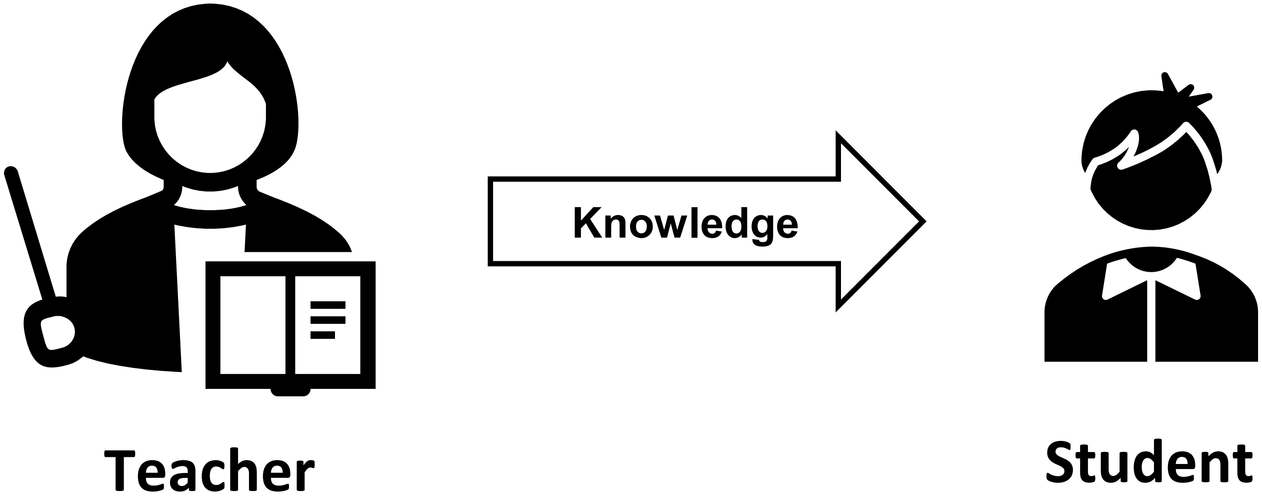
Knowledge Distillation on Graphs: A survey

Yijun Tian[1], Shichao Pei[1], Xialiang Zhang[1],
Chuxu Zhang[2], Nitesh V. Chawla[1]
[1]University of Notre Dame, [2]Brandeis University

DMAIS@CAU
Joohee Cho

What is a Knowledge Distillation?

- ☐ Technique to transfer knowledge from a teacher model to a student model

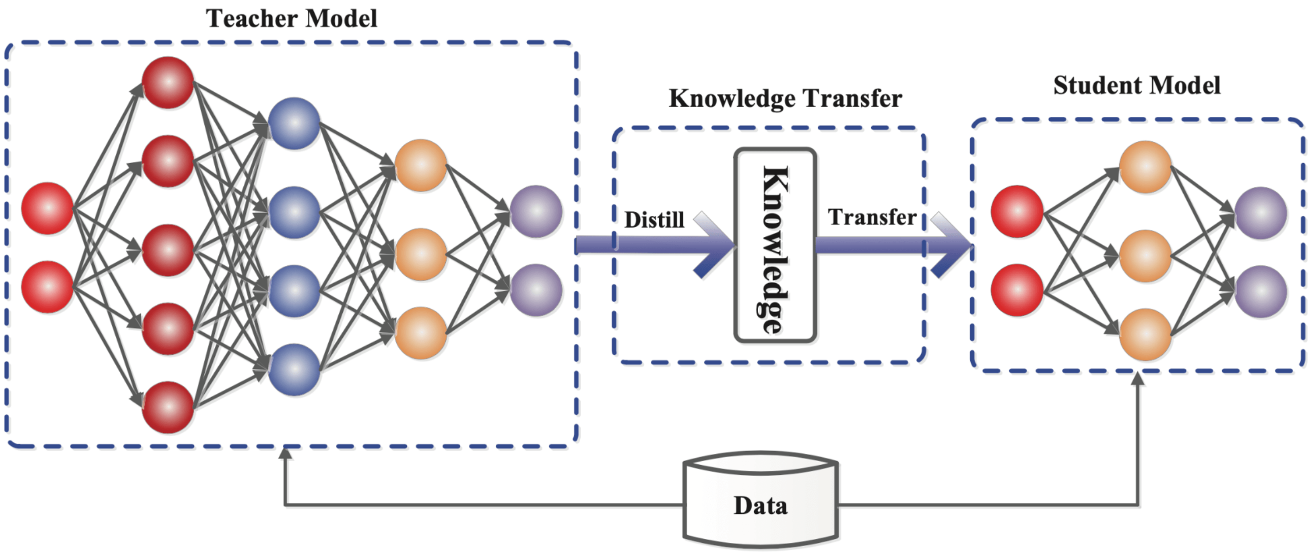


Examples of Graph Knowledge Distillation

Model Name	TinyGNN	FreeKD
Reason to Use Distillation	For compression (Reduce time)	For performance (Make perform better)
What to Distillate	Logits	Logits, Structures
How to Distillate	Direct	Adaptive

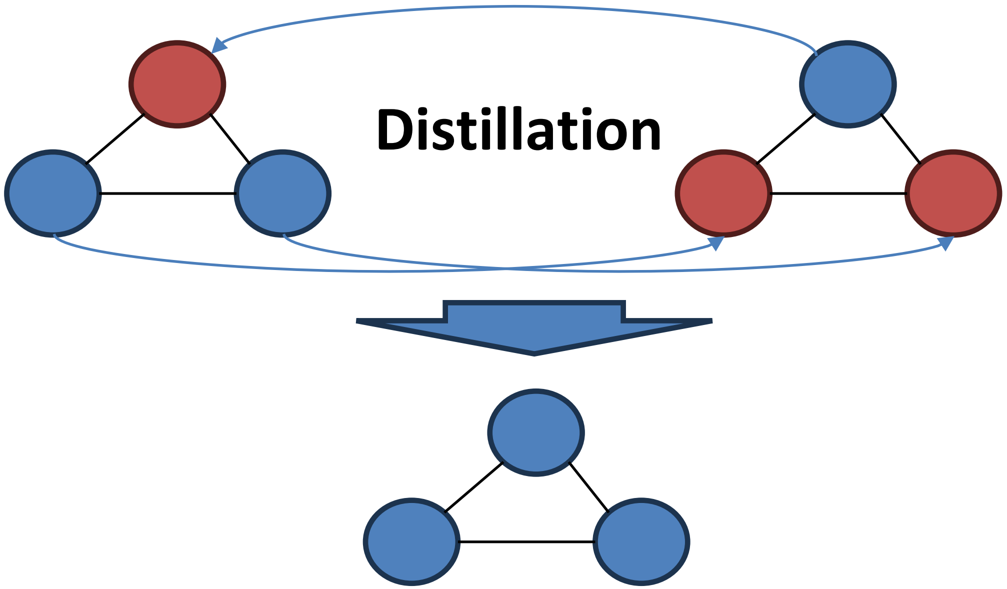
Where to Use Distillation For Graphs - Compression

