GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks

Zemin Liu[1*]
[1]National University of Singapore
Singapore
zeminliu@nus.edu.sg

Xingtong Yu[2*]
[2]University of Science and Technology of China
China
yxt95@mail.ustc.edu.cn

Yuan Fang[3†]
[3]Singapore Management University
Singapore
yfang@smu.edu.sg

Xinming Zhang[2†]
[2]University of Science and Technology of China
China
xinming@ustc.edu.cn

WWW-2023

Code: https://github.com/Starlien95/GraphPrompt

Reported by Dongdong Hu

Introduction

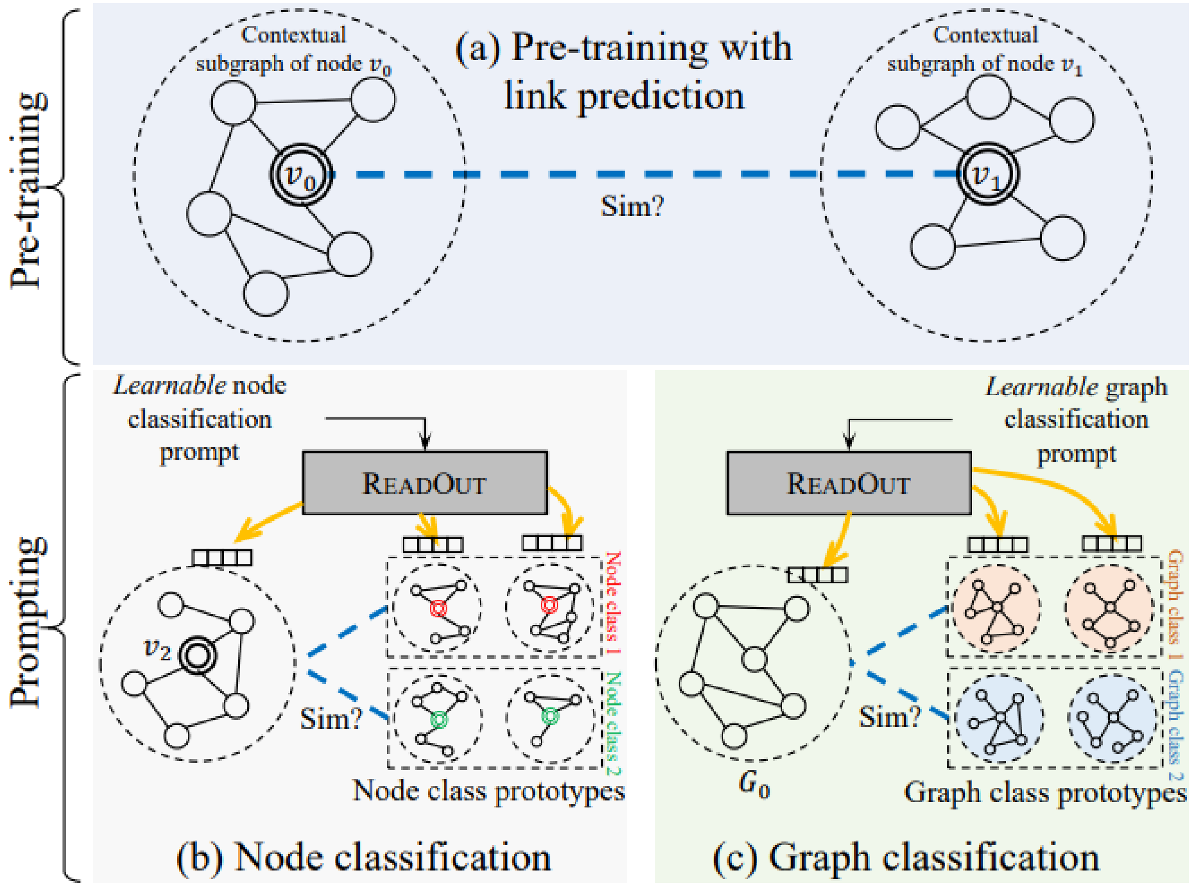


Figure 1: Illustration of the motivation. (a) Pre-training on graphs. (b/c) Downstream node/graph classification.

