Simple and Efficient Heterogeneous Graph Neural Network

Xiaocheng Yang[1], **Mingyu Yan**[*1], **Shirui Pan**[2], **Xiaochun Ye**[1], **Dongrui Fan**[1,3]

[1] State Key Lab of Processors, Institute for Computing Technology, Chinese Academy of Sciences, China
[2] School of Information and Communication Technology, Griffith University, Australia
[3] School of Computer Science and Technology, University of Chinese Academy of Sciences, China
{yangxiaocheng, yanmingyu}@ict.ac.cn, s.pan@griffith.edu.au, {yexiaochun, fandr}@ict.ac.cn

AAAI-2023

Code: https://github.com/ICT-GIMLab/SeHGNN

Reported by Dongdong Hu

Introduction

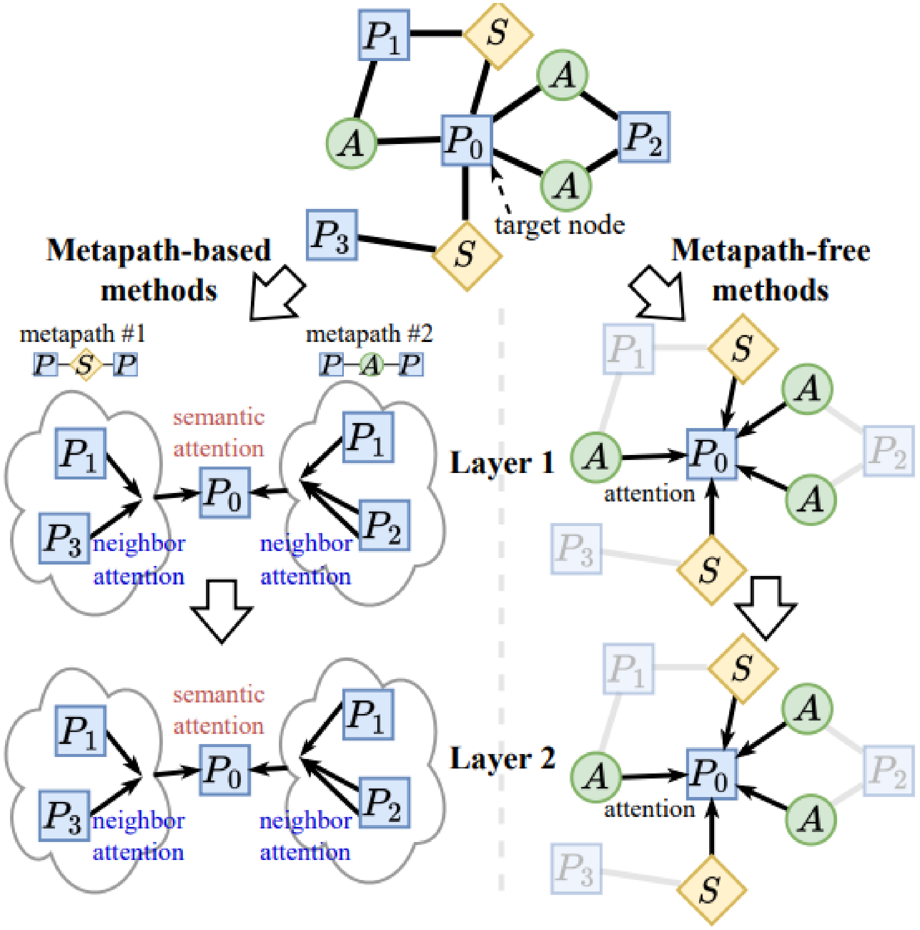


Figure 1: The general architectures of metapath-based methods and metapath-free methods on heterogeneous graphs.

Existing HGNNs inherit many mechanisms from graph neural networks (GNNs) over homogeneous graphs, especially the attention mechanism and the multi-layer structure.