- Distill knowledge from a big GNN model to a small GNN model



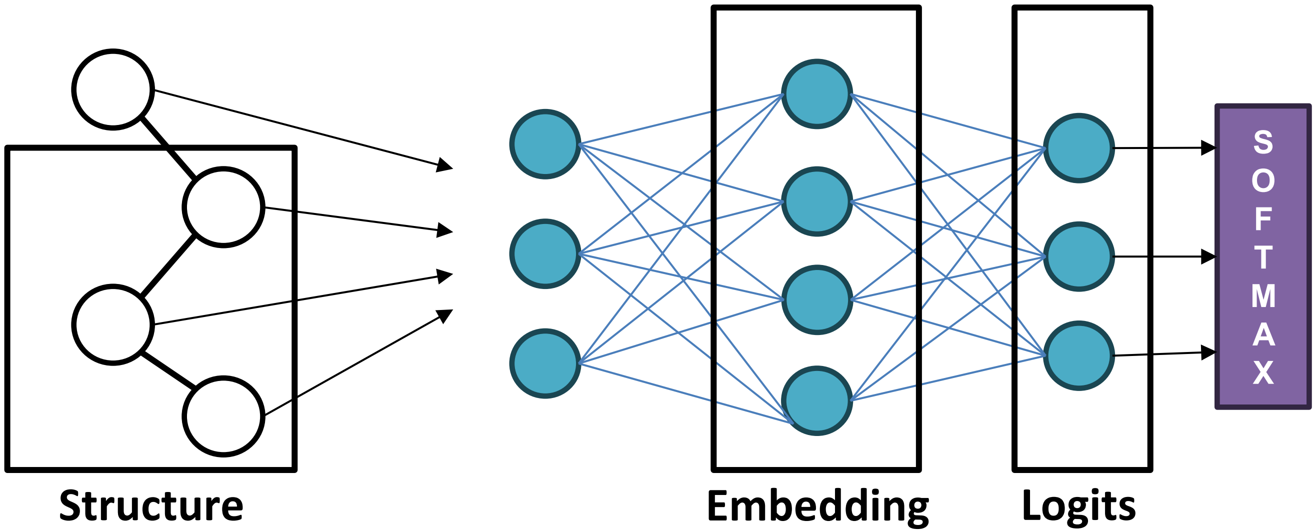
Where to Use Distillation For Graphs - Performance

- ☐ Distill knowledge between similar models to achieve better performances



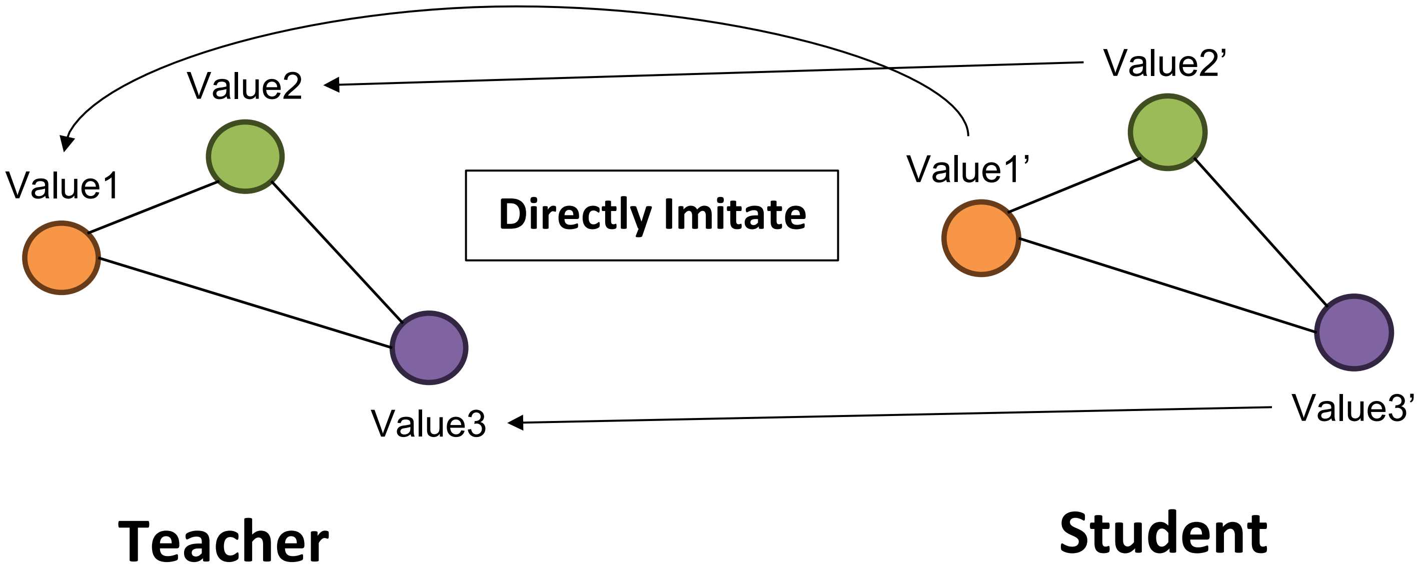
What to Distill From the Teacher Model

- Distill the structure of the graph, output logits, and the embedding space of the models



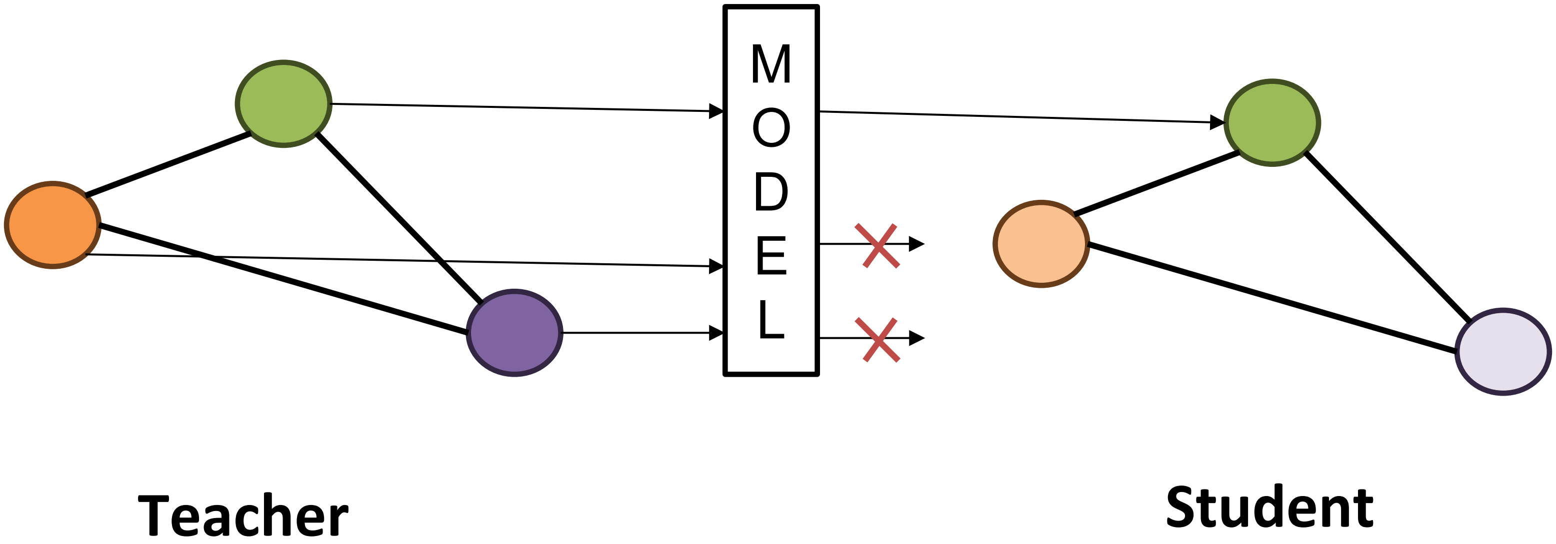
How to Distillate - Direct

- Divergence between models is minimized so that the student model fully mimics the teacher model



