Compressing Deep Graph Neural Networks via Adversarial Knowledge Distillation

Huarui He
CAS Key Laboratory of Technology in GIPAS,
University of Science and Technology of China
Hefei, China
huaruihe@mail.ustc.edu.cn

Jie Wang*
jiewangx@ustc.edu.cn
CAS Key Laboratory of Technology in GIPAS,
University of Science and Technology of China
Institute of Artificial Intelligence,
Hefei Comprehensive National Science Center
Hefei, China

Zhanqiu Zhang
CAS Key Laboratory of Technology in GIPAS,
University of Science and Technology of China
Hefei, China
zqzhang@mail.ustc.edu.cn

Feng Wu
CAS Key Laboratory of Technology in GIPAS,
University of Science and Technology of China
Hefei, China
fengwu@ustc.edu.cn

KDD-2022

gesis Leibniz-Institut für Sozialwissenschaften

Forschungszentrum · Research Center L3S

Reported by Dongdong Hu

Introduction

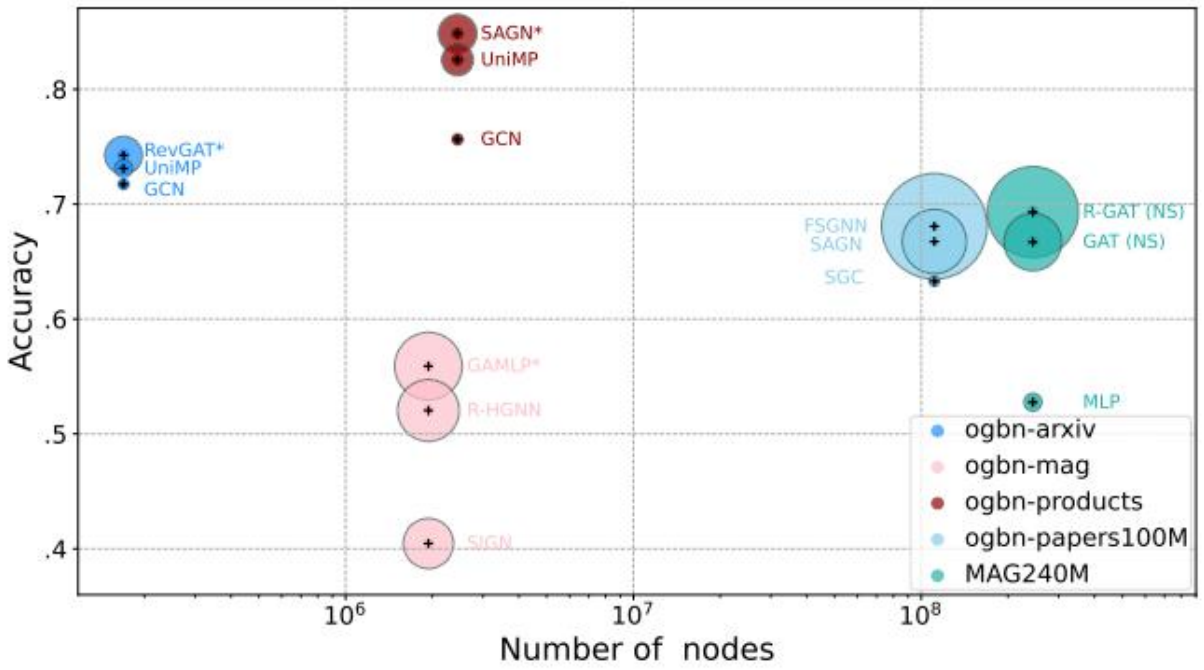

Figure 1: Node classification accuracy v.s. graph size. Each bubble's area is proportional to the number of parameters of a model. Model name with * means the variant. The statistics are collected from OGB leaderboards.

the over-stacked architecture of deep graph models makes it difficult to deploy and rapidly test on mobile or embedded systems.

using the same distance for graphs of various structures may be unfit, and the optimal distance formulation is hard to determine.

