HGEN: Heterogeneous Graph Ensemble Networks

Jiajun Shen[1] , **Yufei Jin**[1] , **Kaibu Feng**[2] , **Yi He**[2] and **Xingquan Zhu**[1]
[1]Dept. of Electrical Engineering and Computer Science, Florida Atlantic University, USA
[2]Department of Data Science, William & Mary, USA
{jshen2024, yjin2021, xzhu3}@fau.edu; {kfeng03, yihe}@wm.edu

Abstract

This paper presents HGEN that pioneers ensemble learning for heterogeneous graphs. We argue that the heterogeneity in node types, nodal features, and local neighborhood topology poses significant challenges for ensemble learning, particularly in accommodating diverse graph learners. Our HGEN framework ensembles multiple learners through a meta-path and transformation-based optimization pipeline to uplift classification accuracy. Specifically, HGEN uses meta-path combined with random dropping to create *Allele Graph Neural Networks (GNNs)*, whereby the base graph learners are trained and aligned for later ensembling. To ensure effective ensemble learning, HGEN presents two key components: 1) a *residual-attention* mechanism to calibrate allele GNNs of different meta-paths, thereby enforcing node embeddings to focus on more informative graphs to improve base learner accuracy, and 2) a *correlation-regularization* term to enlarge the disparity among embedding matrices generated from different meta-paths, thereby enriching base learner diversity. We analyze the convergence of HGEN and attest its higher regularization magnitude over simple voting. Experiments on five heterogeneous networks validate that HGEN consistently outperforms its state-of-the-art competitors by substantial margin. Codes are available at https://github.com/Chrisshen12/HGEN.

1 Introduction

Ensemble learning that combines multiple base models to enhance predictive performance has achieved remarkable success across diverse domains, from weather forecasting [Molteni *et al.*, 1996] and online trading [Sun *et al.*, 2023] to image classification [Yang *et al.*, 2023a] and recent prompt-based ensembling in Large Language Models [Zhang *et al.*, 2024]. While individual models often struggle with generalization and overfitting, ensemble learning harnesses the diversity of base models to improve robustness, reduce variance, and achieve superior accuracy.

Despite its wide adoption, ensemble learning has been mostly studied in the context of *i.i.d.* data, leaving its application to data with complex interdependencies such as graphs relatively underexplored. Early studies such as graph representation ensembling [Goyal *et al.*, 2020] fused node embeddings from multiple graph learning models through concatenation. Subsequent methods such as stacking-based frameworks [Chen *et al.*, 2022] proposed multi-level classifiers to aggregate representations for tasks like link prediction. More recent efforts such as Graph Ensemble Neural Network (GEN) [Duan *et al.*, 2024] integrated ensemble learning directly within Graph Neural Networks (GNNs), performing ensemble operations throughout the training process rather than solely at the prediction stage. To further improve robustness and mitigate overfitting and adversarial attacks, recent methods such as Graph Ensemble Learning (GEL) [Lin *et al.*, 2022] introduced serialized knowledge passing and multilayer DropNode strategies to promote diversity, while GNN-Ensemble [Wei *et al.*, 2023] employed substructure-based training to defend against adversarial perturbations.

While these methods have advanced ensemble learning for homogeneous graphs, they falter in dealing with **heterogeneous graphs**, which are commonly observed in real-world applications, such as social networks [Fu *et al.*, 2020], citation networks [Tang *et al.*, 2008], urban networks [Houssou *et al.*, 2019], and biomedical networks [Jin *et al.*, 2024]. Indeed, ensemble learning on heterogeneous graphs has three unique challenges as follows. 1) *Graph heterogeneity*, where diverse node and edge types necessitate specialized techniques such as meta-path [Zhang *et al.*, 2019] or attention-based [Xiao *et al.*, 2019] models to extract meaningful relationships and patterns, 2) *Base learner accuracy*, where the non-uniform structure of heterogeneous graphs requires accurate base learners to ensure reliable ensemble performance, and 3) *Base learner diversity*, where ensuring diverse learners is crucial for robust and generalized predictions, given the varied nature of heterogeneous graphs.

To address these challenges, we propose HGEN, a novel ensemble learning framework with solid theoretical foundation for heterogeneous graphs. HGEN is tailored to accommodate various mainstream graph learners, such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and GraphSAGE, offering a flexible and efficient solution for complex networked data. Specifically, HGEN simultaneously enhances learner accuracy and diversity by devising a regularized allele GNN framework, which employs

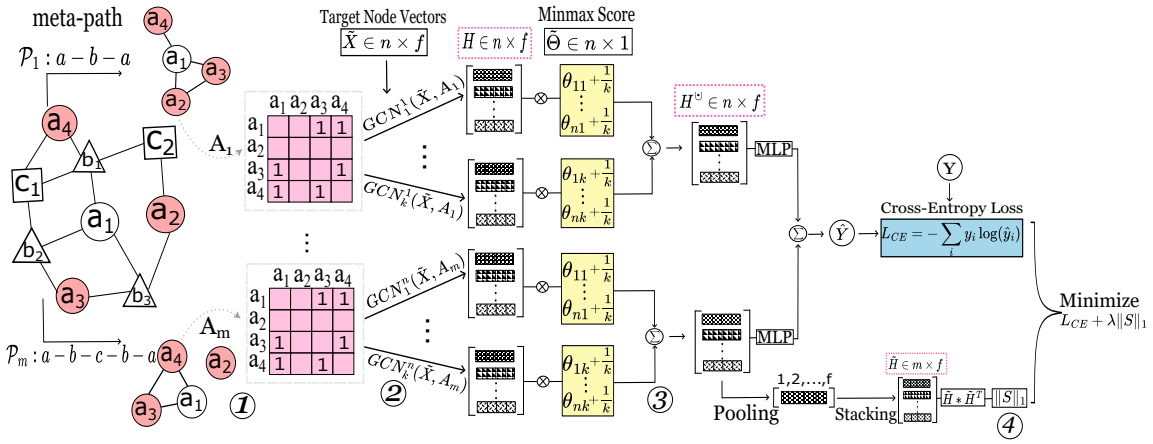

Figure 1: The proposed HGEN for heterogeneous graph ensemble learning. From left to right, ①: a heterogeneous graph is first converted to m meta-graphs (one for each meta-path), with A_m denotes adjacency matrix of the graph from the m^{th} meta-path; ②: node feature dropout is applied to each meta-graph and help train k GNN learners (*i.e.* Allele GNNs); ③: Residual-attention is applied to allele GNNs of each meta-path to consolidate their node embedding features, with one multi-layer-perceptron (MLP) project layer is learned from each meta-path; and ④ a correlation regularizer enforces meta-path's embedding features to be different from each other and ensembles of MLPs to have minimum cross-entropy loss. The combined objective function enforces the GNNs, residual-attention, and MLP project layers to collectively learn for optimized ensemble learning goal.

feature and edge dropping to generate diverse GNNs from heterogeneous graphs. This enhances generalization by encouraging different perspectives of the graph for promoting diversity across base learners. To further improve accuracy, a residual-attention mechanism is incorporated to enable adaptive ensemble weighting, allowing base learners to aggregate allele GNNs derived from perturbed graph structures while favoring more informative graphs. Unlike existing serialized or boosting-based ensemble methods, which can be computationally expensive hence impractical for large graphs, HGEN adopts a bagging strategy, enabling a balance between computational efficiency and scalability through parallelization.