How to Distillate - Adaptive

- Adaptively considers the significance of knowledge before conducting



TinyGNN: Learning Efficient Graph Neural Networks

Bencheng Yan, Chaokun Wang, Gaoyang Guo, Yunkai Lou

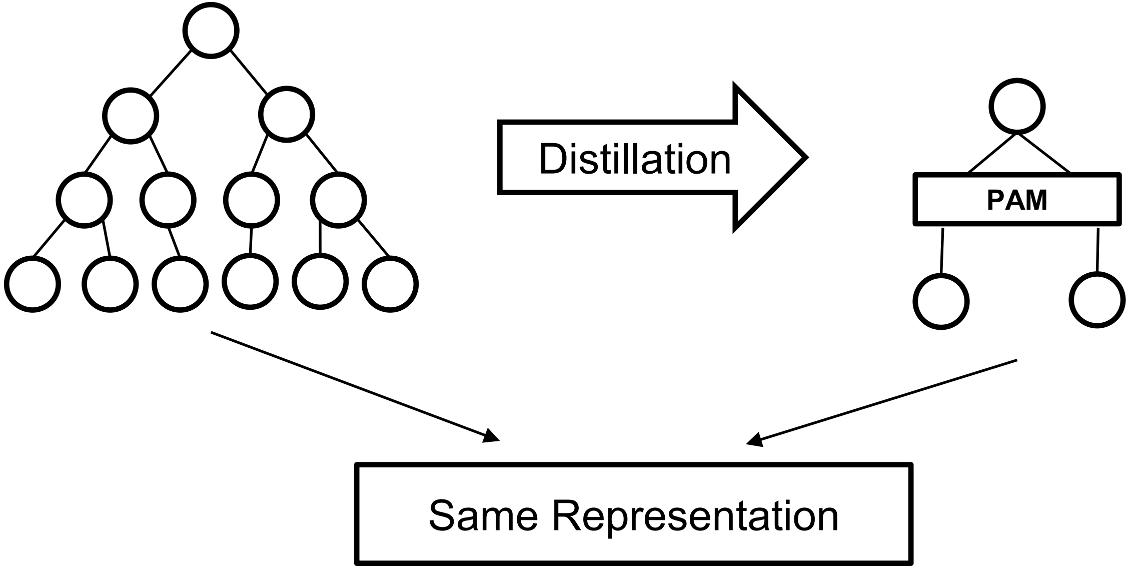
Tsinghua University

DMAIS@CAU

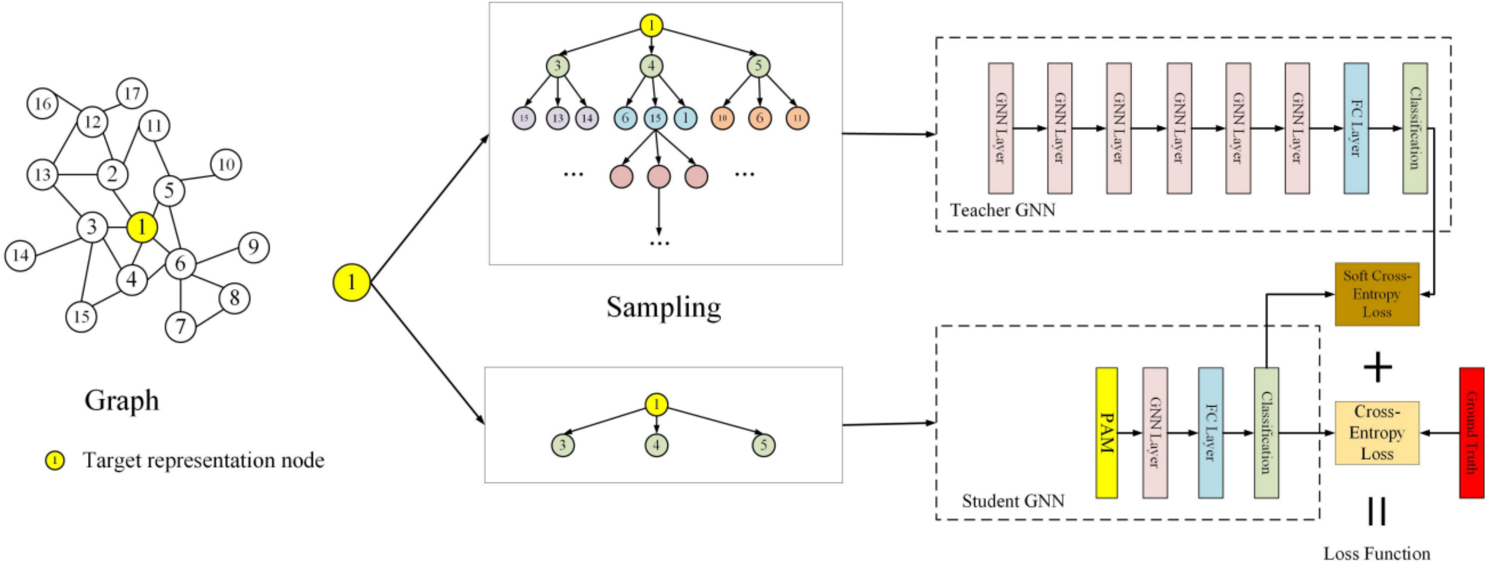
Joohee Cho

Goal of TinyGNN

- Make shallow GNNs to have similar representation powers as deep GNNs

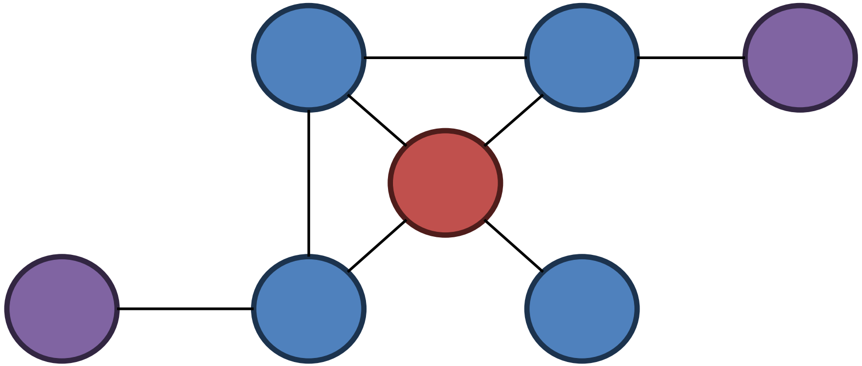


Full Structure of TinyGNN



Relations Between Neighbor Nodes

- Neighborhood nodes are connected to each other in maximum 2 hops



Relations Between Neighbor Nodes (cont.)

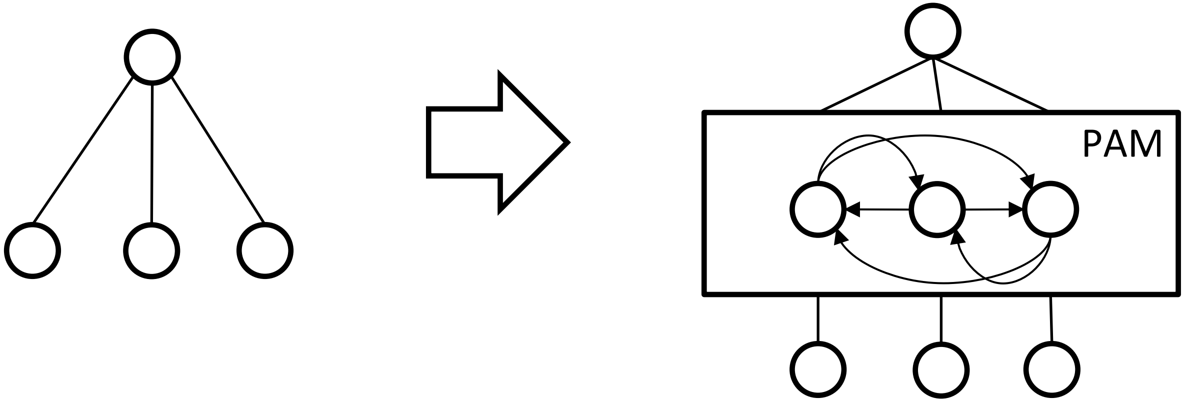
- Many peer nodes are also directly connected to each other

Table 1: Statistics of direct link rate among peer nodes.