GraphPrompt unifies pre-training and downstream tasks into a common task template

However, existing study of prompting on graphs is still limited, lacking a universal treatment to appeal to different downstream task

Method

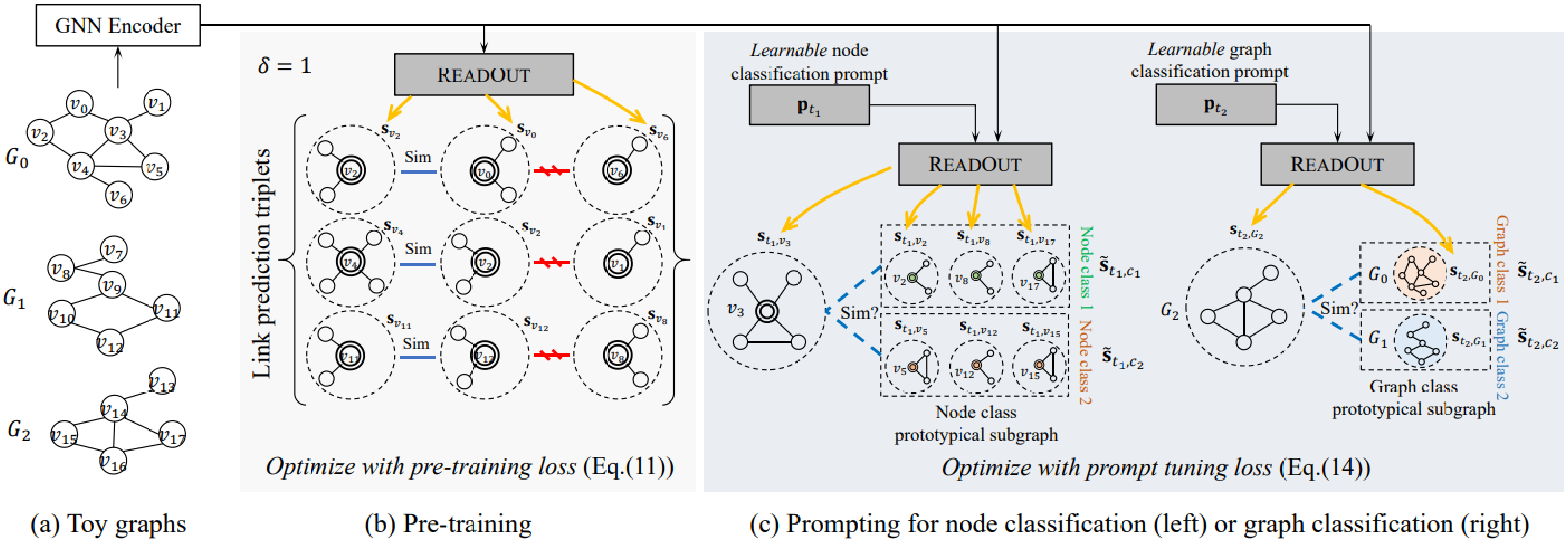


Figure 2: Overall framework of GraphPrompt.

