.61_{\pm 0.48}$	$93.15_{\pm 0.49}$	$57.85_{\pm 0.58}$	$62.79_{\pm 0.70}$	$90.03_{\pm 0.56}$	$90.40_{\pm 0.54}$	$24.76_{\pm 0.47}$	$45.29_{\pm 0.40}$	$89.19_{\pm 1.20}$	$97.93_{\pm 0.16}$
HGT	$93.01_{\pm 0.24}$	$93.49_{\pm 0.25}$	$63.07_{\pm 1.19}$	$67.20_{\pm 1.61}$	$91.12_{\pm 0.76}$	$91.32_{\pm 0.89}$	$27.87_{\pm 0.30}$	$49.19_{\pm 0.63}$	$91.04_{\pm 0.48}$	$98.29_{\pm 0.08}$
SHGN	$94.01_{\pm 0.24}$	$94.20_{\pm 0.31}$	$63.53_{\pm 1.26}$	$67.36_{\pm 0.57}$	$93.42_{\pm 0.44}$	$93.35_{\pm 0.45}$	$22.61_{\pm 0.40}$	$43.68_{\pm 0.71}$	$88.12_{\pm 0.52}$	$97.68_{\pm 0.10}$
HINormer	$93.90_{\pm 0.37}$	$94.25_{\pm 0.38}$	$64.11_{\pm 1.82}$	$67.71_{\pm 1.05}$	$92.66_{\pm 0.73}$	$93.34_{\pm 0.85}$	$25.80_{\pm 0.41}$	$47.57_{\pm 0.61}$	OOM	OOM
DiffGraph	$91.45_{\pm 0.69}$	$91.72_{\pm 0.60}$	$58.84_{\pm 0.59}$	$63.61_{\pm 0.57}$	$93.87_{\pm 0.82}$	$93.85_{\pm 0.81}$	OOM	OOM	OOM	OOM
Heterophily-aware Heterogeneous Model										
HETERO^{2}NET	$93.92_{\pm 0.45}$	$94.36_{\pm 0.57}$	$65.33_{\pm 0.72}$	$69.03_{\pm 0.78}$	$93.96_{\pm 0.88}$	$93.87_{\pm 0.76}$	$33.28_{\pm 0.32}$	$54.66_{\pm 0.25}$	$92.36_{\pm 0.19}$	$98.43_{\pm 0.05}$
HeterSEED (Ours)	$\mathbf{94.48}_{\pm 0.81}$	$\mathbf{94.87}_{\pm 0.73}$	$\mathbf{66.71}_{\pm 0.73}$	$\mathbf{70.60}_{\pm 0.82}$	$\mathbf{94.34}_{\pm 0.81}$	$\mathbf{94.21}_{\pm 0.77}$	$\mathbf{34.99}_{\pm 0.22}$	$\mathbf{56.25}_{\pm 0.25}$	$\mathbf{93.09}_{\pm 0.15}$	$\mathbf{98.60}_{\pm 0.03}$

5.4 Ablation Study (RQ2)

To address **RQ2**, we conduct an ablation study to evaluate the contribution of each component by comparing HeterSEED with several variants that remove or modify a single component: **w/o SHC** removes the pseudo-label–guided structural channel and keeps only the one-hop semantic HGNN backbone; **w/o Dec.** drops the semantics–structure decoupling loss $\mathcal{L}_{\text{dec}}$; **w/o Homo** and **w/o Hetero** disable aggregation over

Table 2: Ablation study on HeterSEED.

Dataset	DBLP		IMDB	
Model \ Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1
w/o SHC	$93.88_{\pm 0.41}$	$94.30_{\pm 0.69}$	$65.79_{\pm 0.52}$	$69.95_{\pm 0.85}$
w/o Dec	$94.16_{\pm 0.75}$	$94.59_{\pm 0.68}$	$66.28_{\pm 0.49}$	$70.29_{\pm 0.66}$
w/o Homo	$93.87_{\pm 0.54}$	$94.23_{\pm 0.58}$	$66.10_{\pm 1.17}$	$70.00_{\pm 0.82}$
w/o Hetero	$93.66_{\pm 0.74}$	$94.13_{\pm 0.74}$	$66.46_{\pm 0.74}$	$70.25_{\pm 0.68}$
w/o Mask	$93.99_{\pm 0.61}$	$94.35_{\pm 0.60}$	$66.11_{\pm 0.47}$	$69.46_{\pm 0.69}$
HeterSEED (Ours)	$\mathbf{94.48}_{\pm 0.81}$	$\mathbf{94.87}_{\pm 0.73}$	$\mathbf{66.71}_{\pm 0.73}$	$\mathbf{70.60}_{\pm 0.82}$

homophilic and heterophilic neighbors in the structural channel, respectively; and **w/o Mask** removes the label-masking mechanism from the input features.

Table 2 reports results on DBLP and IMDB. On both datasets, the full HeterSEED model achieves the best Macro-F1 and Micro-F1, while all ablated variants show clear performance drops. Removing either the structure-aware channel (**w/o SHC**) or its heterophilic branch (**w/o Hetero**) leads to performance degradation, supporting the effectiveness of pseudo-label-guided structural modeling and explicit heterophilic neighborhood aggregation. Removing the decoupling loss (**w/o Dec.**) also consistently hurts performance, indicating that enforcing semantics–structure decoupling is beneficial in practice, in line with our theoretical analysis. Ablation results on the other three datasets exhibit the same trend, i.e., the full HeterSEED model consistently outperforms all variants; detailed results are provided in **Appendix H.1**.

5.5 Performance under Varying Homophily (RQ3)

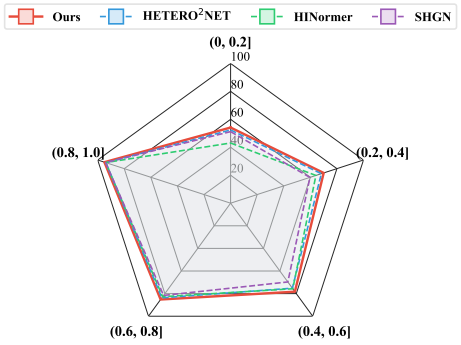


Figure 4: Performance of HeterSEED and three baselines across five homophily intervals on a semi-synthetic benchmark derived from DBLP.

To investigate **RQ3**, we evaluate HeterSEED on a semi-synthetic benchmark derived from DBLP, with homophily manipulated using the SBM-based procedure described in Appendix H.3. For each node, we compute its local homophily ratio (the fraction of neighbors sharing the same label in the metapath-induced graph) and partition nodes into five intervals. We then measure node classification performance within each interval. Figure 4 shows a radar plot comparing HeterSEED with three representative baselines: HETERO^{2}NET, HINormer, and SHGN. HeterSEED consistently achieves the best performance across all homophily ranges, with the most pronounced gains in the strongly heterophilic regimes $(0, 0.2]$ and $(0.2, 0.4]$, where other methods suffer from noisy cross-class aggregation. Even in high-homophily intervals, HeterSEED remains at least on par with the strongest baseline, indicating that semantics–structure decoupling does not sacrifice performance when homophily is high. These observations empirically support our bias-reduction analysis in Proposition 1, showing that explicitly separating and reweighting homophilic and heterophilic neighbors improves robustness under varying degrees of homophily. Detailed experimental settings are provided in **Appendix H.3**.

6 Conclusion

In this paper, we propose HeterSEED, a heterophily-aware semantics–structure decoupling framework for learning on heterogeneous graphs. HeterSEED combines a heterogeneous semantic aggregation channel with a structure-aware heterophily channel that separates homophilic and heterophilic neighborhoods, and fuses them via node-level adaptive gating. We provide theoretical analysis showing that HeterSEED improves expressive power over standard HGNNs and reduces the bias introduced by heterophilic neighbors. Experiments on five heterogeneous graphs, including two large-scale networks, demonstrate that HeterSEED generally achieves leading performance over representative HGNNs and recent heterophily-aware baselines. For future work, it is desirable to extend this semantics–structure decoupling paradigm to heterogeneous hypergraphs under heterophily.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 62622615, No. 62536006, No. 62676036, No. 62576371, No. 62521007, No. U23A20388), the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C03262, No. 2026C02A1239, No. 2026C02A1234), and the Joint Fund of the Science and Technology R&D Program of Henan (No. 235200810031).

References

[1] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[2] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NeurIPS*, pages 6241–6250, 2017.

[3] Rui Bing, Guan Yuan, Mu Zhu, Fanrong Meng, Huifang Ma, and Shaojie Qiao. Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications. *Artificial Intelligence Review*, 56(8):8003–8042, 2023.

[4] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *AAAI*, pages 3950–3957, 2021.

[5] Qin Chen and Guojie Song. Adaptive heterogeneous graph neural networks: Bridging heterophily and heterogeneity. In *CIKM*, pages 312–321, 2025.

[6] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, pages 135–144, 2017.

[7] Chenguang Du, Kaichun Yao, Hengshu Zhu, Deqing Wang, Fuzhen Zhuang, and Hui Xiong. Seq-HGNN: learning sequential node representation on heterogeneous graph. In *SIGIR*, pages 1721–1730, 2023.

[8] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*, pages 2331–2341, 2020.

[9] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[10] Cheng-Hua Gong, Yao Cheng, Jian-Xiang Yu, Can Xu, Cai-Hua Shan, Si-Qiang Luo, and Xiang Li. A survey on learning from graphs with heterophily: Recent advances and future directions. *Frontiers of Computer Science*, 20(2):2002314, 2026.

[11] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *WWW*, pages 2704–2710, 2020.

[12] Taichi Ishiwatari, Yuki Yasuda, Taro Miyazaki, and Jun Goto. Relation-aware graph attention networks with relational position encodings for emotion recognition in conversations. In *EMNLP*, pages 7360–7370, 2020.

[13] Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NeurIPS*, pages 793–800, 2008.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[15] Jianfei Li, Ruigang Zheng, Han Feng, Ming Li, and Xiaosheng Zhuang. Permutation equivariant graph framelets for heterophilous graph learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11634–11648, 2024.

[16] Jintang Li, Zheng Wei, Yuchang Zhu, Ruofan Wu, Huizhe Zhang, Liang Chen, and Zibin Zheng. Heterophily-aware representation learning on heterogeneous graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9):7852–7866, 2025.

[17] Ming Li, Alessio Micheli, Yu Guang Wang, Shirui Pan, Pietro Lió, Giorgio Stefano Gnecco, and Marcello Sanguineti. Guest editorial: Deep neural networks for graphs: Theory, models, algorithms, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 35 (4):4367–4372, 2024.

[18] Zongwei Li, Lianghao Xia, Hua Hua, Shijie Zhang, Shuangyang Wang, and Chao Huang. DiffGraph: Heterogeneous graph diffusion model. In *WSDM*, pages 40–49, 2025.

[19] Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *NeurIPS*, pages 20887–20902, 2021.