Specific contributions of this paper includes the following:

1. HGEN is the first framework to enable ensemble learning for heterogeneous graphs, tackling the unique challenges posed by their complexity and diversity.
2. A novel attention-based aggregation that dynamically adjusts weights based on the contributions of base graph learners is proposed. A residual attention mechanism is to refine ensemble weighting further, enhancing adaptability and improving overall predictive accuracy.
3. Our theoretical analysis substantiates the convergence of HGEN, and that its magnitude of correlation regularization overweighs naïve voting.
4. Empirical study on five real-world heterogeneous networks validates the effectiveness of HGEN over the recent arts, where the datasets establish a new benchmark for heterogeneous graph ensemble learning.

2 Problem Definition

Let $G = (V, E, X, Y)$ denote a heterogeneous graph, where V is the set of nodes, E is the set of edges, X is the feature matrix for the nodes of a certain targeted type, and Y is the one hot encoding label matrix for those target nodes. Denote $\mathcal{T}^v$ as the set of node types and $\mathcal{T}^e$ as the set of edge types, where $t_i \in \mathcal{T}^v$ is the node type i and $e_{i,j} \in \mathcal{T}^e$ is the edge type that connects the node type from t_i to t_j.

Define $\phi : V \mapsto \mathcal{T}^v$ that maps the node from V to its node type, and $\varphi : E \mapsto \mathcal{T}^e$ that maps the edges from E to its edge type. A meta-path $\mathcal{P}$ is a relational sequence $(e_{i,j}e_{j,k}\dots e_{k,j}e_{j,i})$ that is symmetric. Given a specific meta-path $\mathcal{P}_i$, we can construct a homogeneous graph equipped with an adjacency matrix $A_i \in \mathbb{1}^{n\times n}$ where $A_i[j,k] = 1 \iff \exists\, p = (v_j, \dots, v_k)$ such that edges in the path p match the meta-path $\mathcal{P}_i$, formally presented as follows.

$$\text{Gen}(G, \mathcal{P}) = A_i \quad \text{where}$$
$$A_i[j,k] = \begin{cases} 1 & \text{if } \exists p = (v_j, \dots, v_k) \text{ matching } \mathcal{P}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Given graph G and a target node type $t_\iota \in \mathcal{T}^v$, with $V_{t_\iota} \subset V$ being the target node set containing n target nodes, a simple graph learner such as GNN can only learn from a homogeneous graph. Our **goal** is to enable ensemble learning of multiple GNNs to predict the label Y for the target node set V_{t_ι} so to maximize the classification accuracy and AUC values.

3 Proposed Framework

The proposed HGEN ensemble learning framework, in Figure 1, uses meta-paths to extract multiple disparate homogeneous graphs from the heterogeneous graph, such that base graph learners like GCN, GAT, or GraphSAGE can be trained from the extracted graphs to form an ensemble. Two major components of HGEN to enhance base learner accuracy and diversity are presented in Sections 3.1 and 3.3, respectively,

followed by the analysis. Main steps of HGEN are outlined in Algorithm 1 in Appendix.

3.1 HGEN Base Learner Accuracy Enhancement

Allele Graph Neural Network Learning

For each meta-path $\mathcal{P}_i$, we extract a homogeneous graph $G_i = (A_i, X_i)$ consisting of target nodes and its adjacency matrix denoted by A_i and nodal features X_i. Although GNNs under certain conditions such as Lipschitz graph filters are stable [Gama *et al.*, 2020], most GNNs intend have unstable learning outcomes, especially when trained with limited samples with random initialization. This motivates us to devise *allele graph neural networks* for ensemble learning, where alleles refer to variant GNN networks learned from the same source. Therefore, we propose to augment each meta-path graph by using node feature dropout, as follows.

$$\tilde{X}_i = \text{Dropout}(X_i, b) \tag{2}$$

$$H_i^{(0)} = \sigma(\tilde{X}_i W_i^0) \tag{3}$$

where b is the dropout rate, $\sigma(\cdot)$ is the non-linear activation function, X^i is the original node features, W_i^0 is the projection learnable parameters, and $H_i^{(0)}$ is the embedding output prepared for graph learning. Instead of using more sophisticated graph augmentation approaches, our ablation study in Sec. 4 will soon demonstrate that feature dropout outperforms other alternatives. This is also consistent with previous observations where feature dropout often outperforms node or edge dropout [Shu *et al.*, 2022].

After applying base graph learner GNN to augmented meta-path graph $\tilde{G}_i = (A_i, \tilde{X}_i)$, we can obtain one base learner. For each meta-path $\mathcal{P}_i$, we apply different initializations and dropout to obtain k base GNN models. With m meta-paths, a total of $(k * m)$ base GNNs are created. Each single base GNN consists of a single linear projection layer projecting raw feature input X along and multiple graph filtering layers aggregating embeddings with corresponding adjacency matrix A_i. The projection layer is a simple linear layer projecting raw features with a random feature dropout followed by a nonlinear-activation. For the message passing scheme, we used three different backbones: graph convolution layer, graph attention layer, and GraphSAGE layer to show the effects of base model variants. In general, any standard message passing scheme can fit into the framework.

In the following, we outline the j^{th} GNN from the i^{th} meta-path $\mathcal{P}_i$ using GCN, and the layer-wise aggregation of the node information is as follows:

$$H_{i,j}^{(l)} = \sigma(D_i^{-\frac{1}{2}}(A_i + I)D_i^{-\frac{1}{2}}H_{i,j}^{(l-1)}W_{i,j}^{(l)}) \tag{4}$$

with D_i denoting the degree matrix of $(A_i + I)$, $W_{i,j}^{(l)}$ as the learnable parameters for the convolution layer l. With L graph convolution layers in total $l \in \{1, \cdots, L\}$, the GCN outputs of the final node embeddings for the j^{th} GNN from the ith meta-path $\mathcal{P}_i$ is represented as $H_{i,j}^{(L)}$.

3.2 Intra Fusion for each Meta-Path

Residual-Attention for Allele GNN Fusion

The allele GNNs each learns from an augmented meta-path graph, conveying unique information of the underlying meta-path semantics and the network heterogeneity. As the first key step of the ensemble learning process, a fusion mechanism is introduced to consolidate embedding features from allele GNNs to adaptively adjust weight of respective GNNs.

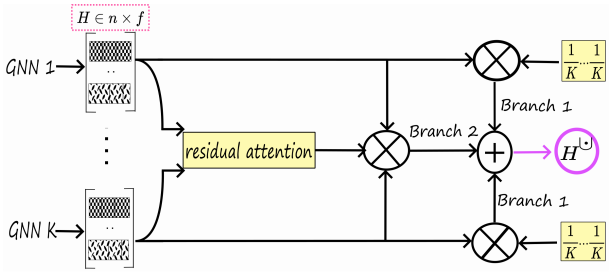

Figure 2: Residual attention concept: instead of learning the attention for each GCN with only one branch, we use the residual mechanism to learn a "Branch 2" representing perturbation deviated from the "Branch 1" which is a simple average ensemble GNN.

