Finding Global Homophily in Graph Neural Networks When Meeting Heterophily

Xiang Li**, Renyu Zhu, Yao Cheng, Weining Qian, **ECNU

Caihua Shan, Dongsheng Li, MSRA

Siqiang Luo, NTU

Introduction

- **Homophilous graphs**
  - Linked nodes tend to share **similar features** or have the **same label**
  - Friendship network, political network, citation network, etc.
- **Heterophilous graphs**
  - Linked nodes tend to have **dissimilar features** or **different labels**
  - Protein structures, ecological food webs, criminal network, etc

GNNs in heterophilous graphs

- *Graph Neural Networks (GNNs)*
  - *A non-linear form of smoothing operation over a node's neighbors*
  - *Works well in homophilous graphs*
  - *But, in heterophilous graphs, errrr…*

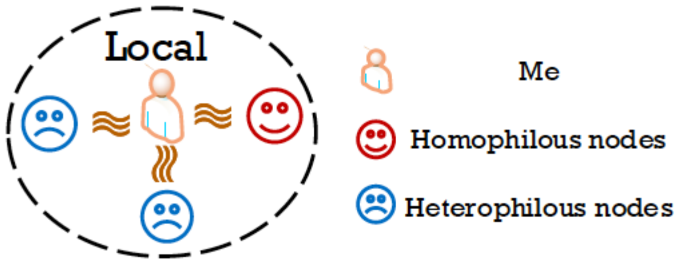


Figure 1. A heterophilous graph example

Nodes with different labels could be given similar representations

SOTAs

- ***Leverage high-pass convolutional filters***

☐ ***High-pass filters:*** *push away a node's embedding from its neighbors'*

☐ ***Low-pass filters:*** *do the opposite!*

- ***Enlarge a node's neighborhood***

☐ ***Jump the locality and find global homophily***

☐ *Set personalized neighborhood size?* ***Big Challenge!***

☐ *Homophilous nodes excluded in the neighborhood are not used!*

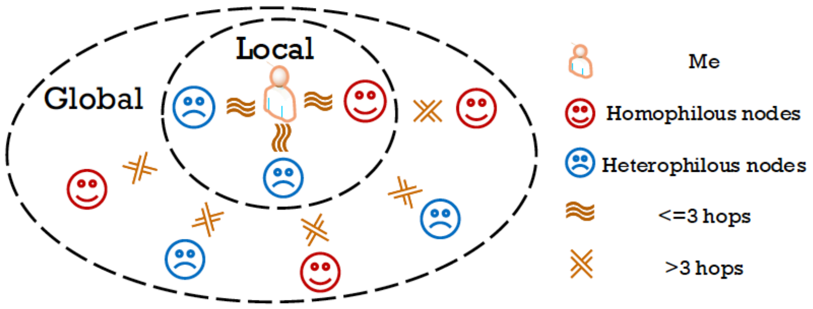


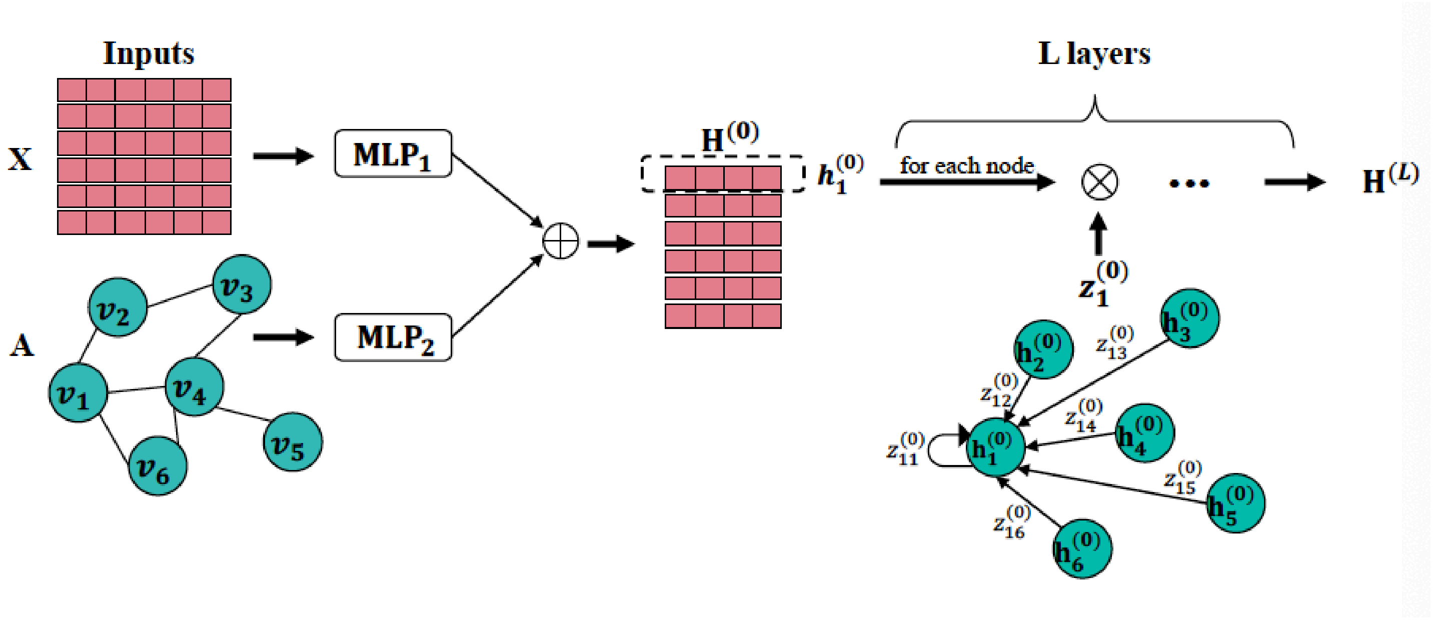
Figure 2. A toy example to show global homophily. All the homophilous nodes express the global homophily of the center user.