[20] Yijian Liu, Hongyi Zhang, Cheng Yang, Ao Li, Yugang Ji, Luhao Zhang, Tao Li, Jinyu Yang, Tianyu Zhao, Juan Yang, Hai Huang, and Chuan Shi. Datasets and interfaces for benchmarking heterogeneous graph neural networks. In *CIKM*, pages 5346–5350, 2023.

[21] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *NeurIPS*, pages 1362–1375, 2022.

[22] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *KDD*, pages 1150–1160, 2021.

[23] Qiheng Mao, Zemin Liu, Chenghao Liu, and Jianling Sun. HINormer: Representation learning on heterogeneous information networks with graph transformer. In *WWW*, pages 599–610, 2023.

[24] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *ICLR*, 2020.

[25] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, pages 593–607, 2018.

[26] Zhixiang Shen and Zhao Kang. When heterophily meets heterogeneous graphs: Latent graphs guided unsupervised representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):10283–10296, 2025.

[27] Chuan Shi, Xiao Wang, and Philip S. Yu. *Heterogeneous graph representation learning and applications*. Singapore: Springer, 2022.

[28] Yizhou Sun and Jiawei Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers, 2012.

[29] Zhen Tao, Ziyue Qiao, Chaoqi Chen, Zhengyi Yang, Lun Du, and Qingqiang Sun. On the cross-type homophily of heterogeneous graphs: Understanding and unleashing. In *CIKM*, pages 2842–2852, 2025.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[31] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020.

[32] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *WWW*, pages 2022–2032, 2019.

[33] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S Yu. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9(2):415–436, 2022.

[34] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. In *AAAI*, pages 10816–10824, 2023.

[35] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. In *NeurIPS*, pages 11983–11993, 2019.

[36] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *KDD*, pages 793–803, 2019.

[37] Songwei Zhao, Yuan Jiang, Zijing Zhang, Yang Yu, and Hechang Chen. GRAIN: Multi-granular and implicit information aggregation graph neural network for heterophilous graphs. In *AAAI*, pages 13383–13391, 2025.

[38] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 38(7):4385–4404, 2026.

[39] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, pages 7793–7804, 2020.

[40] Jiong Zhu, Yujun Yan, Mark Heimann, Lingxiao Zhao, Leman Akoglu, and Danai Koutra. Heterophily and graph neural networks: Past, present and future. *IEEE Data Engineering Bulletin*, 47(2):10–32, 2023.

A Summary of Notation

Table 3 summarizes the main notation used throughout the main text.

Table 3: Summary of main notation used in this paper.

Symbol	Description
$\mathcal{G} = (V, E, T, R, \phi, \psi)$	Heterogeneous graph with node / edge sets, types and mappings
V, E	Sets of nodes and edges
T, R	Sets of node types and relation types
$\phi(v), \psi(e)$	Type of node v and edge e
$\mathbf{x}_v \in \mathbb{R}^{d_{\text{feat}}}$	Input feature of node v
y_v	Ground-truth label of node v
$\hat{y}_v$	Predicted label of node v
y_v^{pseudo}	Pseudo-label of node v (Eq. (3))
$\mathcal{V}_{\text{train}}$	Set of labeled training nodes
$p \in \mathcal{P}$	(Symmetric) metapath, $\mathcal{P}$ the metapath set
$\mathcal{G}_p$	Metapath-induced homogeneous graph for p
$\mathcal{N}_{\mathcal{G}_p}(v)$	Neighbors of v in $\mathcal{G}_p$
$\mathcal{E}^p_{\text{homo}}$, $\mathcal{E}^p_{\text{hetero}}$	Homophilic / heterophilic edge sets under p
$\mathcal{G}_{\text{homo}}$, $\mathcal{G}_{\text{hetero}}$	Aggregated homophilic / heterophilic subgraphs
$\mathcal{N}^v_{\text{homo}}$, $\mathcal{N}^v_{\text{hetero}}$	Homophilic / heterophilic neighbors of v
$\mathbf{h}_v^{(s)}$	Semantic embedding of node v (semantic channel)
$\mathbf{h}_v^{(r)}$	Structural embedding of node v (heterophily channel)
$\mathbf{h}_v$	Final fused embedding of node v (Eq. (9))
$\mathbf{H}^{(s)}, \mathbf{H}^{(r)}$	Matrices of semantic / structural embeddings for a node batch
$\mathbf{W}, \mathbf{W}_r$	Trainable weights in neighborhood-wise convolution
$\omega_{u,v}$, $\tilde{\omega}_{u,v}$	Raw / normalized structural weight from u to v
$\mathbf{h}_{\text{homo}}$, $\mathbf{h}_{\text{hetero}}$	Aggregated homophilic / heterophilic structural messages
$\mathcal{T}_{\text{homo}}(\cdot)$, $\mathcal{T}_{\text{hetero}}(\cdot)$	two-layer MLPs with a shared output dimension
γ_v	Node-level fusion gate for combining $\mathbf{h}_v^{(s)}$ and $\mathbf{h}_v^{(r)}$
$\mathbf{Z}_{\text{label}} \in \mathbb{R}^{(C+1)\times d}$	Label embedding matrix with C classes and one `[MASK]` token
$\tilde{\mathbf{z}}_v^{(\text{label})}$	(Possibly masked) label embedding of node v
α	Coefficient of decoupling regularization $\mathcal{L}_{\text{dec}}$
β	Masking probability in label-guided masking module
$\mathcal{L}_{\text{cls}}$	Supervised classification loss (cross-entropy)
$\mathcal{L}_{\text{dec}}$	Semantics–structure decoupling regularization loss
$\mathcal{L}$	Overall training objective $\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{dec}}$

B Related Work

B.1 Heterogeneous Graph Neural Networks

Heterogeneous graphs are ubiquitous in real-world applications, where multiple node and edge types encode rich semantic information [33, 27]. To model such structures, numerous heterogeneous graph neural networks (HGNNs) have been proposed, which are broadly categorized into metapath-based and metapath-free approaches [3]. Metapath-based methods capture high-order semantic dependencies by propagating and aggregating information along predefined or learned metapaths. Representative examples include MetaPath2Vec [6] for metapath-guided random-walk embeddings, HAN [32] with hierarchical attention over node- and semantic-level contexts, MAGNN [8] with semantic-specific message passing, and GTN [35], which learns soft adjacency matrices to discover informative metapaths. Seq-HGNN [7] further exploits sequential node embeddings to model order-dependent semantics.

Metapath-free HGNNs instead generalize standard message passing to heterogeneous graphs without relying on hand-crafted metapaths. RGCN [25] and RGAT [12] introduce relation-specific transformations and attention for different edge types, SHGN [22] integrates multi-layer attention with learnable edge embeddings, while Transformer-based architectures such as HGT [11] and HINormer [23] employ self-attention to model heterogeneous interactions. DiffGraph [18] adopts latent heterogeneous graph diffusion with cross-view denoising to alleviate noise and capture semantic transitions among heterogeneous relations.

Although these models effectively capture heterogeneous semantics, most are developed under a homophily assumption and mainly rely on feature-similarity-driven aggregation. This design is less reliable on heterogeneous graphs with pronounced heterophily, where neighboring nodes may belong to different semantic categories or play distinct structural roles [26]. Recent work such as HETERO^{2}NET [16] begins to explore heterophily-aware representation learning on heterogeneous graphs by introducing multiple representation branches, and other studies consider heterophily-aware message passing across metapaths [5] or homophily-guided structure optimization [29]. However, these approaches still largely depend on feature-based aggregation or coarse-grained semantic integration and do not explicitly disentangle homophilic and heterophilic structural patterns at a fine granularity, which may limit their ability to exploit discriminative heterophilic structures.

B.2 Graph Neural Networks with Heterophily

Graphs with heterophily, where connected nodes often exhibit dissimilar attributes or labels, pose significant challenges for conventional GNNs [38]. A number of methods have been proposed to address heterophily in homogeneous graphs. H2GCN [39] separates ego- and neighbor-embeddings, leverages higher-order neighborhoods, and combines intermediate representations to better exploit structural signals. Geom-GCN [24] introduces a geometric aggregation scheme that preserves neighborhood structure and captures long-range dependencies in disassortative graphs via a latent geometric space. PEGFAN [15] uses permutation-equivariant Haar-type graph framelets to extract multiscale representations for learning on heterophilous graphs. FAGCN [4] studies low- and high-frequency components of graph signals and proposes a self-gating mechanism to adaptively integrate them. LINKX [19] departs from message passing and directly combines node features and adjacency via scalable linear transformations, achieving strong performance on large non-homophilous graphs. ACM-GCN [21] revisits homophily from a post-aggregation similarity viewpoint and designs an adaptive channel mixing framework that dynamically combines aggregation, diversification, and identity channels. GRAIN [37] aggregates multi-view information at different granularity levels and introduces an adaptive aggregator to fuse local and global signals, improving robustness across homophily regimes.

These approaches, while effective on homogeneous graphs, do not directly extend to heterogeneous settings, where multiple node types and complex semantic relations substantially increase modeling difficulty. Moreover, they generally lack an explicit mechanism to simultaneously capture heterogeneous semantics and heterophilic structural patterns, often relying on unified aggregation or coarse channel fusion. This gap motivates methods that jointly handle heterogeneity and heterophily in a principled manner.

B.3 Positioning of HeterSEED

Our work lies at the intersection of heterogeneous graph learning and heterophilic GNNs. HETERO^{2}NET [16] is the closest prior work, as it explicitly targets heterophily-aware representation learning on heterogeneous graphs via multiple interaction branches. However, it does not explicitly separate homophilic and heterophilic structural patterns, nor does it perform semantics–structure decoupling with node-wise adaptive fusion. LatGRL [26] also introduces a heterophily-aware perspective for heterogeneous graphs, but focuses on unsupervised representation learning, which differs from our semi-supervised node classification setting. SeHGNN [34] focuses on efficient aggregation and fusion of metapath semantics. It does not explicitly partition metapath neighbors into pseudo-label-conditioned homophilic and heterophilic sets for separate count-weighted aggregation. This distinguishes our structural operator from generic multi-channel semantic fusion.

In contrast, HeterSEED adopts a dual-channel design that decouples heterogeneous semantic aggregation from structure-aware heterophily modeling. The heterogeneous semantic aggregation channel captures type-specific, relation-aware semantics via first-order heterogeneous neighborhoods, while the structure-aware heterophily channel explicitly distinguishes homophilic and heterophilic neighborhoods, aggregates them with metapath-based structural weights, and encodes discriminative structural patterns. A node-level adaptive fusion module then assigns fusion weights based on the concatenated semantic and structural embeddings, enabling each node to balance semantic consistency and structural discrimination according to its local context. Extensive experiments on multiple heterogeneous benchmarks show that this semantics–structure decoupling leads to consistent gains over state-of-the-art HGNNs and recent heterophily-aware baselines.

C Additional Details for Theoretical Analysis

In this appendix, we provide detailed proofs and supplementary explanations for the theoretical results stated in Section 4.

