CLNode: Curriculum Learning for Node Classification

Xiaowen Wei
School of Computer Science
Wuhan University
Wuhan, China
weixiaowen@whu.edu.cn

Xiuwen Gong
Faculty of Engineering
The University of Sydney
Sydney, Australia
xiuwen.gong@sydney.edu.au

Yibing Zhan
JD Explore Academy
Beijing, China
zhanyibing@jd.com

Bo Du
School of Computer Science
Wuhan University
Wuhan, China
gunspace@163.com

Yong Luo
School of Computer Science
Wuhan University
Wuhan, China
luoyong@whu.edu.cn

Wenbin Hu*
School of Computer Science
Wuhan University
Wuhan, China
hwb@whu.edu.cn

（WSDM-2023)

code：https://github.com/wxwmd/CLNode

gesis Leibniz-Institut für Sozialwissenschaften

Reported by Zhaoze Gao

1. Introduction
2. Approach
3. Experiments

Introduction

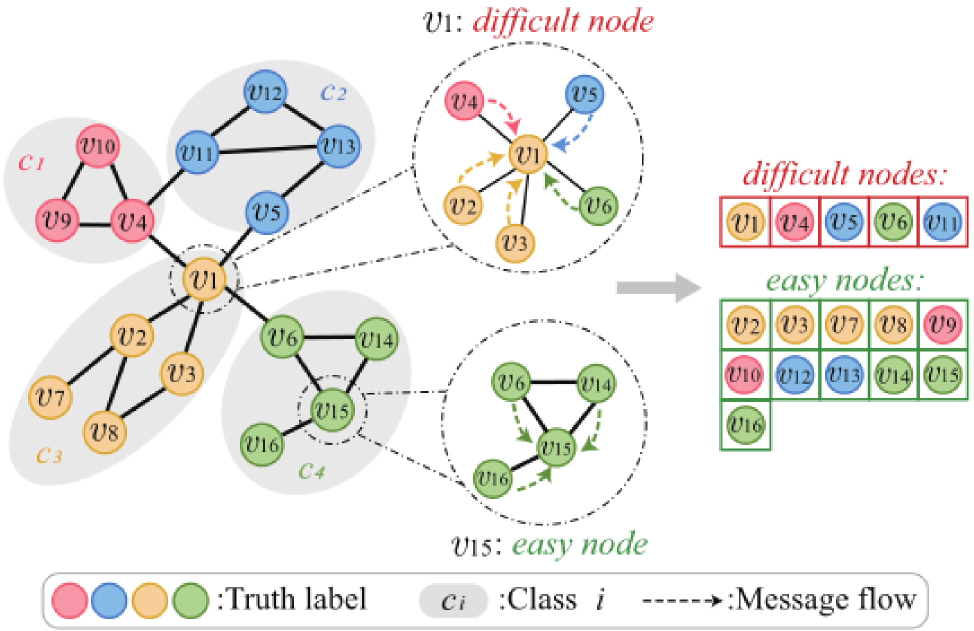

Figure 1: Illustration of node difficulty.

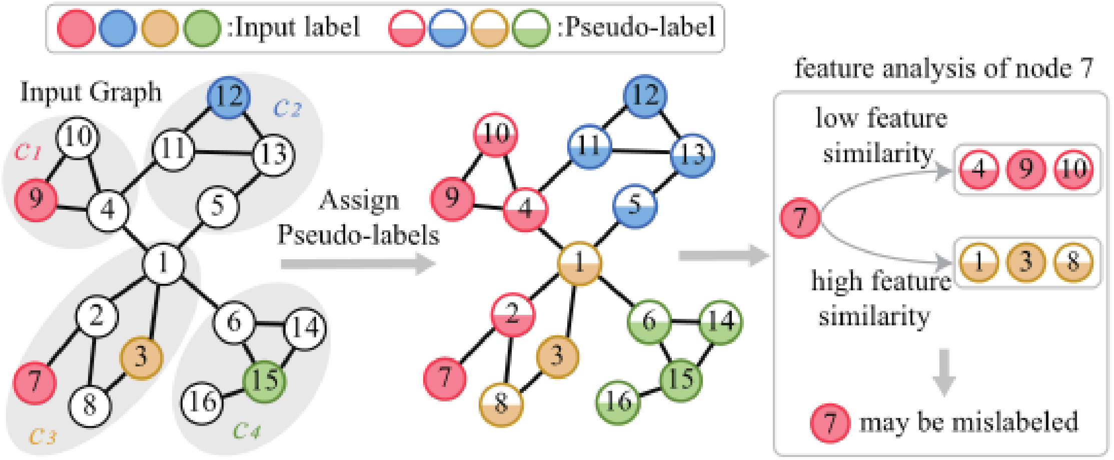

Figure 4: Illustration of the feature-based difficulty measurer.