A Naïve solution

- **To capture global homophily**
  - Add all the nodes to a node's neighborhood
  - **Side effect 1:** quadratic time complexity for neighborhood aggregation
  - **Side effect 2:** more heterophilous nodes included

Can we find global homophily for a node and develop a GNN model that is both effective and efficient for heterophilous graphs?

GloGNN framework



Inputs
X
A
v_1 v_2 v_3 v_4 v_5 v_6
MLP_1
MLP_2
$\mathbf{H}^{(0)}$
$h_1^{(0)}$
L layers
for each node
$\mathbf{H}^{(L)}$
$z_1^{(0)}$
$h_1^{(0)}$ $h_2^{(0)}$ $h_3^{(0)}$ $h_4^{(0)}$ $h_5^{(0)}$ $h_6^{(0)}$
$z_{11}^{(0)}$ $z_{12}^{(0)}$ $z_{13}^{(0)}$ $z_{14}^{(0)}$ $z_{15}^{(0)}$ $z_{16}^{(0)}$

In each layer, we derive a coefficient matrix, based on which a node's embedding is generated by aggregating information from global nodes

Linear characterization

- ***Represent each object by other objects***
  - coefficient matrix Z, embedding matrix H, noise matrix O
  - linear subspace model, in the l-th layer: $H^{(l)} = Z^{(l)} H^{(l)} + O^{(l)}$
- ***Decoupling neighborhood aggregation and feature transformation***
  - transform feature matrix X and adjacency matrix A into

$$H_X^{(0)} = \mathrm{MLP}_1(X),\ H_A^{(0)} = \mathrm{MLP}_2(A)$$

Initial node embedding $\rightarrow$ $H^{(0)} = (1-\alpha) H_X^{(0)} + \alpha H_A^{(0)}$

  - add skip connection and get

$$H^{(l)} = (1-\gamma) Z^{(l)} H^{(l)} + \gamma H^{(0)} + O^{(l)}$$

Linear characterization

- *Local graph structures are also useful!*

$$\min_{Z^{(l)}} \underbrace{\|H^{(l)}-(1-\gamma)Z^{(l)}H^{(l)}-\gamma H^{(0)}\|_F^2}_{\text{Remove noise}}+\beta_1\|Z^{(l)}\|_F^2+\underbrace{\beta_2\|Z^{(l)}-\sum_{k=1}^{K}\lambda_k\hat{A}^k\|_F^2}_{\text{Local graph structure regularization}}$$

- *A closed-form solution is*

$$Z^{(l)*}=\left[(1-\gamma)H^{(l)}(H^{(l)})^T+\beta_2\sum_{k=1}^{K}\lambda_k\hat{A}^k-\gamma(1-\gamma)\underbrace{H^{(0)}(H^{(l)})^T}_{\text{Quadratic time complexity}}\right]\cdot$$

$$\underbrace{\left[(1-\gamma)^2H^{(l)}(H^{(l)})^T+(\beta_1+\beta_2)I_n\right]^{-1}}_{\text{Cubic time complexity}}$$

$H^{(l)} \in R^{n\times d}$
n is # of nodes in the graph
d is the embedding size

Neighborhood aggregation

- *We write neighborhood aggregation as:*

$$H^{(l+1)} = (1-\gamma)Z^{(l)*}H^{(l)} + \gamma H^{(0)}$$

- *But the **time complexity** is a **huge problem!***
- *Based on the Woodbury formula, we transform:*

$$\boxed{\left[(1-\gamma)^2 H^{(l)}(H^{(l)})^T + (\beta_1+\beta_2)I_n\right]^{-1}}$$

$$= \frac{1}{\beta_1+\beta_2}I_n - \frac{1}{(\beta_1+\beta_2)^2}H^{(l)}\left[\frac{1}{(1-\gamma)^2}I_c + \frac{1}{\beta_1+\beta_2}(H^{(l)})^T H^{(l)}\right]^{-1}(H^{(l)})^T$$

$$Z^{(l)*} = \left[(1-\gamma)H^{(l)}(H^{(l)})^T + \beta_2\sum_{k=1}^{K}\lambda_k \hat{A}^k - \gamma(1-\gamma)H^{(0)}(H^{(l)})^T\right]\cdot$$

$$\boxed{\left[(1-\gamma)^2 H^{(l)}(H^{(l)})^T + (\beta_1+\beta_2)I_n\right]^{-1}}$$

Neighborhood aggregation

- *In the l-th layer, calculate Q and then H*
- *Reorder matrix multiplication from right to left*