C.1 Preliminaries and Function Classes

Let $\mathcal{G} = (V, E)$ be a heterogeneous graph with node feature matrix $\mathbf{X} = (\mathbf{x}_v)_{v\in V}$, label vector $(y_v)_{v\in V}$, and a schema with node types T and relation types R. For simplicity, we consider node-level prediction and assume that $y_v \in \{1, \dots, C\}$ unless otherwise specified.

Semantic channel. The heterogeneous semantic aggregation channel of HeterSEED is an HGNN-style message-passing architecture that produces, for each node v, an embedding

$$\mathbf{h}_v^{(s)} = f_s(\mathcal{G}, \mathbf{X}; \theta_s) \in \mathbb{R}^{d_s},$$

where θ_s collects all trainable parameters in the semantic channel. Concretely, f_s is implemented as L_s stacked layers of relation-specific linear transformations, aggregation over type- and relation-specific neighborhoods, followed by pointwise nonlinearities (e.g., ReLU), as described in Section 3.2.

Structure-aware heterophily channel. The structure-aware heterophily channel constructs metapath-induced homogeneous graphs, separates homophilic and heterophilic edges based on pseudo-labels, and aggregates structure-aware signals with metapath-based structural weights and the homophily–heterophily structural fusion layer. This channel produces embeddings

$$\mathbf{h}_v^{(r)} = f_r(\mathcal{G}_{\text{homo}}, \mathcal{G}_{\text{hetero}}, \mathbf{X}; \theta_r) \in \mathbb{R}^{d_r},$$

where $\mathcal{G}_{\text{homo}}$ and $\mathcal{G}_{\text{hetero}}$ are the homophilic and heterophilic graphs constructed in Section 3.3 and θ_r collects all trainable parameters of this channel.

Node-level fusion and classifier. Given $\mathbf{h}_v^{(s)}$ and $\mathbf{h}_v^{(r)}$, the fusion module computes a node-dependent gate

$$\gamma_v = \sigma\big(f_{\text{gate}}([\mathbf{h}_v^{(s)} \| \mathbf{h}_v^{(r)}])\big) \in (0, 1),$$

where f_{gate} is a linear (or shallow MLP) mapping and σ is the sigmoid function. The fused embedding is

$$\mathbf{h}_v = (1 - \gamma_v)\, \mathbf{h}_v^{(s)} + \gamma_v\, \mathbf{h}_v^{(r)} \in \mathbb{R}^d,$$

and a linear classifier g produces logits

$$\mathbf{o}_v = W_{\text{cls}}\mathbf{h}_v + \mathbf{b}_{\text{cls}},$$

followed by softmax to obtain class probabilities.

Function classes. We denote by: (i) $\mathcal{F}_s$ the hypothesis class of models that use only the semantic channel and a linear classifier; i.e., models of the form $v \mapsto g_s(f_s(\mathcal{G}, \mathbf{X}; \theta_s))$, which correspond to standard heterogeneous GNNs; (ii) $\mathcal{F}_{\text{HeterSEED}}$ the hypothesis class of models that use both the semantic channel, the structure-aware heterophily channel, the node-level fusion gate, and a linear classifier; i.e., $v \mapsto g(f_s(\mathcal{G}, \mathbf{X}; \theta_s), f_r(\mathcal{G}_{\text{homo}}, \mathcal{G}_{\text{hetero}}, \mathbf{X}; \theta_r))$. Theorem 1 in the main text compares the expressive power of $\mathcal{F}_s$ and $\mathcal{F}_{\text{HeterSEED}}$ on a specific class of heterophilic heterogeneous graphs.

For risk, unless otherwise stated we write

$$R(f) := \mathbb{E}[\ell(f(v), y_v)],$$

where ℓ is a non-negative loss (e.g., 0-1 loss or squared loss) and the expectation is taken with respect to the data distribution over $(\mathcal{G}, \mathbf{X}, y)$ and a uniformly sampled node v.

C.2 Proof of Theorem 1

Graph construction. Consider a heterogeneous graph with author and paper node types, $\mathcal{T} = \{A, P\}$, and the author–paper relation together with its inverse. We focus on the symmetric metapath APA. Construct $2n$ author nodes

$$A_+ = \{a_1, \ldots, a_n\}, \qquad A_- = \{b_1, \ldots, b_n\},$$

with labels

$$y_{a_i} = +1, \qquad y_{b_i} = -1, \qquad i = 1, \ldots, n.$$

Paper nodes are not classification targets. All authors share identical features:

$$x_{a_i} = x_{b_j} = x_0 \in \mathbb{R}^{d_0}, \qquad \forall i, j.$$

For this construction, take $n = 4$, $d_0 = 1$, and $x_0 = 1$, and assign all paper nodes the same scalar feature. Label-feature augmentation is disabled as stipulated in Theorem 1. Create one distinct paper for each pair $(a_1, a_2), (a_2, a_3), (a_3, a_4), (a_4, a_1)$, two distinct papers for each pair $(b_1, b_2), (b_3, b_4)$, and one distinct paper for every cross-class pair (a_i, b_j). Each paper is incident to exactly its two coauthors, with both relation directions available for message passing. There are 24 papers: every author has six paper neighbors, and every paper has two author neighbors. Including self-neighbors in the APA projection, the local homophily ratios of positive and negative authors are $3/7$ and $1/3$, respectively, both below $1/2$. Throughout this proof, R denotes the 0–1 classification error under a uniform choice of the eight authors.

Limitations of standard HGNNs. We consider the semantic message-passing class $\mathcal{F}_s$ specified in Section 3.2, without additional structural or label-feature encodings. Its layer update has the form

$$h_v^{(l+1)} = \sigma\left(W_{\text{self}}^{(l)} h_v^{(l)} + \sum_{r \in \mathcal{R}} \frac{1}{|\mathcal{N}_r(v)|} \sum_{u \in \mathcal{N}_r(v)} W_r^{(l)} h_u^{(l)} \right),$$

with $h_v^{(0)} = x_v$, where empty-relation contributions are taken to be zero. The transformations are shared within the corresponding node and relation types. Since features are constant within each node type and all nodes of a given type have the same relation-specific degrees, the following representation-collapse property holds.

Lemma 1 (Representation collapse under symmetry)**.** *On the graph constructed above, every semantic HGNN $f_s \in \mathcal{F}_s$ of finite depth and width satisfies*

$$h_{a_i}^{(L)} = h_{b_j}^{(L)}, \qquad \forall i, j,$$

where L is its number of layers.

Proof. The base case holds because the initial features are constant within each node type. Assume that, at layer l, all authors share a common representation and all papers share a common representation. Every author then receives six identical paper representations, while every paper receives two identical author representations. The shared relation-specific transformations, aggregation, self-transformation, and nonlinearity therefore preserve equality within each node type at layer $l+1$. Induction establishes the claim for every finite depth. □

Implication for classification. Let $h^{(L)}$ denote the common last-layer author embedding. The classifier produces identical logits

$$o_v = W_{\text{cls}} h^{(L)} + b_{\text{cls}}$$

for all authors, and therefore predicts the same label for all of them. Since the positive and negative classes have equal size,

$$R(f_s) \geq \frac{1}{2} =: \varepsilon_0 > 0.$$

This proves item (i) of Theorem 1. We next establish the positive direction on the same graph.

Metapath-based structural distinction. We examine the APA counts without changing the graph's edges or features. Retaining self-neighbors as in Definition 3 gives $C_{APA}(v,v) = 6$ for every author. Each positive author has two other same-class neighbors with count one each, whereas each negative author has one other same-class neighbor with count two. Every author has four opposite-class neighbors, each with count one. All remaining distinct-author counts are zero. The structure-aware weights are

$$\omega_{u,v} = C_{APA}(u,v), \qquad \widetilde{\omega}_{u,v} = \frac{\exp(\omega_{u,v})}{\sum_{u' \in \mathcal{N}_{APA}(v)} \exp(\omega_{u',v})},$$

where $\mathcal{N}_{APA}(v) = \{u : C_{APA}(u,v) > 0\}$ includes v. These weights retain the different distributions of within-class APA counts in the two author classes.

Pseudo-labels and homophily/heterophily separation. For the positive direction, condition on a fixed pseudo-label assignment that induces the correct homophilic and heterophilic partition. The resulting subgraphs G_{homo} and G_{hetero} therefore agree with the true label-based partition. This is an explicit condition of the expressiveness comparison, not a claim that the initial classifier or iterative pseudo-labeling procedure necessarily recovers this partition.

The structural branch aggregates

$$h_{\text{homo}}(v) = \sum_{u \in \mathcal{N}^v_{\text{homo}}} \widetilde{\omega}_{u,v} x_u, \qquad h_{\text{hetero}}(v) = \sum_{u \in \mathcal{N}^v_{\text{hetero}}} \widetilde{\omega}_{u,v} x_u,$$

and applies the branch transformations and sigmoid gate:

$$h_v^{(r)} = \sigma(g)\, T_{\text{homo}}(h_{\text{homo}}(v)) + \big(1 - \sigma(g)\big)\, T_{\text{hetero}}(h_{\text{hetero}}(v)).$$

Here, a scalar structural embedding suffices. Since $x_u = 1$ for every author,

$$h_{\text{homo}}(v) = \begin{cases} S_+ := \dfrac{e^6 + 2e}{e^6 + 6e}, & v \in A_+, \\ S_- := \dfrac{e^6 + e^2}{e^6 + e^2 + 4e}, & v \in A_-. \end{cases}$$

Both expressions have the form $N/(N+4e)$. Because $e^2 > 2e$, we have $S_- > S_+$. Choose $\sigma(g) = 1/2$, $T_{\text{homo}}(t) = 2t$, and $T_{\text{hetero}}(t) = 0$. These maps are realizable by the stated finite-width ReLU MLPs on the nonnegative inputs, and yield

$$h_v^{(r)} = \begin{cases} S_+, & v \in A_+, \\ S_-, & v \in A_-. \end{cases}$$

Thus, the structural channel distinguishes the two classes under the assumed exact partition.

Fusing with the semantic channel. Choose a one-dimensional semantic output, whose common value on all authors is denoted by c. Set the node-level fusion gate to $\gamma_v \equiv 1/2$, which is realized by a zero sigmoid logit. The fused embedding is

$$h_v = (1 - \gamma_v)c + \gamma_v h_v^{(r)} = \begin{cases} c/2 + S_+/2, & v \in A_+, \\ c/2 + S_-/2, & v \in A_-. \end{cases}$$

Since $S_- > S_+$, the affine classifier

$$\widehat{y}_v = \operatorname{sign}\left(\frac{c}{2} + \frac{S_+ + S_-}{4} - h_v\right)$$

correctly predicts every author label. All parameters and both gate logits are finite. Therefore, under the exact partition assumed above, $R(f) = 0$, proving item (ii) and completing the proof of Theorem 1.

