GRAIN: Multi-Granular and Implicit Information Aggregation Graph Neural Network for Heterophilous Graphs

Songwei Zhao[1,2], **Yuan Jiang**[3*], **Zijing Zhang**[1,2], **Yang Yu**[1,2], **Hechang Chen**[1,2*]

[1]School of Artificial Intelligence, Jilin University, Changchun
[2]Engineering Research Center of Knowledge-Driven Human-Machine Intelligence, Jilin University, Changchun
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore
{zhaosw22, zijing23, yyu23}@mails.jlu.edu.cn, yuan005@ntu.edu.sg, chenhc@jlu.edu.cn

Abstract

Graph neural networks (GNNs) have shown significant success in learning graph representations. However, recent studies reveal that GNNs often fail to outperform simple MLPs on heterophilous graph tasks, where connected nodes may differ in features or labels, challenging the homophily assumption. Existing methods addressing this issue often overlook the importance of information granularity and rarely consider implicit relationships between distant nodes. To overcome these limitations, we propose the Granular and Implicit Graph Network (GRAIN), a novel GNN model specifically designed for heterophilous graphs. GRAIN enhances node embeddings by aggregating multi-view information at various granularity levels and incorporating implicit data from distant, non-neighboring nodes. This approach effectively integrates local and global information, resulting in smoother, more accurate node representations. We also introduce an adaptive graph information aggregator that efficiently combines multi-granularity and implicit data, significantly improving node representation quality, as shown by experiments on 13 datasets covering varying homophily and heterophily. GRAIN consistently outperforms 12 state-of-the-art models, excelling on both homophilous and heterophilous graphs.

Introduction

Graph Neural Networks (GNNs) (Welling and Kipf 2016) are a specialized class of deep neural networks designed to process and analyze graph-structured data. GNNs capitalize on the inherent properties of graphs, where entities are represented as nodes and their relationships as edges, to effectively capture complex interdependencies between entities. By employing iterative message-passing and aggregation mechanisms, GNNs iteratively update each node's representation by combining its features with those of its neighbors. This process enables GNNs to learn sophisticated and informative embeddings that are highly effective for a variety of graph-based machine learning tasks, such as node classification (He et al. 2024), link prediction (Lu et al. 2023), and graph classification (Zhao et al. 2024), often surpassing the performance of traditional neural networks. GNNs have also demonstrated remarkable success across a broad spectrum of real-world applications, including social network analysis (Zhang et al. 2022), recommendation systems (Agrawal et al. 2024), and drug discovery (Liu et al. 2023). However, the primary reason GNNs excel in many tasks—their reliance on the homophily assumption—also presents a significant limitation. The homophily assumption presupposes that connected nodes tend to share similar attributes or labels, providing additional context for information aggregation. While this assumption works well in homophilous graphs, it fails to capture the complexities of heterophilous graphs, where connected nodes may have dissimilar features or labels.

As a result, increasing research shows that GNNs do not always outperform traditional deep neural networks in graph tasks, particularly in heterophilous settings. In some cases, even simple multi-layer perceptrons (MLPs) can surpass GNNs (Liu, Wang, and Ji 2021; Chien et al. 2020). This discrepancy is largely due to the heterophily problem, where the homophily assumption breaks down, and GNNs struggle to effectively aggregate information from connected nodes with differing attributes. Recently, scholars have increasingly focused on heterophily and proposed models to address this issue. For instance, Chanpuriya and Musco (2022) proposes the Adaptive Simple Graph Convolution, which selects different filters for each feature, demonstrating its adaptability to both homophilous and heterophilous graph structures in experiments. GBK-GNN (Du et al. 2022) captures both homophilous and heterophilous information through bi-kernel feature transformation and introduces a selection gate to choose the appropriate core for a given node. Xiao et al. (2023) addresses the spatial heterophily in urban graphs by designing a rotation-scaling spatial aggregation module and a heterophily-sensitive spatial interaction module. LRGNN (Liang et al. 2024) tackles label relation prediction in heterophilous graphs by solving an approximation problem of the global label relation matrix for signed graphs, making the proposed model applicable in both homophilous and heterophilous settings.

Despite achieving good results in addressing heterophily, existing works generally underperform on homophily (Chien et al. 2020). Moreover, they often overlook the importance of aggregation at various granularity levels in node representations, where coarse-grained information captures the overall position or influence of nodes (e.g., users) within

*Corresponding author.

the network, while fine-grained information focuses on their direct interactions with closely connected nodes. Additionally, these methods rarely consider implicit relationships between distant nodes that are not neighbors but may share common interests or characteristics. Such implicit information is crucial for generating smooth embeddings by capturing underlying connections beyond neighbors.

Incorporating the aforementioned multi-view information into the aggregation process presents significant challenges. First, real-world graph data is inherently complex and diverse, with varying attributes and types, making it difficult to accurately determine the appropriate range of coarse and fine-grained information for each node. Second, even if the correct granularity levels can be identified, the task of effectively aggregating and training nodes within these ranges remains daunting. Different nodes necessitate tailored granularity ranges, which implies the need for varying network depths during training, thereby substantially increasing model complexity. Lastly, integrating implicit relationships between distant nodes into the aggregation process is a crucial challenge that remains unresolved.