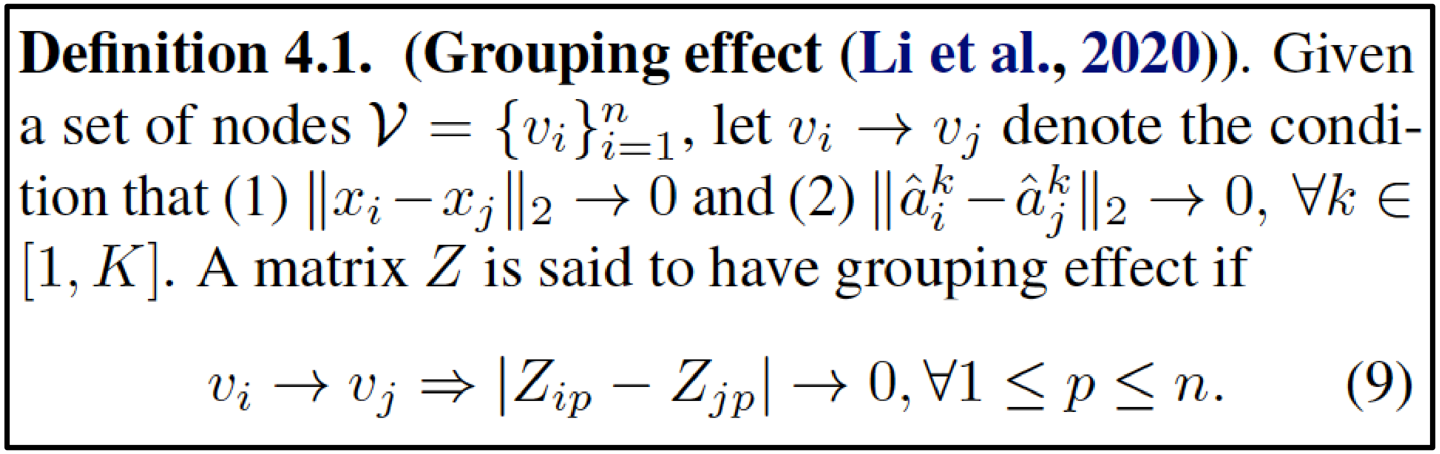
$$Q^{(l+1)} = \frac{1-\gamma}{\beta_1+\beta_2} H^{(l)} - \frac{1-\gamma}{(\beta_1+\beta_2)^2} H^{(l)} \cdot \left[\frac{1}{(1-\gamma)^2} I_c + \frac{1}{\beta_1+\beta_2} (H^{(l)})^T H^{(l)}\right]^{-1} (H^{(l)})^T H^{(l)}$$

$$H^{(l+1)} = (1-\gamma) H^{(l)} (H^{(l)})^T Q^{(l+1)} + \beta_2 \sum_{k=1}^{K} \lambda_k \hat{A}^k Q^{(l+1)} - \gamma(1-\gamma) H^{(0)} (H^{(l)})^T Q^{(l+1)} + \gamma H^{(0)}$$

- ***Derive a linear time complexity!***

Theoretical explanation

- ***Grouping effect***



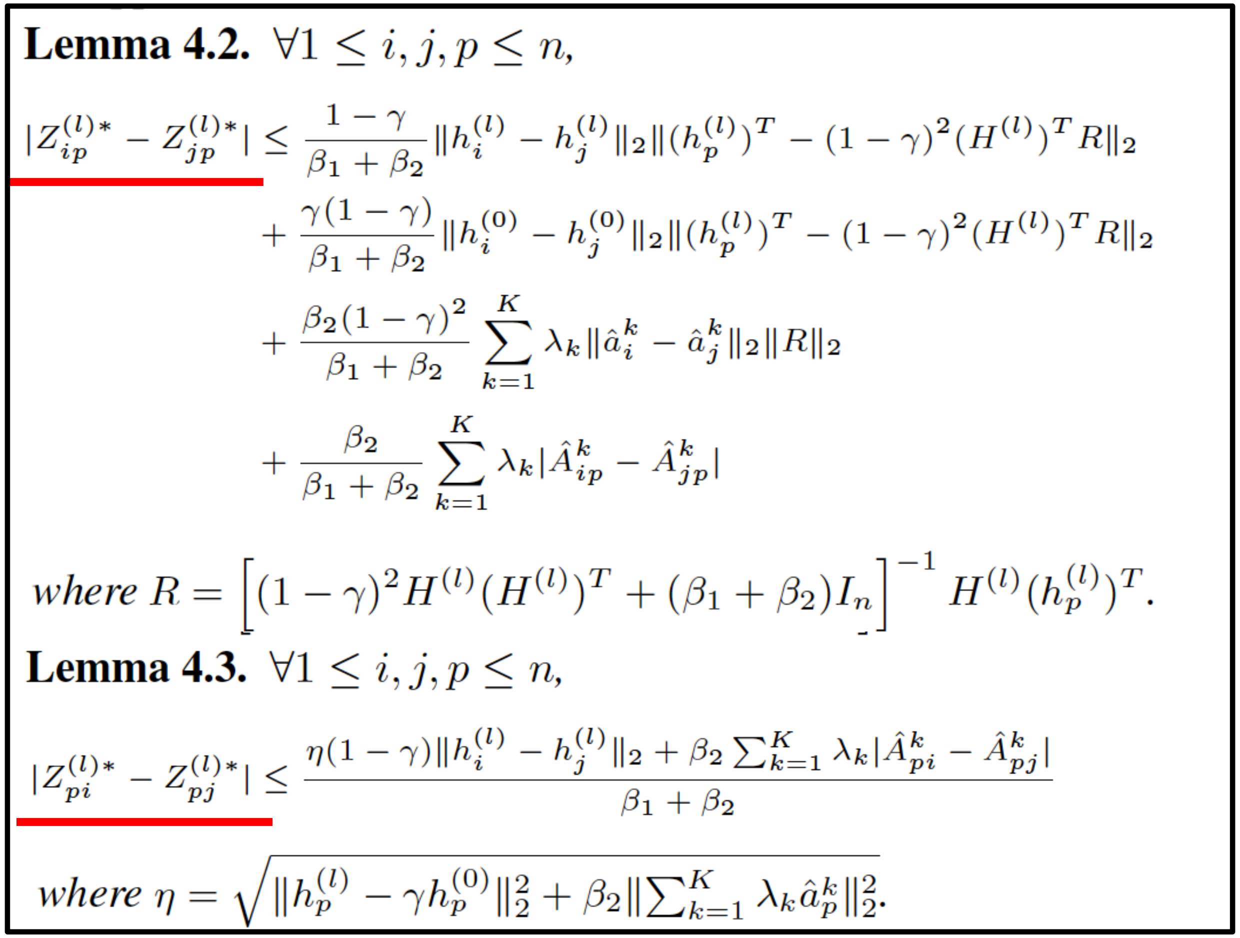
Definition 4.1. **(Grouping effect (Li et al., 2020)).** Given a set of nodes $\mathcal{V} = \{v_i\}_{i=1}^n$, let $v_i \to v_j$ denote the condition that (1) $\|x_i - x_j\|_2 \to 0$ and (2) $\|\hat{a}_i^k - \hat{a}_j^k\|_2 \to 0$, $\forall k \in [1, K]$. A matrix Z is said to have grouping effect if