Take-home Message. Theorem 1 provides an existence example in which the structural channel distinguishes classes that the specified semantic message-passing class cannot separate. The comparison is conditional on an exact pseudo-label-induced partition and concerns representational capacity; it does not establish pseudo-label recoverability, optimization convergence, or uniform superiority across heterogeneous graphs.

C.3 Proof of Proposition 1

Semantic HGNN as a linear smoother. For clarity, we specialize to a single-layer linear semantic HGNN with scalar output; the argument can be extended to deeper networks with Lipschitz nonlinearities. Let the prediction of a purely semantic HGNN f_s at node v be written as $f_s(v) = \sum_{u \in \mathcal{N}(v)} a_{v,u}\, y_u$, where the coefficients $a_{v,u} \geq 0$ and $\sum_{u \in \mathcal{N}(v)} a_{v,u} = 1$. This covers the case where f_s aggregates neighbor labels (or proxy signals) linearly with normalized weights, as in many Laplacian smoothing interpretations.

We partition the neighborhood $\mathcal{N}(v)$ into homophilic and heterophilic parts according to the *true* labels:

$$\mathcal{N}_{\text{homo}}(v) := \{u \in \mathcal{N}(v) : y_u = y_v\}, \quad \mathcal{N}_{\text{hetero}}(v) := \{u \in \mathcal{N}(v) : y_u \neq y_v\}.$$

Define the heterophilic mass

$$q_v := \sum_{u \in \mathcal{N}_{\text{hetero}}(v)} a_{v,u}, \quad 1 - q_v = \sum_{u \in \mathcal{N}_{\text{homo}}(v)} a_{v,u}.$$

Bias computation. Assume for now that the neighbor labels satisfy $y_u = y_v$ for $u \in \mathcal{N}_{\text{homo}}(v)$ and $y_u = -y_v$ for $u \in \mathcal{N}_{\text{hetero}}(v)$. Then

$$f_s(v) = \sum_{u \in \mathcal{N}_{\text{homo}}(v)} a_{v,u} y_u + \sum_{u \in \mathcal{N}_{\text{hetero}}(v)} a_{v,u} y_u = (1 - q_v) y_v + q_v(-y_v) = (1 - 2q_v) y_v.$$

The conditional expectation of $f_s(v)$ given y_v is thus $\mathbb{E}[f_s(v) \mid y_v] = (1 - 2q_v) y_v$, and the squared bias is

$$\operatorname{Bias}_v^2(f_s) := \big(\mathbb{E}[f_s(v) \mid y_v] - y_v\big)^2 = \big((1 - 2q_v) y_v - y_v\big)^2 = (-2q_v y_v)^2 = 4q_v^2.$$

HeterSEED with homophily/heterophily separation. For HeterSEED, the structure-aware heterophily channel and node-level fusion induce new aggregation coefficients $\tilde{a}_{v,u}$ and an effective heterophilic mass

$$\tilde{q}_v := \sum_{u \in \mathcal{N}_{\text{hetero}}(v)} \tilde{a}_{v,u}, \quad 1 - \tilde{q}_v := \sum_{u \in \mathcal{N}_{\text{homo}}(v)} \tilde{a}_{v,u}.$$

The prediction at node v can be written as $f(v) = \sum_{u \in \mathcal{N}(v)} \tilde{a}_{v,u}\, y_u$, and, under the same label model, its bias becomes $\operatorname{Bias}_v^2(f) = 4\tilde{q}_v^2$.

Thus the difference in squared bias between f_s and f at node v is

$$\operatorname{Bias}_v^2(f_s) - \operatorname{Bias}_v^2(f) = 4(q_v^2 - \tilde{q}_v^2) = 4(q_v - \tilde{q}_v)(q_v + \tilde{q}_v).$$

Effect of pseudo-label accuracy and Lipschitz gate. The separation of homophilic and heterophilic neighbors in HeterSEED relies on pseudo-labels y_v^{pseudo}. By assumption, these pseudo-labels are δ-accurate:

$$\mathbb{P}(y_v^{\text{pseudo}} \neq y_v) \leq \delta.$$

On nodes where $y_v^{\text{pseudo}} = y_v$ and similarly for neighbors, the homophilic and heterophilic partitions match the true label-based partition. In this case, the design of the structure-aware heterophily channel ensures that heterophilic neighbors receive strictly smaller weights than in the purely semantic HGNN, i.e.,

$$\tilde{q}_v \leq q_v - \eta,$$

for some $\eta > 0$ that depends on the structural weights and the fusion mechanism.

On the other hand, when $y_v^{\text{pseudo}} \neq y_v$ (or some neighbor pseudo-label is incorrect), the separation may be imperfect. However, the impact of such nodes on $\tilde{q}_v$ is controlled by the Lipschitz constant L of the gate f_{gate} and the boundedness of the embeddings, which implies that the deviation $|q_v - \tilde{q}_v|$ on these nodes is at most $O(L)$.

Combining both cases, we obtain

$$\mathbb{E}\big[(q_v - \tilde{q}_v)^2\big] \geq c_1\, \mathbb{E}[q_v^2] - c_2\delta,$$

for some constants $c_1, c_2 > 0$. Intuitively, on the $(1-\delta)$ fraction of nodes with correct pseudo-labels, $q_v - \tilde{q}_v$ is guaranteed to be at least η, while on the δ fraction of nodes with incorrect pseudo-labels, the error is bounded by a constant depending on L.

From bias to risk. Using the expressions for the squared biases, we have

$$\mathbb{E}[\text{Bias}_v^2(f_s)] - \mathbb{E}[\text{Bias}_v^2(f)] = 4\,\mathbb{E}[q_v^2 - \tilde{q}_v^2].$$

Since $q_v^2 - \tilde{q}_v^2 = (q_v - \tilde{q}_v)(q_v + \tilde{q}_v)$ and $q_v + \tilde{q}_v \leq 2$, we can lower-bound $q_v^2 - \tilde{q}_v^2 \geq \frac{1}{2}(q_v - \tilde{q}_v)^2$ up to a constant factor. Therefore,

$$\mathbb{E}[\text{Bias}_v^2(f_s)] - \mathbb{E}[\text{Bias}_v^2(f)] \geq c\,\mathbb{E}[(q_v - \tilde{q}_v)^2] - C\delta,$$

for suitable constants $c, C > 0$.

The total squared-loss risk $R(\cdot)$ can be decomposed into bias and variance contributions. Under our assumptions (bounded labels, bounded aggregation weights, and Lipschitz architectures), the variance terms of f_s and f are of the same order and do not offset the bias improvement. Consequently, the same inequality holds for the risks:

$$\mathbb{E}[R(f_s)] - \mathbb{E}[R(f)] \geq c\,\mathbb{E}[(q_v - \tilde{q}_v)^2] - C\delta,$$

which establishes Proposition 1.

Take-home Message. Proposition 1 quantifies a second, complementary benefit of HeterSEED: beyond being more expressive, it is also less biased under heterophily. In feature-similarity-driven HGNNs, heterophilic neighbors behave like structured noise that systematically pulls predictions away from the true label, and the magnitude of this effect grows with the heterophily ratio q_v. By explicitly separating homophilic and heterophilic neighbors and re-weighting them through structure-aware aggregation and node-level gating, HeterSEED effectively reduces the effective heterophilic mass from q_v to $\tilde{q}_v$, thereby shrinking the bias term in the risk. This result explains why, in our experiments, HeterSEED particularly benefits nodes that sit in highly heterophilic local neighborhoods, where conventional HGNNs tend to suffer the most.

C.4 Generalization Benefit of Semantics–Structure Decoupling

Beyond improving expressiveness and reducing bias under heterophily, the semantics–structure decoupling in HeterSEED also has a regularizing effect on the hypothesis class. In this subsection, we adopt a margin-based viewpoint to relate the capacity of linear predictors on top of HeterSEED to the covariance of the fused representations, and discuss how the decoupling regularization $\mathcal{L}_{\text{dec}}$ impacts this capacity.

Formulation and basic assumptions. For ease of exposition, we consider a binary classification setting with labels $y_v \in \{-1, +1\}$ and a linear classifier on the fused embeddings. Let $\mathbf{h}_v^{(s)}, \mathbf{h}_v^{(r)} \in \mathbb{R}^d$ denote the semantic and structural embeddings of node v, and define the fused embedding

$$\mathbf{h}_v = (1 - \gamma_v)\,\mathbf{h}_v^{(s)} + \gamma_v\,\mathbf{h}_v^{(r)},$$

where $\gamma_v = \sigma(f_{\text{gate}}([\mathbf{h}_v^{(s)} \| \mathbf{h}_v^{(r)}]))$ as in the main text. A linear classifier with weight vector $\mathbf{w}$ predicts

$$f(v) = \text{sign}(\mathbf{w}^\top \mathbf{h}_v).$$

We make the following standard assumptions:

(A1) The fused embeddings are bounded: $\|\mathbf{h}_v\|_2 \le B_h$ for all nodes v.

(A2) The classifier weights are bounded: $\|\mathbf{w}\|_2 \le B_w$.

Let $\mathcal{S} = \{v_1, \ldots, v_n\}$ be a sample of n labeled nodes, and denote the fused embeddings on this sample by a matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$ with rows $\mathbf{h}_{v_i}^\top$. The empirical covariance of the fused representations is

$$\Sigma = \frac{1}{n} \mathbf{H}^\top \mathbf{H}.$$

Rademacher complexity and margin bound. Using standard Rademacher complexity arguments for linear predictors (see, e.g., [1]), one obtains the upper bound

$$\mathfrak{R}_n(\mathcal{F}) \;\le\; \frac{B_w}{\sqrt{n}} \sqrt{\mathrm{Tr}(\Sigma)}.$$

Together with classical margin-based generalization bounds for linear classifiers [13, 2], this implies that for any margin $\gamma > 0$ and any confidence parameter $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the draw of the training sample we have

$$R(f) \;\le\; \hat{R}_\gamma(f) + \mathcal{O}\left(\frac{B_w}{\gamma} \sqrt{\frac{\mathrm{Tr}(\Sigma)}{n}} \right) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\log(1/\delta)}{n}} \right),$$

where $\hat{R}_\gamma(f)$ is the empirical γ-margin error and $R(f)$ is the population risk.

Intuitive Explanation. In HeterSEED, each fused embedding $\mathbf{h}_v$ is obtained by combining a semantic embedding $\mathbf{h}_v^{(s)}$ and a structural embedding $\mathbf{h}_v^{(r)}$ through a node-level gating mechanism. The decoupling regularization $\mathcal{L}_{\text{dec}}$ explicitly discourages strong statistical dependence between these two channels (for instance, via a penalty on their empirical cross-covariance), encouraging them to capture complementary rather than redundant directions in the representation space. From a covariance-based viewpoint, this reduces redundancy in the fused representation matrix $\mathbf{H}$ and tends to concentrate the variance of Σ on fewer, more informative directions, i.e., it lowers the effective dimensionality of the fused embeddings. In light of the above Rademacher complexity bound, this contraction of the effective representation space acts as a regularizer on the hypothesis class of linear predictors on top of HeterSEED, which helps explain the improved generalization behaviour observed in our experiments. We do not attempt to derive an explicit closed-form bound in terms of $\mathcal{L}_{\text{dec}}$ itself, but this analysis clarifies that semantics–structure decoupling plays a regularizing role, rather than merely adding more parameters.