To tackle the challenges of representing heterophilous graphs, we introduce the Granular and Implicit Graph Network (GRAIN), which achieves smoother and more comprehensive representations through multi-granularity information aggregation. First, we model graph representation learning as a Markov Decision Process (MDP), enabling systematic optimization of granularity selection based on long-term rewards, and efficiently incorporate Reinforcement Learning (RL) methods to solve this problem. Our RL framework is specifically designed to output continuous values, enabling flexible adjustment of granularity levels for different nodes. This customization allows for tailored information aggregation that seamlessly integrates both local and global perspectives, thereby enhancing the depth and richness of the node embeddings. Additionally, we introduce a novel node information aggregator that synergizes multi-granularity with implicit relationships between distant nodes, resulting in more refined and accurate node representations.

To validate the effectiveness of the proposed model, we conducted extensive experiments on 13 datasets with varying homophily rates, comparing GRAIN with 12 SOTA models. The results show that GRAIN not only performs excellently on homophilous datasets but also demonstrates superior performance on heterophilous datasets. The key contributions of this paper are as follows:

- We propose a novel GNN model that integrates multi-granularity information and implicit relationships, achieving smoother and more accurate node embeddings across a variety of graph structures.
- We devise two core components: Intelligent Granularity Perceiver, which systematically explores various granularities and implicit information while optimizing long-term rewards, and Multi-view Aggregator, which integrates this information into node representations for a cohesive synthesis of diverse data perspectives.
- Extensive experiments on 13 benchmark datasets show that GRAIN excels in both homophilous and heterophilous settings, significantly outperforming 12 state-of-the-art methods and demonstrating strong generalization and broad applicability.

Preliminaries

Formal problem definition. Suppose that $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ represents an undirected graph, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the set of nodes with $|\mathcal{V}| = n$, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of undirected edges. Each edge $e_{ij} \in \mathcal{E}$ indicates a connection between node v_i and node v_j. We use $\mathbf{A}_{ij}$ to denote the adjacency matrix of graph $\mathcal{G}$, where $\mathbf{A}_{ij} = 1$ if there is an edge between v_i and v_j, and 0 otherwise. We denote $\mathcal{N}_k(v)$ as the k-th order neighbors of node v. Assume each node has d-dimensional features, then the feature vectors of all nodes are represented as $\mathbf{X} \in \mathbb{R}^{\mathbf{n}\times\mathbf{d}}$, and their labels $\mathbf{Y}$ are composed of one-hot vectors. The objective of graph representation learning is to map the input graph features into a low-dimensional vector space. According to the message-passing mechanism of GNNs, our goal is to learn node representations by considering the coarse and fine-grained information within the k-th order neighbors $\mathcal{N}_k(v)$ and the implicit information between nodes. The learned embeddings are thus more likely to be smooth and preserved in the low-dimensional vector space, making them ready for downstream tasks.

Based on the above definitions of the notations, we define the aggregation optimization problem in graph representation learning as follows. Given a graph network, we model the aggregation process of this network as a MDP. We aim to optimize this process to learn an information granularity selection policy π. This policy enables the network to dynamically learn and adjust the information granularity while also capturing the implicit information between nodes, thereby improving the overall performance of GNNs.

Graph Neural Network. Many popular GNN variants have been proposed to achieve smooth node representations effectively. One of the most prominent and widely used variants is the Graph Convolutional Network (GCN) (Kipf and Welling 2016). GCNs aggregate information from a node's neighbors to update its representation. At each layer, a node's features are combined with those of its neighbors, using normalized connections to account for different node degrees. This aggregated information is then transformed through learnable weights and passed through an activation function like ReLU. The process iteratively refines the node's embedding, capturing both local features and structural information from the graph. This allows GCNs to effectively learn meaningful representations in graph-structured data.

Reinforcement Learning for MDP. Reinforcement learning algorithms, such as Q-learning and policy gradient methods, commonly solve MDP problems by learning the optimal policy through environmental interactions. This study aims to consider coarse-grained information and implicit information between nodes, such as long-term dependencies and potential connections. Therefore, we utilize an Actor-Critic type RL algorithm to learn and adjust attention to different granularities of information. Additionally, we use the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm (Fujimoto, Hoof, and Meger 2018), designed for

continuous action spaces, to learn the implicit information between nodes, thereby enhancing the expressive capacity of embeddings, detailed in the Appendix A.2. In order to avoid overestimating the true value, the method takes the minimum between the two target networks, which gives the target update of the algorithm:

$$target = reward + \gamma \cdot \min_{i \in \{1,2\}} Q_{\theta_i'}(s', a'), \quad (1)$$

where γ denotes the discount factor, θ' denotes the parameters of target the critic network, Q refers to the Q-value estimated by the target critic network, s' and a' are the next state and action, respectively.