$$v_i \to v_j \Rightarrow |Z_{ip} - Z_{jp}| \to 0, \forall 1 \leq p \leq n. \tag{9}$$

Condition (1) -> similar feature vectors
Condition (2) -> similar local graph structures

Theoretical explanation

- *For two objects v_i and v_j, we have*



Lemma 4.2. $\forall 1 \leq i, j, p \leq n,$

$$\begin{aligned}\underline{|Z_{ip}^{(l)*} - Z_{jp}^{(l)*}|} \leq\ & \frac{1-\gamma}{\beta_1+\beta_2}\|h_i^{(l)} - h_j^{(l)}\|_2 \|(h_p^{(l)})^T - (1-\gamma)^2 (H^{(l)})^T R\|_2 \\ & + \frac{\gamma(1-\gamma)}{\beta_1+\beta_2}\|h_i^{(0)} - h_j^{(0)}\|_2 \|(h_p^{(l)})^T - (1-\gamma)^2 (H^{(l)})^T R\|_2 \\ & + \frac{\beta_2(1-\gamma)^2}{\beta_1+\beta_2}\sum_{k=1}^{K} \lambda_k \|\hat{a}_i^k - \hat{a}_j^k\|_2 \|R\|_2 \\ & + \frac{\beta_2}{\beta_1+\beta_2}\sum_{k=1}^{K} \lambda_k |\hat{A}_{ip}^k - \hat{A}_{jp}^k|\end{aligned}$$

$$\text{where } R = \left[(1-\gamma)^2 H^{(l)}(H^{(l)})^T + (\beta_1+\beta_2) I_n\right]^{-1} H^{(l)}(h_p^{(l)})^T.$$

Lemma 4.3. $\forall 1 \leq i, j, p \leq n,$

$$\underline{|Z_{pi}^{(l)*} - Z_{pj}^{(l)*}|} \leq \frac{\eta(1-\gamma)\|h_i^{(l)} - h_j^{(l)}\|_2 + \beta_2 \sum_{k=1}^{K} \lambda_k |\hat{A}_{pi}^k - \hat{A}_{pj}^k|}{\beta_1+\beta_2}$$

$$\text{where } \eta = \sqrt{\|h_p^{(l)} - \gamma h_p^{(0)}\|_2^2 + \beta_2 \|\textstyle\sum_{k=1}^{K} \lambda_k \hat{a}_p^k\|_2^2}.$$

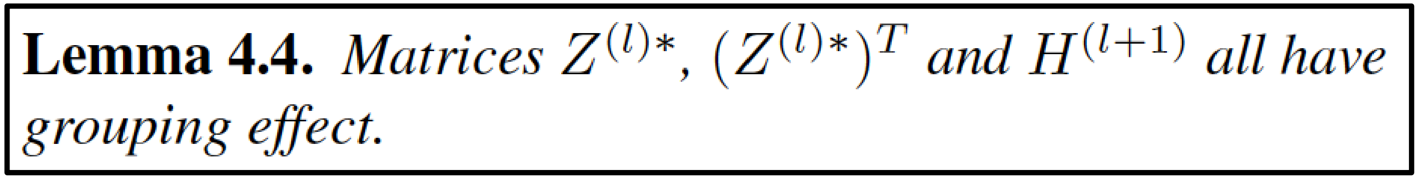
華東師範大學 EAST CHINA NORMAL UNIVERSITY

Microsoft

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Theoretical explanation

- *Finally, we induce*



Lemma 4.4. *Matrices $Z^{(l)*}$, $(Z^{(l)*})^T$ and $H^{(l+1)}$ all have grouping effect.*

- *Based on the lemma, if v_i->v_j, they have*

☐ *Similar coefficient vectors*

☐ *Similar roles in characterizing other nodes*

☐ *Similar embedding vectors*

Experiments

- ***Datasets***

☐ *15 benchmark datasets*

☐ *Varying Sizes (9 small-scale and 6 large-scale)*

☐ *Varying homophilies/heterophilies*

- ***Baselines***

☐ *Standard methods: MLP*

☐ *General GNNs: GCN, GAT, MixHop, GCNII*

☐ *Heterophilous-graph-oriented methods: H_2GCN, WRGAT, GPR-GNN, GGCN, ACM-GCN, LINKX*