These mechanisms bring excessive complexity, but seldom work studies whether they are really effective on heterogeneous graphs

Method

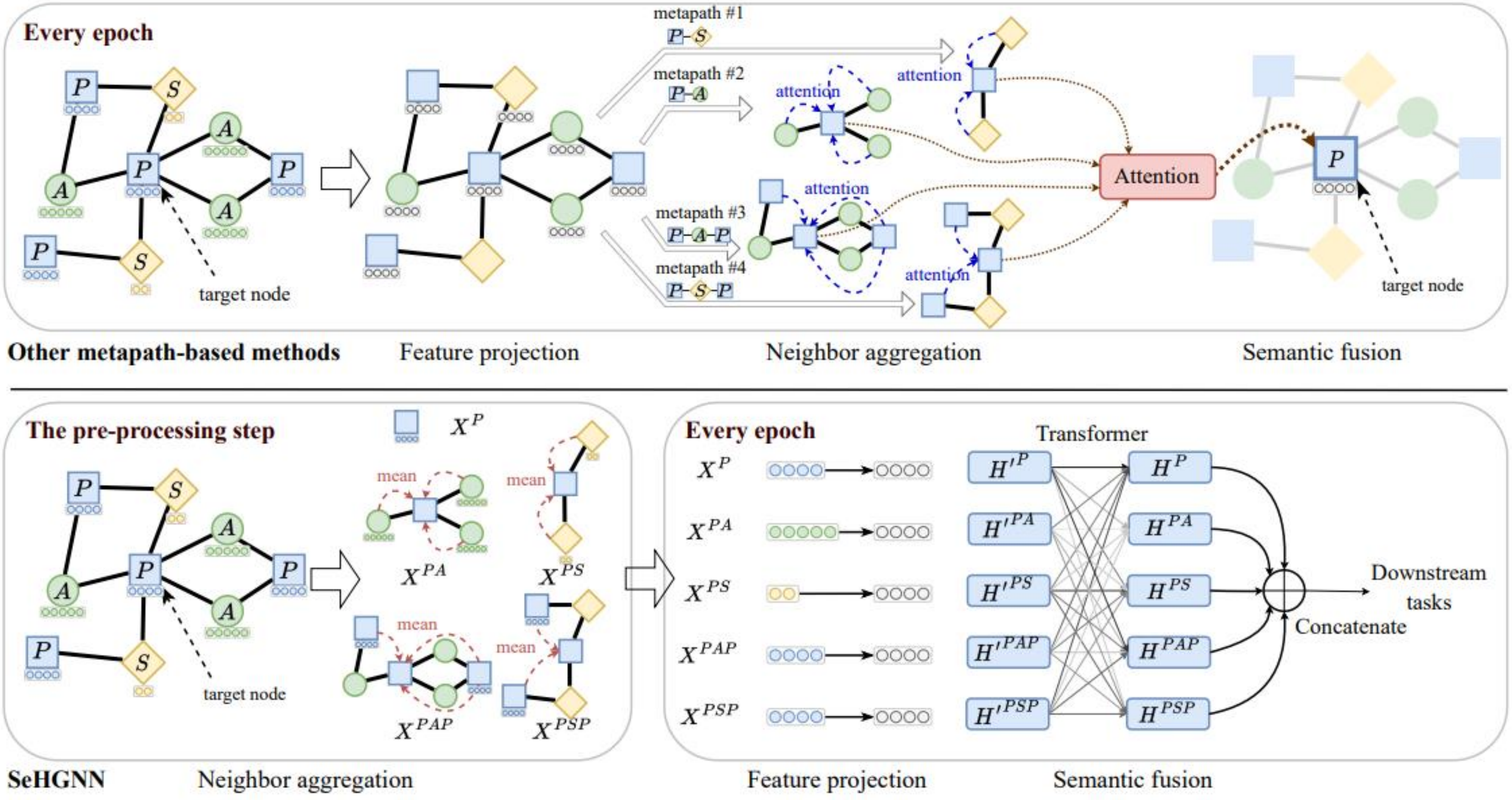
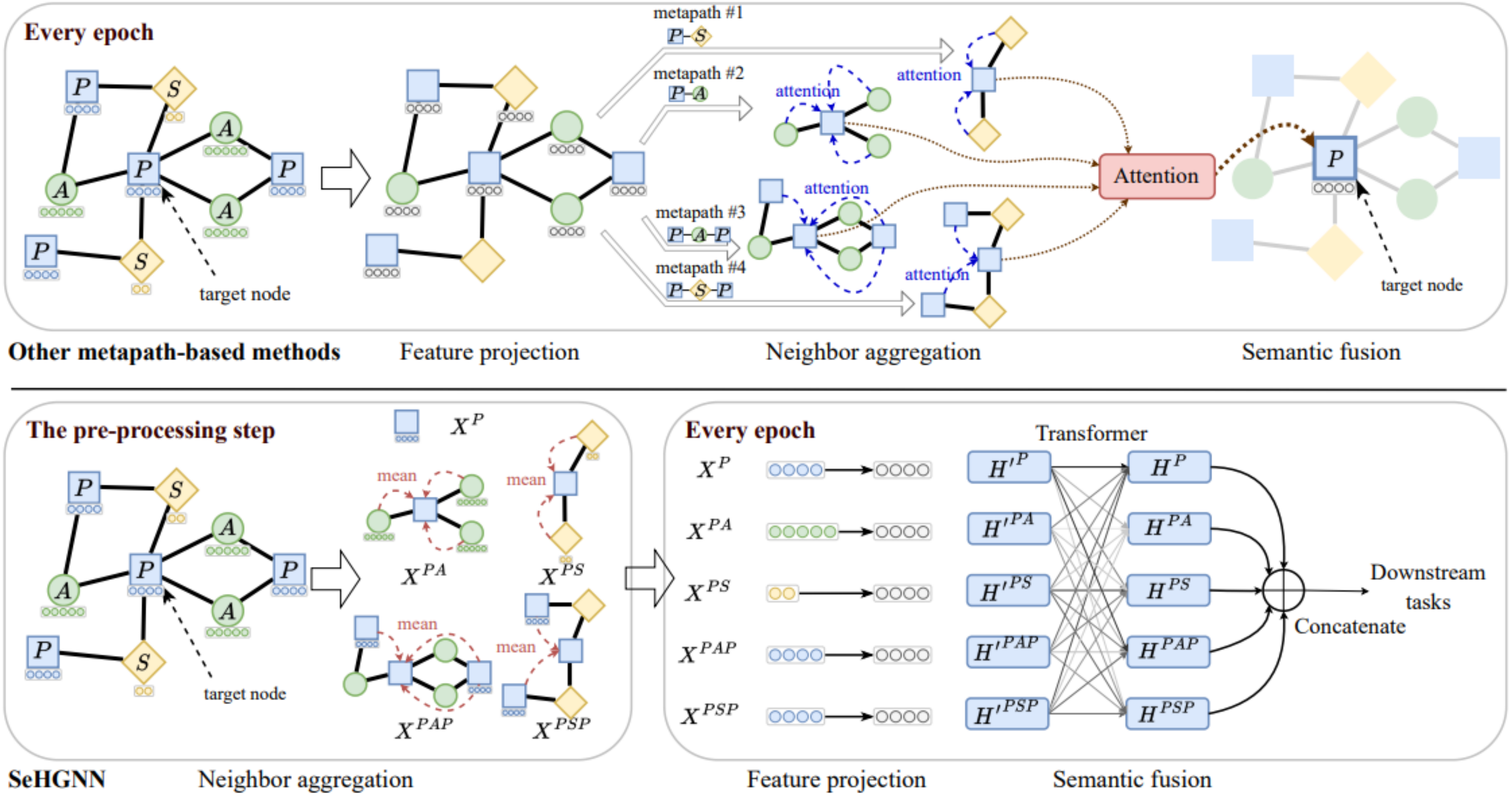