While we employ TD3 to solve the MDP in graph representation learning in this study, we also compare value-based and policy-based deep RL methods in Appendix C.3. The experimental results highlight the advantages of the Actor-Critic framework in addressing our problem, particularly in its ability to explore implicit information between graph nodes. These findings further validate the flexibility of our proposed framework.

Methodology

Figure 1 illustrates the overall framework of our proposed GRAIN. Our framework consists of two main components: the Intelligent Granularity Perceiver Module and the Multi-view Aggregator Module. The perceiver automatically learns and adjusts attention to different granularities of information from a global perspective. The aggregator utilizes the learned policies to integrate different granularities of information, enhancing the embedding of nodes. The algorithm's pseudo-code is presented in the Appendix B.

Intelligent Granularity Perceiver

We discuss how to formulate the process of learning the optimal information granularity selection policy as a MDP. The critical components of an MDP include states, actions, rewards, and the transfer probability from the current state to the next state. Therefore, we need to define these components in the context of the graph network: 1) *State* ($\mathcal{S}$): In the context of our graph network, we define the state $s_t \in \mathcal{S}$ as the current representation of the node at the t-th iteration. 2) *Action* ($\mathcal{A}$): The action $a_t \in \mathcal{A}$ of the t-th iteration is defined as the number of hops for the current node to aggregate information. This action determines the scope of the neighborhood information that will be considered when updating the node's representation. 3) *Reward function* ($\mathcal{R}$): The reward r_t at the t-th iteration is defined as the average performance of the predictive task over the past few steps.

Based on the above definitions, we model the input graph network as an interactive environment. The process of information granularity selection involves the following three steps: 1) *Select the Initial Node*: From the input graph data, select a starting node and use its features as the current state s_t. 2) *Generate Action*: Utilize the RL policy to generate an action a_t to guide the node in aggregating information from its neighbors, which determines the number of hops for the current node's aggregation. 3) *Sample*: Sample a neighbor within the a_t hops as the next time step's node, and use its input features as the next state s_{t+1} in the MDP. Through the described process, we can effectively represent the node embedding learning process in the graph as a MDP.

It is challenging to automatically learn and adjust the attention to different information granularities for each node as graph data becomes more complex. Implicit relationships between nodes are crucial for node representation quality, but integrating this information during aggregation remains a significant challenge. RL algorithms (Mnih et al. 2013, 2015), with their decision-making processes based on environmental feedback, are well-suited for complex and dynamic environments where traditional optimization algorithms struggle to achieve optimal solutions. Therefore, we propose leveraging RL to optimize this process. To account for the implicit information between nodes and facilitate the model's better understanding of the relationships and semantic similarities among nodes, we employ an Actor-Critic RL algorithm to adapt attention to different levels of granularity in the information. Additionally, we consider using the TD3 algorithm to capture implicit information between nodes, thereby enhancing node embedding.

The pivotal factor guiding the TD3 algorithm to learn the proposed MDP in the graph is the reward function, which we define as follows:

$$\mathcal{R}(s_t, a_t) = \frac{\varphi \cdot \sum_{l=t-\vartheta}^{t} \left[\mathcal{F}(s_t, a_t) - \mathcal{F}(s_l, a_l)\right]}{\vartheta + 1}, \quad (2)$$

where φ is a hyperparameter that plays a role in determining the strength of the reward signal, ϑ specifies the number of historical steps considered in the optimization process, and $\mathcal{F}$ denotes the evaluation metric for the graph node classification task. The purpose of calculating the reward function over a certain range of historical windows is to encourage the agent to jointly consider the performance of the most recent ϑ time steps, which can lead to better policy exploration. During training, our framework learns using nodes from the train set and utilizes the node classification accuracy on the validation set as the evaluation metric.

Based on reward function (2), the target value is computed using Equation (1), and the mean square error loss is calculated against the current Q value. Following this, we update the actor network to ensure more stable training.

Multi-View Aggregator

In this subsection, we discuss how information granularity selection strategies can be used to enhance node representation learning through a novel aggregation function. The proposed model explicitly incorporates this strategy by mapping node embeddings to continuous hop counts, as shown in the aggregator part of Figure 1. This aggregation function enables the target node to consider different granularities and implicit information, forming a specialized GNN architecture. Unlike previous aggregators, our approach enables capturing more relevant information, offering excellent compatibility and flexibility, thereby enhancing graph representation learning.