Method

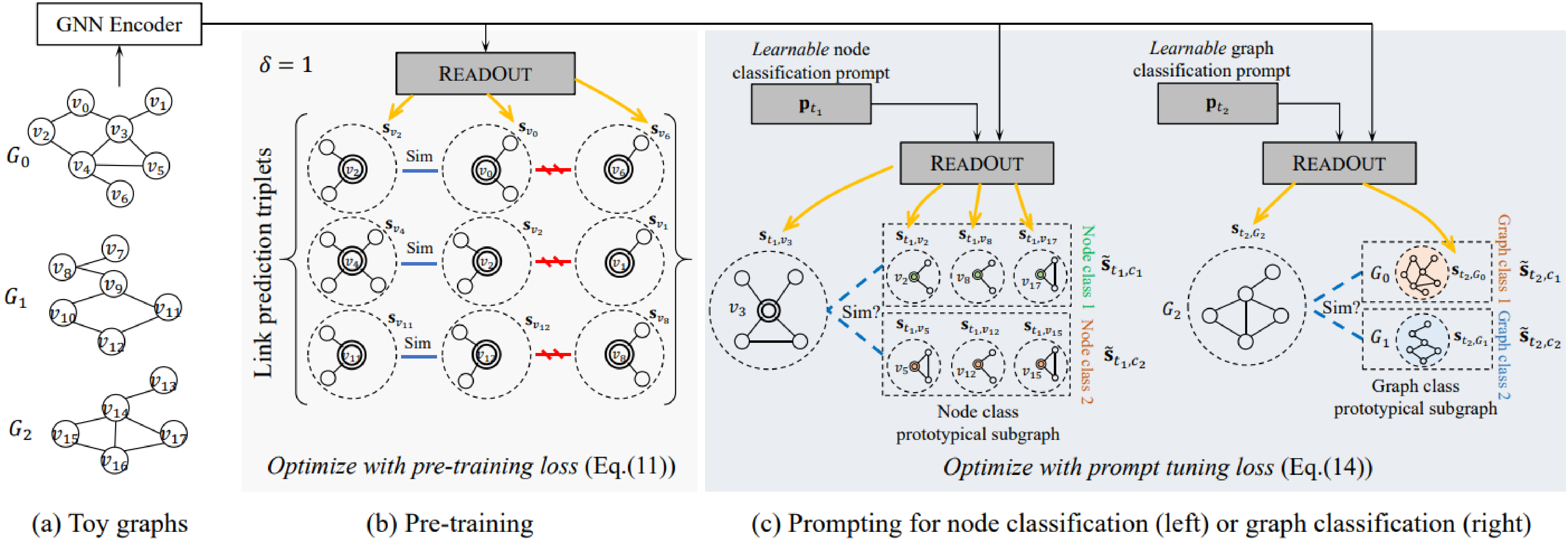


Figure 2: Overall framework of GraphPrompt.

$G = (V, E)$

$\mathbf{X} \in \mathbb{R}^{|V| \times d}$

$\mathcal{G} = \{G_1, G_2, \ldots, G_N\}$

k-shot classification

$\ell_i \in C \quad G_i \in \mathcal{G}$

$$\mathbf{h}_v^l = \text{AGGR}(\mathbf{h}_v^{l-1}, \{\mathbf{h}_u^{l-1} : u \in \mathcal{N}_v\}; \theta^l), \tag{1}$$

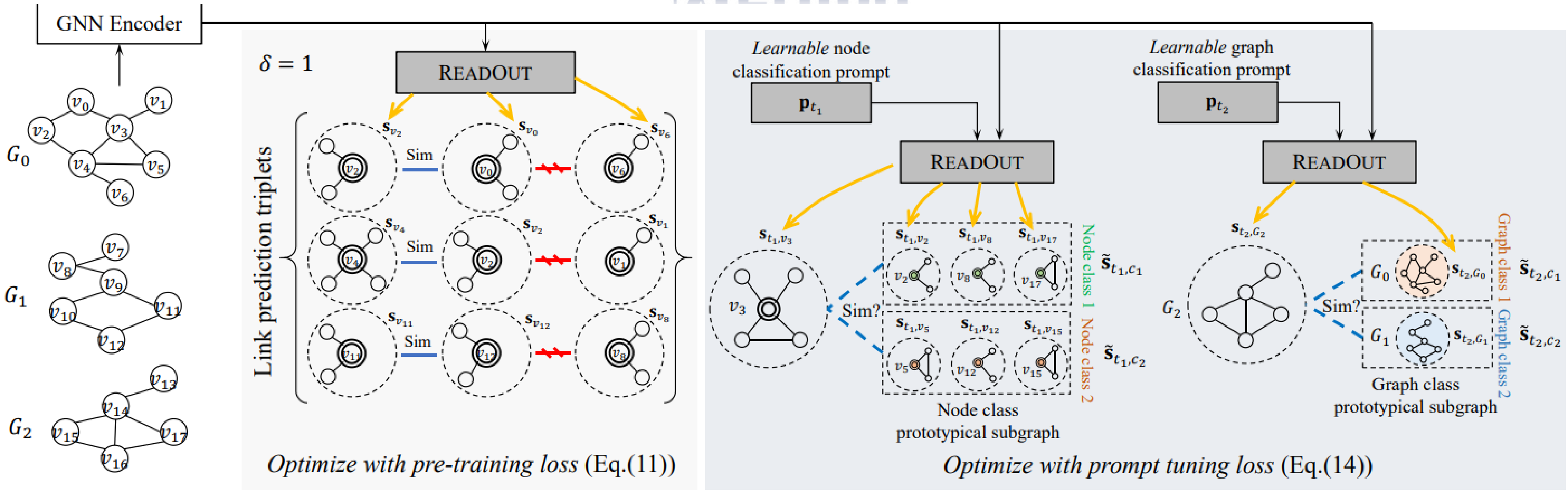
$$V(S_v) = \{d(u,v) \leq \delta \mid u \in V\}, \text{ and} \tag{2}$$