Classification accuracy (I)

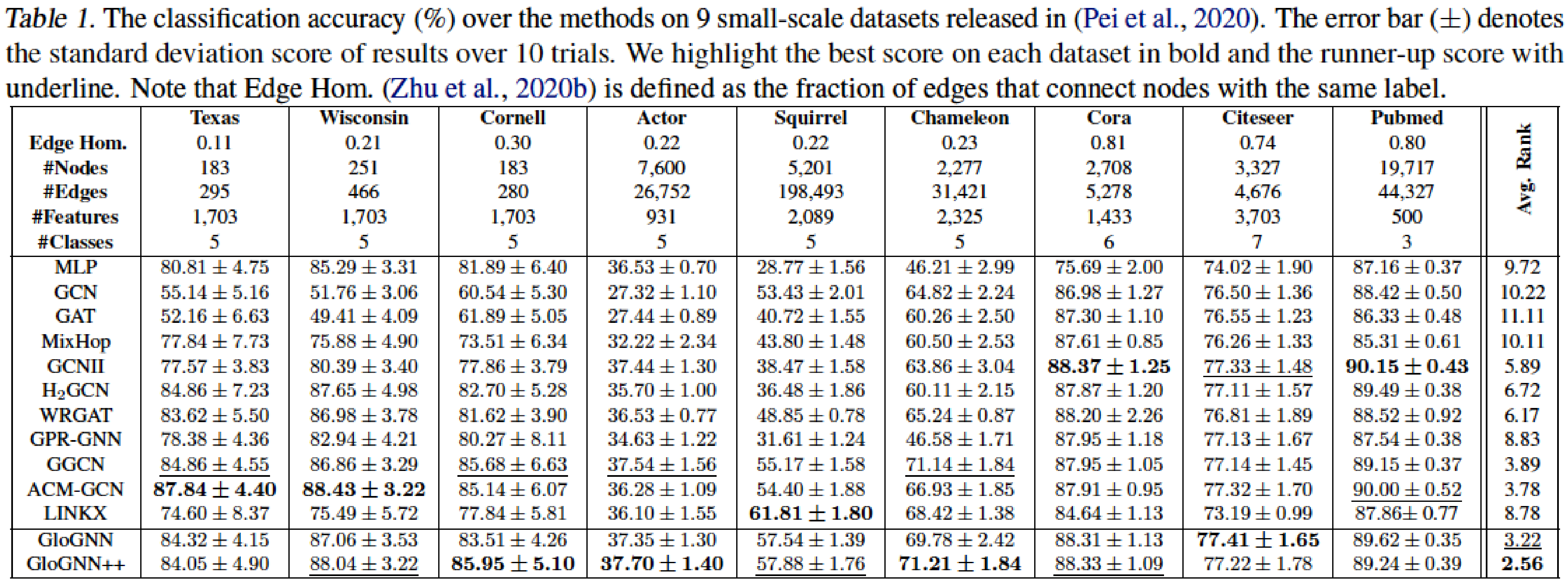


Table 1. The classification accuracy (%) over the methods on 9 small-scale datasets released in (Pei et al., 2020). The error bar (±) denotes the standard deviation score of results over 10 trials. We highlight the best score on each dataset in bold and the runner-up score with underline. Note that Edge Hom. (Zhu et al., 2020b) is defined as the fraction of edges that connect nodes with the same label.

	Texas	**Wisconsin**	**Cornell**	**Actor**	**Squirrel**	**Chameleon**	**Cora**	**Citeseer**	**Pubmed**	**Avg. Rank**
Edge Hom.	0.11	0.21	0.30	0.22	0.22	0.23	0.81	0.74	0.80	
#Nodes	183	251	183	7,600	5,201	2,277	2,708	3,327	19,717	
#Edges	295	466	280	26,752	198,493	31,421	5,278	4,676	44,327	
#Features	1,703	1,703	1,703	931	2,089	2,325	1,433	3,703	500	
#Classes	5	5	5	5	5	5	6	7	3	
MLP	80.81 ± 4.75	85.29 ± 3.31	81.89 ± 6.40	36.53 ± 0.70	28.77 ± 1.56	46.21 ± 2.99	75.69 ± 2.00	74.02 ± 1.90	87.16 ± 0.37	9.72
GCN	55.14 ± 5.16	51.76 ± 3.06	60.54 ± 5.30	27.32 ± 1.10	53.43 ± 2.01	64.82 ± 2.24	86.98 ± 1.27	76.50 ± 1.36	88.42 ± 0.50	10.22
GAT	52.16 ± 6.63	49.41 ± 4.09	61.89 ± 5.05	27.44 ± 0.89	40.72 ± 1.55	60.26 ± 2.50	87.30 ± 1.10	76.55 ± 1.23	86.33 ± 0.48	11.11
MixHop	77.84 ± 7.73	75.88 ± 4.90	73.51 ± 6.34	32.22 ± 2.34	43.80 ± 1.48	60.50 ± 2.53	87.61 ± 0.85	76.26 ± 1.33	85.31 ± 0.61	10.11
GCNII	77.57 ± 3.83	80.39 ± 3.40	77.86 ± 3.79	37.44 ± 1.30	38.47 ± 1.58	63.86 ± 3.04	**88.37 ± 1.25**	77.33 ± 1.48	**90.15 ± 0.43**	5.89
H_2GCN	84.86 ± 7.23	87.65 ± 4.98	82.70 ± 5.28	35.70 ± 1.00	36.48 ± 1.86	60.11 ± 2.15	87.87 ± 1.20	77.11 ± 1.57	89.49 ± 0.38	6.72
WRGAT	83.62 ± 5.50	86.98 ± 3.78	81.62 ± 3.90	36.53 ± 0.77	48.85 ± 0.78	65.24 ± 0.87	88.20 ± 2.26	76.81 ± 1.89	88.52 ± 0.92	6.17
GPR-GNN	78.38 ± 4.36	82.94 ± 4.21	80.27 ± 8.11	34.63 ± 1.22	31.61 ± 1.24	46.58 ± 1.71	87.95 ± 1.18	77.13 ± 1.67	87.54 ± 0.38	8.83
GGCN	84.86 ± 4.55	86.86 ± 3.29	85.68 ± 6.63	37.54 ± 1.56	55.17 ± 1.58	71.14 ± 1.84	87.95 ± 1.05	77.14 ± 1.45	89.15 ± 0.37	3.89
ACM-GCN	**87.84 ± 4.40**	**88.43 ± 3.22**	85.14 ± 6.07	36.28 ± 1.09	54.40 ± 1.88	66.93 ± 1.85	87.91 ± 0.95	77.32 ± 1.70	90.00 ± 0.52	3.78
LINKX	74.60 ± 8.37	75.49 ± 5.72	77.84 ± 5.81	36.10 ± 1.55	**61.81 ± 1.80**	68.42 ± 1.38	84.64 ± 1.13	73.19 ± 0.99	87.86± 0.77	8.78
GloGNN	84.32 ± 4.15	87.06 ± 3.53	83.51 ± 4.26	37.35 ± 1.30	57.54 ± 1.39	69.78 ± 2.42	88.31 ± 1.13	**77.41 ± 1.65**	89.62 ± 0.35	3.22
GloGNN++	84.05 ± 4.90	88.04 ± 3.22	**85.95 ± 5.10**	**37.70 ± 1.40**	57.88 ± 1.76	**71.21 ± 1.84**	88.33 ± 1.09	77.22 ± 1.78	89.24 ± 0.39	**2.56**