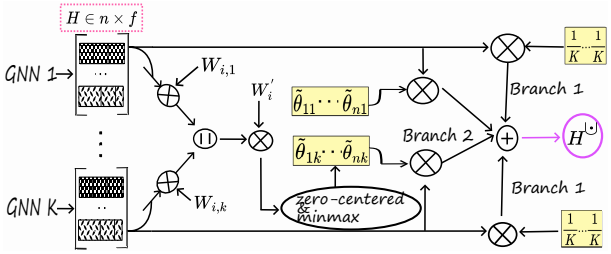

Figure 3: Residual attention computation: k GNNs first being compressed to attention space through projection weight $W_{i,j}$ and then being concatenated and fed to a shared projection weight W_i' to learn k Θ attention scores. Using minmax normalization (refer to Eq 5), the final fusion residual attention $\tilde{\Theta}$ is obtained. Final fusion representation $H^{\cup}$ is the summation of "Branch 1" and "Branch 2".

Instead of directly learning GNN weights, we introduce a residual-attention mechanism which borrows residual network's [He *et al.*, 2015] unique strength of learning identity mapping that is hard to directly learn from highly nonlinear functions. The residual mechanism provides a good precondition or initial points that guarantee easy learning on recovering shallow layer results. Meanwhile, it is known that the attention mechanism is highly non-linear and could be difficult to learn a uniform weight distribution. Inspired by the residual mechanism and its success in image recognition [Wang *et al.*, 2017], we propose a residual-attention approach for the first ensemble stage that ensures an easy learning of allele GNN aggregation, which is shown to be a simple mechanism serving as a good precondition analogy to identity mapping.

Specifically, for each meta-path $\mathcal{P}_i$, our model leverages multiple single GNN to learn the embeddings. To aggregate the final embeddings of multiple GNNs, a residual-attention based fusion mechanism is designed as follows:

$$\Theta = \|_{j=1}^{k}\{(H_{i,j}^{(L)})W_{i,j}\}W_i^{'}, \quad \text{and} \tag{5}$$

$$\bar{\Theta} = \text{Mean}(\Theta), \;\; \hat{\Theta} = \Theta - \bar{\Theta}, \;\; \tilde{\Theta} = \frac{(\hat{\Theta} - \hat{\Theta}^{\downarrow})}{(\hat{\Theta}^{\uparrow} - \hat{\Theta}^{\downarrow})}$$

where $\Theta \in \mathbb{R}^{n \times k}$ is the learned residual attention with

meta-path $\mathcal{P}_i$, where n denotes number of target nodes and k denotes the number of allele GNNs for each meta-path. $\bar{\Theta}$ is the row average of learned Θ. $\hat{\Theta}$ is the broadcasting difference between Θ and $\bar{\Theta}$, which is zero-centered. $\Theta^{\uparrow} = \max(\hat{\Theta})$ and $\Theta^{\downarrow} = \min(\hat{\Theta})$ are the corresponding maximal and minimal attention scores over row. $\tilde{\Theta}$ is the final minimax version of residual attention. The min-max residual attention adaptively adjusts the weights/influence of each base learner. Strong learners making more correct predictions are promoted with higher weights, while weaker learners have lower weights, allowing the model to mitigate the impact of errors without explicit error detection.

Residual-Attention Fusion. In order to fuse embeddings from allele GNNs which are derived from the same meta-path graph, we employ a residual-attention mechanisms, as defined in Eq. (6), where $\tilde{\Theta}[:, i] + \frac{1}{k}$ is the node-wise attention and $+$ is the broadcasting plus over nodes as follows.

$$H_i^{\cup} = \sum_{j=1}^{k} (\tilde{\Theta}[:, j] + \frac{1}{k}) \cdot H_{i,j}^{(L)} \tag{6}$$

As the results of the residual-attention, $H_i^{\cup}$ is used to denote the aggregated node embeddings for meta-path $\mathcal{P}_i$ after residual-attention fusion. It is worth noting that the residual-attention in Eq. (6) does not involve any learnable parameters. The attention in Eq. (5) learns respective parameters to regulate the residual-attention fusion.

3.3 HGEN Base Learner Diversity Enhancement

After obtaining the final embedding $H_i^{\cup}$ for each meta-path $\mathcal{P}_i$, we project the embedding to the output class dimension with a linear layer decoder. Finally, we use summation to bag the prediction for each meta-path to obtain final prediction:

$$\hat{Y} = \sum_{i=1}^{m} \text{MLP}(H_i^{\cup}), \tag{7}$$

where a mean pooling operator $\text{MP}(\cdot)$ is used to obtain graph embeddings $\tilde{H}_i \in R^f$ over $H_i^{\cup}$ for each meta-path $\mathcal{P}_i$. Denote the stacked version of all meta-paths $\tilde{H}_i$ as $\tilde{H} \in \mathbb{R}^{m \times f}$. A correlation matrix is computed quantifying the inter correlation among meta-paths:

$$S = \tilde{H} * \tilde{H}^T \tag{8}$$

where $S \in \mathbb{R}^{m \times m}$ is the correlation matrix for evaluating inter meta-path relation. We applied L_1 norm on S as regularization loss to ensure more independent embeddings learned among all meta-paths. The final objective function of the HGEN is then defined by Eq. (9).

$$\ell = -\sum_i y_i \log(\hat{y}_i) + \lambda \|S\|_1 \tag{9}$$

We evaluated the impact of correlation regularization term by varying its weight coefficient λ as indicated in Eq. (9). A larger λ enforces greater sparsity in the correlation matrix, encouraging base learners trained from different meta-path graphs to be more independent. To assess this effect, we examined the ensemble performance across different meta-paths for $\lambda \in \{0, 0.1, 0.5\}$, as shown in Fig. 4(c) . The violin plots visualize the distribution of ensemble performance. As λ increases, the spread (*i.e.*, width) of the distributions becomes larger, suggesting that models trained from individual meta-paths produce increasingly diverse predictions. This validates that the correlation-regularization term positively contributes to diversity between ensemble models.

3.4 Theoretical Analysis

Complexity. Algorithm 1 in Appendix lists the major steps of HGEN which takes a heterogeneous graph as input, learns allele GNNs from meta-path graphs, and outputs predictions for target nodes. Given m meta-paths and k allele GNNs for each meta-path, with an average of e number of edges for each meta-path. Denote training a single GNN time complexity is T with T at least linearly scaled to $\mathcal{O}(nf + e)$. The asymptotic complexity of the HGEN is then at least $\mathcal{O}(m * (kT) + m * (k * nf)$. With $m * k * nf$ roughly the time for the residual attention computation and $m * (kT)$ for each individual GNN training. Since $nf \leq T$, we should approximately have $c * m * (kT)$ time complexity and asymptotically $\mathcal{O}(m * (kT))$ time complexity. With the assumption that $m, k \ll T$, the total complexity of the ensemble model is linearly scalable.