To validate the effectiveness of the proposed framework, we integrate the information granularity selection strategy

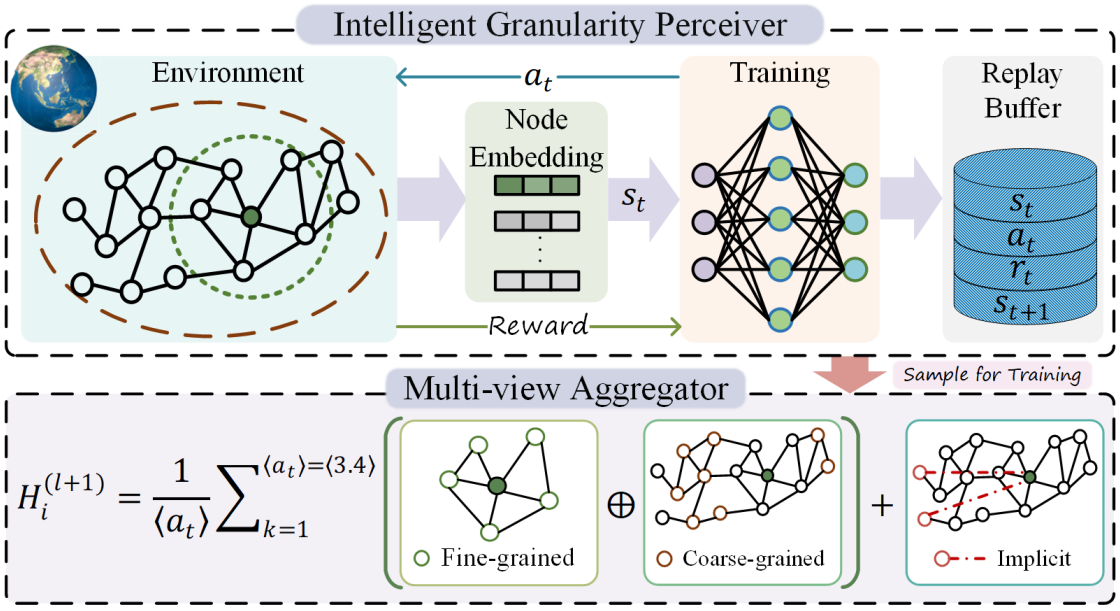

Figure 1: An illustration of our proposed framework. The key idea of the model is to explore different levels of granularity information and implicit information of the target nodes in the graph through the intelligent granularity perceiver. We aim to smooth the representation by aggregating the different information into embedding the nodes through the multi-view aggregator.

and novel aggregation function into a basic GCN, deliberately choosing this simpler architecture to isolate and clearly demonstrate the impact of our approach. By replacing the original fixed receptive field and aggregation function, our framework adaptively customizes the number of aggregation hops for each node at time step t, enabling nodes to gather information at varying granularities from a global perspective and capture implicit information from latent connections and semantic similarities. This adaptive approach enhances each node's ability to learn relevant features in its unique context. The proposed aggregation function is constructed as follows:

$$h_v^1 = \sigma\left(\sum\nolimits_{u_1 \in \mathcal{N}_1(v)} \hat{a}_{u_1 v} \mathbf{X}_u\right),$$

$$\vdots$$

$$h_v^{k=\langle a_t \rangle} = \sigma\left(\sum\nolimits_{u_k \in \mathcal{N}_k(v)} \hat{a}_{u_k u_{k-1}} h_v^{k-1}\right),$$

$$h_v = \frac{1}{\langle a_t \rangle} h_v^{\langle a_t \rangle} + \left(a_t - \lfloor a_t \rfloor\right) \hat{a}_{u_k u_{k-1}} h_v^{k-1} + \left(\lceil a_t \rceil - a_t\right) \hat{a}_{u_{k+1} u_k} h_v^k,$$

$$output = \log\_\mathrm{softmax}\left(\sigma\left(\mathrm{Dropout}\left(h_v\right)\right)\right)$$

where h_v^k is the feature vector of node v at the k-th layer, $\mathbf{X}_u$ denotes the direct neighbor representation vector of node u. The aggregation hop count $k = \langle a_t \rangle$ is determined by the policy function π at time step t. Since a_t is continuous, we round to the nearest integer. The $\hat{a}_{u_1 v}$ denotes the normalized adjacency matrix for the 1-hop neighbors of node v. In the formula for obtaining h_v, the first term represents the integrated information in coarse and fine granularity, while the latter terms capture the implicit information of potential connections and semantic similarity with the target node. The functions $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ represent the floor and ceiling functions, respectively. This formulation allows each node to dynamically adjust the aggregated granularity information, and enhances the ability to learn implicit information.

Efficient GNN Aggregation

In this section, we introduce key techniques to enhance the model's training efficiency. Specifically, we employ an adaptive information granularity selection strategy for each node, which, while effective, results in a time-intensive construction of GNN aggregation at each time step. This leads to significant training overhead, posing a challenge in implementing the proposed model. Given that the number of hidden units in each node's aggregation layer is N, the parameter count required for training at each step can be substantial, calculated as $\frac{N \cdot (2 + \langle a_t \rangle)(\langle a_t \rangle + 1)}{2}$. This large parameter count considerably impacts training efficiency.