Method

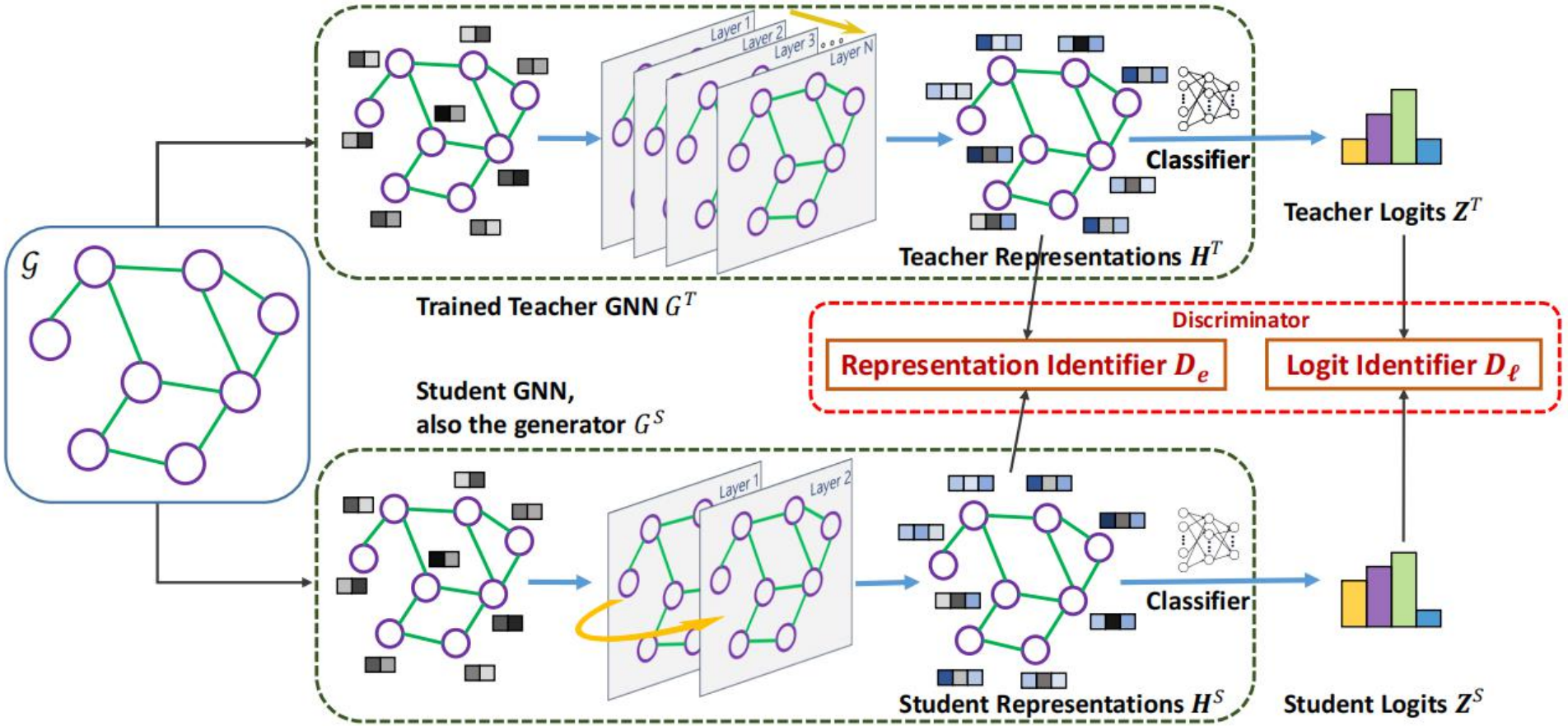

Figure 2: Illustration of the proposed adversarial knowledge distillation framework GraphAKD.