Data Sets	Facebook	Chameleon	Squirrel	AliGraph
rate (%)	32.81	42.94	46.91	20.69

Peer-Aware Module for Student Model

- Add module to update representation of nodes considering its peers



Peer-Aware Module for Student Model (cont.)

- Apply self-attention to aggregate peer information

$$\alpha_{i,j} = \frac{exp(a_{i,j})}{\sum_k exp(a_{i,k})}$$

$$a_{i,j} = \frac{dot(W_a h_j, W_a h_i)}{\sqrt{d}}$$

$$h_i^* = \sum_j \alpha_{i,j} \cdot (W_v \cdot h_j)$$

Neighbor Distillation Strategy

- Both teacher and student model learns by the loss between the true label and its prediction

$$L_{CE}^{t} = \sum_{i=1}^{N} y_i \log(\widehat{y_i^t}) + (1 - y_i)\log(1 - \widehat{y_i^t})$$

Teacher Model Cross-Entropy Loss

$$L_{CE}^{t} = \sum_{i=1}^{N} y_i \log(\widehat{y_i^t}) + (1 - y_i)\log(1 - \widehat{y_i^t})$$

Student Model Cross-Entropy Loss

Neighbor Distillation Strategy (cont.)

- Student model additionally uses soft cross-entropy loss between teacher model's logits and the logits of itself

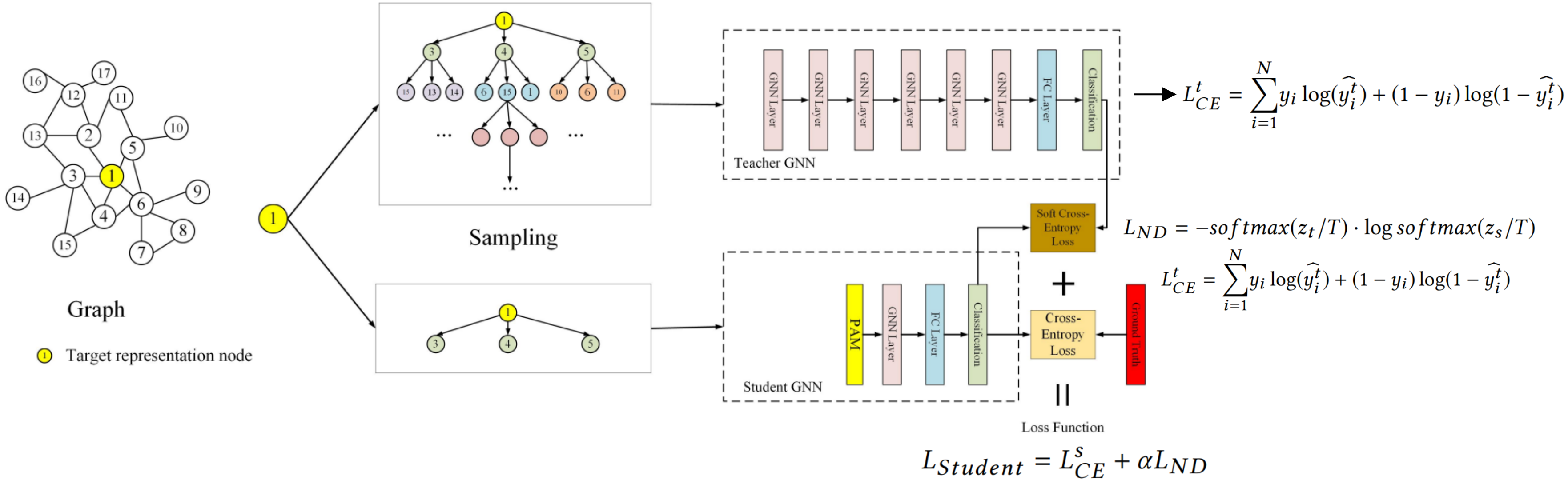
$$L_{ND} = -softmax(z_t/T) \cdot \log softmax(z_s/T)$$

Soft Cross-Entropy Loss of Logits

$$L_{Student} = L_{CE}^{s} + \alpha L_{ND}$$

Final Loss Function of the Student Model

Full Algorithm of the Model



Experiments – Datasets

Table 2: Data Sets Information.

Data Sets	#Nodes	#Edges	#Class
Facebook	22,470	171,002	4
Chameleon	2,277	31,421	10
Squirrel	5,201	198,493	6
AliGraph	46,800	946,175	7

Experiments – Node Classification (cont.)

Table 4: The inference time for each model on different data sets.

model	Facebook	Chameleon	Squirrel	AliGraph
GNN_3 (Teacher)	18.1999 (1×)	1.2715 (1×)	3.2022 (1×)	106.7618 (1×)
GNN_2 (Base)	0.8654 (21.03×)	0.0856 (14.85×)	0.164 (19.53×)	3.4994 (30.51×)
GNN_1 (Base)	0.2703 (67.33×)	0.03 (40.50×)	0.0447 (67.13×)	0.5081 (210.12×)
$TinyGNN_1$	0.4314 (42.19×)	0.0332 (38.30×)	0.0897 (35.70×)	0.8434 (126.59×)
$TinyGNN_2$	2.3542 (7.73×)	0.1059 (12.00×)	0.3112 (10.29×)	5.0072 (21.32×)

Experiments – Node Classification

Table 3: Results for node classification. The best results for one- or two-layer GNNs are in bold, respectively. GNN_3 **(Teacher) is the teacher GNN.** GNN_1 **(Base) and** GNN_2 **(Base) are direct trained on each dataset without using NDS and PAM.**

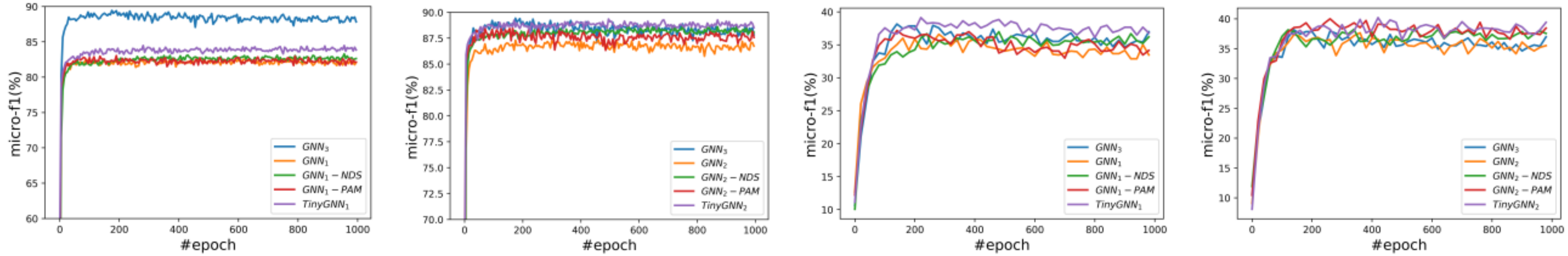
Model	Facebook		Chameleon		Squirrel		AliGraph	
	Micro-f1(%)	Macro-f1(%)	Micro-f1(%)	Macro-f1(%)	Micro-f1(%)	Macro-f1(%)	Micro-f1(%)	Macro-f1(%)
GNN_3 (Teacher)	89.43	88.69	39.12	38.16	30.68	30.41	60.34	53.75
GNN_2 (Base)	87.69	86.83	38.73	36.92	30.47	30.08	49.98	43.45
GNN_1 (Base)	82.75	81.65	36.88	35.79	29.83	29.24	33.77	25.22
GNN_1-*NDS*	83.32	82.11	37.46	36.42	30.73	30.38	35.30	24.88
GNN_1-*PAM*	83.15	81.79	38.05	36.56	31.32	31.08	40.19	32.05
TinyGNN$_1$	**84.56**	**83.41**	**39.71**	**38.46**	**32.09**	**32.00**	**42.34**	**32.37**
GNN_2-*NDS*	88.90	88.15	39.51	38.21	30.98	30.52	54.72	46.97
GNN_2-*PAM*	88.56	87.72	40.98	39.62	32.69	31.95	57.81	50.40
TinyGNN$_2$	**89.40**	**88.66**	**41.17**	**40.01**	**33.33**	**33.17**	**61.12**	**54.17**