To address this issue, we introduce a parameter sharing mechanism to reduce parameters. During aggregation, each node's layer shares a set of parameters, enabling reuse. The aggregation function is expressed in Equation (3), reducing the training parameters to $N \cdot \langle a_t \rangle$:

$$\mathbf{H}_i^{(l+1)} = \frac{1}{\langle a_t \rangle} \sum\nolimits_{k=1}^{\langle a_t \rangle} \left[(1-\alpha)\hat{\mathbf{A}}_i^k \mathbf{H}^{(l)} + \alpha \mathbf{H}_i\right] + (a_t - \lfloor a_t \rfloor)\hat{\mathbf{A}}_i^{\langle a_t \rangle} \mathbf{H}^{(l)} + (\lceil a_t \rceil - a_t)\hat{\mathbf{A}}_i^{\langle a_t \rangle + 1} \mathbf{H}^{(l)}, \quad (3)$$

which not only lowers storage requirements but also decreases the model's computational load, thereby enhancing training efficiency. $\hat{\mathbf{A}}$ equals $\mathbf{A} + \mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the

Type	Method	Pubmed	Citeseer	Cora	Cornell	Actor	Wisconsin	Texas	Chameleon	Avg. Drop (↓)
Tradition	MLP	81.02	73.44	76.72	83.02	37.02	83.78	81.82	42.64	11.32
	GCN	83.77	75.51	84.28	62.26	30.69	53.27	51.11	44.42	28.33
	GAT	85.80	77.91	84.92	72.31	31.84	65.01	69.81	44.63	17.01
	GraphSAGE	85.78	76.65	84.74	71.68	32.31	78.63	70.73	41.57	14.89
Homophily	MixHop	86.16	78.18	85.89	76.71	35.39	77.51	73.17	45.71	11.47
	APPNP	85.98	78.79	85.04	75.61	32.13	73.75	71.95	37.86	15.09
	GCNII	86.92	77.64	85.56	76.92	32.66	71.25	74.93	42.23	13.62
Heterophily	GPR-GNN	86.76	76.69	85.57	82.93	35.66	80.17	79.27	51.42	7.66
	GGCN	86.88	76.72	85.25	88.89	37.81	82.64	83.33	55.16	4.37
	ACM-GNN	86.91	79.27	86.72	86.11	37.61	88.25	84.62	56.04	2.85
	FE-GNN	86.68	79.81	85.41	87.78	36.92	88.54	87.69	53.61	2.69
	GNN-SATA	OOM	78.46	86.83	87.81	**39.66**	87.62	86.59	56.12	2.36
Ours	GRAIN	**87.04**	**81.25**	**88.52**	**90.12**	38.89	**89.01**	**87.69**	**56.43**	–

Table 1: The average test classification accuracy (%) across all methods on eight real-world datasets. Drop (↓) denotes how much the performance of baselines drop relative to our method. The best results are in bold. OOM refers to "out-of-memory".

identity matrix. We incorporate the representation of target node i into the formula and use the parameter α to balance the aggregated and node representations. This approach ensures that the newly obtained representation reflects both macro- and micro-level relationships while preserving critical features during aggregation. The Equation (3) corresponds to the three components depicted in the lower parts of Figure 1. The first term in the summation represents the integration of coarse and fine granularities, while the sum of the latter two terms captures the implicit information between nodes. When dealing with a continuous action space, we take the floor of the action to determine the range of coarse-grained information. The implicit information denotes the data between nodes within the ceiling of the action range and the target node.

Beside parameter sharing, we introduce a buffering mechanism to further improve the model's efficiency. The buffering mechanism stores the states, actions, and rewards collected during exploration at each time step, avoiding redundant computations computations during training data collection. Once the buffer reaches the batch size, the data stored in the buffer is used to train the GNN, thus accelerating the inference process of the framework. By combining the buffering mechanism with parameter sharing, we achieve a dual enhancement of the model's efficiency, significantly improving the practicality of the proposed framework.

Experiments

In this section, we conduct a comprehensive evaluation of the GRAIN algorithm by comparing it against 12 baseline methods across 13 diverse datasets. The analysis of the experimental results highlight the effectiveness and strong generalization ability of GRAIN.

Datasets and Setup. The proposed GRAIN is evaluated on both homophilous and heterophilous datasets. For homophily data, we use three citation networks (Sen et al. 2008), i.e., Cora, Citeseer, and Pubmed, two Amazon co-purchase networks (Shchur et al. 2018), specifically Computers and Photo, and one co-authorship network (Coauthor CS) (Shchur et al. 2018). For heterophily data, we use seven network datasets (Pei et al. 2020), including Cornell, Wisconsin, Texas, Film, Actor (Tang et al. 2009), Squirrel, and Chameleon. For the GNN module, we set the batch size to 128 and use *Relu* as the activation function. The complete details of datasets, experimental settings, and full results are provided in Appendix C.

Comparative Algorithms. We compare GRAIN with 12 classical and advanced baseline methods to validate the effectiveness of the proposed approach. These methods include a two-layer MLP, classical GNN methods such as GCN (Welling and Kipf 2016), GAT (Veličković et al. 2017), and GraphSAGE (Hamilton, Ying, and Leskovec 2017); advanced methods designed for homophily: including MixHop (Abu-El-Haija et al. 2019) utilizes aggregation at different hop counts, APPNP (Gasteiger, Bojchevski, and Günnemann 2018) introduces personalized information propagation, and GCNII (Chen et al. 2020) mitigates over-smoothing by using initial residuals and identity mapping. For handling heterophily, we compare with state-of-the-art methods such as GPR-GNN (Chien et al. 2020), which adaptively learns the weights of Generalized PageRank to optimize features, GGCN (Yang et al. 2021), which proposes structural and feature-based edge correction, ACM-GCN (Luan et al. 2021), which adaptively exploits aggregation, diversification, and identity channels to address harmful heterophily, FE-GNN (Sun et al. 2023), which constructs features using Chebyshev polynomials or monomials, and GNN-SATA (Yang et al. 2024), which utilizes soft association to identify correlations between features and structure.