Method

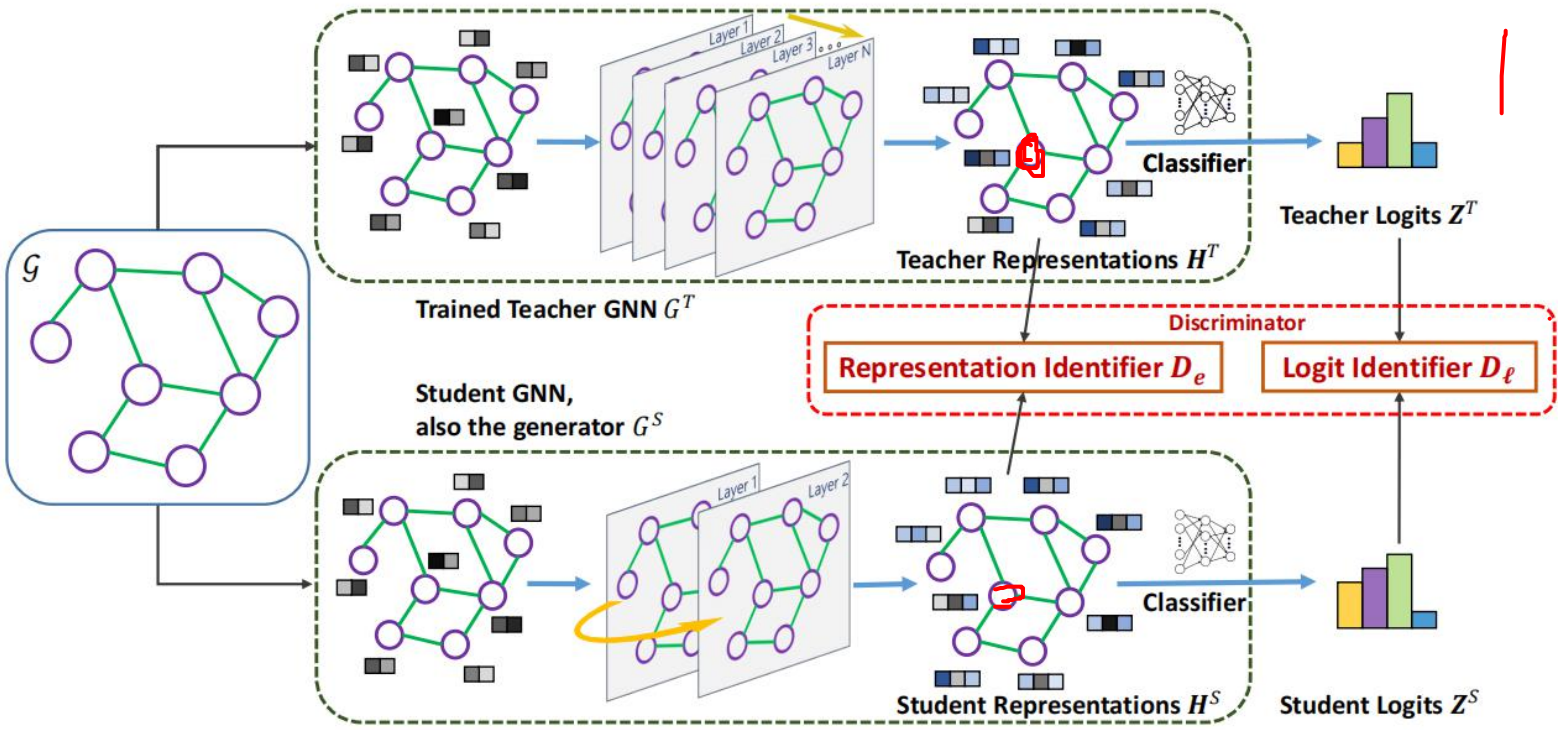

Figure 2: Illustration of the proposed adversarial knowledge distillation framework GraphAKD.

$$\mathbf{H}^T \in \mathbb{R}^{|\mathcal{V}| \times d^T}$$

$$\mathbf{H}^S \in \mathbb{R}^{|\mathcal{V}| \times d^S}$$

$$\mathbf{s}^T \in \mathbb{R}^{d^T} \qquad \mathbf{s}^S \in \mathbb{R}^{d^S}$$

$$d^S = d^T = d.$$

D_e $\{\mathbf{h}_v, \mathbf{h}_u\}$ $\{\mathbf{h}_v, \mathbf{s}\}$: "Real/Fake"

$$D_e^{local}(\mathbf{h}_v^T, \mathbf{h}_u^T) = \langle \mathbf{h}_v^T, \mathbf{W}^{local}\mathbf{h}_u^T \rangle \in [0,1],$$

$$D_e^{local}(\mathbf{h}_v^S, \mathbf{h}_u^S) = \langle \mathbf{h}_v^S, \mathbf{W}^{local}\mathbf{h}_u^S \rangle \in [0,1], \quad \forall\, (v,u) \in \mathcal{E},$$

$$D_e^{global}(\mathbf{h}_v^{T/S}, \mathbf{s}_{\mathcal{G}}^T) = \langle \mathbf{h}_v^{T/S}, \mathbf{W}^{global}\mathbf{s}_{\mathcal{G}}^T \rangle \in [0,1],$$

$$D_e^{global}(\mathbf{h}_v^{T/S}, \mathbf{s}_{\mathcal{G}}^S) = \langle \mathbf{h}_v^{T/S}, \mathbf{W}^{global}\mathbf{s}_{\mathcal{G}}^S \rangle \in [0,1], \quad \forall\, v \in \mathcal{V} \subset \mathcal{G},$$

where $\mathbf{W}^{local}$ and $\mathbf{W}^{global}$ are learnable diagonal matrices.