D Algorithm Implementation

Pseudocode for HeterSEED. The overall procedure of HeterSEED is summarized in **Algorithm 1**. We first apply the *Label-Guided Masking* mechanism, where node features are augmented with (possibly masked) label embeddings to support label propagation under limited supervision; if masking is disabled, the original features are used. Then the *Heterogeneous Semantic Aggregation Channel* performs stacked HGNN-based message passing over 1-hop heterogeneous neighborhoods to obtain semantic representations for all nodes. To explicitly handle heterophily, the *Structure-Aware Weight Computation* module computes metapath-based structural counts and normalizes them into edge weights, after which the *Structure-Aware Heterophily Channel* separates each node's neighbors into homophilic and heterophilic sets and aggregates them through the homophily–heterophily fusion layer to produce structural embeddings. Finally, the *node-level adaptive fusion* module combines semantic and structural embeddings via a learned gate to obtain the final node representations, which are optimized using the classification loss and the decoupling regularization for downstream tasks such as node classification.

E Computational Complexity Analysis

In this section, we analyze the computational complexity of HeterSEED in both full-batch and mini-batch training regimes, highlighting how the additional structure-aware channel affects scalability compared with standard HGNNs.

Algorithm 1 Implementation for HeterSEED

Input: Heterogeneous graph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, meta-paths $\mathcal{P}$, node features $\mathbf{X}$, training labels $\mathbf{Y}_{\text{train}}$
Parameters: Number of layers L, decoupling coefficient α, label masking rate β
Output: Prediction probabilities $\{\hat{\mathbf{y}}_v\}_{v \in \mathcal{V}_t}$

while not converged **do**
  Step 1: Label-Guided Masking
  for each node $i \in \mathcal{V}$ **do**
    Sample $m_i \sim \text{Bernoulli}(1-\beta)$
    $\tilde{\mathbf{z}}_i^{(\text{label})} \leftarrow \begin{cases} \mathbf{Z}_{\text{label}}[\ell_i], & m_i = 1, \\ \mathbf{Z}_{\text{label}}[C+1], & m_i = 0 \end{cases}$
    $\tilde{\mathbf{x}}_i \leftarrow \mathbf{x}_i + g(\tilde{\mathbf{z}}_i^{(\text{label})})$
  end for
  Step 2: Heterogeneous Semantic Encoding
  Initialize $\mathbf{H}^{(0)} \leftarrow \tilde{\mathbf{X}}$
  for $l = 0, \ldots, L-1$ **do**
    $\mathbf{h}_v^{(l+1)} \leftarrow \sigma\Big(\text{NeighborAggregation}(\{\mathbf{W}\mathbf{h}_u^{(l)} \mid u \in \mathcal{N}(v)\})\Big)$
  end for
  Obtain semantic representations $\mathbf{h}_v^{(s)}$
  Step 3: Structure-Aware Weight Computation
  for each meta-path $p \in \mathcal{P}$ **do**
    Compute connection counts $C_p(u, v)$
  end for
  Normalize structure-aware weights $\tilde{\omega}_{u,v}$ (see Eq. 7)
  Step 4: Heterophilic Structural Encoding
  for each target node $v \in \mathcal{V}_t$ **do**
    Identify homophilic neighbors $\mathcal{N}_{\text{homo}}^v$ and heterophilic neighbors $\mathcal{N}_{\text{hetero}}^v$
    $\mathbf{h}_{\text{homo}}^v \leftarrow \sum_{u \in \mathcal{N}_{\text{homo}}^v} \tilde{\omega}_{u,v} \mathbf{x}_u$
    $\mathbf{h}_{\text{hetero}}^v \leftarrow \sum_{u \in \mathcal{N}_{\text{hetero}}^v} \tilde{\omega}_{u,v} \mathbf{x}_u$
    $\mathbf{h}_v^{(r)} \leftarrow \textsc{HomoHeteroGate}(\mathbf{h}_{\text{homo}}^v, \mathbf{h}_{\text{hetero}}^v)$ (using two-layer MLPs)
  end for
  Step 5: Adaptive Fusion
  for each $v \in \mathcal{V}_t$ **do**
    $\mathbf{h}_v \leftarrow (1-\gamma_v)\mathbf{h}_v^{(s)} + \gamma_v \mathbf{h}_v^{(r)}$
  end for
  Step 6: Loss Computation
  $\mathcal{L}_{\text{dec}} \leftarrow \text{DecoupleLoss}(\mathbf{H}^{(s)}, \mathbf{H}^{(r)})$ (see Eq. 10)
  $\mathcal{L}_{\text{cls}} \leftarrow \textsc{CrossEntropy}(\{\mathbf{h}_v\}, \mathbf{Y}_{\text{train}})$ (see Eq. 11)
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{dec}}$ (Eq. 12)
  Update model parameters by minimizing $\mathcal{L}$
end while
$\hat{\mathbf{y}}_v \leftarrow \textsc{Softmax}(\mathbf{h}_v), \quad \forall v \in \mathcal{V}_t$
return $\{\hat{\mathbf{y}}_v\}_{v \in \mathcal{V}_t}$

Full-batch training. In the full-batch setting, HeterSEED processes the entire heterogeneous graph at each iteration. The total cost is dominated by three components: (i) message passing over graph edges, (ii) node-wise linear transformations, and (iii) meta-path–based structural aggregation.

For L layers with hidden dimension d, message passing costs $\mathcal{O}(L|\mathcal{E}|d)$, where $|\mathcal{E}|$ is the number of edges, and node-wise transformations cost $\mathcal{O}(L|V|d^2)$. Aggregating over M meta-path–induced adjacency structures adds $\mathcal{O}(M\|\mathcal{A}\|d)$, where $\|\mathcal{A}\|$ is the number of non-zero entries in all meta-path adjacency matrices. Thus the per-iteration complexity is

$$\mathbf{T}_{\text{full}} = \mathcal{O}\big(L|\mathcal{E}|d + L|V|d^2 + M\|\mathcal{A}\|d\big).$$

Since all computations are carried out on the full graph, both time and memory scale linearly with graph size, which is practical for medium-scale datasets (e.g., DBLP, IMDB, ACM).

Mini-batch training. For large-scale heterogeneous graphs, we adopt mini-batch training. Given batch size b, L layers, and an average of s sampled neighbors per layer, the receptive field of each target node is $\mathcal{O}(s^L)$. Message passing within a mini-batch therefore costs $\mathcal{O}(bs^L d)$, and node-wise transformations cost $\mathcal{O}(Lbs^{L-1}d^2)$.

Instead of precomputing dense meta-path adjacency matrices, we dynamically construct sparse meta-path connections within each batch using $(\text{row}, \text{col}, w)$ triplets. Let $\|\mathcal{A}_B\|$ be the number of such connections in one batch; the corresponding cost is $\mathcal{O}(M\|\mathcal{A}_B\|d)$. Since an epoch involves roughly $|V|/b$ batches, the per-epoch complexity is

$$\mathbf{T}_{\text{mini}} = \mathcal{O}\big(|V|s^L d + L|V|s^{L-1}d^2 + \tfrac{|V|}{b}M\|\mathcal{A}_B\|d\big).$$

In practice $\|\mathcal{A}_B\| \ll \|\mathcal{A}\|$, so dynamic sparse meta-path construction substantially reduces both computation and memory compared with full-batch meta-path aggregation, making HeterSEED scalable on million-node, hundred-million-edge heterogeneous graphs.

F Further Details for Benchmark Datasets

Table 4: Detailed statistics of the datasets used in our experiments. #Nodes and #Edges denote the total numbers of nodes and edges, respectively; $\mathcal{T}_V$ and $\mathcal{T}_E$ are the numbers of node and edge types; #Class is the number of target classes; F_{target} is the feature dimension of target nodes; h denotes the graph-level homophily ratio.

Dataset	#Nodes	#Edges	$\mathcal{T}_V$	$\mathcal{T}_E$	#Class	F_{target}	h
DBLP	26,128	239,566	4	6	4	334	0.81
IMDB	21,420	86,642	4	6	5	3066	0.16
ACM	10,942	547,872	4	8	3	1902	0.59
MAG	1,939,743	21,111,007	4	4	349	128	0.21
RCDD	13,806,619	157,814,864	7	7	2	256	0.45

We conduct experiments on several widely used heterogeneous graph benchmarks, all of which are publicly available through the HGB[2] platform. Below we briefly describe the characteristics of each dataset; detailed statistics, including graph scale, heterogeneity, and homophily ratios, are summarized in Table 4.

- DBLP[3] is a bibliographic network in the computer science domain, consisting of four node types: authors, papers, terms, and venues. The graph contains six relation types, including paper–term, paper–venue, and paper–author interactions, as well as their reverse directions. The task is node classification on author nodes, where each author is assigned to one of four research areas: database, data mining, artificial intelligence, and information retrieval.
- IMDB[4] is a heterogeneous information network describing the movie industry. It contains four node types (movies, directors, actors, and keywords) and six relation types such as movie–director, movie–actor, and movie–keyword, together with their inverse relations. Each movie node may belong to multiple genres, and the task is to predict its associated categories, including action, comedy, drama, romance, and thriller.
- ACM[5] is a heterogeneous citation network introduced in [32], comprising four node types: authors, papers, terms, and subjects. Relations include paper citation, authorship, topic assignment, and term association, along with their reverse edges. The objective is to classify paper nodes into three research fields: database, wireless communication, and data mining.
- MAG[6] is a large-scale academic heterogeneous network composed of multiple entity types, including papers, fields of study, and authors. It encodes rich semantic relations such as paper–field associations and paper–author collaborations, together with corresponding reverse edges. In our experiments, we focus on paper classification, where each paper node is assigned to a research domain.

[2] https://www.biendata.xyz/hgb/

[3] http://web.cs.ucla.edu/~yzsun/data/

[4] https://www.kaggle.com/karrrimba/

[5] http://dl.acm.org/

[6] https://ogb.stanford.edu/docs/nodeprop/

- RCDD[7] is a real-world heterogeneous graph constructed for risk commodity detection on Alibaba's e-commerce platform. The dataset contains multiple node types (e.g., items, type-f objects, type-b objects, and other auxiliary entities) that capture complex interactions in online transactions, as well as diverse item–object relations and their reverse edges. The task is to classify item nodes into risky and non-risky categories. It should be note that RCDD is particularly challenging. It is large-scale and exhibits severe class imbalance, where negative ("black") and positive ("white") samples are distributed at an approximate ratio of 10:1. Moreover, the graph structure is noisy: malicious users may deliberately construct seemingly benign relations by spoofing devices, addresses, or other identifiers. Consequently, many connected nodes share low or even zero attribute similarity, leading to pronounced attribute heterophily. These properties make RCDD a demanding benchmark for heterogeneous graph representation learning under heterophily.