Comparative Results

In this section, we first define the homophily rate to classify two types of networks: homophily and heterophily. Here, we use edge homophily (Zhu et al. 2020) as a metric to measure the proportion of connections between nodes with the same label in the graph, defined as follows:

$$\mathcal{H} = \frac{|\{e_{ij} \mid e_{ij} \in \mathcal{E} \wedge \mathbf{Y}_i = \mathbf{Y}_j\}|}{|\mathcal{E}|}, \tag{4}$$

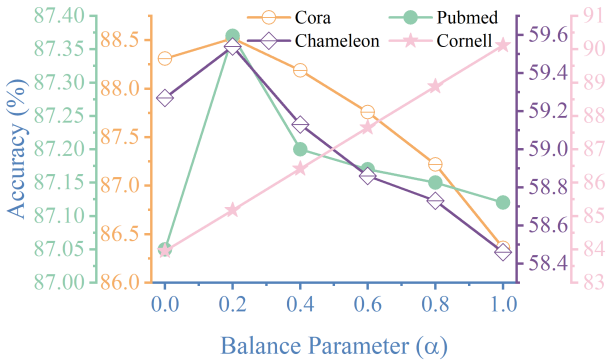

Figure 2: Analysis of balancing parameters (α) for different datasets, where α influences the proportion of aggregation chosen for coarse- and fine-grained information.

The homophily rate ranges from 0 to 1. A large $\mathcal{H}$ value ($\mathcal{H} \to 1$) indicates the graph is homophilous, i.e., the target node and its neighboring nodes are likely to belong to the same class. On the contrary, a small $\mathcal{H}$ value ($\mathcal{H} \to 0$) indicates the graph is low in homophily, or what we refer to as heterophily.

In the experiment, our goal is to validate the effectiveness of the proposed GRAIN method on different types of datasets, particularly heterophilous datasets. Table 1 displays the classification accuracy results on eight datasets, including three homophily and five heterophily, where the bold results represent the best scores in each dataset column. The last column of the table indicates the improvement of the proposed method compared to each row's method. In addition, we present the results of comparative experiments on five more datasets in the Appendix C.2. From the table, we can draw the following conclusions:

1) MLP, which relies solely on node features, performs unexpectedly well on highly heterophilous data, surpassing three typical GNNs and homophily-generalized models. This outcome is likely due to MLP's exclusion of structural information, thereby avoiding the noise introduced by aggregating heterophilous nodes. This suggests that simply employing heterophilous information without careful design does not necessarily result in high-quality representations. Moreover, the MLP's disregard for structural information leads to the poorest performance on homophilous datasets, underscoring the necessity of incorporating structural features via effective node representation learning.

2) Models designed for homophily also perform well on small-scale heterophilous datasets, mainly due to their enhanced node representations, which take into account more comprehensive information such as multi-hop neighbor information and tailoring features. However, their performance decreases on large-scale heterophilous datasets. These GNN methods heavily rely on the homophily assumption, and the aggregation process introduces noise due to the interference of heterophilous features, resulting in poor classification performance.

3) The methods for heterophilous networks, such as GNN-SATA, have shown relatively good results, especially on the Actor dataset. However, these methods may not consistently outperform others in homophilous networks. This could be because these methods are tailored for handling intricate relationships in heterophilous networks, such as conflicting node labels and dissimilar feature distributions.

4) Notably, GRAIN demonstrates outstanding performance across heterophilous and homophilous datasets. Compared to the best baseline, it achieves an overall performance improvement of 2.36%, and perform best across almost all datasets. This is primarily because our method considers different granularities of information and implicit features during the aggregation process, focusing on enhancing node representations. By considering different granularities of information, the model effectively learns structural information and avoid interference of heterophilous features, resulting in better embedding representations. In the Appendix C.3, we further ablate the effects of different information on the model with more types of RL methods.

Sensitivity Analysis

This section analyzes the hyperparameter α in the constructed aggregation function. The role of parameter α is to assess the importance of granularity information during the aggregation process. A smaller α indicates that coarse and fine-grained information has a greater impact on the representation, aiding in smoother node embeddings. The parameter α range is between 0 and 1, and we conduct a comparative analysis using equal intervals.

As illustrated in the line graph of Figure 2, it is evident that variations in α significantly impact the model across both homophily (Cora and Pubmed) and heterophily (Chameleon and Cornell). Whether in heterophilous or homophilous settings, the model performs relatively better when α is set to 0.2. This suggests that granularity information plays a crucial role in achieving smooth node representations, primarily because it allows for the aggregation of more homophilous data within a certain range. However, when α is set too low, the critical features of the node itself may not be effectively utilized, leading to a decline in performance. Additionally, incorporating excessive coarse-grained information in the smaller Cornell network may introduce aggregation noise from heterogeneous data.