Experiments – Different Teacher and Student Models

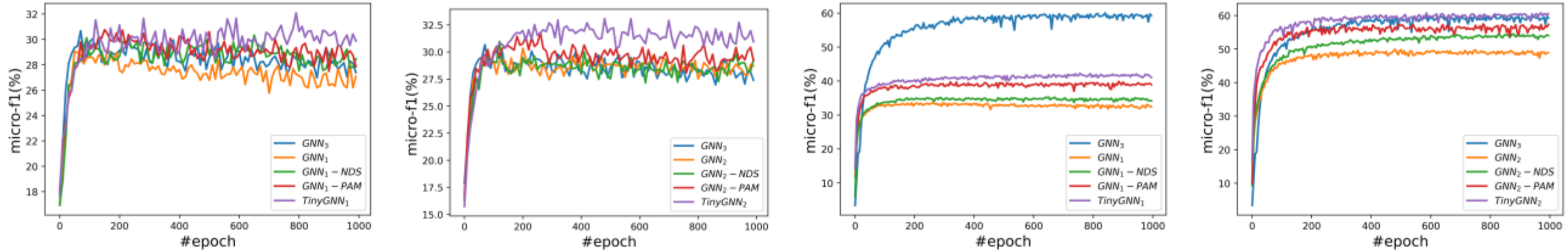
Table 5: Performance comparison with different teacher and different student GNNs. The best results for GNNs with the same layer number are in bold.

		Facebook		Chameleon		Squirrel		AliGraph	
Base		Micro-f1(%)	Macro-f1(%)	Micro-f1(%)	Macro-f1(%)	Micro-f1(%)	Macro-f1(%)	Micro-f1(%)	Macro-f1(%)
GNN_1		82.75	81.65	36.88	35.79	29.83	29.24	33.77	25.22
GNN_2		87.69	86.83	38.73	36.92	30.47	30.08	49.98	43.45
GNN_3		89.43	88.69	39.12	38.16	30.68	30.41	60.34	53.75
GNN_4		90.21	89.53	38.73	36.57	29.70	28.94	64.37	58.35
Teacher	Student								
GNN_4	$TinyGNN_1$	**84.99**	**83.75**	39.22	37.72	**32.86**	**32.48**	**42.67**	**32.39**
GNN_3	$TinyGNN_1$	84.56	83.41	**39.71**	**38.46**	32.09	32.00	42.34	32.37
GNN_2	$TinyGNN_1$	83.04	81.73	38.24	36.12	32.22	31.66	41.43	31.85
GNN_4	$TinyGNN_2$	**89.89**	**89.24**	40.68	39.22	32.44	32.29	**61.28**	**54.26**
GNN_3	$TinyGNN_2$	89.40	88.66	**41.17**	**40.01**	**33.33**	**33.17**	61.12	54.17
GNN_4	$TinyGNN_3$	**90.83**	**90.25**	**41.46**	**40.46**	**31.58**	**31.40**	**67.07**	**60.83**

Experiments – Convergence



(a) Micro-f1 of 1-layer GNNs on Facebook (b) Micro-f1 of 2-layer GNNs on Facebook (c) Micro-f1 of 1-layer GNNs on Chameleon (d) Micro-f1 of 2-layer GNNs on Chameleon



(e) Micro-f1 of 1-layer GNNs on Squirrel (f) Micro-f1 of 2-layer GNNs on Squirrel (g) Micro-f1 of 1-layer GNNs on AliGraph (h) Micro-f1 of 2-layer GNNs on AliGraph

Figure 4: The Micro-f1 of GNNs in different epoch on different data sets.

FreeKD: Free-direction Knowledge Distillation for Graph Neural Networks

Kaituo Feng, Changsheng Li, Ye Yuan, Guoren Wang

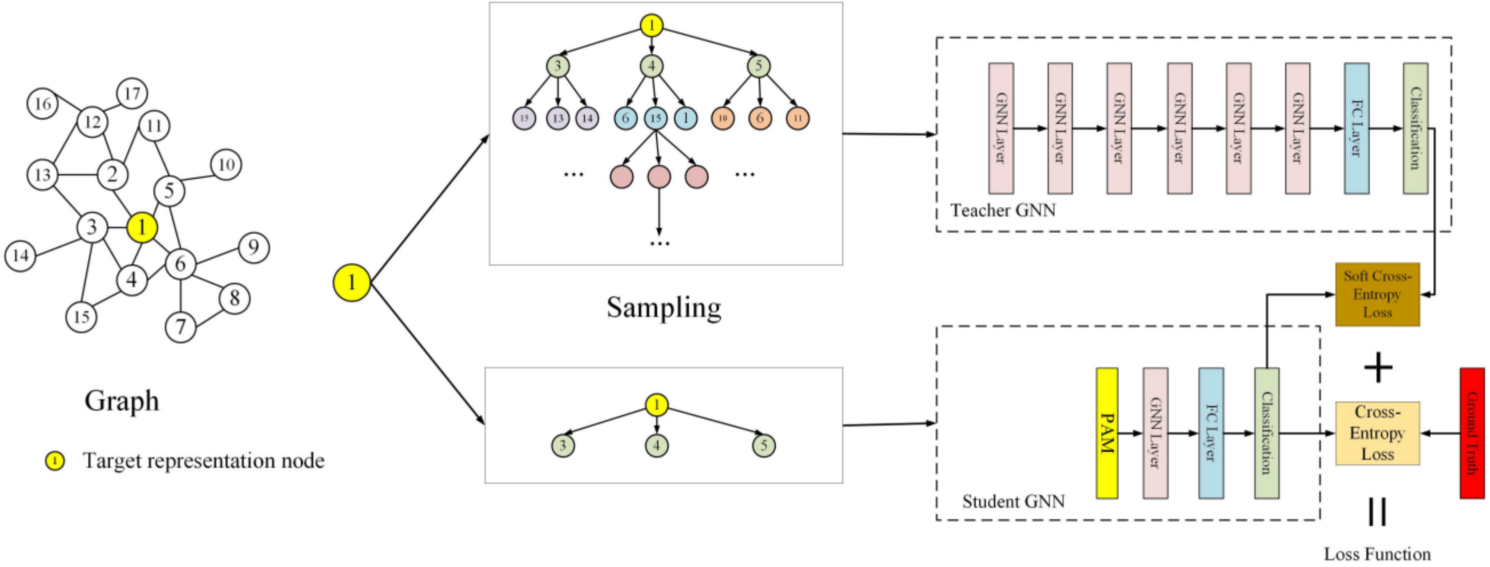
Beijing Institute of Technology

DMAIS@CAU

Joohee Cho

Limitation of Previous Work

- Deep models gets into over-smoothing problems, and shallow models cannot perfectly imitate their representations