Approach

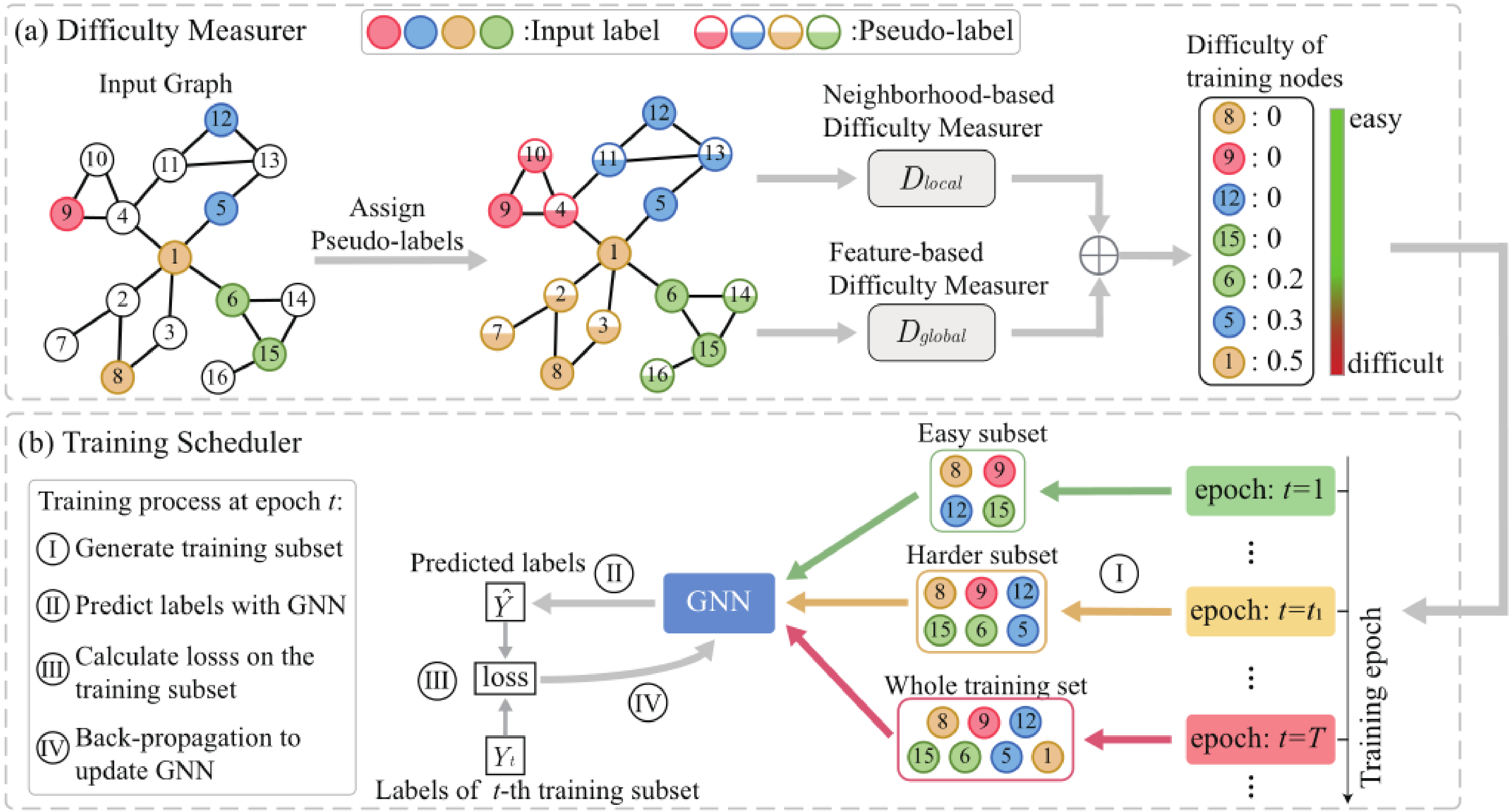

Figure 3: An overall framework of the proposed CLNode.