Method

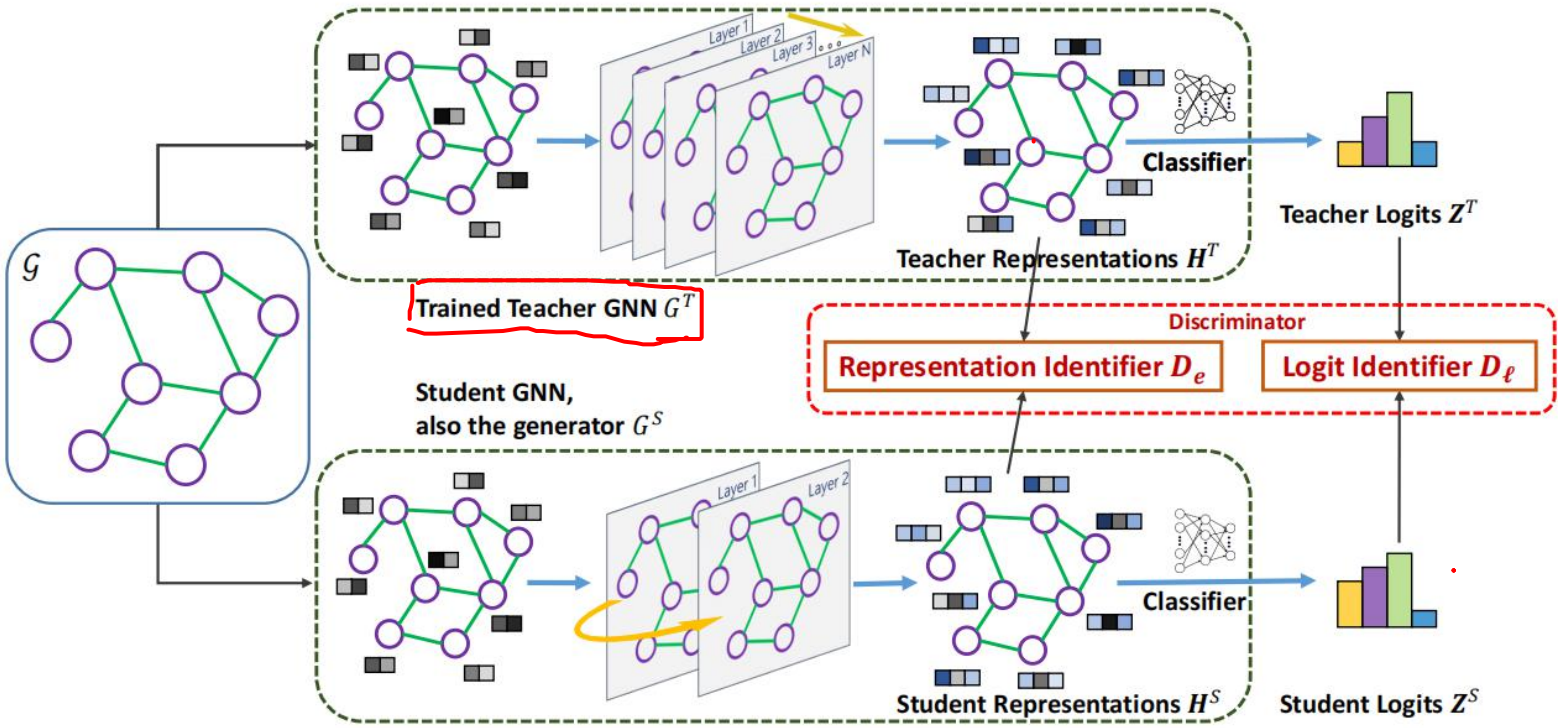

Figure 2: Illustration of the proposed adversarial knowledge distillation framework GraphAKD.