Node Representations on Different Models

- Given two models of a graph, some nodes of a model outperform the corresponding nodes in another model while the others do not

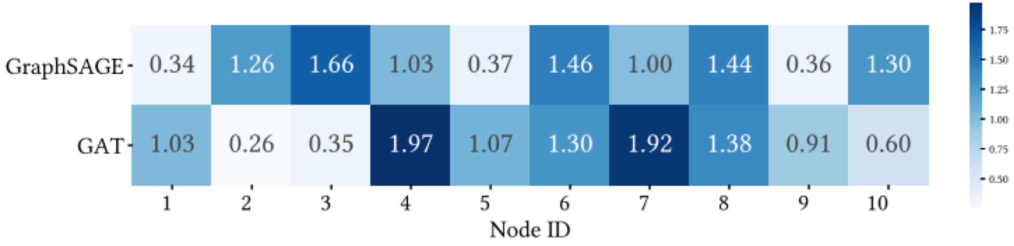
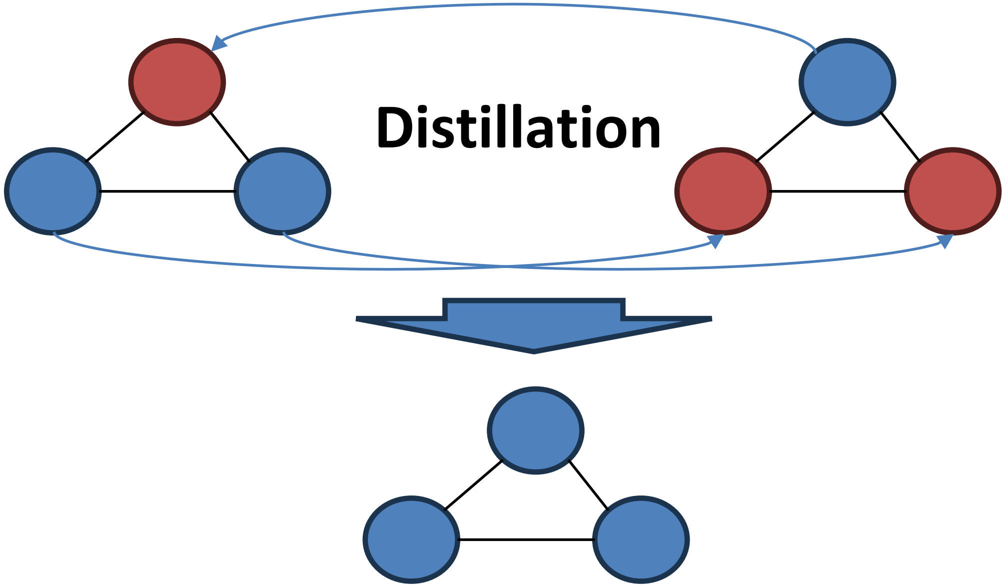


Figure 1: Cross entropy losses for nodes with ID from 1 to 10 on the Cora dataset obtained by two typical GNN models, GraphSAGE [13] and GAT [33], after training 20 epochs. The value in each block denotes the corresponding loss.

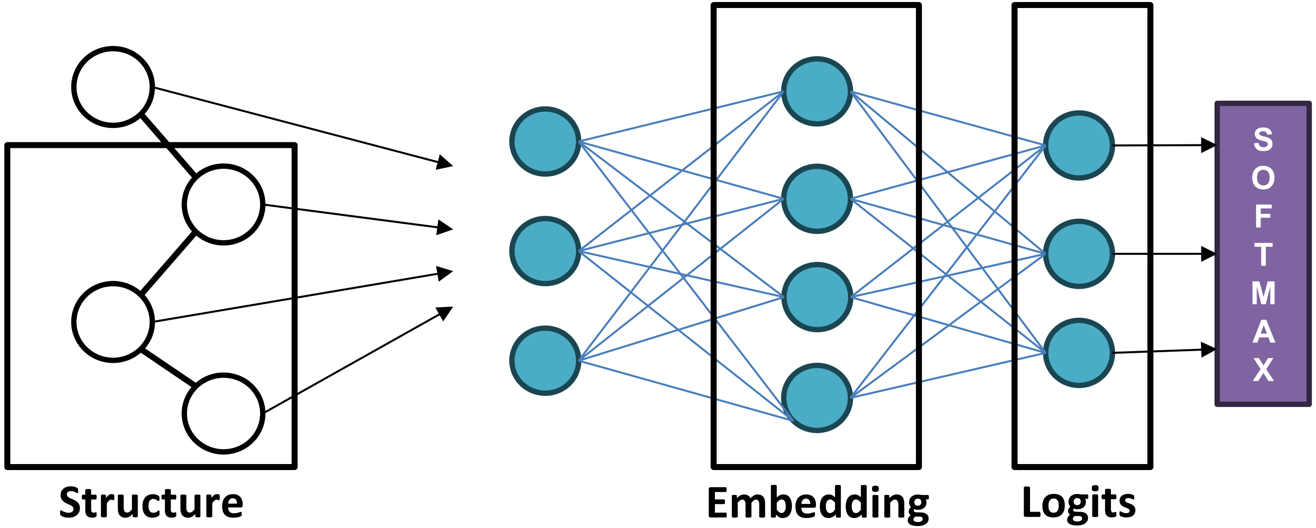
Idea of FreeKD Model

- Each graph distills the other graph the nodes that have great performance



What to Distillate From the Teacher Model

- ☐ Distill structure and the logits that performs better than the other model

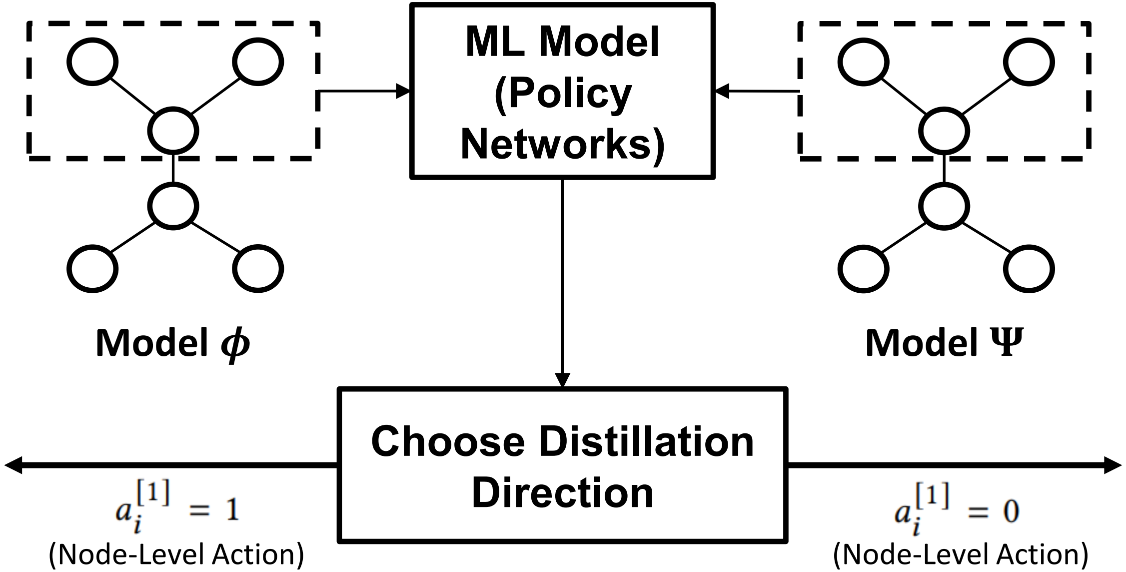


What to Consider

- ☐ Which model to regard as a teacher for each batch
- ☐ How to distill logits of the nodes
- ☐ Which nodes' structures to distillate
- ☐ How to distill structures of the nodes