Approach

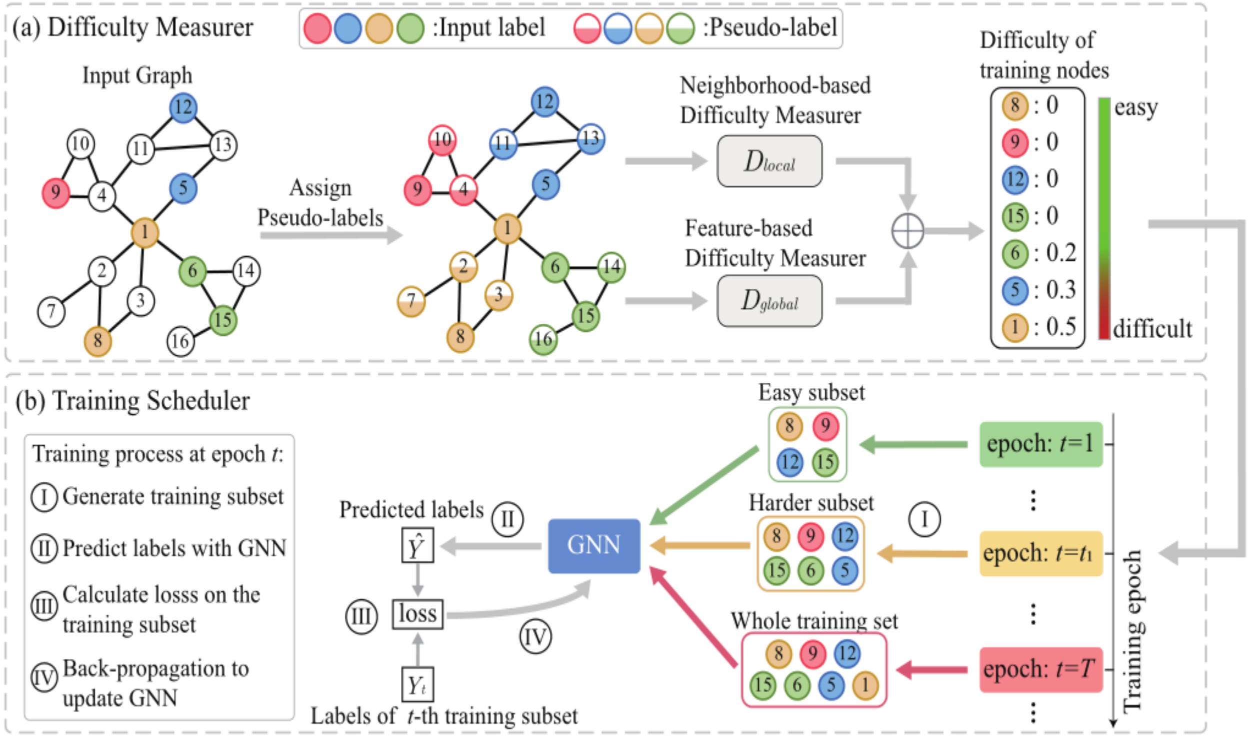

Figure 3: An overall framework of the proposed CLNode.

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, X) \qquad \mathcal{N}(i)$$

$$\mathcal{V}_L = \{v_1, ..., v_l\}$$

$$h_i^l = \text{UPDATE}(h_i^{l-1}, \text{AGGREGATE}(\{h_j^{l-1} | j \in \mathcal{N}(i)\})). \tag{1}$$

$$H = f_1(\mathcal{G}), \tag{2}$$

$$Y_P = MLP(H), \tag{3}$$

$$\tilde{Y}[i] = \begin{cases} Y_L[i], & i \in \mathcal{V}_L \\ Y_P[i], & otherwise. \end{cases} \tag{4}$$