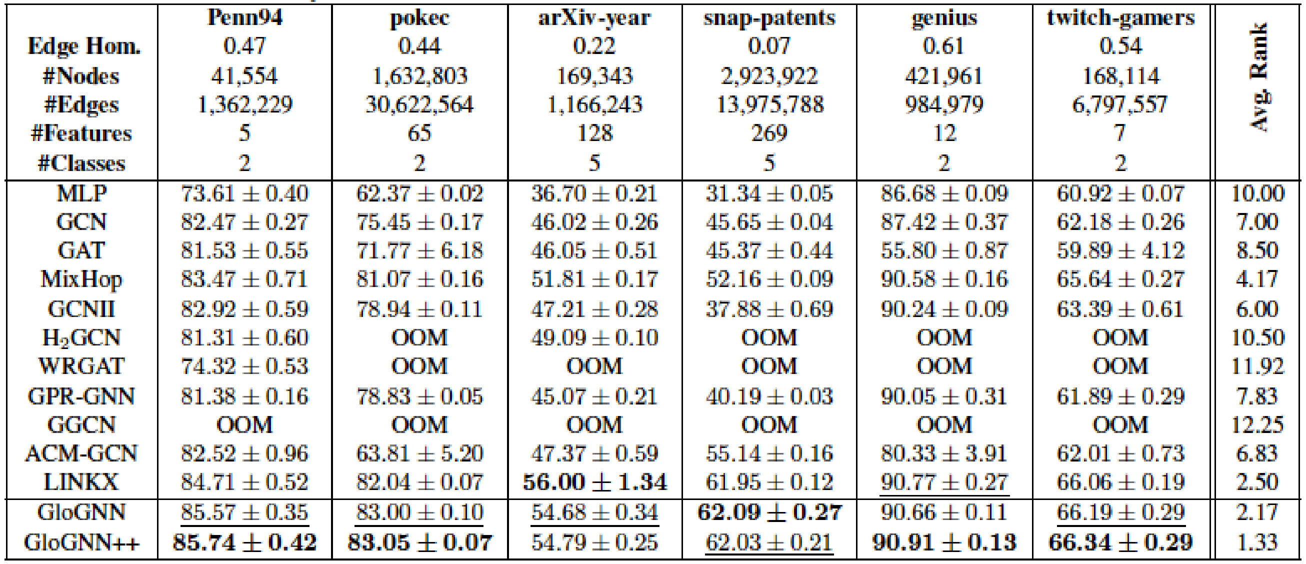
Standard method: MLP

General GNNs: GCN, GAT, MixHop, GCNII

Heterophilous-graph-oriented methods: H_2GCN, WRGAT, GPR-GNN, GGCN, ACM-GCN, LINKX

Classification accuracy (II)

Table 2. The classification results (%) over the methods on 6 large-scale datasets released in (Lim et al., 2021). Note that we compare the AUC score on genius as in (Lim et al., 2021). For other datasets, we show the classification accuracy. The error bar (±) denotes the standard deviation score of results over 5 trials. We highlight the best score on each dataset in bold and the runner-up score with underline. Note that OOM refers to the out-of-memory error.