We report the wallclock runtime performance by assessing the average training time per epoch across ensemble sizes in Fig. 4(b) . We observe a linear increase of runtime *w.r.t.* ensemble size in all five datasets. Besides, on larger graphs, the increasing trend of average training time per epoch *w.r.t.* increasing number of nodes and edges is also linear. The two results suggest scalability of HGEN in both ensemble size and graph size, attesting its practicability for large-scale graph learning. We also conduct experiments by using a large Freebase dataset containing 40,000 nodes and four different magnitude datasets, with the results illustrated in Fig. 4(d) . We observe that the attention in HGEN could introduce slight overhead, but it remains constant and does not scale up with the graph size. If we remove this attention mechanism, the runtime of HGEN will be on a par with the naive ensemble baseline. The extra runtime is mainly required from computing the attention, it will remain manageable for large graphs.

Convergence & Superiority Than Naive Voting. In the following, we derive a Theorem and two remarks, which assert the convergence of HGEN, and its superiority compared to simple voting based ensemble. Detailed derivations are reported in Supplement D.

Theorem 1. *Denoted by $L' = -\sum_i y_i \log(y_i)$ the cross-entropy loss accumulated by by minimizing the objective Eq. (9) over t training iterations. The cumulative loss ℓ at the t-th iteration satisfies:*

$$\begin{aligned} \ell &= L' + \|S\|_1 \\ &\leq t \cdot L'(W_t, W_*^{mlp}) + \frac{\|W_t^{mlp} - W_*^{mlp}\|}{2\eta'} \\ &+ m \cdot f \cdot \max |h_{ij}^{\cup}|^2, \end{aligned}$$

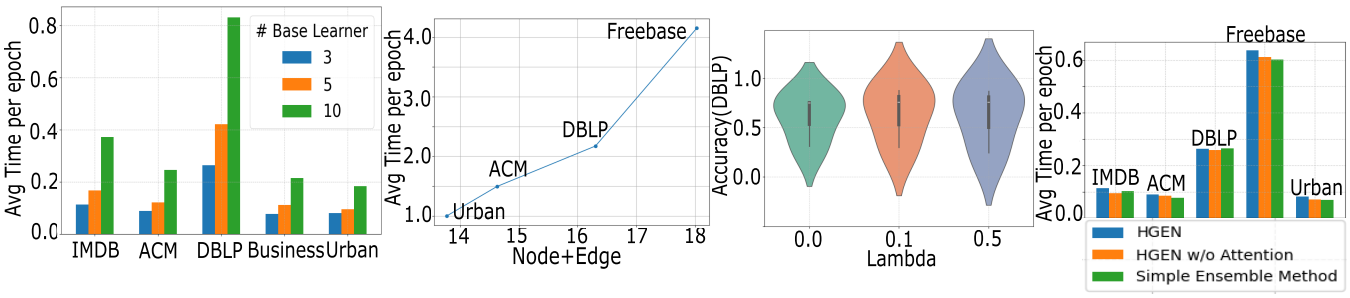

Figure 4: (a) Average runtime per epoch *w.r.t.* different number of base learners on five datasets. (b) Log Average runtime per epoch using three base learners increases with number of nodes plus number of edges in log term. (c) Ensemble accuracy of each meta-path *w.r.t.* different λ values. (d) Runtime comparison on HGEN, HGEN w/o attention, and simple ensemble model on five datasets, where Freebase is the largest to attest their scalability.

where W_t and W_t^{mlp} represent the learnable parameters of base graph learner and the weights of MLP at the t-th iteration, respectively; W_*^{mlp} is the optimal MLP weights and η' the step size. L is the Lipschitz constant. Denoted by $h_{ij}^{\cup}$ the (i,j)-th entry of the fused node embedding matrix $H_j^{\cup}$, whose magnitude is bounded by:

$$h_{ij}^{\cup} \in \Big(\min\Big(\mu - 4\sigma - \sqrt{t}\,\eta L, (k+1)(\mu - 4\sigma - \sqrt{t}\,\eta L)\Big), \max\Big(\mu + 4\sigma + \sqrt{t}\,\eta L, (k+1)(\mu + 4\sigma + \sqrt{t}\,\eta L)\Big)\Big).$$

Remark 1. *Consider the immediate loss $l_t(W_t)$ that holds $l_t(W_t) \leq L'(W_t, W_*^{mlp}) + \frac{\|W_t^{mlp} - W_*^{mlp}\|}{2\eta' t} + \frac{m \cdot f \cdot \max|h_{ij}^{\cup}|^2}{t}$. We observe $\lim_{t\to\infty}(L'(W_t, W_*^{mlp}) - l_t(W_t)) = 0$, where $l_t(W_t)$ is approaching its optimal status as t goes larger. This suggests that the learning process of using MLP to ensemble the resultant node embeddings of base graph learners is stabilizing and improving over time, indicating convergence.*

Remark 2. *The θ-weighted ensemble strategy in* HGEN *enjoys a magnified term $\|S\|_1$ hence strengthen the regularization effect through the aggregated embedding , because* range$(h_{ij}^{\cup}) \geq$ range$(\tilde{h}_{ij}^{\cup})$*, where $\tilde{h}_{ij}^{\cup} \in (\mu - 4\sigma - \sqrt{t}\eta L, \mu + 4\sigma + \sqrt{t}\eta L)$ is the (i,j)-th entry of the fused node embedding matrix generated from naïve voting, and $\widetilde{\Theta}_{i,j} \in [0,1]$, $\forall i,j$. This means that* HGEN *provides a broader range of possible embedding values compared to the naïve voting, thereby resulting in more flexible and informative embeddings, which helps improve overall performance and generalization.*

4 Experiment

4.1 Benchmark Datasets

Five heterogeneous graphs from real applications are used as benchmark datasets. Their statistics and detailed descriptions are deferred to Supplement B of Appendix due to page limits.

4.2 Baselines

We compare our HGEN with some state-of-art baselines including heterogeneous graph embedding models.

- **HAN** [Xiao *et al.*, 2019] leverages node- and semantic-level attention mechanisms to learn heterogeneous node embeddings from different meta-paths.
- **Ensemble-GNN** is a variant of the state-of-the-art ensemble learning method for homogeneous graphs [Wei *et al.*, 2023], which combines predictions from multiple GNNs through voting. To make the method [Wei *et al.*, 2023] working for hoterogeneous graphs, we use meta-paths to generate homogenerous graphs, and apply the method [Wei *et al.*, 2023] for ensemble learning (predictions are generated using GCN, GraphSAGE, and GAT). After training, the predictions across all GNNs and meta-paths are aggregated through voting to determine the final prediction.
- **Transformer-GNN** uses a transformer architecture to combine embeddings from different GNNs (GCN, GraphSAGE, and GAT) for each meta-path. Once predictions are generated for a meta-path, they are stacked and integrated to produce the final prediction, ensuring an effective fusion of heterogeneous graph information.
- **SeHGNN** [Yang *et al.*, 2023b] is a heterogeneous graph representation learning that precomputes neighbor aggregation using a lightweight mean aggregator and avoids repeated computations during training. SeHGNN extends the receptive field with long meta-paths and fuses features through a transformer-based module.
- **NaiveWeighting-GNN** is a variant of the proposed HGEN. It uses the same architecture and loss function (and regularizer) as HGEN to guide learning but replaces the residual-attention fusion of HGEN using a simple mean average to aggregate all allele GNNs. Its comparision to HGEN can demonstrate the advantage of the residual-attention fusion, compared to simple voting.

