Are we really making much progress? Revisiting, benchmarking, and refining heterogeneous graph neural networks

Qingsong Lv*†, Ming Ding*†, Qiang Liu♣, Yuxiang Chen†, Wenzheng Feng†, Siming He♢, Chang Zhou‡, Jianguo Jiang♣, Yuxiao Dong¶, Jie Tang†§

† Tsinghua University ♣ Chinese Academy of Sciences
‡ Alibaba Group ♢ University of Pennsylvania ¶ Microsoft

June, 2021

1 Preliminaries

2 Motivation

3 Methodology

4 Experiments

5 References

1 Preliminaries

2 Motivation

3 Methodology

4 Experiments

5 References

Heterogeneous Graph

- $G = \{V, E, \phi, \psi\}$
- V: set of nodes; E: set of edges.
- Each node v has a type $\phi(v)$; Each edge e has a type $\psi(e)$.
- Assume $T_v = \{\phi(v) : \forall v \in V\}$ and $T_e = \{\psi(e) : \forall e \in E\}$.
- When $|T_v| = |T_e| = 1$, the graph degenerates into an ordinary homogeneous graph. Otherwise, G is a heterogeneous graph.

Figure 1: Homogeneous Graph and Heterogeneous Graph illustration.

Graph Neural Networks

- GCN: $H^{(l)} = \sigma(\hat{A}H^{(l-1)}W^{(l)})$
- GAT: $\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(a^T[Wh_i \| Wh_j]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(a^T[Wh_i \| Wh_k]\right)\right)}$
- Homogeneous GNN $\rightarrow$ Heterogeneous GNN

Meta-Paths

- A meta-path [1, 2] is a pre-defined node and edge types pattern.
- $\mathcal{P} \triangleq n_1 \xrightarrow{r_1} n_2 \xrightarrow{r_2} \cdots \xrightarrow{r_l} n_{l+1}$, where $r_i \in T_e$ and $n_i \in T_v$.
- For example, "user $\xrightarrow{\text{buy}}$ item $\xleftarrow{\text{buy}}$ user $\xrightarrow{\text{buy}}$ item" is a meta-path, and "user 3 $\xrightarrow{\text{buy}}$ item 1 $\xleftarrow{\text{buy}}$ user 1 $\xrightarrow{\text{buy}}$ item 4" is an instance of the meta-path.

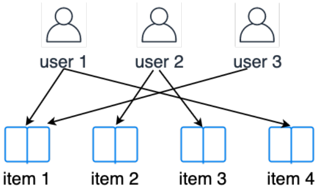

Figure 2: An Example of User-Item Graph.

1 Preliminaries

2 Motivation

3 Methodology

4 Experiments

5 References

Issues with Current HGNN Research

- Experiment settings
  - Improper settings for homogeneous baselines
  - Biased performance reporting for multiple runs
  - Data leakage
- Datasets:
  - Various train/test split and preprocessing steps in different papers (even with a same dataset)
- Pipelines:
  - Various designs for components outside HGNNs

Issues Demonstration

Table 1: Reproduction of Heterogeneous GNNs with simple GCN and GAT as baselines—all reproduction experiments use official codes and the same dataset, settings, hyperparameters as the original paper. The line with star (*) are results reported in the paper, and the lines without star are our reproduction. "-" means the results are not reported in the original paper. We mark the reproduction terms with >1 point gap compared to the reported results by ↑ and ↓. We also keep the standard variance terms above 1.

	HAN [3]		GTN [4]			RSHN [6]			HetGNN [5]			
Dataset	ACM		DBLP	ACM	IMDB	AIFB	MUTAG	BGS	MC (10%)		MC (30%)	
Metric	Macro-F1	Micro-F1	Macro-F1	Macro-F1	Macro-F1	Accuracy	Accuracy	Accuracy	Macro-F1	Micro-F1	Macro-F1	Micro-F1
model*	91.89	91.85	94.18	92.68	60.92	97.22	82.35	93.10	97.8	97.9	98.1	98.2
GCN*	89.31	89.45	87.30	91.60	56.89	-	-	-	-	-	-	-
GAT*	90.55	90.55	93.71	92.33	58.14	91.67	72.06	66.32	96.2	96.3	96.5	96.5
model	90.94	90.96	92.95↓	92.28	57.53±2.22↓	97.22	**82.35**	93.10	97.06	97.11	97.34	97.37
GCN	**92.25**↑	**92.29**↑	91.48↑	92.28	**59.11±1.73**↑	97.22	79.41	96.55	91.88	92.04	95.37	95.57
GAT	92.08↑	92.15↑	**94.18**	**92.49**	58.86±1.73	**100**↑	80.88↑	**100**↑	**98.25**↑	**98.30**↑	**98.42**↑	**98.50**↑