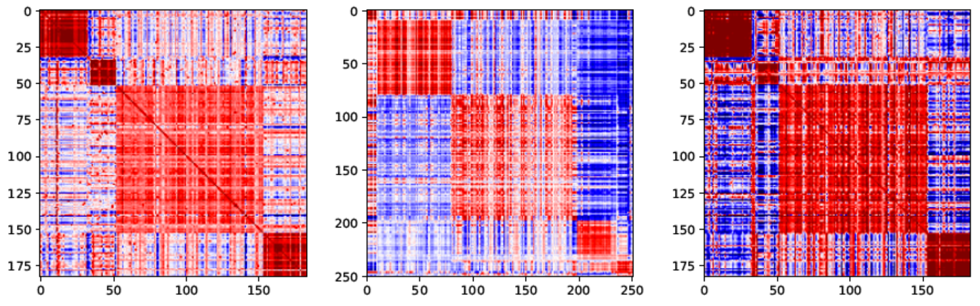


	Penn94	pokec	arXiv-year	snap-patents	genius	twitch-gamers	Avg. Rank
Edge Hom.	0.47	0.44	0.22	0.07	0.61	0.54	
#Nodes	41,554	1,632,803	169,343	2,923,922	421,961	168,114	
#Edges	1,362,229	30,622,564	1,166,243	13,975,788	984,979	6,797,557	
#Features	5	65	128	269	12	7	
#Classes	2	2	5	5	2	2	
MLP	73.61 ± 0.40	62.37 ± 0.02	36.70 ± 0.21	31.34 ± 0.05	86.68 ± 0.09	60.92 ± 0.07	10.00
GCN	82.47 ± 0.27	75.45 ± 0.17	46.02 ± 0.26	45.65 ± 0.04	87.42 ± 0.37	62.18 ± 0.26	7.00
GAT	81.53 ± 0.55	71.77 ± 6.18	46.05 ± 0.51	45.37 ± 0.44	55.80 ± 0.87	59.89 ± 4.12	8.50
MixHop	83.47 ± 0.71	81.07 ± 0.16	51.81 ± 0.17	52.16 ± 0.09	90.58 ± 0.16	65.64 ± 0.27	4.17
GCNII	82.92 ± 0.59	78.94 ± 0.11	47.21 ± 0.28	37.88 ± 0.69	90.24 ± 0.09	63.39 ± 0.61	6.00
H_2GCN	81.31 ± 0.60	OOM	49.09 ± 0.10	OOM	OOM	OOM	10.50
WRGAT	74.32 ± 0.53	OOM	OOM	OOM	OOM	OOM	11.92
GPR-GNN	81.38 ± 0.16	78.83 ± 0.05	45.07 ± 0.21	40.19 ± 0.03	90.05 ± 0.31	61.89 ± 0.29	7.83
GGCN	OOM	OOM	OOM	OOM	OOM	OOM	12.25
ACM-GCN	82.52 ± 0.96	63.81 ± 5.20	47.37 ± 0.59	55.14 ± 0.16	80.33 ± 3.91	62.01 ± 0.73	6.83
LINKX	84.71 ± 0.52	82.04 ± 0.07	**56.00 ± 1.34**	61.95 ± 0.12	$\underline{90.77 \pm 0.27}$	66.06 ± 0.19	2.50
GloGNN	$\underline{85.57 \pm 0.35}$	$\underline{83.00 \pm 0.10}$	$\underline{54.68 \pm 0.34}$	**62.09 ± 0.27**	90.66 ± 0.11	$\underline{66.19 \pm 0.29}$	2.17
GloGNN++	**85.74 ± 0.42**	**83.05 ± 0.07**	54.79 ± 0.25	$\underline{62.03 \pm 0.21}$	**90.91 ± 0.13**	**66.34 ± 0.29**	1.33

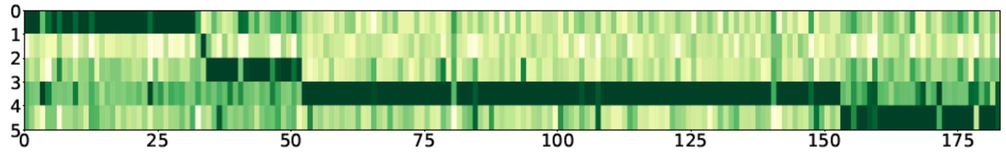
Standard method: MLP
General GNNs: GCN, GAT, MixHop, GCNII
Heterophilous-graph-oriented methods: H_2GCN, WRGAT, GPR-GNN, GGCN, ACM-GCN, LINKX

Grouping effect

- *The grouping effect of coefficient matrix Z: block-diagonal property*

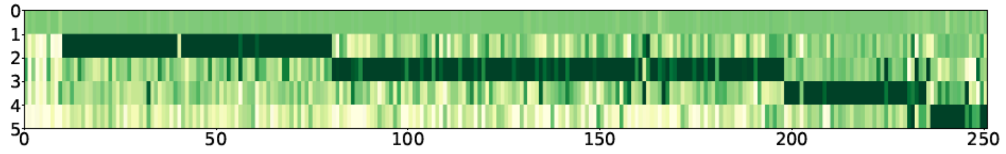


(a) Texas (b) Wisconsin (c) Cornell

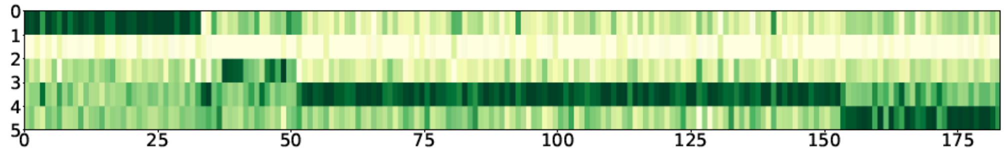


(d) Texas

- *The grouping effect of embedding matrix H: nodes in the same class have similar embedding vectors*