Approach

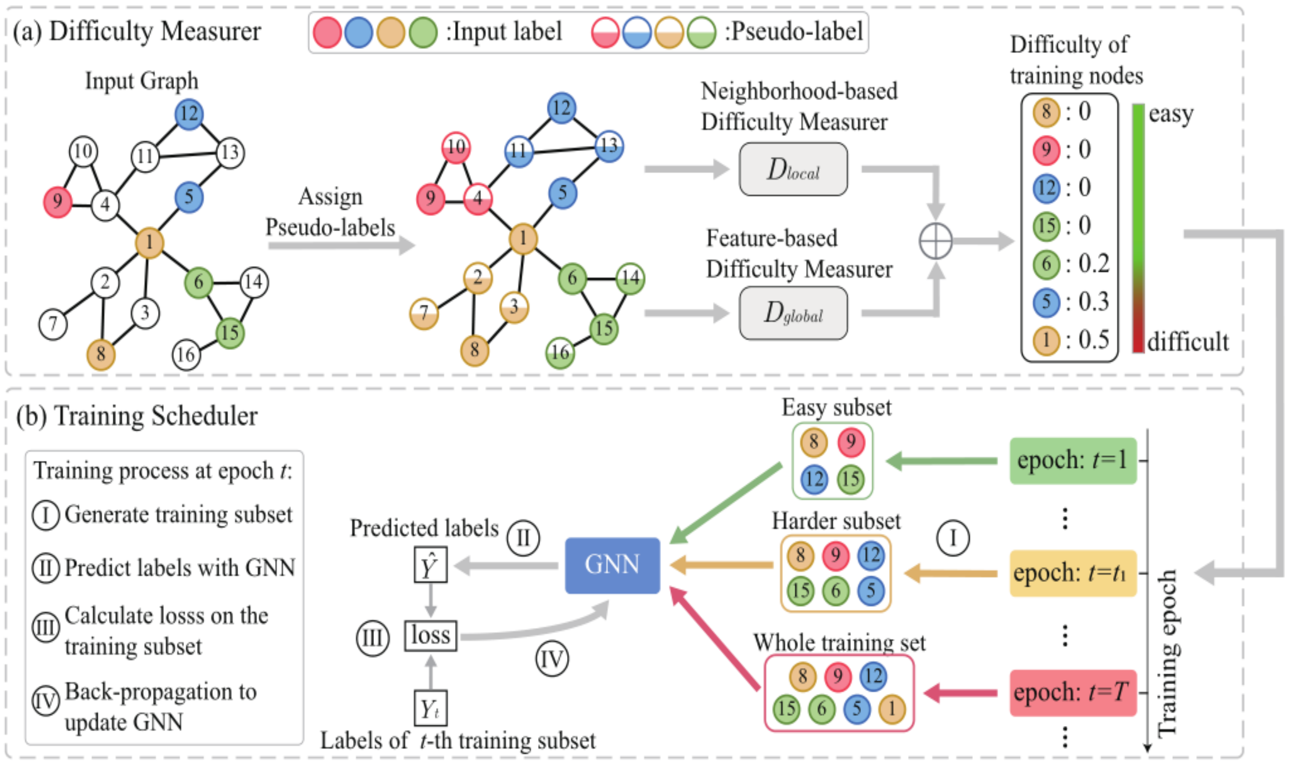

Figure 3: An overall framework of the proposed CLNode.

$$P_c(u) = \frac{|\{\tilde{Y}[v] = c \mid v \in \hat{\mathcal{N}}(u)\}|}{|\hat{\mathcal{N}}(u)|}, \tag{5}$$

$$D_{local}(u) = -\sum_{c \in C} P_c(u)\, log(P_c(u)), \tag{6}$$

$$\mathcal{V}_c = \{v \mid \tilde{Y}[v] = c\}, \tag{7}$$

$$h_c = \text{AVG}(h_v \mid v \in \mathcal{V}_c), \tag{8}$$

$$S(u) = \frac{exp(h_u \cdot h_{c_u})}{\max_{c \in C} exp(h_u \cdot h_c)}, \tag{9}$$