4.3 Implementation Details

We perform a grid search with selected range of hyperparameters including hidden dimension, layer size, dropping rate, number of individual GNN, and control rate for regularizer. We choose Adam [Kingma and Ba, 2014] as our optimizer. We fix the learning rate, weight decay, the number of epochs and apply early stopping mechanism. For each method, we report the average accuracy and roc-auc score across five random seeds. All experiments are run on desktop workstations equipped with Nvidia GeForce RTX 2080 Ti.

Base Learner:		GCN			GraphSAGE			GAT	
Dataset	Model	Accuracies	AUC	Model	Accuracies	AUC	Model	Accuracies	AUC
IMDB	HAN_{GCN}	$0.540^{*}_{\pm 0.0160}$	$0.720^{*}_{\pm 0.0072}$	HAN_{SAGE}	$0.523^{*}_{\pm 0.0059}$	$0.697_{\pm 0.0121}$	HAN_{GAT}	$0.563^{*}_{\pm 0.0219}$	$0.747^{*}_{\pm 0.0140}$
	SeH_{GCN}	$0.536^{*}_{\pm 0.0031}$	$0.712^{*}_{\pm 0.008}$	SeH_{SAGE}	$0.398^{*}_{\pm 0.0101}$	$0.568^{*}_{\pm 0.0117}$	SeH_{GAT}	$0.419^{*}_{\pm 0.0101}$	$0.591^{*}_{\pm 0.0123}$
	GCN-Ensemble	$0.551^{*}_{\pm 0.0406}$	$0.747^{*}_{\pm 0.0184}$	SAGE-Ensemble	$0.589^{*}_{\pm 0.0055}$	$0.760^{*}_{\pm 0.0021}$	GAT-Ensemble	$0.583^{*}_{\pm 0.0034}$	$0.754^{*}_{\pm 0.0017}$
	$\text{NaiveWeighting}_{GCN}$	$0.596^{*}_{\pm 0.0031}$	$0.774^{*}_{\pm 0.0029}$	$\text{NaiveWeighting}_{SAGE}$	$0.587^{*}_{\pm 0.0067}$	$0.762^{*}_{\pm 0.0063}$	$\text{NaiveWeighting}_{GAT}$	$0.595_{\pm 0.0034}$	$0.770_{\pm 0.0030}$
	Transformer_{GCN}	$0.595^{*}_{\pm 0.0075}$	$0.771^{*}_{\pm 0.0034}$	$\text{Transformer}_{SAGE}$	$0.584^{*}_{\pm 0.0072}$	$0.757^{*}_{\pm 0.0047}$	Transformer_{GAT}	$0.591^{*}_{\pm 0.0036}$	$0.767_{\pm 0.0034}$
	HGEN_{GCN}	$\mathbf{0.604}_{\pm 0.0033}$	$\mathbf{0.776}_{\pm 0.0010}$	HGEN_{SAGE}	$\mathbf{0.605}_{\pm 0.0040}$	$\mathbf{0.775}_{\pm 0.0032}$	HGEN_{GAT}	$\mathbf{0.600}_{\pm 0.0021}$	$\mathbf{0.769}_{\pm 0.0034}$
ACM	HAN_{GCN}	$0.839^{*}_{\pm 0.0183}$	$0.973^{*}_{\pm 0.0015}$	HAN_{SAGE}	$0.880^{*}_{\pm 0.0174}$	$0.977^{*}_{\pm 0.0032}$	HAN_{GAT}	$0.872^{*}_{\pm 0.0107}$	$0.965^{*}_{\pm 0.0066}$
	SeH_{GCN}	$0.794^{*}_{\pm 0.0168}$	$0.923^{*}_{\pm 0.0145}$	SeH_{SAGE}	$0.753^{*}_{\pm 0.0213}$	$0.914^{*}_{\pm 0.0121}$	SeH_{GAT}	$0.730^{*}_{\pm 0.0577}$	$0.898^{*}_{\pm 0.0385}$
	GCN-Ensemble	$0.766^{*}_{\pm 0.0088}$	$0.969^{*}_{\pm 0.0030}$	SAGE-Ensemble	$0.802^{*}_{\pm 0.0308}$	$0.984_{\pm 0.0012}$	GAT-Ensemble	$0.825^{*}_{\pm 0.0078}$	$0.978_{\pm 0.0005}$
	$\text{NaiveWeighting}_{GCN}$	$0.892^{*}_{\pm 0.0089}$	$0.977_{\pm 0.0017}$	$\text{NaiveWeighting}_{SAGE}$	$0.909^{*}_{\pm 0.0113}$	$0.984_{\pm 0.0009}$	$\text{NaiveWeighting}_{GAT}$	$0.892_{\pm 0.0062}$	$0.978_{\pm 0.0005}$
	Transformer_{GCN}	$0.898^{*}_{\pm 0.0103}$	$0.977_{\pm 0.0019}$	$\text{Transformer}_{SAGE}$	$0.912^{*}_{\pm 0.0017}$	$0.983^{*}_{\pm 0.0008}$	Transformer_{GAT}	$0.905_{\pm 0.0041}$	$0.978_{\pm 0.0007}$
	HGEN_{GCN}	$\mathbf{0.909}^{*}_{\pm 0.0016}$	$\mathbf{0.977}_{\pm 0.0016}$	HGEN_{SAGE}	$\mathbf{0.923}_{\pm 0.0022}$	$\mathbf{0.984}_{\pm 0.0006}$	HGEN_{GAT}	$\mathbf{0.908}_{\pm 0.0007}$	$\mathbf{0.977}_{\pm 0.0011}$
DBLP	HAN_{GCN}	$0.868^{*}_{\pm 0.0247}$	$0.970^{*}_{\pm 0.0097}$	HAN_{SAGE}	$0.891^{*}_{\pm 0.0161}$	$0.975^{*}_{\pm 0.0048}$	HAN_{GAT}	$0.900^{*}_{\pm 0.0100}$	$0.982^{*}_{\pm 0.0020}$
	SeH_{GCN}	$0.809^{*}_{\pm 0.0175}$	$0.878^{*}_{\pm 0.0355}$	SeH_{SAGE}	$0.780^{*}_{\pm 0.0266}$	$0.926^{*}_{\pm 0.0155}$	SeH_{GAT}	$0.878^{*}_{\pm 0.0094}$	$0.971^{*}_{\pm 0.0026}$
	GCN-Ensemble	$0.925^{*}_{\pm 0.0123}$	$0.990_{\pm 0.0014}$	SAGE-Ensemble	$0.930^{*}_{\pm 0.0022}$	$0.990_{\pm 0.0006}$	GAT-Ensemble	$0.867^{*}_{\pm 0.0439}$	$0.977^{*}_{\pm 0.0046}$