This phenomenon arises primarily due to the disparity in correlation between topology and node attributes, where even nodes with similar features may not be connected. The proposed aggregation function effectively integrates information across different granularities, preventing under-smoothing in node representations and ensuring more consistent and accurate embeddings in the latent space.

Visualization of Learned Representations

To intuitively validate the performance of the proposed model in node classification tasks, we employed the t-SNE method to visualize the node embeddings in a two-dimensional space, as depicted in Figure 3. In these visualizations, nodes with the same color belong to the same category. The results clearly illustrate that our proposed GRAIN model significantly outperforms the raw data and other methods in learning more distinct and well-separated

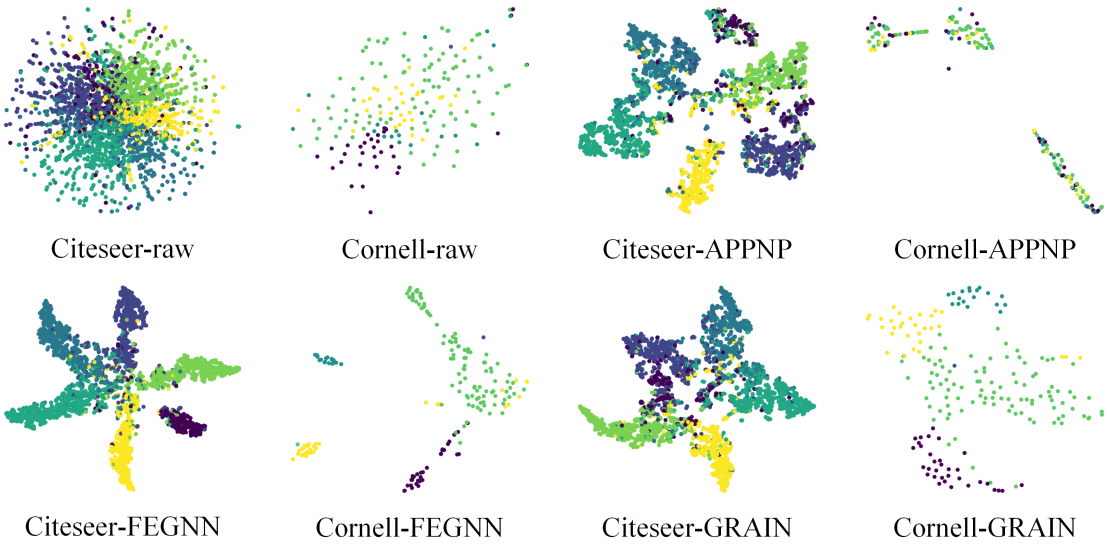

Figure 3: The visualization of classification results of the proposed model for different datasets.

embeddings for different categories. Specifically, GRAIN excels in capturing the inherent structure of the graph, as evidenced by the formation of clear, consistent clusters that are well-aligned with the underlying categories. This demonstrates GRAIN's superior capability in learning meaningful and expressive graph representations, which translates into enhanced performance in node classification tasks. The sharp separation and distinct clustering of nodes in the two-dimensional space visually affirm that GRAIN is highly effective in distinguishing between categories.

Related Works

Several studies (Wang et al. 2020; Luan et al. 2021; Yang et al. 2024) have focused on addressing heterophily in GNNs. For instance, Abu-El-Haija et al. (2019) recognizes that popular GNNs fail to learn general neighborhood mixing relationships, MixHop addresses this issue by mixing features of neighbors at different distances. Geom-GCN (Pei et al. 2020) maps the graph to a continuous latent space via node embeddings, then defines geometric relationships and constructs structural neighbors for aggregation to handle heterophily. H_2GCN (Zhu et al. 2020) adapts to heterophilous networks by designing ego- and neighbor-embedding separation, higher-order neighbors, and combination of intermediate representations. Yang et al. (2022) iteratively updates representations of topology and attributes by simultaneously capturing semantic information and removing common information, thereby improving performance on heterophilous data. Luan et al. (2022) proposes an Adaptive Channel Mixing (ACM) framework, which adaptively utilizes aggregation, diversification, and identity channels to intelligently extract richer localized information and adapt to the heterogeneity of different nodes. Ordered-GNN (Song et al. 2023) proposes an effective message-passing strategy, using specific blocks of neurons for messages passed within specific hops to address heterophily.

These methods offer effective techniques for tackling heterophily in graph neural networks, advancing research, and providing new insights for managing heterogeneous relationships in complex networks. However, they often overlook the impact of aggregation granularity on node representation and fail to consider the implicit information between nodes in the graph. In contrast, our proposed GRAIN achieves multi-granularity information aggregation, resulting in smoother representations and providing a more generalized solution for heterophilous graphs.