$$D_{global}(u) = 1 - S(u). \tag{10}$$

$$D(u) = D_{local}(u) + \alpha \cdot D_{global}(u), \tag{11}$$

Approach

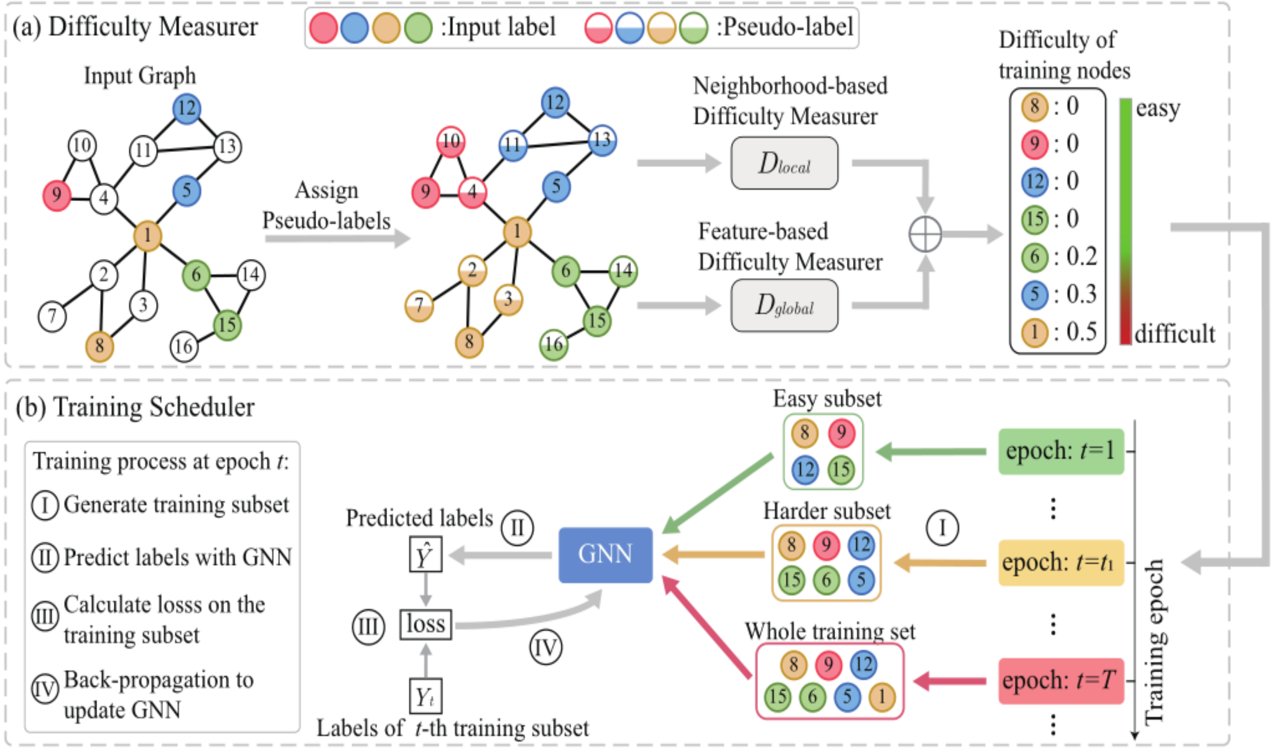

Figure 3: An overall framework of the proposed CLNode.

- linear:

$$g(t) = min(1, \lambda_0 + (1 - \lambda_0) * \frac{t}{T}). \tag{12}$$

- root:

$$g(t) = min(1, \sqrt{\lambda_0^2 + (1 - \lambda_0^2) * \frac{t}{T}}). \tag{13}$$

- geometric:

$$g(t) = min(1, 2^{log_2 \lambda_0 - log_2 \lambda_0 * \frac{t}{T}}). \tag{14}$$

Experiments

Table 1: Statistics of five benchmark datasets.

Dataset	Nodes	Edges	Features	Classes	Label rate
Cora	2708	5429	1433	7	2%
CiteSeer	3327	4732	3703	6	2%
PubMed	19717	88648	500	3	0.1%
A-Computers	13381	245778	767	10	1%
A-Photo	7487	119043	745	8	1%