In practice, we use short symmetric meta-paths anchored at the target node type, following standard heterogeneous graph benchmark conventions and a simple schema-driven rule: we choose short, semantically meaningful symmetric paths that capture the primary typed interactions around the target nodes while keeping the structural channel efficient and reproducible. For the large-scale MAG and RCDD datasets in particular, we intentionally use a small set of short symmetric meta-paths to balance structural expressiveness and scalability.

The specific meta-paths selected for each dataset are summarized in Table 5.

Table 5: Summary of selected symmetric meta-paths for each dataset.

Dataset	Target Node	Selected Symmetric Meta-paths
DBLP	Author	A-P-A
IMDB	Movie	M-A-M, M-D-M, M-K-M
ACM	Paper	P-A-P, P-S-P, P-T-P
MAG	Paper	P-A-P, P-F-P
RCDD	Item	I-F-I, I-B-I

G Further Details for the Experimental Studies

G.1 Additional Details for Baseline Methods

To evaluate the effectiveness of HeterSEED, we compare it with representative baselines from four categories: vanilla GNN models, heterogeneous GNN models under homophily, homogeneous GNN models tailored for heterophily, and heterophily-aware HGNNs.

Vanilla GNN models. These methods are designed under the homophily assumption and mainly rely on feature-similarity-based message passing. They serve as standard references for performance degradation under heterophily.

- **GCN** [14] performs neighborhood aggregation via shared linear transformations over normalized adjacency, which is effective on homophilous graphs but vulnerable to noisy neighbors under heterophily.
- **GAT** [30] extends GCN with attention-based neighbor weighting, but still implicitly assumes that similar neighbors are informative and thus may struggle on disassortative graphs.

Homogeneous GNN models under heterophily. These models are specifically designed to alleviate the limitations of conventional GNNs on graphs with low homophily.

- **LINKX** [19] abandons message passing and separately transforms node features and adjacency, enabling scalable learning on large non-homophilous graphs.
- **FAGCN** [4] introduces a frequency-adaptive graph convolution with a self-gating mechanism to integrate low- and high-frequency signals across different homophily regimes.

[7] https://zenodo.org/record/8103003

- **ACM-GCN** [21] proposes adaptive channel mixing over aggregation, diversification, and identity channels to handle diverse heterophily patterns.
- **GRAIN** [37] aggregates multi-view information at different granularities and incorporates implicit signals from distant nodes, fusing local and global information for robust node representations.

HGNN models under homophily. To benchmark performance on heterogeneous graphs, we include several representative HGNNs: RGCN [25], RGAT [12], HAN [32], HGT [11], SHGN [22], HINormer [23], and DiffGraph [18].

- **RGCN** [25] extends GCN to multi-relational graphs via relation-specific transformations.
- **RGAT** [12] generalizes GAT with relation-aware attention to capture inter- and intra-type dependencies.
- **HAN** [32] uses hierarchical attention over nodes and metapaths to learn semantic-aware node embeddings.
- **HGT** [11] is a heterogeneous graph transformer with type-specific parameters and heterogeneous attention for large-scale graphs.
- **SHGN** [22] provides a strong, reproducible HGNN baseline via standardized preprocessing and hyperparameters, showing that carefully tuned simple architectures can be highly competitive.
- **HINormer** [23] adopts a transformer-style architecture with large-range aggregation and dedicated encoders to capture both structural and semantic information.
- **DiffGraph** [18] performs latent heterogeneous graph diffusion with cross-view denoising to mitigate noise and model semantic transitions among heterogeneous relations.

Heterophily-aware HGNN model. This category explicitly accounts for both heterogeneity and heterophily. A representative baseline is **HETERO^{2}NET** [16], which detects heterophily via metapaths and employs masked metapath prediction together with masked label prediction to improve robustness on low-homophily heterogeneous graphs.

G.2 Experimental Setup and Hyperparameter Settings

All experiments are implemented in PyTorch and run on a single NVIDIA RTX A6000 GPU with 48GB GPU memory. For each dataset, hyperparameters are tuned based on validation performance. The learning rate is selected from $\{1, 5\} \times \{10^{-4}, 10^{-3}\}$ (i.e., $\{1\text{e}{-4}, 5\text{e}{-4}, 1\text{e}{-3}, 5\text{e}{-3}\}$), the hidden dimension from $\{128, 256\}$, and the dropout ratio from $\{0.3, 0.5, 0.7, 0.9\}$. The number of layers of HeterSEED is fixed to 2 in all experiments. The number of training epochs is chosen from $\{50, 100\}$ according to validation performance, depending on dataset scale. We use the Adam optimizer to optimize all the trainable model parameters, which are randomly initialized by the Xavier uniform distribution [9].

For the large-scale MAG and RCDD datasets, we adopt mini-batch training with batch size 1024. The neighbor sampling size is fixed to $[15, 15]$ for efficiency and is not included in the hyperparameter search. Channel-related hyperparameters are tuned as follows: the decoupling coefficient α is selected from $\{0.2, 0.3, 0.4, 0.5, 0.6\}$, and the masked label rate β from $\{0.6, 0.7, 0.8, 0.9, 1.0\}$. For the MAG dataset, we set $\beta = 1$ to match the configuration of HETERO^{2}NET for a fair comparison. Under this setting, all observed training labels are fully masked during training, meaning that the model does not directly use ground-truth label embeddings on MAG.

The final hyperparameter configurations that yield optimal performance for each dataset are reported in Table 6, enabling accurate reproduction of the experimental setup and fair comparison of performance results.

H Further Experimental Results

H.1 Additional Results for Ablation Study

Table 7 reports the complete ablation results on all five datasets. The full HeterSEED model achieves the highest mean Macro-F1 and Micro-F1 throughout. Removing the structure-aware channel or

Table 6: Hyperparameter settings for the five datasets used in the experiments.

Dataset	Hyperparameter Setting	
DBLP	Learning rate: 1e-3 Hidden Size: 128 Dropout ratio: 0.7 Epochs: 50	Layers: 2 Alpha: 0.2 Beta: 0.7
IMDB	Learning rate: 5e-3 Hidden Size: 128 Dropout ratio: 0.9 Epochs: 50	Layers: 2 Alpha: 0.2 Beta: 0.6
ACM	Learning rate: 5e-3 Hidden Size: 128 Dropout ratio: 0.9 Epochs: 100	Layers: 2 Alpha: 0.3 Beta: 0.7
MAG	Learning rate: 5e-3 Hidden Size: 256 Dropout ratio: 0.3 Epochs: 50 Batch size: 1024	Layers: 2 Alpha: 0.4 Beta: 1.0 Num neighbors: [15, 15]
RCDD	Learning rate: 5e-3 Hidden Size: 256 Dropout ratio: 0.7 Epochs: 100 Batch size: 1024	Layers: 2 Alpha: 0.2 Beta: 0.7 Num neighbors: [15, 15]

Table 7: Ablation study on HeterSEED.

Dataset	**DBLP**		**IMDB**		**ACM**		**MAG**		**RCDD**	
Model \ Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
w/o SHC	$93.88_{\pm 0.41}$	$94.30_{\pm 0.69}$	$65.79_{\pm 0.52}$	$69.95_{\pm 0.85}$	$93.86_{\pm 0.72}$	$93.75_{\pm 0.69}$	$33.10_{\pm 0.33}$	$54.34_{\pm 0.20}$	$92.60_{\pm 0.21}$	$98.51_{\pm 0.05}$
w/o Dec	$94.16_{\pm 0.75}$	$94.59_{\pm 0.68}$	$66.28_{\pm 0.49}$	$70.29_{\pm 0.66}$	$93.96_{\pm 0.90}$	$93.85_{\pm 0.82}$	$33.33_{\pm 0.54}$	$54.30_{\pm 0.23}$	$92.49_{\pm 0.23}$	$98.49_{\pm 0.05}$
w/o Homo	$93.87_{\pm 0.54}$	$94.23_{\pm 0.58}$	$66.10_{\pm 1.17}$	$70.00_{\pm 0.82}$	$93.98_{\pm 1.11}$	$94.09_{\pm 1.12}$	$34.00_{\pm 0.21}$	$55.83_{\pm 0.16}$	$92.65_{\pm 0.13}$	$98.53_{\pm 0.02}$
w/o Hetero	$93.66_{\pm 0.74}$	$94.13_{\pm 0.74}$	$66.46_{\pm 0.74}$	$70.25_{\pm 0.68}$	$94.18_{\pm 0.94}$	$94.09_{\pm 0.96}$	$33.97_{\pm 0.43}$	$55.91_{\pm 0.21}$	$92.50_{\pm 0.19}$	$98.45_{\pm 0.04}$
w/o Mask	$93.99_{\pm 0.61}$	$94.35_{\pm 0.60}$	$66.11_{\pm 0.47}$	$69.46_{\pm 0.69}$	$93.73_{\pm 1.12}$	$93.62_{\pm 1.11}$	$28.31_{\pm 0.47}$	$50.61_{\pm 0.46}$	$92.61_{\pm 0.17}$	$98.32_{\pm 0.08}$
HeterSEED (Ours)	$\mathbf{94.48}_{\pm 0.81}$	$\mathbf{94.87}_{\pm 0.73}$	$\mathbf{66.71}_{\pm 0.73}$	$\mathbf{70.60}_{\pm 0.82}$	$\mathbf{94.34}_{\pm 0.81}$	$\mathbf{94.21}_{\pm 0.77}$	$\mathbf{34.99}_{\pm 0.22}$	$\mathbf{56.25}_{\pm 0.25}$	$\mathbf{93.09}_{\pm 0.15}$	$\mathbf{98.60}_{\pm 0.03}$

either of its two branches reduces the reported mean performance, supporting the usefulness of separate homophilic and heterophilic aggregation. The decoupling loss and label-masking mechanism also improve the reported means, although the relative contribution of each component varies across datasets and metrics. In particular, removing label masking produces the largest drops in both metrics on MAG.

H.2 Disentangling Label Injection and Structural Modeling

To clearly attribute the performance gains of HeterSEED, we conduct a series of controlled experiments to explicitly disentangle the contributions of label embedding, label-aware propagation, the masking mechanism, and our heterophily-aware structural modeling. The detailed comparison results are summarized in Table 8 and Table 9.

First, **HGNN-LE** removes both the structural branch and the masking mechanism, retaining only the label embedding module appended to the base heterogeneous semantic channel. The substantial performance gap between this variant and the full model across the datasets—particularly on highly heterophilic graphs such as IMDB—provides strong evidence that simple label embeddings alone are insufficient to account for the observed improvements.