How to Choose a Teacher Model

- ☐ Set predicted probabilities and the cross-entropy losses of batches as the input of RL based ML Model



Node-level Distillation

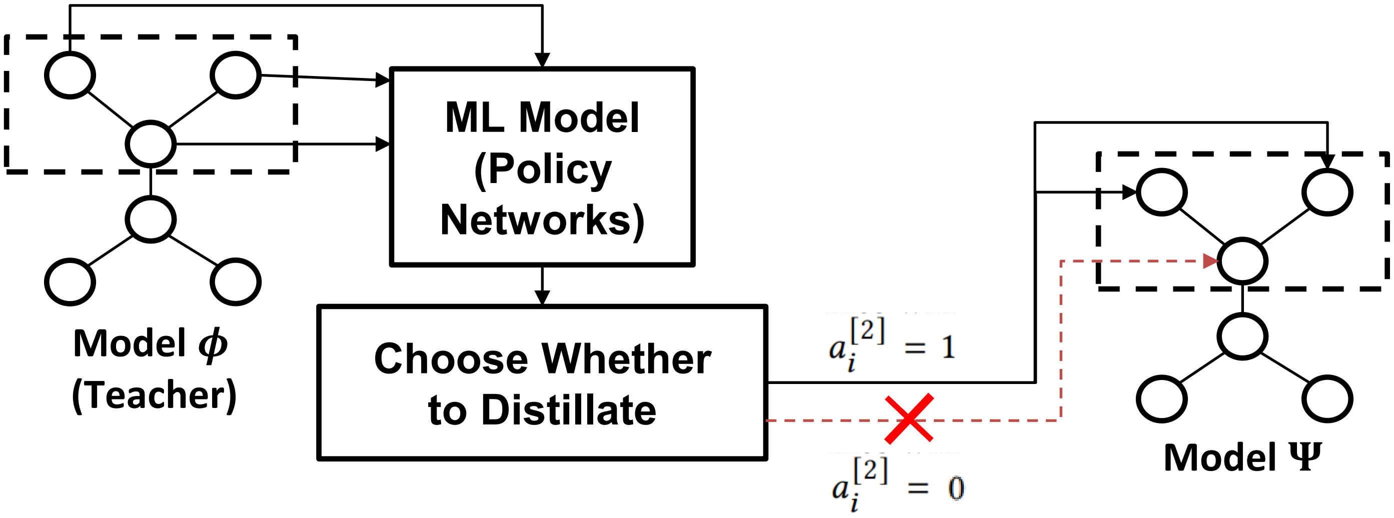
- ☐ Minimize the KL divergence of the predictions of both graphs

$$L^{\Psi}_{node} = \sum_{i=1}^{N} (1 - a_i^{[1]}) KL(\mathbf{p}_i^{\Phi} || \mathbf{p}_i^{\Psi})$$

$$L^{\Phi}_{node} = \sum_{i=1}^{N} a_i^{[1]} KL(\mathbf{p}_i^{\Psi} || \mathbf{p}_i^{\Phi}),$$

How to Use Structure Information

- ☐ Set input of node-level distillation and the center similarities of the models as the input of the ML Model

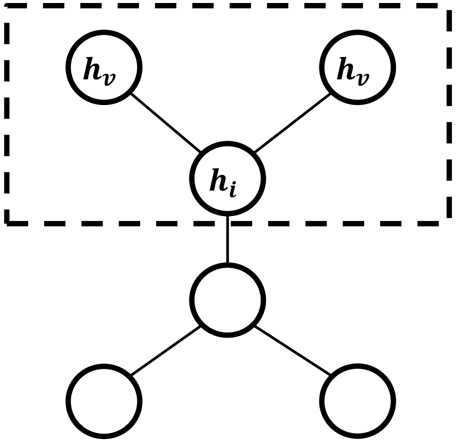


How to Use Structure Information (cont.)

- Center similarity compares the average similarities between a node and its neighbors

$$\mathbf{u}_i = \begin{cases} \left(\frac{1}{|\mathbf{M}_i^{\Phi}|} \sum_{v \in \mathbf{M}_i^{\Phi}} g(\mathbf{h}_i^{\Phi}, \mathbf{h}_v^{\Phi}), \frac{1}{|\mathbf{M}_i^{\Phi}|} \sum_{v \in \mathbf{M}_i^{\Phi}} g(\mathbf{h}_i^{\Psi}, \mathbf{h}_v^{\Psi})\right), \text{if } a_i^{[1]} = 0 \\ \left(\frac{1}{|\mathbf{M}_i^{\Psi}|} \sum_{v \in \mathbf{M}_i^{\Psi}} g(\mathbf{h}_i^{\Psi}, \mathbf{h}_v^{\Psi}), \frac{1}{|\mathbf{M}_i^{\Psi}|} \sum_{v \in \mathbf{M}_i^{\Psi}} g(\mathbf{h}_i^{\Phi}, \mathbf{h}_v^{\Phi})\right), \text{if } a_i^{[1]} = 1 \end{cases}$$

$$\mathbf{s}_i^{[2]} = CONCAT(\mathbf{s}_i^{[1]}, \mathbf{u}_i)$$



Structure-level Distillation

- Minimize the KL divergence of the similarities of both graphs

$$\hat{s}_{ij}^{\Phi} = \frac{e^{g\left(\mathbf{h}_i^{\Phi},\mathbf{h}_j^{\Phi}\right)}}{\sum_{v\in\mathbf{M}_i^{\Phi}} e^{g\left(\mathbf{h}_i^{\Phi},\mathbf{h}_v^{\Phi}\right)}}, \quad \hat{s}_{ij}^{\Psi} = \frac{e^{g\left(\mathbf{h}_i^{\Psi},\mathbf{h}_j^{\Psi}\right)}}{\sum_{v\in\mathbf{M}_i^{\Phi}} e^{g\left(\mathbf{h}_i^{\Psi},\mathbf{h}_v^{\Psi}\right)}}$$

Similarity

$$L_{struct}^{\Psi} = \sum_{i=1}^{N} (1 - a_i^{[1]}) a_i^{[2]} KL(\hat{\mathbf{s}}_i^{\Phi} || \hat{\mathbf{s}}_i^{\Psi})$$

$$L_{struct}^{\Phi} = \sum_{i=1}^{N} a_i^{[1]} a_i^{[2]} KL(\bar{\mathbf{s}}_i^{\Psi} || \bar{\mathbf{s}}_i^{\Phi})$$

Loss Function

Loss Function of the Model

- Minimize the summation of cross-entropy losses, node-level losses and structure-level losses