Figure 2: The architecture of SeHGNN compared to previous metapath-based methods. The example is based on ACM dataset with node types author (A), paper (P), and subject (S). This figure exhibits aggregation of 0-hop metapath P (the target node itself), 1-hop metapaths PA, PS, and 2-hop metapaths PAP, PSP.

Method



$$m_i = \{\mathbf{z}_i^{\mathcal{P}} = \frac{1}{||S^{\mathcal{P}}||} \sum_{p(i,j)\in S_{\mathcal{P}}} \mathbf{x}_j : \mathcal{P} \in \Phi_X\},$$

where $S^{\mathcal{P}}$ is the set of all metapath instances corresponding to metapath $\mathcal{P}$ and $p(i,j)$ is one metapath instance with the target node i and the source node j.

$$X^c = \{{x_0^c}^T; {x_1^c}^T; \ldots; {x_{||V^c||-1}^c}^T\} \in \mathbb{R}^{||V^c||\times d^c}$$

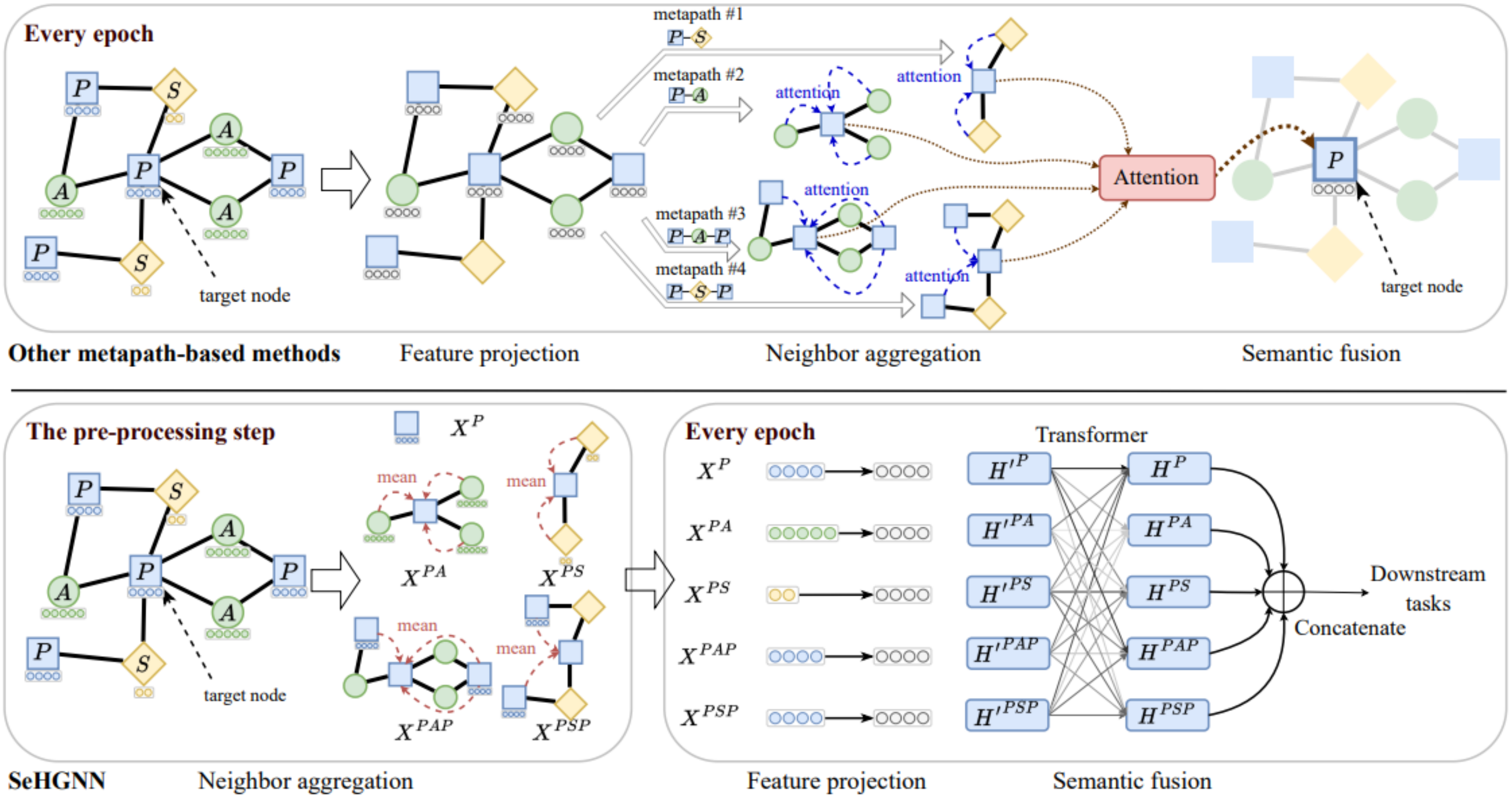
be the raw feature matrix of all nodes belonging to type c, '

$$X^{\mathcal{P}} = \hat{A}_{c,c_1}\hat{A}_{c_1,c_2}\ldots\hat{A}_{c_{l-1},c_l}X^{c_l},$$

where $\mathcal{P} = cc_1c_2\ldots c_l$ is a l-hop metapath, and $\hat{A}_{c_i,c_{i+1}}$ is the row-normalized form of adjacency matrix $A_{c_i,c_{i+1}}$ between node type c_i and c_{i+1}.

$$Y^{\mathcal{P}} = \mathrm{rm\_diag}(\hat{A}^{\mathcal{P}})Y^c,\ \hat{A}^{\mathcal{P}} = \hat{A}_{c,c_1}\hat{A}_{c_1,c_2}\ldots\hat{A}_{c_{l-1},c},$$