Second, we benchmark against **propagation-style heterogeneous baselines**, including SeHGNN, HINormer+, and DiffGraph+. Notably, their "+" variants are further equipped with our label-guided masking strategy to ensure a fair comparison. Despite this enhanced supervision interface, these models consistently underperform HeterSEED. This indicates that neither label-aware propagation

Table 8: Disentanglement of label injection vs. structural modeling.

Dataset	**DBLP**		**IMDB**		**ACM**	
Model \ Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
HGNN-LE	$91.19_{\pm 0.78}$	$91.84_{\pm 0.64}$	$47.58_{\pm 4.18}$	$57.80_{\pm 2.65}$	$93.67_{\pm 0.81}$	$93.55_{\pm 0.78}$
SeHGNN	$93.62_{\pm 0.20}$	$94.06_{\pm 0.20}$	$65.67_{\pm 0.49}$	$68.40_{\pm 0.40}$	$93.11_{\pm 0.53}$	$93.20_{\pm 0.51}$
HINormer+	$93.94_{\pm 0.20}$	$94.43_{\pm 0.26}$	$65.52_{\pm 1.96}$	$68.33_{\pm 1.44}$	$94.01_{\pm 0.76}$	$93.97_{\pm 0.93}$
DiffGraph+	$92.54_{\pm 0.93}$	$92.79_{\pm 0.84}$	$59.57_{\pm 1.54}$	$64.57_{\pm 1.14}$	$94.18_{\pm 0.87}$	$94.18_{\pm 0.98}$
HeterSEED-NM	$91.87_{\pm 0.47}$	$92.43_{\pm 0.40}$	$48.99_{\pm 4.29}$	$58.63_{\pm 2.79}$	$93.91_{\pm 1.44}$	$93.72_{\pm 1.40}$
HeterSEED (Full)	$\mathbf{94.48}_{\pm 0.81}$	$\mathbf{94.87}_{\pm 0.73}$	$\mathbf{66.71}_{\pm 0.73}$	$\mathbf{70.60}_{\pm 0.82}$	$\mathbf{94.34}_{\pm 0.81}$	$\mathbf{94.21}_{\pm 0.77}$

Table 9: Disentanglement of label injection vs. structural modeling on large-scale datasets.

Dataset	**MAG**		**RCDD**	
Model \ Metric	Macro-F1	Micro-F1	Macro-F1	Micro-F1
HGNN-LE	$9.34_{\pm 0.32}$	$29.96_{\pm 1.02}$	$92.02_{\pm 0.77}$	$98.15_{\pm 0.20}$
HeterSEED-NM	$9.83_{\pm 0.41}$	$30.85_{\pm 0.97}$	$92.42_{\pm 0.26}$	$98.44_{\pm 0.07}$
HeterSEED (Full)	$\mathbf{34.99}_{\pm 0.22}$	$\mathbf{56.25}_{\pm 0.25}$	$\mathbf{93.09}_{\pm 0.15}$	$\mathbf{98.60}_{\pm 0.03}$

nor architectural variations on conventional message passing can match the effectiveness of our structural decoupling approach.

Third, HeterSEED-NM disables stochastic label masking while retaining label embeddings and the remaining model components. Its effect varies across datasets. On IMDB, Macro-F1 decreases from 66.71 to 48.99, a reduction of 17.72 percentage points. This is consistent with a stronger need to regularize the use of observed label embeddings on this dataset. The comparison suggests that stochastic masking can reduce over-reliance on visible training labels; it does not by itself identify the cause of the performance decrease.

Finally, by comparing the full model with the variants in our core ablation study (Table 7) that remove the pseudo-label-guided structural components (e.g., w/o SHC, w/o Dec, w/o Homo, w/o Hetero, w/o Mask), we observe a substantial performance degradation, especially on heterophilic datasets. This confirms that the heterophily-aware structural channel is the cornerstone for capturing informative neighborhood patterns.

Taken together, these controlled experiments provide a clear and definitive attribution of our model's performance: the gains of HeterSEED do not merely stem from injecting or propagating label information. While label embeddings offer auxiliary feature augmentation, it is the pseudo-label-guided structural modeling that plays the decisive role. Specifically, pseudo-labels serve strictly as structural cues to partition neighbors into homophilic and heterophilic groups, thereby enabling effective heterophily-aware aggregation. Importantly, we clarify that these pseudo-labels are not propagated as explicit supervision signals, but are utilized exclusively to guide the structural disambiguation within the structure channel.

H.3 Performance under Varying Homophily

This section describes the construction of the semi-synthetic DBLP benchmark used in Section 5.5. Concretely, for a target node type (e.g., *author*) and a given meta-path scheme $\mathcal{P}$ (e.g., $A \to P \to A$), we apply an SBM-based label injection mechanism to control the Metapath-based Label Homophily (MLH). We introduce a swap ratio $\rho \in [0, 1]$ and generate two injection modes: (i) *high-homophily injection*, where with probability ρ all nodes in a meta-path–induced clique share the same (random) label, mimicking strong community coherence; and (ii) *low-homophily injection*, where labels within each clique are shuffled to enforce diversity, simulating adversarial heterophily. By varying ρ and the injection mode, we obtain a family of semi-synthetic benchmarks that span five MLH intervals $(0, 0.2], (0.2, 0.4], \ldots, (0.8, 1.0]$.
Figure 5 reports the Micro-F1 scores of HeterSEED and three strong HGNN baselines across these MLH intervals. The performance of all methods generally improves as MLH increases, but HeterSEED consistently achieves the highest accuracy in every interval. In particular, in the low-homophily regime (e.g., MLH $\in (0, 0.2]$), where conventional HGNNs are most vulnerable,

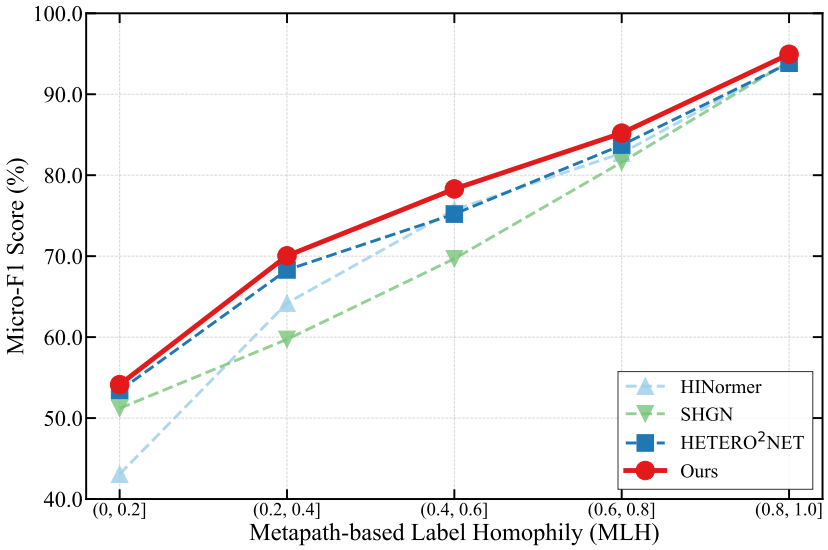


Figure 5: Micro-F1 of HeterSEED and three baselines across five homophily intervals on the semi-synthetic DBLP benchmark. The same results are presented as a radar plot in Figure 4.

HeterSEED still maintains a clear margin over the strongest baseline HETERO^{2}NET (54.13% vs. 53.39%).

This trend is consistent with our theoretical results in Section 4: Theorem 1 shows that the semantics–structure decoupling in HeterSEED yields a strictly richer hypothesis class on heterophilic heterogeneous graphs, while Proposition 1 demonstrates that separating homophilic and heterophilic neighbors reduces the prediction bias induced by heterophilic edges. When MLH is small, these advantages are most pronounced, leading to larger empirical gains; as MLH increases, the bias term diminishes and the performance gap narrows but remains positive. Overall, the semi-synthetic experiments empirically support the theoretical insights that HeterSEED both enhances expressiveness and mitigates heterophily-induced bias.

H.4 Empirical Validation of Semi-Synthetic Benchmarks

To ensure that the semi-synthetic benchmarks used in our robustness analysis behave in a controlled and meaningful way, we perform a sanity check on the DBLP dataset. Figures 6a and 6b report the APA metapath homophily H and the corresponding Macro-F1 / Micro-F1 of HeterSEED under different label perturbation intensities ρ.

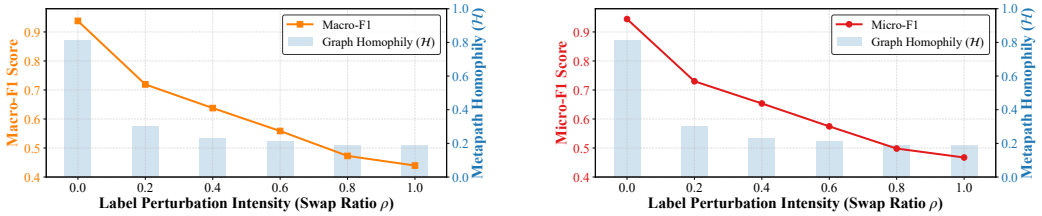
Observation 1 (Controllability of label homophily). As the swap ratio ρ increases from 0.0 to 1.0, the observed APA homophily H monotonically decreases from about 0.80 to a stable floor around 0.20. This confirms that our SBM-based label injection mechanism reliably drives the graph from a homophilous to a heterophilous regime.

Observation 2 (Performance sensitivity). Both Micro-F1 and Macro-F1 degrade smoothly as H decreases, indicating that stronger heterophily indeed makes the prediction task more challenging. Similar trends are observed for baseline HGNNs (omitted for brevity), supporting the view that heterophily is a primary bottleneck for conventional heterogeneous GNNs.

These observations validate that the semi-synthetic MLH-controlled benchmarks used in Section 5.5 faithfully reflect the impact of varying homophily and provide a sound basis for the interval-based robustness evaluation in the main text.

H.5 Average Precision on the RCDD Dataset

Figure 7 reports the Average Precision (AP) of HeterSEED and the evaluated baselines on the large-scale RCDD dataset. **HeterSEED** attains the highest AP and thus ranks first, while HETERO^{2}NET is consistently the second-best method; the remaining models (SHGN, HGT, RGAT, HAN, RGCN, ACM-GCN, FAGCN, GAT, GCN) lag further behind. In particular, the margin between HeterSEED



(a) Macro-F1 Performance

(b) Micro-F1 Performance

Figure 6: Effect of label perturbation intensity ρ on APA metapath homophily H and model performance on DBLP. Panels (a) and (b) show Macro-F1 and Micro-F1 of HeterSEED and the corresponding APA homophily H, respectively.

and the strongest baseline is noticeable, and HeterSEED also exhibits small variance across runs, indicating stable performance. Together with the F1 results in the main text, this AP comparison suggests that the proposed semantics–structure decoupling and heterophily-aware aggregation enable HeterSEED to cope with the severe heterophily and class imbalance of RCDD more effectively than existing HGNNs and heterophily-aware baselines.