Experiments

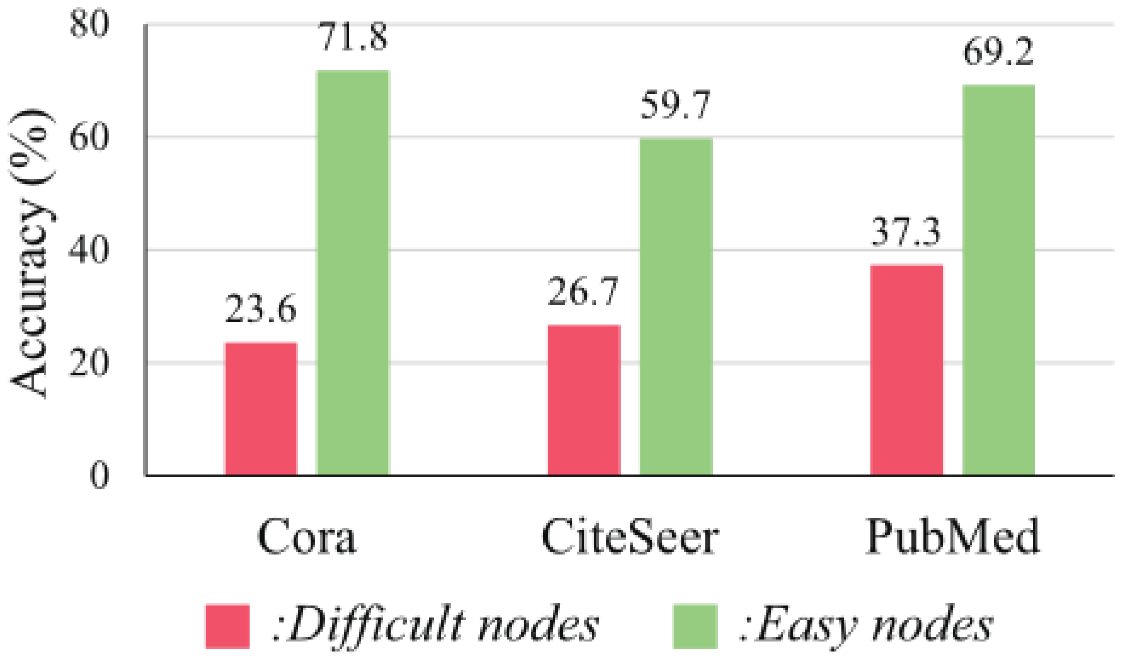

Figure 2: Accuracy of GCN trained on *difficult nodes* or *easy nodes*.

Experiments

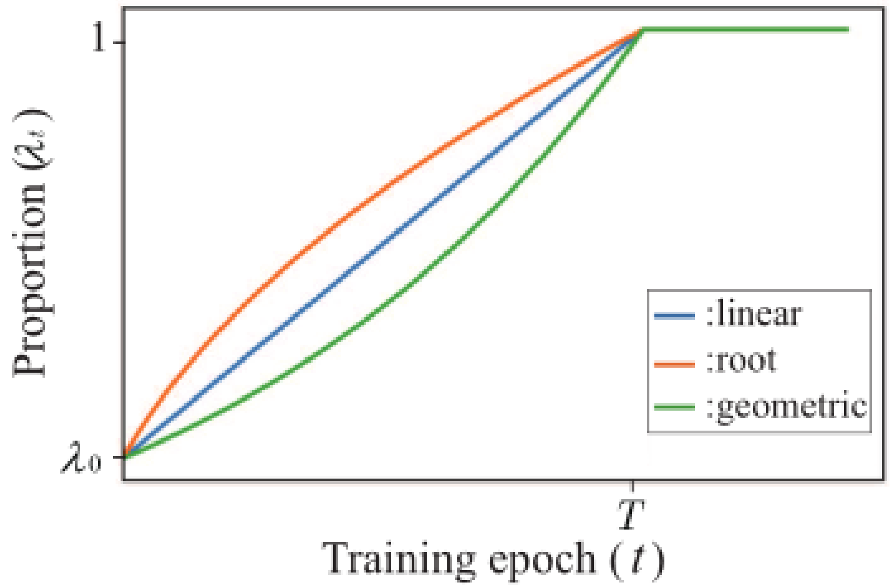

Figure 5: Visualization of three pacing functions.