Conclusion

In this paper, we introduce the GRAIN model to address the challenges posed by heterophilous graphs, which can achieve smoother node embedding by aggregating multi-view information. Our approach effectively explores different granularity levels information, ensuring that both fine- and coarse-grained details are incorporated into the node embeddings, resulting in a richer representation. Further, by incorporating the implicit relationships between distant nodes, GRAIN enhances the graph's contextual understanding, resulting in smoother and more accurate node embeddings. We demonstrate the effectiveness of GRAIN in extensive comparative experiments on 13 datasets.

While our approach significantly improves training efficiency, substantial computational challenges remain when scaling to much larger networks. The iterative training required to learn the appropriate information granularity for different nodes can be resource-intensive. Additionally, the next state is randomly selected in the process, which does not account for optimality guarantees. In future work, we aim to develop more scalable optimization techniques to better manage complex graphs. Additionally, we plan to integrate more nuanced information, such as multi-scale and temporal features, via advanced aggregation functions.

Acknowledgments

This work is partially supported in part by the National Natural Science Foundation of China (No. 62476110, No. U2341229); the National Key R&D Program of China (No. 2021ZD0112500); the Key R&D Project of Jilin Province (No. 20240304200SF); and the International Cooperation Project of Jilin Province (No. 20220402009GH).

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Ver Steeg, G.; and Galstyan, A. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, 21–29. PMLR.

Agrawal, N.; Sirohi, A. K.; Kumar, S.; et al. 2024. No Prejudice! Fair Federated Graph Neural Networks for Personalized Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10, 10775–10783.

Chanpuriya, S.; and Musco, C. 2022. Simplified graph convolution with heterophily. *Advances in Neural Information Processing Systems*, 35: 27184–27197.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, 1725–1735. PMLR.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2020. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*.

Du, L.; Shi, X.; Fu, Q.; Ma, X.; Liu, H.; Han, S.; and Zhang, D. 2022. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022*, 1550–1558.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, D.; Liu, S.; Ge, M.; Yu, Z.; Xu, G.; and Feng, Z. 2024. Improving Distinguishability of Class for Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11, 12349–12357.

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

Liang, L.; Hu, X.; Xu, Z.; Song, Z.; and King, I. 2024. Predicting global label relationship matrix for graph neural networks under heterophily. *Advances in Neural Information Processing Systems*, 36.

Liu, M.; Wang, Z.; and Ji, S. 2021. Non-local graph neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 44(12): 10270–10276.

Liu, Y. L.; Wang, Y.; Vu, O.; Moretti, R.; Bodenheimer, B.; Meiler, J.; and Derr, T. 2023. Interpretable chirality-aware graph neural network for quantitative structure activity relationship modeling in drug discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12, 14356–14364.

Lu, Y.; Yang, D.; Wang, P.; Rosso, P.; and Cudre-Mauroux, P. 2023. Schema-Aware Hyper-Relational Knowledge Graph Embeddings for Link Prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2021. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*.

Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2022. Revisiting heterophily for graph neural networks. *Advances in neural information processing systems*, 35: 1362–1375.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Song, Y.; Zhou, C.; Wang, X.; and Lin, Z. 2023. Ordered gnn: Ordering message passing to deal with heterophily and over-smoothing. *arXiv preprint arXiv:2302.01524*.

Sun, J.; Zhang, L.; Chen, G.; Xu, P.; Zhang, K.; and Yang, Y. 2023. Feature expansion for graph neural networks. In *International Conference on Machine Learning*, 33156–33176. PMLR.

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 807–816.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, X.; Zhu, M.; Bo, D.; Cui, P.; Shi, C.; and Pei, J. 2020. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining*, 1243–1253.

Welling, M.; and Kipf, T. N. 2016. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

Xiao, C.; Zhou, J.; Huang, J.; Xu, T.; and Xiong, H. 2023. Spatial heterophily aware graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2752–2763.

Yang, L.; Zhou, W.; Peng, W.; Niu, B.; Gu, J.; Wang, C.; Cao, X.; and He, D. 2022. Graph neural networks beyond compromise between attribute and topology. In *Proceedings of the ACM Web Conference 2022*, 1127–1135.

Yang, Y.; Liu, T.; Wang, Y.; Zhou, J.; Gan, Q.; Wei, Z.; Zhang, Z.; Huang, Z.; and Wipf, D. 2021. Graph neural networks inspired by classical iterative algorithms. In *International Conference on Machine Learning*, 11773–11783. PMLR.

Yang, Y.; Sun, Y.; Wang, S.; Guo, J.; Gao, J.; Ju, F.; and Yin, B. 2024. Graph Neural Networks with Soft Association between Topology and Attribute. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 9260–9268.

Zhang, Y.; Gao, S.; Pei, J.; and Huang, H. 2022. Improving social network embedding via new second-order continuous graph neural networks. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2515–2523.

Zhao, Z.; Wang, P.; Wen, H.; Zhang, Y.; Zhou, Z.; and Wang, Y. 2024. A twist for graph classification: Optimizing causal information flow in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15, 17042–17050.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33: 7793–7804.