	$\text{NaiveWeighting}_{GCN}$	$0.932_{\pm 0.0017}$	$0.990^{*}_{\pm 0.0007}$	$\text{NaiveWeighting}_{SAGE}$	$0.931_{\pm 0.0021}$	$0.989_{\pm 0.0007}$	$\text{NaiveWeighting}_{GAT}$	$0.919^{*}_{\pm 0.0071}$	$0.984_{\pm 0.0016}$
	Transformer_{GCN}	$0.913^{*}_{\pm 0.0179}$	$0.948^{*}_{\pm 0.0114}$	$\text{Transformer}_{SAGE}$	$0.928^{*}_{\pm 0.0025}$	$0.988^{*}_{\pm 0.0014}$	Transformer_{GAT}	$0.902^{*}_{\pm 0.0103}$	$0.983^{*}_{\pm 0.0013}$
	HGEN_{GCN}	$\mathbf{0.932}_{\pm 0.0020}$	$\mathbf{0.991}_{\pm 0.0003}$	HGEN_{SAGE}	$\mathbf{0.936}_{\pm 0.0021}$	$\mathbf{0.989}_{\pm 0.0015}$	HGEN_{GAT}	$\mathbf{0.928}_{\pm 0.0031}$	$\mathbf{0.987}_{\pm 0.0018}$
Business	HAN_{GCN}	$0.717^{*}_{\pm 0.0022}$	$0.782^{*}_{\pm 0.0015}$	HAN_{SAGE}	$0.720^{*}_{\pm 0.0030}$	$0.779^{*}_{\pm 0.0029}$	HAN_{GAT}	$0.692^{*}_{\pm 0.0193}$	$0.744^{*}_{\pm 0.0296}$
	SeH_{GCN}	$0.702^{*}_{\pm 0.0134}$	$0.759^{*}_{\pm 0.0163}$	SeH_{SAGE}	$0.678^{*}_{\pm 0.0037}$	$0.708^{*}_{\pm 0.0093}$	SeH_{GAT}	$0.597^{*}_{\pm 0.0927}$	$0.587^{*}_{\pm 0.1487}$
	GCN-Ensemble	$0.708^{*}_{\pm 0.0012}$	$0.775^{*}_{\pm 0.0006}$	SAGE-Ensemble	$0.710^{*}_{\pm 0.0026}$	$0.772^{*}_{\pm 0.0008}$	GAT-Ensemble	$0.705^{*}_{\pm 0.0038}$	$0.774^{*}_{\pm 0.0016}$
	$\text{NaiveWeighting}_{GCN}$	$0.715^{*}_{\pm 0.0035}$	$0.770_{\pm 0.0329}$	$\text{NaiveWeighting}_{SAGE}$	$0.720^{*}_{\pm 0.0030}$	$0.784^{*}_{\pm 0.0022}$	$\text{NaiveWeighting}_{GAT}$	$0.712^{*}_{\pm 0.0038}$	$0.778^{*}_{\pm 0.0051}$
	Transformer_{GCN}	$0.719^{*}_{\pm 0.0023}$	$0.786^{*}_{\pm 0.0018}$	$\text{Transformer}_{SAGE}$	$0.721^{*}_{\pm 0.0064}$	$0.783^{*}_{\pm 0.0045}$	Transformer_{GAT}	$0.713^{*}_{\pm 0.0021}$	$0.780^{*}_{\pm 0.0021}$
	HGEN_{GCN}	$\mathbf{0.725}_{\pm 0.0042}$	$\mathbf{0.788}_{\pm 0.0011}$	HGEN_{SAGE}	$\mathbf{0.732}_{\pm 0.0019}$	$\mathbf{0.787}_{\pm 0.0018}$	HGEN_{GAT}	$\mathbf{0.726}_{\pm 0.0059}$	$\mathbf{0.785}_{\pm 0.0027}$
Urban	HAN_{GCN}	$0.231^{*}_{\pm 0.0155}$	$0.596^{*}_{\pm 0.0090}$	HAN_{SAGE}	$0.502^{*}_{\pm 0.0178}$	$0.811^{*}_{\pm 0.0061}$	HAN_{GAT}	$0.368^{*}_{\pm 0.0650}$	$0.765_{\pm 0.0470}$
	SeH_{GCN}	$0.204^{*}_{\pm 0.0037}$	$0.458^{*}_{\pm 0.0062}$	SeH_{SAGE}	$0.329^{*}_{\pm 0.0276}$	$0.779^{*}_{\pm 0.0175}$	SeH_{GAT}	$0.206^{*}_{\pm 0.0000}$	$0.487^{*}_{\pm 0.0637}$
	GCN-Ensemble	$0.206^{*}_{\pm 0.0000}$	$0.416^{*}_{\pm 0.0026}$	SAGE-Ensemble	$0.454^{*}_{\pm 0.0185}$	$0.832^{*}_{\pm 0.0052}$	GAT-Ensemble	$0.300^{*}_{\pm 0.0207}$	$0.761^{*}_{\pm 0.0134}$
	$\text{NaiveWeighting}_{GCN}$	$0.246^{*}_{\pm 0.0367}$	$0.532^{*}_{\pm 0.0798}$	$\text{NaiveWeighting}_{SAGE}$	$0.538^{*}_{\pm 0.0108}$	$0.831_{\pm 0.0125}$	$\text{NaiveWeighting}_{GAT}$	$0.444^{*}_{\pm 0.0578}$	$0.815_{\pm 0.016}$
	Transformer_{GCN}	$0.201^{*}_{\pm 0.0045}$	$0.457^{*}_{\pm 0.0034}$	$\text{Transformer}_{SAGE}$	$0.500^{*}_{\pm 0.0477}$	$0.815^{*}_{\pm 0.0091}$	Transformer_{GAT}	$0.383_{\pm 0.0531}$	$0.787^{*}_{\pm 0.0212}$
	HGEN_{GCN}	$\mathbf{0.289}_{\pm 0.0000}$	$\mathbf{0.612}_{\pm 0.0090}$	HGEN_{SAGE}	$\mathbf{0.591}_{\pm 0.0142}$	$\mathbf{0.850}_{\pm 0.0115}$	HGEN_{GAT}	$\mathbf{0.451}_{\pm 0.0353}$	$\mathbf{0.813}_{\pm 0.0106}$

Table 1: Performance comparisons between baselines and our proposed method equipped with GCN, GraphSAGE, and GAT graph base learners across five heterogeneous datasets. Accuracies (ACC) and AUC values are reported over 5 different initialization status. Superscript * indicates that HGEN is statistically significantly better than this method at 95% confidence level using the performance metrics.