Experiments

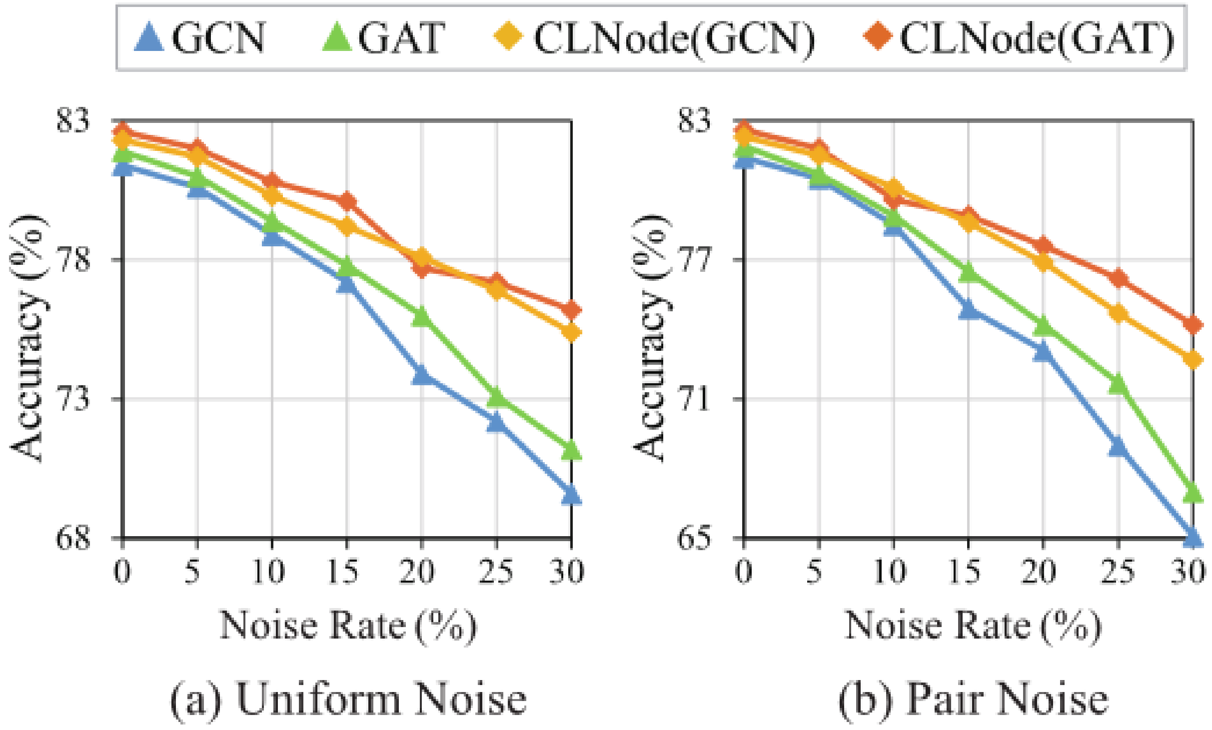

Figure 6: Accuracy (%) on Cora with two kinds of label noise.

Experiments

Table 2: Node classification performance (Accuracy (%)±Std) on five datasets.

	Method	Cora	CiteSeer	PubMed	A-Computers	A-Photo
GCN	Original	73.5±0.8	62.8±2.6	64.3±2.9	79.0±3.7	89.1±0.8
	+CLNode	**77.0±0.7**	**65.5±2.3**	**65.9±1.3**	**84.7±0.5**	**90.8±1.0**
	(Improv.)	3.5%	2.7%	1.6%	5.7%	1.7%
GraphSAGE	Original	70.1±2.3	57.4±3.7	61.3±1.4	71.7±2.4	83.0±2.6
	+CLNode	**72.1±1.4**	**60.3±3.1**	**64.1±3.8**	**77.5±1.6**	**87.5±1.2**
	(Improv.)	2.0%	2.9%	2.8%	5.8%	4.5%
GAT	Original	74.2±1.2	63.7±2.8	64.6±2.5	80.2±0.8	89.4±1.8
	+CLNode	**77.1±1.1**	**65.3±2.6**	**68.2±2.6**	**82.6±1.1**	**90.1±1.1**
	(Improv.)	2.9%	1.6%	3.6%	2.4%	0.7%
SuperGAT	Original	74.4±4.3	**64.8±3.3**	67.4±4.3	81.2±2.0	87.3±2.0
	+CLNode	**75.5±2.7**	63.0±3.2	**72.2±3.0**	**83.4±2.4**	**88.8±1.2**
	(Improv.)	1.1%	-	4.8%	2.2%	1.5%
JK-Net	Original	74.0±1.5	62.1±3.7	**66.0±1.7**	83.2±1.3	89.2±0.7
	+CLNode	**76.8±0.8**	**63.6±1.2**	**71.5±3.2**	**84.4±1.0**	**90.4±0.9**
	(Improv.)	2.8%	1.5%	5.5%	1.2%	1.2%
GCNII	Original	76.2±4.0	64.5±4.3	70.8±6.1	79.8±1.8	87.4±2.1
	+CLNode	**77.8±2.1**	**66.5±2.2**	**71.3±4.6**	**82.2±1.5**	**89.3±2.0**
	(Improv.)	1.6%	2.0%	0.5%	2.4%	1.9%