$$L^{\Phi} = L_{CE}^{\Phi} + \mu L_{node}^{\Phi} + \rho L_{struct}^{\Phi}$$

$$L^{\Psi} = L_{CE}^{\Psi} + \mu L_{node}^{\Psi} + \rho L_{struct}^{\Psi}$$

Rewards of the Policy Network

- The RL model (policy network) aims to maximize its rewards

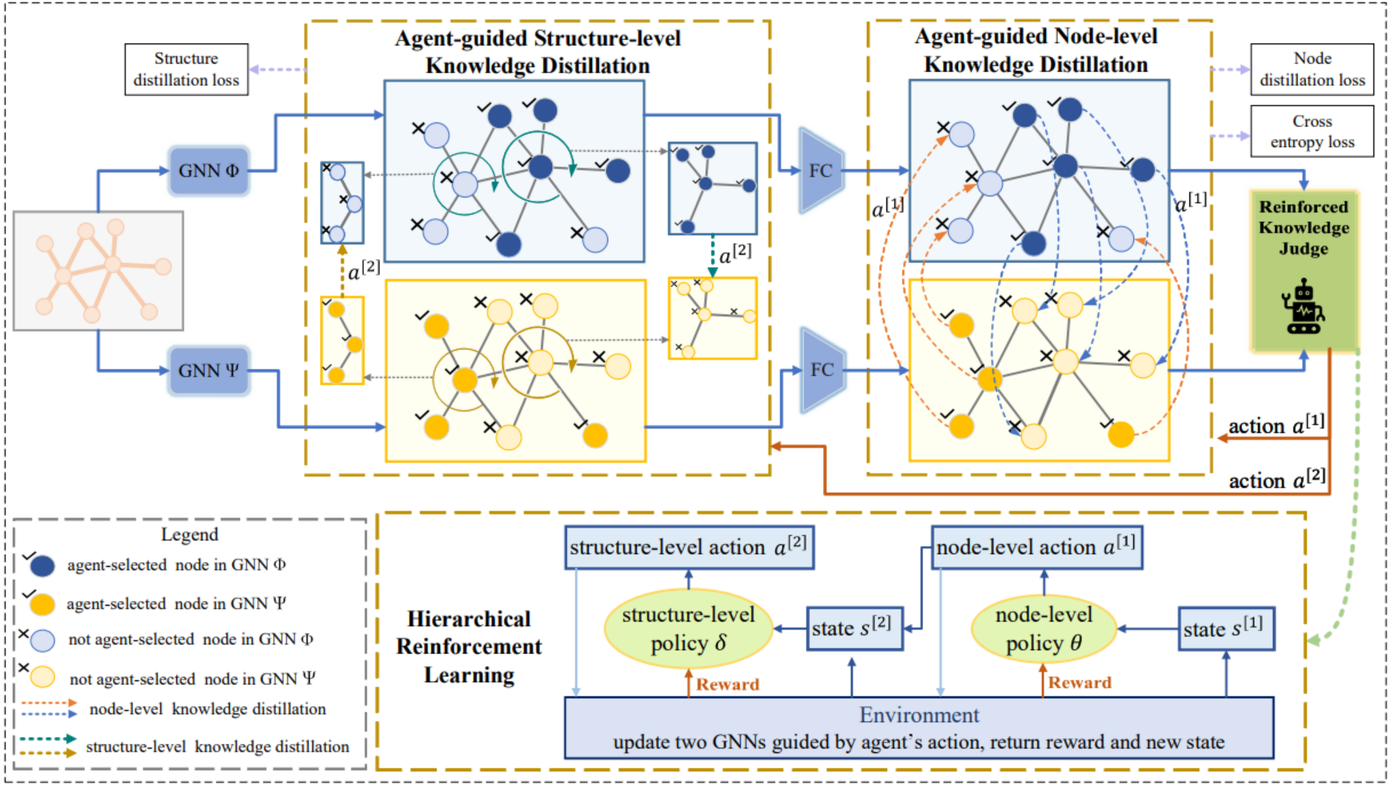
$$R_i = -\frac{\sum_{u \in \mathbf{B}} (L_{CE}^{\Phi}(u) + L_{CE}^{\Psi}(u))}{|\mathbf{B}|} - \gamma \frac{\sum_{v \in \mathcal{N}_i} (L_{CE}^{\Phi}(v) + L_{CE}^{\Psi}(v))}{|\mathcal{N}_i|}$$

Reward of a Node

$$\nabla_{\theta,\delta} J = \frac{1}{|\mathbf{B}|} \sum_{i \in \mathbf{B}} (R_i - b_i) \nabla_{\theta,\delta} \log(\pi_\theta(\mathbf{s}_i^{[1]}, a_i^{[1]}) \pi_\delta(\mathbf{s}_i^{[2]}, a_i^{[2]}))$$

Cumulative Rewards

Full Model



Structure distillation loss
Agent-guided Structure-level Knowledge Distillation
Agent-guided Node-level Knowledge Distillation
Node distillation loss
Cross entropy loss
GNN Φ
GNN Ψ
FC
FC
$a^{[1]}$
$a^{[2]}$
Reinforced Knowledge Judge
action $a^{[1]}$
action $a^{[2]}$
Legend
agent-selected node in GNN Φ
agent-selected node in GNN Ψ
not agent-selected node in GNN Φ
not agent-selected node in GNN Ψ
node-level knowledge distillation
structure-level knowledge distillation
Hierarchical Reinforcement Learning
structure-level action $a^{[2]}$
node-level action $a^{[1]}$
structure-level policy δ
state $s^{[2]}$
node-level policy θ
state $s^{[1]}$
Reward
Reward
Environment
update two GNNs guided by agent's action, return reward and new state

Experiments - Performance

			Cora		Chameleon		Citeseer		Texas	
Method	Basic Model		F1 Score (↑Impv.)		F1 Score (↑Impv.)		F1 Score (↑Impv.)		F1 Score (↑Impv.)	
	Φ	Ψ	Φ	Ψ	Φ	Ψ	Φ	Ψ	Φ	Ψ
GCN	-	-	85.12	-	33.09	-	75.42	-	57.57	-
GSAGE	-	-	85.36	-	48.77	-	76.56	-	76.22	-
GAT	-	-	85.45	-	40.29	-	75.66	-	57.84	-
FreeKD	GCN	GCN	86.53(↑**1.41**)	86.62(↑**1.50**)	37.61(↑**4.52**)	37.70(↑**4.61**)	77.28(↑**1.86**)	77.33(↑**1.91**)	60.28(↑**2.71**)	60.55(↑**2.98**)
FreeKD	GSAGE	GSAGE	86.41(↑**1.05**)	86.55(↑**1.19**)	49.89(↑**1.12**)	49.85(↑**1.08**)	77.78(↑**1.22**)	77.58(↑**1.02**)	78.76(↑**2.54**)	77.85(↑**1.63**)
FreeKD	GAT	GAT	86.46(↑**1.01**)	86.68(↑**1.23**)	43.96(↑**3.67**)	44.42(↑**4.13**)	77.13(↑**1.47**)	77.42(↑**1.76**)	61.18(↑**3.34**)	61.36(↑**3.52**)
FreeKD	GCN	GAT	86.65(↑**1.53**)	86.72(↑**1.27**)	35.58(↑**2.49**)	43.79(↑**3.53**)	77.39(↑**1.97**)	77.58(↑**1.92**)	61.06(↑**3.49**)	60.38(↑**2.54**)
FreeKD	GCN	GSAGE	86.26(↑**1.14**)	86.76(↑**1.40**)	35.39(↑**2.30**)	49.89(↑**1.12**)	77.08(↑**1.66**)	77.68(↑**1.12**)	60.61(↑**3.04**)	77.58(↑**1.36**)
FreeKD	GAT	GSAGE	86.67(↑**1.22**)	86.84(↑**1.48**)	43.96(↑**3.67**)	49.87(↑**1.10**)	77.24(↑**1.58**)	77.62(↑**1.06**)	62.45(↑**4.61**)	78.36(↑**2.14**)

Experiments – Ablation Study

Method	Network		Chameleon F1 Score		Cora F1 Score	
	Φ	Ψ	Φ	Ψ	Φ	Ψ
GCN	-	-	33.09	-	85.12	-
FreeKD-node	GCN	GCN	36.35	36.42	86.17	86.03
FreeKD-w.o.-judge	GCN	GCN	35.33	35.27	85.83	85.76
FreeKD-loss	GCN	GCN	35.86	35.79	85.89	85.97
FreeKD-all-neighbors	GCN	GCN	36.85	36.73	86.21	86.26
FreeKD-all-structures	GCN	GCN	36.53	36.62	86.13	86.07
FreeKD	GCN	GCN	37.61	37.70	86.53	86.62