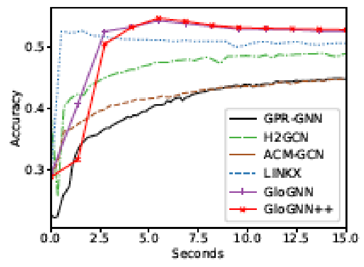
(e) Wisconsin



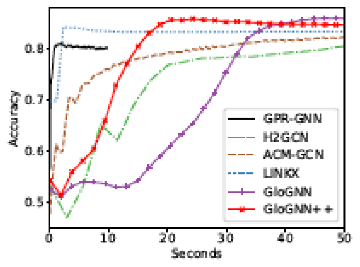
(f) Cornell

Efficiency study

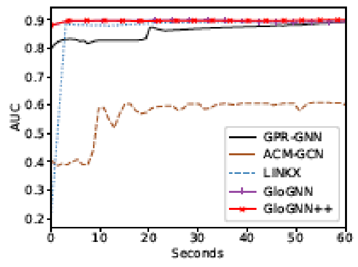
- *LINKX (– -) is simply based on MLP and runs fast*
- *GPR-GNN (—) runs fast but performs poorly*
- *GloGNN (—) converges fast to the best/runner-up results*



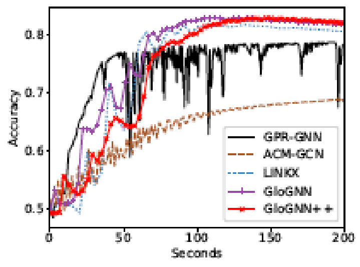
(a) arXiv-year



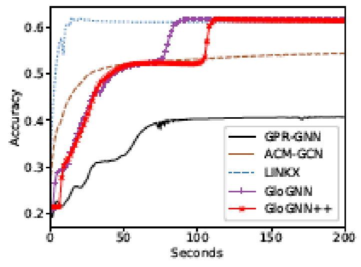
(b) Penn94



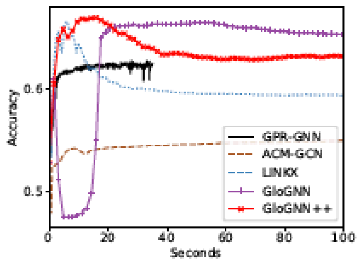
(c) genius



(d) pokec



(e) snap-patents

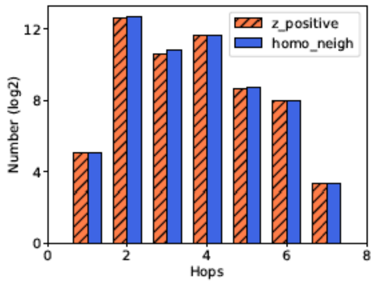


(f) twitch-gamers

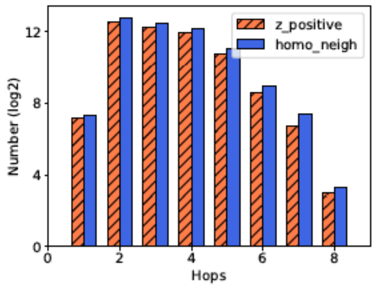
華東師範大學 EAST CHINA NORMAL UNIVERSITY   Microsoft   NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Global homophily

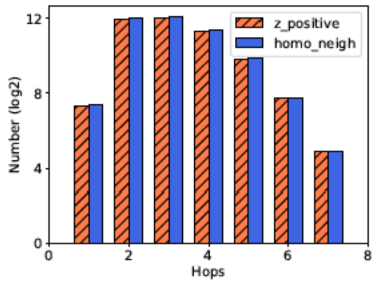
- ***# of k-hop neighbors with the same label***
- ***# of positive Z values***



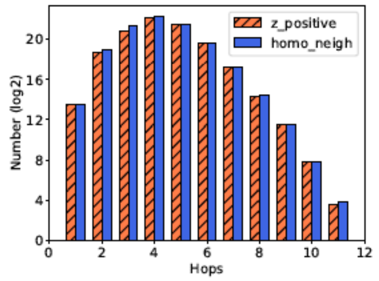
(a) Texas



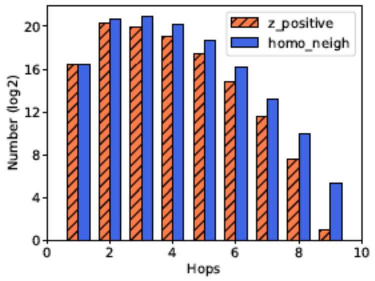
(b) Wisconsin



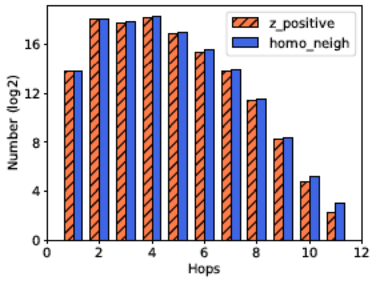
(c) Cornell



(d) Actor



(e) Squirrel



(f) Chameleon

華東師範大學 EAST CHINA NORMAL UNIVERSITY Microsoft NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Conclusion

- *We proposed two effective and efficient GNN models: GloGNN and GloGNN++*
- *We theoretically proved the effectiveness of the models*
- *We automatically combined low-pass and high-pass convolutional filters in neighborhood aggregation*
- *We showed the superiority of our models against 11 other competitors on 15 benchmark datasets*

Thank you!