The Representation Identifier

$$\min_{G^S} \max_{D_e} \mathcal{J}^{local} + \mathcal{J}^{global}, \tag{1}$$

$\mathcal{J}^{local}$

$$\frac{1}{|\mathcal{E}|} \sum_{(v,u)\in\mathcal{E}} \Big(\log \mathrm{P}(\mathrm{Real} \mid D_e^l(\mathbf{h}_v^T, \mathbf{h}_u^T)) + \log \mathrm{P}(\mathrm{Fake} \mid D_e^l(\mathbf{h}_v^S, \mathbf{h}_u^S)) \Big),$$

$\mathcal{J}^{global}$

$$\frac{1}{2|\mathcal{V}|} \sum_{v\in\mathcal{V}} \Big(\log \mathrm{P}(\mathrm{Real} \mid D_e^g(\mathbf{h}_v^T, \mathbf{s}_{\mathcal{G}}^T)) + \log \mathrm{P}(\mathrm{Fake} \mid D_e^g(\mathbf{h}_v^S, \mathbf{s}_{\mathcal{G}}^T)) + \log \mathrm{P}(\mathrm{Real} \mid D_e^g(\mathbf{h}_v^S, \mathbf{s}_{\mathcal{G}}^S)) + \log \mathrm{P}(\mathrm{Fake} \mid D_e^g(\mathbf{h}_v^T, \mathbf{s}_{\mathcal{G}}^S)) \Big),$$

if $\mathbf{W}^{local} = \mathbf{I}$ and $\widehat{\mathbf{h}}_v = \mathbf{h}_v / \|\mathbf{h}_v\|_2$, then

$$\langle \widehat{\mathbf{h}}_v, \mathbf{W}^{local} \widehat{\mathbf{h}}_u \rangle = \mathrm{cosine\_sim}(\mathbf{h}_v, \mathbf{h}_u), \ \forall\ (v,u) \in \mathcal{E}.$$