$$E(S_v) = \{(u,u') \in E \mid u \in V(S_v), u' \in V(S_v)\}, \tag{3}$$

$\Theta = \{\theta^1, \theta^2, \ldots\}$

$\mathbf{h}_v$

Method



(a) Toy graphs (b) Pre-training (c) Prompting for node classification (left) or graph classification (right)

$$\mathrm{sim}(\mathbf{s}_v, \mathbf{s}_a) > \mathrm{sim}(\mathbf{s}_v, \mathbf{s}_b). \tag{4}$$

$$\tilde{\mathbf{s}}_c = \frac{1}{k} \sum_{(v_i,\ell_i)\in D, \ell_i = c} \mathbf{s}_{v_i}. \tag{5}$$

$$\ell_j = \arg\max_{c\in C} \mathrm{sim}(\mathbf{s}_{v_j}, \tilde{\mathbf{s}}_c). \tag{6}$$

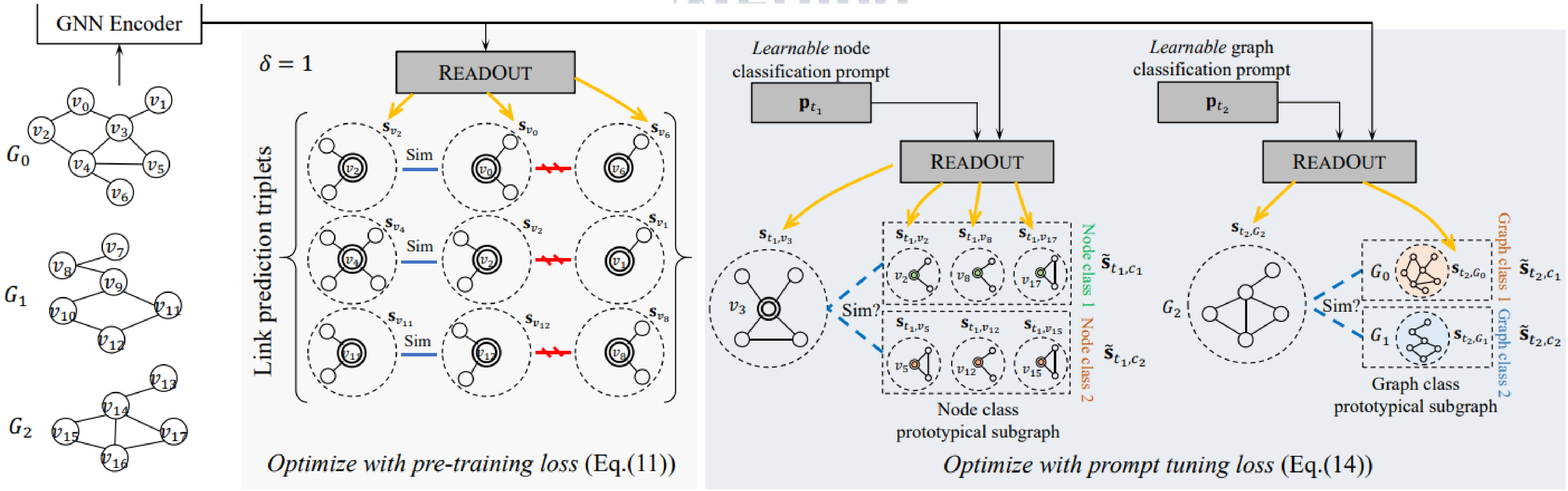
$$\tilde{\mathbf{s}}_c = \frac{1}{k} \sum_{(G_i,L_i)\in \mathcal{D}, L_i = c} \mathbf{s}_{G_i}. \tag{7}$$