4.4 Results and Analysis

Variants and Baseline Comparison. Table 1 reports the results of the experiment on five datasets with different baselines, our proposed method, and variants over three individual message passing backbones, including graph convolution network (GCN), graph attention network (GAT), and GraphSAGE. Within the same message passing scheme, it can be observed that our proposed HGEN consistently performs better over other baselines with 95% confidence level on IMDB, Business, ACM, and Urban datasets, and scores on top along with the GCN-Ensemble method on DBLP dataset, proving the superiority of our framework.

Compared to all baselines (excluding NaiveWeighting GNN, which is HGEN's variant), HGEN performs consistently better over all datasets with 95% confidence level. For GCN backbones, HGEN beats all other variants over IMDB, Business, ACM, and Urban dataset and performs on top along with weighted GCN over DBLP dataset. For GAT backbones, our methods is on par with Transformer_{GAT} and weighted GAT over Urban and ACM datasets and outperforms others in the rest of datasets. This shows the advantage and necessity of our individual components.

Note, HGEN is statistically significantly better than GNN-Ensemble on 25 out of 30 occasions (across all five datasets). Although both of them are graph ensemble learning methods, GNN-ensemble's base learners are not regulated by the global objective function and there is no constraint to enhance base learner diversities. This results in suboptimal base learners with low accuracies and diversity.

Comparing HGEN and its variant $\text{NaiveWeighting}_{GNN}$, the results show that HGEN is statistically significantly better than NaiveWeighting-GNN on 17 out of 30 occasions (across all five datasets), asserting the advantage of the proposed residual-attention for allele GNN fusions, compared to naive voting. This is also consistent with our theoretical analysis in Section 3.3 which asserts that the Θ-weighted ensemble strategy strengthens the regularization effects due to its larger range of $h_{ij}^{\cup}$, compared to naive voting.

Ablation Study on Allele GNNs & Meta-paths. In order to validate the impact of allele GNNs on HGEN's ensemble learning results, we report mean and variance for allele GNNs leaner and HGEN in Figures 5(a) and 5(b), where the blue violin plots represents allele GNN models' accuracies (mean and variance), and the the orange violin plots show the corresponding final accuracy of HGEN. For each dataset, the results are reported by increasing the number of meta-paths. Figures 5(a) and 5(b) show that the variance of the blue plots is significantly larger than that of the orange plots. This suggests that individual allele GNN models exhibits larger variability and diversity in their predictions, a preferable setting for ensemble learning. As a result, the final accuracy, after ensemble, shows a more stable and consistent result.

As we add more meta-paths, more GNNs are obtained. Each GNN model learns different aspects of the data, improving the overall robustness of the model. However, while individual GNNs show more diversity, their performance can still be limited by the inherent biases or weaknesses of each model. Meanwhile, when comparing individual GNN accuracy and HGEN final accuracy, we observe that the ensemble results have a much higher accuracy, showing the success of our residual attention ensemble approach. Using attention mechanisms, HGEN learns dynamic weights to each meta-

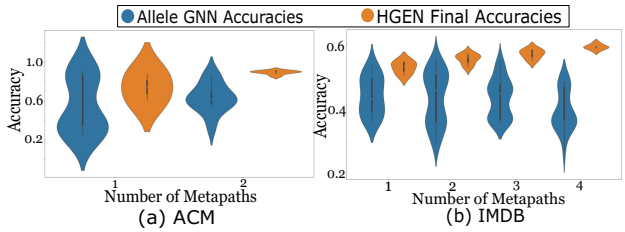

Figure 5: Impact of allele GNNs and meta-paths on the ensemble learning results (using GCN as the base learner). Blue violin plots show allele GNNs' mean and variance whereas orange violin plots show HGEN's mean and variance.

path, focusing on capturing and fusing individual GNN's advantage in the learning process. By enforcing diversity across models from different meta-paths through the $\|S\|_1$, HGEN strengthens their embedding fusion range$(h_{ij}^{\cup})$ as we have theoretically analyzed in Section 3.3, and therefore improves the final prediction results.

Dataset	Model	ACC	AUC
IMDB	HGEN feature drop+regularizer	**0.605**$_{\pm 0.0040}$	**0.775**$_{\pm 0.0032}$
	HGEN feature drop	0.600$_{\pm 0.0112}$	0.773$_{\pm 0.0071}$
	HGEN edge drop+regularizer	0.582$_{\pm 0.0143}$	0.755$_{\pm 0.0168}$
	HGEN feature drop+edge drop+regularizer	0.595$_{\pm 0.0034}$	0.771$_{\pm 0.0050}$
ACM	HGEN feature drop+regularizer	**0.923**$_{\pm 0.0022}$	**0.984**$_{\pm 0.0006}$
	HGEN feature drop	0.909$_{\pm 0.0083}$	0.983$_{\pm 0.0014}$
	HGEN edge drop+regularizer	0.914$_{\pm 0.0067}$	0.983$_{\pm 0.0005}$
	HGEN feature drop+edge drop+regularizer	0.912$_{\pm 0.0142}$	0.983$_{\pm 0.0014}$
DBLP	HGEN feature drop+regularizer	**0.936**$_{\pm 0.0021}$	**0.989**$_{\pm 0.0015}$
	HGEN feature drop	0.933$_{\pm 0.0030}$	0.988$_{\pm 0.0004}$
	HGEN edge drop+regularizer	0.916$_{\pm 0.0220}$	0.985$_{\pm 0.0060}$
	HGEN feature drop+edge drop+regularizer	0.921$_{\pm 0.0168}$	0.987$_{\pm 0.0044}$
Business	HGEN feature drop+regularizer	**0.732**$_{\pm 0.0019}$	**0.787**$_{\pm 0.0018}$
	HGEN feature drop	0.725$_{\pm 0.0043}$	0.787$_{\pm 0.0021}$
	HGEN edge drop+regularizer	0.724$_{\pm 0.0045}$	0.788$_{\pm 0.0032}$
	HGEN feature drop+edge drop+regularizer	0.723$_{\pm 0.0059}$	0.785$_{\pm 0.0047}$
Urban	HGEN feature drop+regularizer	**0.591**$_{\pm 0.0142}$	**0.850**$_{\pm 0.0115}$
	HGEN feature drop	0.553$_{\pm 0.0117}$	0.837$_{\pm 0.0122}$
	HGEN edge drop+regularizer	0.537$_{\pm 0.0185}$	0.842$_{\pm 0.0083}$
	HGEN feature drop+edge drop+regularizer	0.548$_{\pm 0.0348}$	0.845$_{\pm 0.0059}$

Table 2: Ablation study results *w.r.t.* regularizer, feature Dropout, edge Dropout (using GraphSAGE as base learners). Node Dropout was left out due to its significant inferior performance.

Ablation Study on Augmentations & Regularizer.
Table 2 reports the results of HGEN using different augmentations (feature/edge dropout), combined with the regularizer across various datasets. Feature dropping diversifies individual GNNs by enabling a broader range of learning process, which improves the models' capacity for generalization. The regularizer ensures predictive consistency by enforcing constraints which further enhances model performance. These combined effects make the framework particularly suitable for handling heterogeneous graphs. We observed that the Urban dataset benefits significantly from the regularizer and feature dropping, showcasing their utility in domains characterized by high heterogeneity and noise.