Method

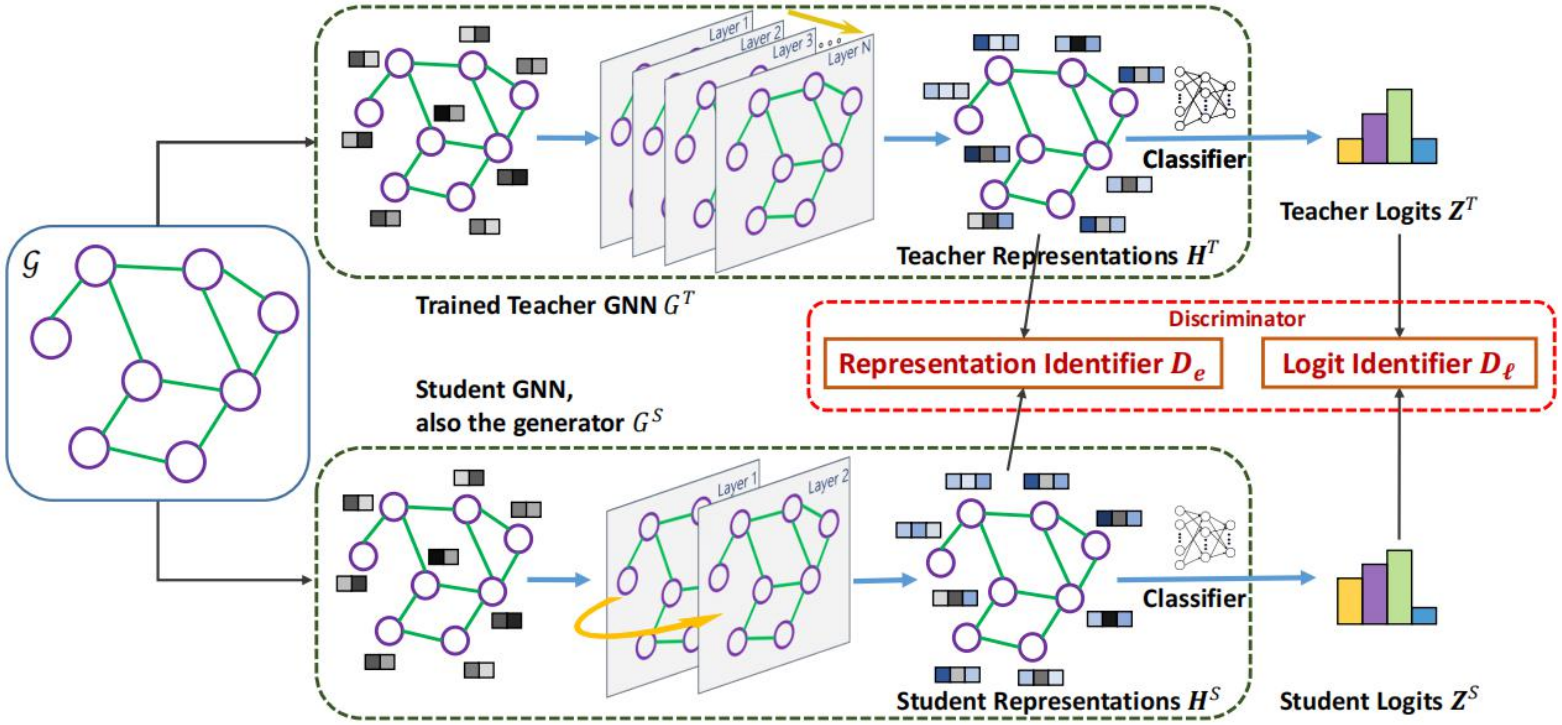

Figure 2: Illustration of the proposed adversarial knowledge distillation framework GraphAKD.

The Logit Identifier

$$\max_{D} \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \Big(\log \mathrm{P}(\mathrm{Real} \mid D(\mathbf{z}_v^T)) + \log \mathrm{P}(\mathrm{Fake} \mid D(\mathbf{z}_v^S)) \Big). \quad (2)$$

As Xu et al. [52] pointed out that the plain version is slow and unstable,

$$\max_{D_\ell} \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \Big(\log \mathrm{P}(\mathrm{Real} \mid D_\ell(\mathbf{z}_v^T)) + \log \mathrm{P}(\mathrm{Fake} \mid D_\ell(\mathbf{z}_v^S)) + \log \mathrm{P}(y_v \mid D_\ell(\mathbf{z}_v^T)) + \log \mathrm{P}(y_v \mid D_\ell(\mathbf{z}_v^S)) \Big). \quad (3)$$

$$\min_{G^S} \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \Big(\log \mathrm{P}(\mathrm{Real} \mid D_\ell(\mathbf{z}_v^T)) + \log \mathrm{P}(\mathrm{Fake} \mid D_\ell(\mathbf{z}_v^S)) - \Big[\log \mathrm{P}(y_v \mid D_\ell(\mathbf{z}_v^T)) + \log \mathrm{P}(y_v \mid D_\ell(\mathbf{z}_v^S)) \Big] + \|\mathbf{z}_v^S - \mathbf{z}_v^T\|_1 \Big). \quad (4)$$

we use an MLP with residual connections as our logit identifier D_ℓ

Experiments

Table 2: Statistics of the eight node classification benchmarks.

Datasets	#Nodes	#Edges	#Feat.	Data Split
Cora [3, 34]	2,708	5,429	1,433	140/500/1K
CiteSeer [40]	3,327	4,732	3,703	120/500/1K
PubMed [35]	19,717	44,338	500	60/500/1K
Flickr [33, 58]	89,250	899,756	500	44K/22K/22K
Arxiv [23]	169,343	1,166,243	128	90K/29K/48K
Reddit [19, 58]	232,965	23,213,838	602	153K/23K/55K
Yelp [58]	716,847	13,954,819	300	537K/107K/71K
Products [23]	2,449,029	61,859,140	100	196K/39K/2M

Compared to existing knowledge distillation models for GNNs [53, 59], the logit identifier relaxes the rigid coupling between student and teacher. Besides, the adversarial training approach relieves the pain for hand-engineering the loss.

Method

Table 3: Performance on Node Classification (metric: F1-micro (%)). "O. Perf." and "R. Perf." refer to performance reported in original papers and reproduced by our own, respectively. Higher of these two columns are underlined. "Perf. Impv." and "#Params Decr." refer to the absolute improvement of student performance (w.r.t. the underlined results) and the relative decrease of teacher parameters, respectively. Results of previous work are mainly taken from [58], [12], and OGB Leaderboards. We report the average performance and std. across 10 random seeds.