Experiments

Table 3: Accuracy (%) on Cora under different label rates.

	Method	1%	2%	3%
GCN	Original	62.4±2.7	73.5±0.8	78.6±0.6
	+CLNode	**66.9±1.2**	**77.0±0.7**	**79.7±0.6**
GraphSage	Original	54.8±3.0	70.1±2.3	76.0±0.8
	+CLNode	**61.8±2.6**	**72.1±1.4**	**77.7±1.5**
GAT	Original	65.2±2.4	74.2±1.2	78.8±1.0
	+CLNode	**68.5±2.0**	**77.1±1.1**	**79.9±0.5**
SuperGAT	Original	65.5±6.0	74.4±4.3	**78.7±1.6**
	+CLNode	**67.9±3.2**	**75.5±2.7**	78.5±2.4
JK-Net	Original	67.5±1.7	74.0±1.5	77.4±1.4
	+CLNode	**69.4±1.4**	**76.8±0.8**	**78.8±0.3**
GCNII	Original	68.5±3.9	76.2±4.0	79.0±2.2
	+CLNode	**71.2±3.8**	**77.8±2.1**	**80.2±2.0**

Experiments

Table 4: Comparisons between different difficulty measurers.

	Method	Cora	CiteSeer	PubMed
GCN	original	69.6	55.3	69.4
	+CLNode(local)	74.8	61.8	74.2
	+CLNode(global)	72.3	62.5	73.2
	+CLNode	**75.4**	**63.1**	**74.4**

Table 5: Comparisons between different pacing functions.

	Pacing Function	Cora	CiteSeer	PubMed
CLNode	linear	74.8	62.7	74.2
	root	74.5	62.5	73.9
	geometric	**75.4**	**63.1**	**74.4**

Experiments

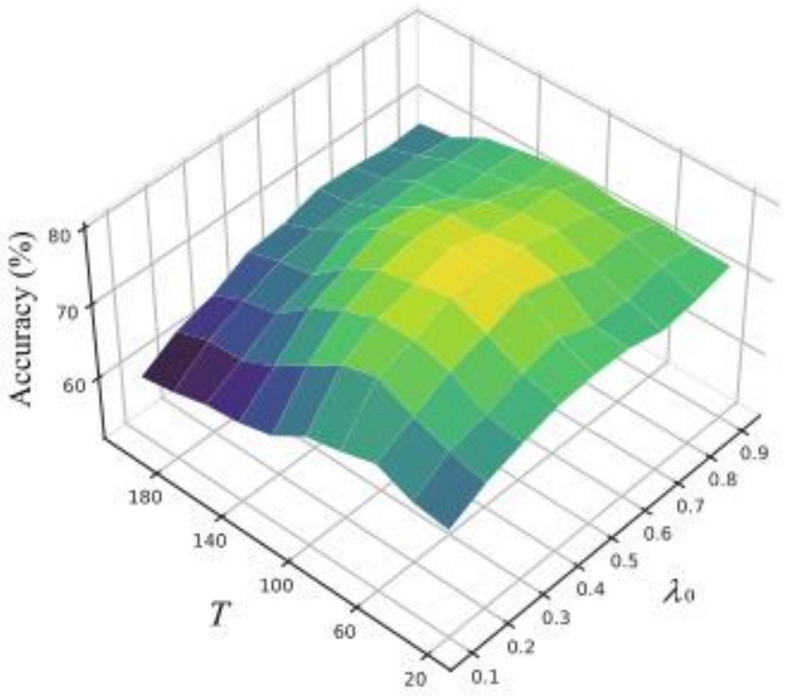

Figure 7: Parameter sensitivity analysis on Cora.

Thank you！