Method



$$H'^{\mathcal{P}} = \mathrm{MLP}_{\mathcal{P}}(\mathrm{X}^{\mathcal{P}}).$$

$$q^{\mathcal{P}_i} = W_Q h'^{\mathcal{P}_i},\ k^{\mathcal{P}_i} = W_K h'^{\mathcal{P}_i},\ v^{\mathcal{P}_i} = W_V h'^{\mathcal{P}_i},\ \mathcal{P}_i \in \Phi,$$

$$\alpha_{(\mathcal{P}_i,\mathcal{P}_j)} = \frac{\exp\left(q^{\mathcal{P}_i} \cdot {k^{\mathcal{P}_j}}^T\right)}{\sum_{\mathcal{P}_t \in \Phi} \exp\left(q^{\mathcal{P}_i} \cdot {k^{\mathcal{P}_t}}^T\right)},$$

$$h^{\mathcal{P}_i} = \beta \sum_{\mathcal{P}_j \in \Phi} \alpha_{(\mathcal{P}_i,\mathcal{P}_j)} v^{\mathcal{P}_j} + h'^{\mathcal{P}_i},$$

where W_Q, W_K, W_V, β are trainable parameters shared for all metapaths.

$$\mathrm{Pred} = \mathrm{MLP}([h^{\mathcal{P}_1} || h^{\mathcal{P}_2} || \ldots || h^{\mathcal{P}_{|\Phi|}}]).$$

Experiments

	DBLP		ACM	
	macro-f1	micro-f1	macro-f1	micro-f1
HAN	92.59	93.06	90.30	90.15
HAN*	92.75	93.23	90.61	90.48
HAN†	92.19	92.66	89.78	89.67
HGB	94.15	94.53	93.09	93.03
HGB*	94.20	94.58	93.11	93.05
HGB†	93.77	94.15	92.32	92.27

Table 1: Experiments to analyze the effects of two kinds of attentions. * means removing neighbor attention and † means removing semantic attention.

	DBLP		ACM	
network	macro-f1	micro-f1	macro-f1	micro-f1
(1,)	79.43	80.16	89.81	90.03
(1,1)	85.06	86.69	90.79	90.87
(2,)	88.18	88.83	91.64	91.67
(1,1,1)	88.38	89.37	87.95	88.84
(3,)	93.33	93.72	92.67	92.64
(1,1,1,1)	89.55	90.44	88.62	88.93
(2,2)	91.88	92.35	92.57	92.53
(4)	**93.60**	**94.02**	**92.82**	**92.79**

Table 2: Experiments to analyze the effects of different combinations of the number of layers and the maximum metapath hop. e.g., the structure (1,1,1) means a three-layer network with all metapaths no more than 1 hop in each layer.

Finding 1: Semantic attention is essential while neighbor attention is not necessary.

Finding 2: Models with single-layer structure and long metapaths perform better than those with multi-layers and short metapaths.

Experiments

		DBLP		IMDB		ACM		Freebase	
		macro-f1	micro-f1	macro-f1	micro-f1	macro-f1	micro-f1	macro-f1	micro-f1
1st	RGCN	91.52±0.50	92.07±0.50	58.85±0.26	62.05±0.15	91.55±0.74	91.41±0.75	46.78±0.77	58.33±1.57
	HetGNN	91.76±0.43	92.33±0.41	48.25±0.67	51.16±0.65	85.91±0.25	86.05±0.25	-	-
	HAN	91.67±0.49	92.05±0.62	57.74±0.96	64.63±0.58	90.89±0.43	90.79±0.43	21.31±1.68	54.77±1.40
	MAGNN	93.28±0.51	93.76±0.45	56.49±3.20	64.67±1.67	90.88±0.64	90.77±0.65	-	-
2nd	RSHN	93.34±0.58	93.81±0.55	59.85±3.21	64.22±1.03	90.50±1.51	90.32±1.54	-	-
	HetSANN	78.55±2.42	80.56±1.50	49.47±1.21	57.68±0.44	90.02±0.35	89.91±0.37	-	-
	HGT	93.01±0.23	93.49±0.25	63.00±1.19	67.20±0.57	91.12±0.76	91.00±0.76	29.28±2.52	60.51±1.16
	HGB	94.01±0.24	94.46±0.22	63.53±1.36	67.36±0.57	93.42±0.44	93.35±0.45	47.72±1.48	**66.29±0.45**
3rd	SeHGNN	**95.06±0.17**	**95.42±0.17**	**67.11±0.25**	**69.17±0.43**	**94.05±0.35**	**93.98±0.36**	**51.87±0.86**	65.08±0.66
4th	Variant#1	93.61±0.51	94.08±0.48	64.48±0.45	66.58±0.42	93.06±0.18	92.98±0.18	33.23±1.39	57.60±1.17
	Variant#2	94.66±0.27	95.01±0.24	65.27±0.60	66.68±0.52	93.46±0.43	93.38±0.44	46.82±1.12	64.08±1.43
	Variant#3	94.86±0.14	95.24±0.13	66.63±0.34	68.21±0.32	93.95±0.48	93.87±0.50	50.71±0.44	63.41±0.47
	Variant#4	94.52±0.05	94.93±0.06	64.99±0.54	66.65±0.50	93.88±0.63	93.80±0.64	35.48±1.36	60.03±1.13

Table 3: Experiment results on the four datasets from HGB benchmark, where "-" means that the models run out of memory.