1 Preliminaries

2 Motivation

3 Methodology
Heterogeneous Graph Benchmark (HGB)
A Simple but Strong Baseline (Simple-HGN)

4 Experiments

5 References

1. Preliminaries
2. Motivation
3. Methodology
   - Heterogeneous Graph Benchmark (HGB)
   - A Simple but Strong Baseline (Simple-HGN)
4. Experiments
5. References

HGB

HGB standardizes heterogeneous experiment settings for all HGNNs for fair comparison.

- We collect 11 widely-recognized *medium-scale* datasets on 3 tasks with predefined meta-paths from previous works
- We run all datasets for all methods 5 times and report the average score and standard deviation
- We design a unified pipeline for each task to reveal the ability of HGNN module and eliminate variation from other components

Datasets

Table 2: Statistics of HGB datasets.

Node Classification	#Nodes	#Node Types	#Edges	#Edge Types	Target	#Classes
DBLP	26,128	4	239,566	6	author	4
IMDB	21,420	4	86,642	6	movie	5
ACM	10,942	4	547,872	8	paper	3
Freebase	180,098	8	1,057,688	36	book	7
Link Prediction					Target	
Amazon	10,099	1	148,659	2	product-product	
LastFM	20,612	3	141,521	3	user-artist	
PubMed	63,109	4	244,986	10	disease-disease	

Recommendation	Amazon-book	LastFM	Movielens	Yelp-2018
#Users	70,679	23,566	37,385	45,919
#Items	24,915	48,123	6,182	45,538
#Interactions	846,434	3,034,763	539,300	1,183,610
#Entities	113,487	106,389	24,536	136,499
#Relations	39	9	20	42
#Triplets	2,557,746	464,567	237,155	1,853,704

Pipelines

We use "feature preprocessing → HGNN encoder → downstream decoder" pipeline in HGB, and the whole pipeline is trained in an *end-to-end* fashion.

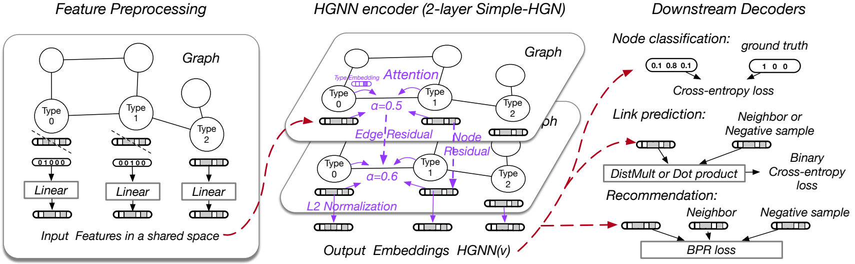

Figure 3: HGB Pipelines

1 Preliminaries

2 Motivation

3 Methodology
Heterogeneous Graph Benchmark (HGB)
A Simple but Strong Baseline (Simple-HGN)

4 Experiments

5 References

Simple-HGN

Simple-HGN uses GAT as backbone, and adding three simple yet effective components:

- Relation-aware attention weight calculation:
$$\hat{\alpha}_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^T[Wh_i \| Wh_j \| W_r r_{\psi(\langle i,j \rangle)}]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\mathrm{LeakyReLU}\left(a^T[Wh_i \| Wh_k \| W_r r_{\psi(\langle i,k \rangle)}]\right)\right)}$$
- Residual connection for nodes edges
- L_2 normalization for output representations

1 Preliminaries

2 Motivation

3 Methodology

4 Experiments

5 References

Node Classification

Table 3: Node classification benchmark. Vacant positions ("-") mean that the models run out of memory on large graphs.