	Teacher			Vanilla Student			Student trained with GraphAKD			
Datasets	Model	Perf.	#Params	Model	O. Perf.	R. Perf.	Perf.	#Params	Perf. Impv. (%)	#Params Decr.
Cora	GCNII	85.5	616,519	GCN	81.5	78.3 ±0.9	83.6 ±0.8	96,633	2.1	84.3%
CiteSeer	GCNII	73.4	5,144,070	GCN	71.1	68.6 ±1.1	72.9 ±0.4	1,016.156	1.8	80.2%
PubMed	GCNII	80.3	1,177,603	GCN	79.0	78.1 ±1.0	81.3 ±0.4	195,357	2.3	83.4%
Flickr	GCNII	56.20	1,182,727	GCN	49.20	49.63 ±1.19	52.95 ±0.24	196,473	3.32	83.4%
Arxiv	GCNII	72.74	2,148,648	GCN	71.74	71.43 ±0.13	73.05 ±0.22	242,426	1.31	88.7%
Reddit	GCNII	96.77	691,241	GCN	93.30	94.12 ±0.04	95.15 ±0.02	234,655	1.03	66.1%
Yelp	GCNII	65.14	2,306,660	Cluster-GCN	59.15	59.63 ±0.51	60.63 ±0.42	431,950	1.00	81.3%
Products	GAMLP	84.59	3,335,831	Cluster-GCN	76.21	74.99 ±0.76	81.45 ±0.47	682,449	5.24	79.5%

Experiments

Table 4: Comparison with other distillation algorithms.

Datasets	Student	KD [21]	FitNet [37]	LSP [55]	GraphAKD
Cora	81.5	83.2	82.4	81.7	**83.6**
CiteSeer	71.1	71.4	71.6	68.8	**72.9**
PubMed	79.0	80.3	**81.3**	80.8	**81.3**
Flickr	49.20	50.58	50.69	50.02	**52.95**
Arxiv	71.74	73.03	71.83	OOM	**73.05**
Reddit	93.30	94.01	94.99	OOM	**95.15**
Yelp	59.15	59.14	59.92	49.24	**60.63**
Products	76.21	79.19	76.57	70.86	**81.45**

Experiments

Table 5: Graph classification on Molhiv [23] (metric: ROC-AUC (%)) and Molpcba [23] (metric: AP (%)). Results of teacher and student are taken from OGB Leaderboards. We report the average performance and std. across 10 random seeds.

Dataset	Molhiv		Molpcba	
Teacher	HIG with DeeperGCN		HIG with Graphormer	
Student	GCN	GIN	GCN	GIN
Teacher	84.03 ±0.21	84.03 ±0.21	31.67 ±0.34	31.67 ±0.34
Student	76.06 ±0.97	75.58 ±1.40	20.20 ±0.24	22.66 ±0.28
KD [21]	74.98 ±1.09	75.08 ±1.76	21.35 ±0.42	23.56 ±0.16
FitNet [37]	79.05 ±0.96	77.93 ±0.61	21.25 ±0.91	23.74 ±0.19
GraphAKD	**79.46** ±0.97	**79.16** ±1.50	**22.56** ±0.23	**25.85** ±0.17

Experiments

Table 6: Comparison of efficiency between student GNNs and the corresponding teachers.

	#Params		GPU Memory		Inference time	
Datasets	Teacher	Student	Teacher	Student	Teacher	Student
Cora	0.6M	0.1M	0.22G	0.03G	40.3ms	4.1ms
PubMed	1.2M	0.2M	1.23G	0.33G	57.3ms	5.7ms
Flickr	1.2M	0.2M	2.79G	1.49G	309.7ms	11.9ms
Yelp	2.3M	0.4M	6.28G	4.73G	3.0s	1.5s
Products	3.3M	0.7M	6.25G	6.20G	16.1s	7.0s

Table 7: Ablation studies on the impacts of each identifier.

Datasets	Cora	PubMed	Flickr	Yelp	Products	Molhiv
Teacher	85.5	80.3	56.20	65.14	84.59	84.03
Student	81.5	79.0	49.20	59.15	76.21	75.58
Only D_e	82.9	80.6	52.20	59.63	81.13	78.28
Only D_ℓ	82.3	81.0	52.52	60.03	79.76	78.09
GraphAKD	**83.6**	**81.3**	**52.95**	**60.63**	**81.45**	**79.16**

Experiments

(a) Vanilla GCN (b) Student GCN trained with GraphAKD (c) Teacher GCNII

Figure 3: t-SNE embeddings of the nodes in the Cora dataset from the vanilla GCN embeddings (left), embeddings from the student GCN that trained by GraphAKD (middle), and GCNII (right). The Silhouette scores [38] of the embeddings learned by three models are 0.2196, 0.2638, and 0.3033, respectively.

Thanks