Experiments

	Feature projection	Neighbor aggregation	Semantic fusion	Total
SeHGNN	$O(NKD^2)$	-	$O(NK^2D^2)$	$O(NK^2D^2)$
HAN	$O(NKD^2)$	$O(NK\mathcal{E}_1D^2)$	$O(NKD^2)$	$O(NK\mathcal{E}_1D^2)$
HGB	$O(NLD^2)$	$O(N\mathcal{E}_2D^2)$		$O(N\mathcal{E}_2D^2)$

Table 5: Theoretical complexity of SeHGNN, HAN and HGB in every training mini-batch.

Methods	Validation accuracy	Test accuracy
RGCN	48.35±0.36	47.37±0.48
HGT	49.89±0.47	49.27±0.61
NARS	51.85±0.08	50.88±0.12
SAGN	52.25±0.30	51.17±0.32
GAMLP	53.23±0.23	51.63±0.22
HGT+emb	51.24±0.46	49.82±0.13
NARS+emb	53.72±0.09	52.40±0.16
GAMLP+emb	55.48±0.08	53.96±0.18
SAGN+emb+ms	55.91±0.17	54.40±0.15
GAMLP+emb+ms	57.02±0.41	55.90±0.27
SeHGNN	55.95±0.11	53.99±0.18
SeHGNN+emb	56.56±0.07	54.78±0.17
SeHGNN+ms	58.70±0.08	56.71±0.14
SeHGNN+emb+ms	**59.17±0.09**	**57.19±0.12**

Table 4: Experiment results on the large-scale dataset ogbn-mag compared with other methods on the OGB leaderboard, where "emb" means using extra embeddings and "ms" means using multi-stage training.

Experiments

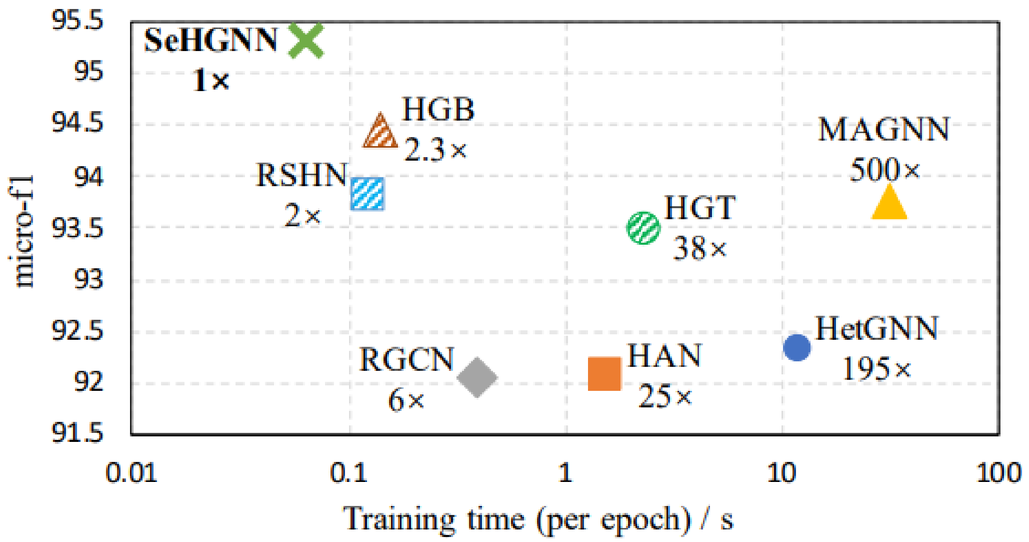


Figure 3: Micro-f1 scores and time consumption of different HGNNs on DBLP dataset. Numbers below model names exhibit the ratio of time consumption relative to SeHGNN. e.g., "6x" below RGCN means RGCN costs 6 times of time.

Thanks