	DBLP		IMDB		ACM		Freebase	
	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
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
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
GTN	93.52±0.55	93.97±0.54	60.47±0.98	65.14±0.45	91.31±0.70	91.20±0.71	-	-
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	-	-
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	-	-
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	-	-
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
GCN	90.84±0.32	91.47±0.34	57.88±1.18	64.82±0.64	92.17±0.24	92.12±0.23	27.84±3.13	60.23±0.92
GAT	93.83±0.27	93.39±0.30	58.94±1.35	64.86±0.43	92.26±0.94	92.19±0.93	40.74±2.58	65.26±0.80
Simple-HGN	**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**

Link Prediction

Table 4: Link prediction benchmark. Vacant positions ("-") are due to lack of meta-paths on those datasets.

	Amazon		LastFM		PubMed	
	ROC-AUC	MRR	ROC-AUC	MRR	ROC-AUC	MRR
RGCN	86.34±0.28	93.92±0.16	57.21±0.09	77.68±0.17	78.29±0.18	90.26±0.24
GATNE	77.39±0.50	92.04±0.36	66.87±0.16	85.93±0.63	63.39±0.65	80.05±0.22
HetGNN	77.74±0.24	91.79±0.03	62.09±0.01	83.56±0.14	73.63±0.01	84.00±0.04
MAGNN	-	-	56.81±0.05	72.93±0.59	-	-
HGT	88.26±2.06	93.87±0.65	54.99±0.28	74.96±1.46	80.12±0.93	90.85±0.33
GCN	92.84±0.34	**97.05±0.12**	59.17±0.31	79.38±0.65	80.48±0.81	90.99±0.56
GAT	91.65±0.80	96.58±0.26	58.56±0.66	77.04±2.11	78.05±1.77	90.02±0.53
Simple-HGN	**93.40±0.62**	96.94±0.29	**67.59±0.23**	**90.81±0.32**	**83.39±0.39**	**92.07±0.26**

Knowledge-aware Recommendation

Table 5: Knowledge-aware recommendation benchmark. GCN and GAT are not included, because they are already very similar to KGCN and KGAT-. (MovieLens dataset is omitted here due to the space constraint.)

	Amazon-Book		LastFM		Yelp-2018	
	recall@20	ndcg@20	recall@20	ndcg@20	recall@20	ndcg@20
KGCN	0.1464±0.0002	0.0769±0.0002	0.0819±0.0002	0.0705±0.0002	0.0683±0.0003	0.0431±0.0003
KGNN-LS	0.1448±0.0003	0.0759±0.0001	0.0806±0.0003	0.0695±0.0002	0.0671±0.0003	0.0422±0.0002
KGAT	0.1507±0.0003	0.0802±0.0004	0.0877±0.0003	0.0749±0.0003	0.0697±0.0002	0.0450±0.0001
KGAT—	0.1486±0.0003	0.0790±0.0002	0.0890±0.0002	0.0762±0.0002	0.0715±0.0001	0.0460±0.0001
Simple-HGN	**0.1587±0.0011**	**0.0854±0.0005**	**0.0917±0.0006**	**0.0797±0.0003**	**0.0732±0.0003**	**0.0466±0.0003**

1. Preliminaries
2. Motivation
3. Methodology
4. Experiments
5. References

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

[2] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu.
Pathsim: Meta path-based top-k similarity search in heterogeneous information networks.
PVLDB, 4(11):992–1003, 2011.

[3] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu.
Heterogeneous graph attention network.
In *WWW'19*, 2019.

[4] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim.
Graph transformer networks.
In *NeurIPS'19*, 2019.

[5] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla.
Heterogeneous graph neural network.
In *KDD'19*, pages 793–803, 2019.

[6] Shichao Zhu, Chuan Zhou, Shirui Pan, Xingquan Zhu, and Bin Wang.
Relation structure-aware heterogeneous graph neural network.
In *ICDM'19*, 2019.

Thank You!