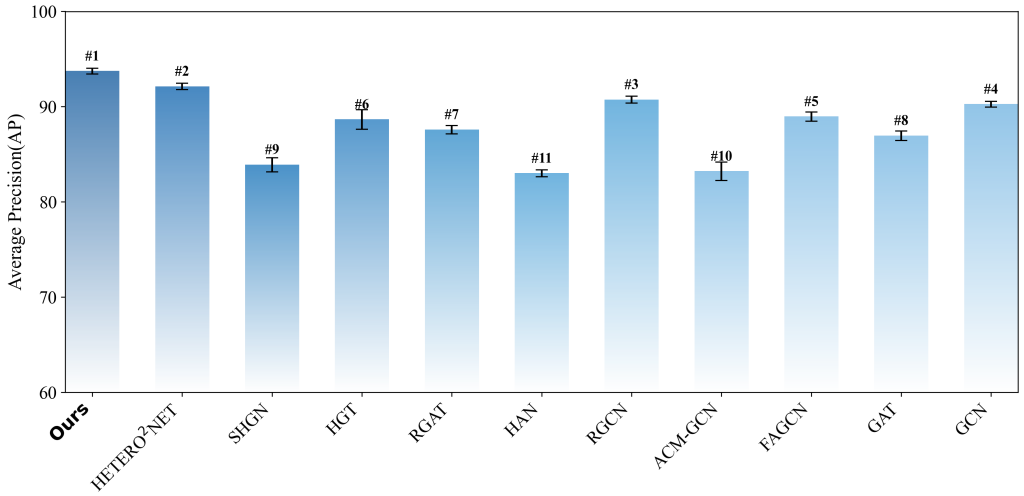


Figure 7: Average Precision (AP) of different models on the RCDD dataset.

H.6 Additional Results of Hyperparameter Sensitivity Analysis

We systematically study hyperparameter sensitivity of HeterSEED on three representative heterogeneous datasets: DBLP, IMDB, and ACM, as shown in Figure 8. In addition, we evaluate large-scale datasets in Figure 9. It is worth noting that on large-scale datasets, the value ranges of Micro-F1 and Macro-F1 differ significantly; therefore, Figure 9 adopts dual y-axes to separately present the two metrics. In contrast, since both metrics fall within similar ranges on the benchmark datasets, Figure 8 uses a single shared y-axis for clarity.

For each dataset, we analyze the effect of two key hyperparameters: the decoupling coefficient α in the objective function $\mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{dec}}$, and the masked-label rate β in the label-guided masking mechanism. Specifically, the left subfigures report the performance variation with respect to α, while the right subfigures show the effect of β.

Across all datasets, HeterSEED exhibits low sensitivity to both α and β: performance curves remain smooth, and only minor fluctuations appear under extreme settings. These results are consistent with the observations in the main text and further indicate that HeterSEED is robust and reliable across heterogeneous graphs of different scales and structural characteristics.

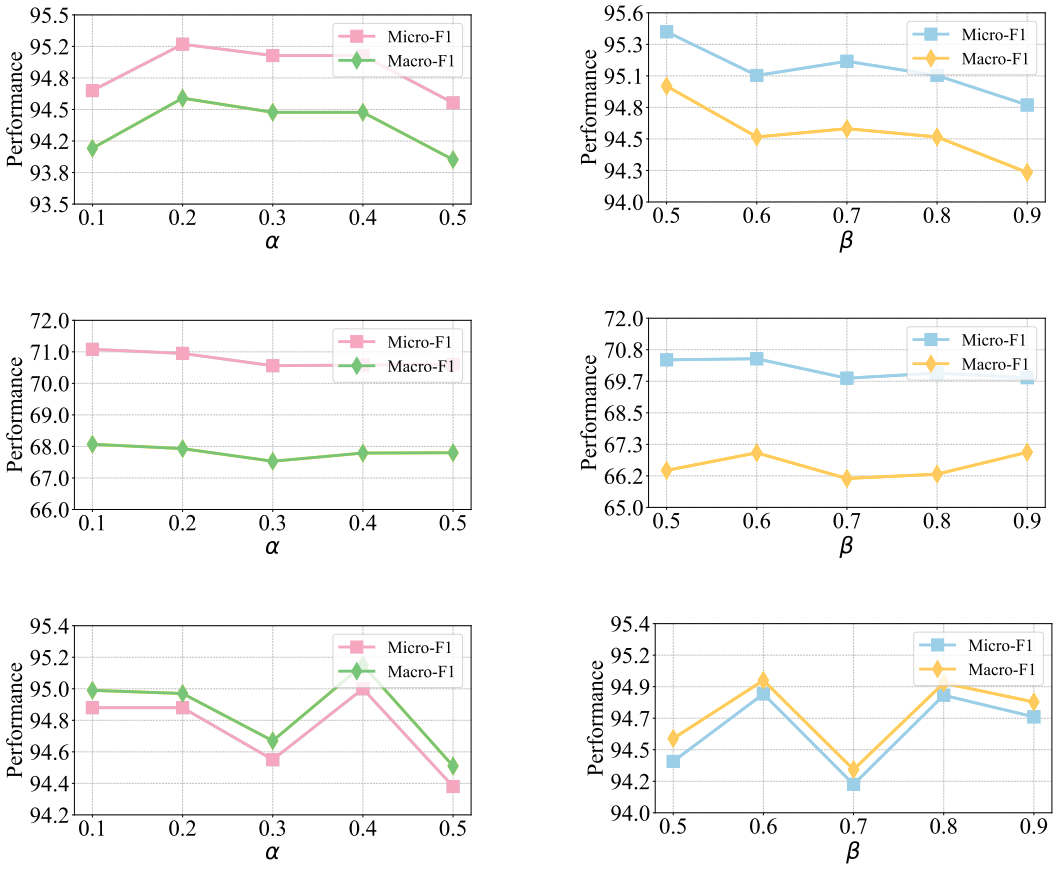


Figure 8: Hyperparameter sensitivity of HeterSEED on DBLP (top), IMDB (middle), and ACM (bottom). For each dataset, the left plot shows the performance versus the decoupling coefficient α, and the right plot shows the effect of the masked label rate β.

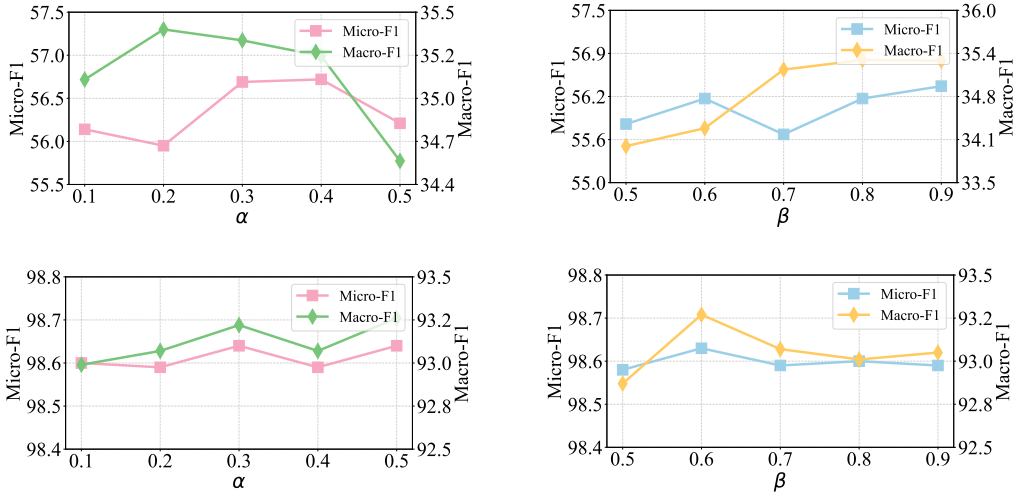


Figure 9: Hyperparameter sensitivity of HeterSEED on the large-scale datasets: MAG (top) and RCDD (bottom). The left plot reports Micro-F1 and Macro-F1 versus the decoupling coefficient α, and the right plot reports the effect of the masked label rate β.

I Limitations and Broader Impacts

Limitations. HeterSEED is currently developed for semi-supervised node classification on heterogeneous graphs under heterophily, and its present formulation is most natural when a small set of informative symmetric metapaths is available. In particular, the structure-aware channel uses pseudo-labels to separate homophilic and heterophilic neighborhoods. Although the iterative pseudo-label updating strategy makes the framework reasonably robust in practice, its reliability can further benefit from confidence-aware pseudo-label refinement in extremely low-label or highly noisy settings. In addition, the current implementation assumes manually specified symmetric metapaths, which are standard and effective on the benchmarks considered here, but automatic metapath discovery or selection may further broaden applicability to domains with richer or less well-understood schemas. These considerations primarily delineate the current scope of the method and suggest natural directions for future work, including confidence-aware pseudo-label refinement, adaptive metapath learning, and extensions of semantics–structure decoupling to heterogeneous hypergraphs under heterophily.

Broader Impacts. The proposed HeterSEED framework advances heterogeneous graph representation learning by improving robustness to heterophily through explicit semantics–structure decoupling. This can benefit a range of applications, such as recommendation, social network analysis, and knowledge graph mining, where heterogeneous relations often involve both homophilic and heterophilic patterns. Beyond predictive performance, the explicit separation of semantic and structural channels, together with node-level adaptive fusion, may also provide a more transparent view of whether a prediction is supported primarily by local semantics or by higher-order structural evidence. As with other general-purpose graph learning methods, deployment in real systems should remain attentive to standard considerations such as data quality, sampling bias, privacy, fairness, and appropriate human oversight in high-impact settings. Overall, we view HeterSEED as a methodological contribution intended to broaden the modeling toolkit for heterogeneous graphs rather than a system tailored to any specific consequential decision-making pipeline.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See the abstract and Section 1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See AppendixI.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: See Appendix C.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide the information needed to reproduce the main experimental results: the model specification and training objective in Section 3, pseudocode of HeterSEED in Appendix D (Algorithm 1), dataset descriptions and public sources in Appendix F, evaluation metrics and standard train/validation/test splits in Section 5.2, baseline descriptions in Appendix G.1, and the hyperparameter search space together with the final dataset-specific settings in Appendix G.2 and Table 6.

Guidelines:

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

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

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

   Answer: [No]

   Justification: All the datasets are publicly available, and the supplemental material includes pseudocode for HeterSEED as well as detailed experimental and hyperparameter settings to facilitate reproduction.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: See Appendix G.2.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: See the experimental results in Table 1, Table 2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Appendix E.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: All the experiments of this work conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: See Appendix I.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have properly cited existing assets in our paper.

    Guidelines:

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

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

13. **New assets**

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

    Answer: [N/A]

    Justification: This work does not release new assets and only uses publicly available datasets such as ACM and DBLP.

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

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

    Answer: [N/A]

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

    Guidelines:

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

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Guidelines:

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