4.5 Case Study Analysis

To demonstrate why and how HGEN outperforms baselines, we carry out case studies on each benchmark dataset to compare samples on which HGEN makes correct classification whereas rivals (*i.e.* HAN [Xiao *et al.*, 2019]) make mistakes,

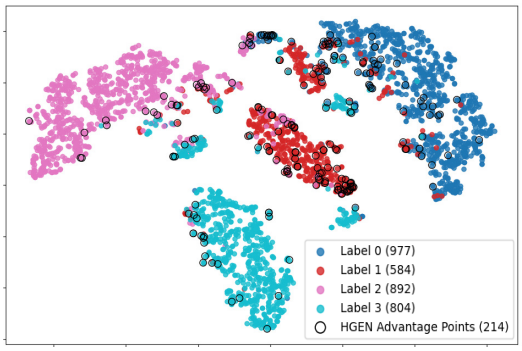

Figure 6: Case study on the DBLP dataset explaining why and how HGEN outperforms HAN using t-visualization. Points represent samples in the DBLP dataset, colored-coded based on their ground-truth labels. The circled points are correctly classified by HGEN but misclassified by HAN. There are 214 circled points.

and report the case study on DBLP in Figure 6 (case studies of rest datasets are reported in Supplement C). For each dataset, points represent samples which are color-coded based on their ground-truth class. Circled points are correctly classified by HGEN but misclassified by HAN.

Taking DBLP dataset in Figure 6 as an example, it can be observed that the circled points are mostly around the edges of the clusters, meaning boundary or difficult cases for separation. The proposed HGEN can correctly identify boundary points for each class that the HAN method incorrectly classify. Similar phenomena can be observed from other four datasets. In fact, this is one of the frequently observed advantages that are naturally brought about by ensemble learning [Ross *et al.*, 2020; Polikar, 2006].

5 Conclusion

This paper proposed HGEN, a novel ensemble learning framework for heterogeneous graphs. Unlike existing ensemble methods that focus mainly on homogeneous graphs, HGEN addresses the unique challenges posed by graph heterogeneity, including various node and edge types, and the need for accurate and diverse base learners. By leveraging a regularized allele GNN framework, HGEN enhances generalization through feature dropping techniques, promoting diversity among base learners. The residual-attention mechanism further enables adaptive ensemble weighting, ensuring improved predictive performance through dynamic aggregation of allele GNNs.

Ethical Statement

This is basic ML research and entails no ethical issues.

Acknowledgments

This work has been supported in part by the National Science Foundation (NSF) under Grant Nos IIS-2236578, IIS-2236579, IIS-2302786, IIS-2441449, IOS-2430224, and IOS-2446522, the Science Center for Marine Fisheries (SCeMFiS), and the Commonwealth Cyber Initiative (CCI).

References

[Chen *et al.*, 2022] Yen-Liang Chen, Chen-Hsin Hsiao, and Chia-Chi Wu. An ensemble model for link prediction based on graph embedding. *Decision Support Systems*, 157:113753, 2022.

[Duan *et al.*, 2024] Rui Duan, Chungang Yan, Junli Wang, and Changjun Jiang. Graph ensemble neural network. *Information Fusion*, 110:102461, 2024.

[Fu *et al.*, 2020] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of The Web Conference 2020*, WWW '20. ACM, April 2020.

[Gama *et al.*, 2020] Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability properties of graph neural networks. *IEEE Transactions on Signal Processing*, 68:5680–5695, 2020.

[Goyal *et al.*, 2020] Palash Goyal, Sachin Raja, Di Huang, Sujit Rokka Chhetri, Arquimedes Canedo, Ajoy Mondal, Jaya Shree, and CV Jawahar. Graph representation ensemble learning. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 24–31, 2020.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[Houssou *et al.*, 2019] Noudéhouénou L. J. Houssou, Jeanloup Guillaume, and Armelle Prigent. A graph based approach for functional urban areas delineation. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 652–658, New York, NY, USA, 2019. Association for Computing Machinery.

[Jin *et al.*, 2024] Yufei Jin, Heng Lian, Yi He, and Xingquan Zhu. Hgdl: Heterogeneous graph label distribution learning. In *Proc. of the The Thirty-Eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, NeurIPS '24, 2024.

[Kingma and Ba, 2014] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[Lin *et al.*, 2022] Qi Lin, Shuo Yu, Ke Sun, Wenhong Zhao, Osama Alfarraj, Amr Tolba, and Feng Xia. Robust graph neural networks via ensemble learning. *Mathematics*, 10(8), 2022.

[Molteni *et al.*, 1996] F. Molteni, R. Buizza, T. N. Palmer, and T. Petroliagis. The ecmwf ensemble prediction system: Methodology and validation. *Quarterly Journal of the Royal Meteorological Society*, 122(529):73–119, 1996.

[Polikar, 2006] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[Ross *et al.*, 2020] Andrew Ross, Weiwei Pan, Leo Celi, and Finale Doshi-Velez. Ensembles of locally independent prediction models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5527–5536, 2020.

[Shu *et al.*, 2022] Juan Shu, Bowei Xi, Yu Li, Fan Wu, Charles Kamhoua, and Jianzhu Ma. Understanding dropout for graph neural networks. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 1128–1138, New York, NY, USA, 2022. Association for Computing Machinery.

[Sun *et al.*, 2023] Shuo Sun, Xinrun Wang, Wanqi Xue, Xiaoxuan Lou, and Bo An. Mastering stock markets with efficient mixture of diversified trading experts. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2109–2119. Association for Computing Machinery, 2023.

[Tang *et al.*, 2008] Jie Tang, Jing Zhang, Limin Yao, Juan-Zi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Knowledge Discovery and Data Mining*, 2008.

[Wang *et al.*, 2017] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. *arXiv preprint arXiv:1704.06904*, 2017.

[Wei *et al.*, 2023] Wenqi Wei, Mu Qiao, and Divyesh Jadav. Gnn-ensemble: Towards random decision graph neural networks. In *2023 IEEE International Conference on Big Data (BigData)*, pages 956–965, 2023.

[Xiao *et al.*, 2019] Wang Xiao, Ji Houye, Shi Chuan, Wang Bai, Cui Peng, Yu P., and Ye Yanfang. Heterogeneous graph attention network. *WWW*, 2019.

[Yang *et al.*, 2023a] Sai Yang, Fan Liu, Delong Chen, and Jun Zhou. Few-shot classification via ensemble learning with multi-order statistics. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1631–1639. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[Yang *et al.*, 2023b] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:10816–10824, 06 2023.

[Zhang *et al.*, 2019] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 793–803, New York, NY, USA, 2019. Association for Computing Machinery.

[Zhang *et al.*, 2024] Chenrui Zhang, Lin Liu, Chuyuan Wang, Xiao Sun, Hongyu Wang, Jinpeng Wang, and Mingchen Cai. Prefer: Prompt ensemble learning via feedback-reflect-refine. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19525–19532, 2024.