$$L_j = \arg\max_{c\in C} \mathrm{sim}(\mathbf{s}_{G_j}, \tilde{\mathbf{s}}_c). \tag{8}$$

$$y = \arg\max_{c\in Y} \mathrm{sim}(\mathbf{s}_x, \tilde{\mathbf{s}}_c). \tag{9}$$

$$\mathbf{s}_x = \textsc{ReadOut}(\{\mathbf{h}_v : v \in V(S_x)\}). \tag{10}$$

Method



(a) Toy graphs (b) Pre-training (c) Prompting for node classification (left) or graph classification (right)

$$\mathcal{L}_{\text{pre}}(\Theta) = -\sum_{(v,a,b)\in\mathcal{T}_{\text{pre}}} \ln \frac{\exp(\text{sim}(\mathbf{s}_v, \mathbf{s}_a)/\tau)}{\sum_{u\in\{a,b\}} \exp(\text{sim}(\mathbf{s}_v, \mathbf{s}_u)/\tau)}, \quad (11)$$

$$\Theta_0 = \arg\min_\Theta \mathcal{L}_{\text{pre}}(\Theta)$$

$$\mathbf{s}_{t,x} = \text{READOUT}(\{\mathbf{p}_t \odot \mathbf{h}_v : v \in V(S_x)\}), \quad (12)$$

$$\mathbf{s}_{t,x} = \text{READOUT}(\{\mathbf{P}_t \mathbf{h}_v : v \in V(S_x)\}). \quad (13)$$

$$\mathcal{L}_{\text{prompt}}(\mathbf{p}_t) = -\sum_{(x_i,y_i)\in\mathcal{T}_t} \ln \frac{\exp(\text{sim}(\mathbf{s}_{t,x_i}, \tilde{\mathbf{s}}_{t,y_i})/\tau)}{\sum_{c\in Y} \exp(\text{sim}(\mathbf{s}_{t,x_i}, \tilde{\mathbf{s}}_{t,c})/\tau)}, \quad (14)$$

Note that, the prompt tuning loss is only parameterized by the learnable prompt vector $\mathbf{p}_t$, without the GNN weights.

Experiments

Table 1: Summary of datasets.

	Graphs	Graph classes	Avg. nodes	Avg. edges	Node features	Node classes	Task (N/G)
Flickr	1	-	89,250	899,756	500	7	N
PROTEINS	1,113	2	39.06	72.82	1	3	N, G
COX2	467	2	41.22	43.45	3	-	G
ENZYMES	600	6	32.63	62.14	18	3	N, G
BZR	405	2	35.75	38.36	3	-	G

Experiments

Table 2: Accuracy evaluation on node classification.

All tabular results are in percent, with best **bolded** and runner-up underlined.

Methods	Flickr 50-shot	PROTEINS 1-shot	ENZYMES 1-shot
GCN	9.22 ± 9.49	59.60 ± 12.44	61.49 ± 12.87
GraphSAGE	13.52 ± 11.28	59.12 ± 12.14	61.81 ± 13.19
GAT	16.02 ± 12.72	58.14 ± 12.05	60.77 ± 13.21
GIN	10.18 ± 5.41	60.53 ± 12.19	63.81 ± 11.28
DGI	17.71 ± 1.09	54.92 ± 18.46	63.33 ± 18.13
GraphCL	18.37 ± 1.72	52.00 ± 15.83	58.73 ± 16.47
GPPT	18.95 ± 1.92	50.83 ± 16.56	53.79 ± 17.46
GraphPrompt	**20.21** ± 11.52	**63.03** ± 12.14	**67.04** ± 11.48

Table 3: Accuracy evaluation on graph classification.

Methods	PROTEINS 5-shot	COX2 5-shot	ENZYMES 5-shot	BZR 5-shot
GCN	54.87 ± 11.20	51.37 ± 11.06	20.37 ± 5.24	56.16 ± 11.07
GraphSAGE	52.99 ± 10.57	52.87 ± 11.46	18.31 ± 6.22	57.23 ± 10.95
GAT	48.78 ± 18.46	51.20 ± 27.93	15.90 ± 4.13	53.19 ± 20.61
GIN	58.17 ± 8.58	51.89 ± 8.71	20.34 ± 5.01	57.45 ± 10.54
InfoGraph	54.12 ± 8.20	54.04 ± 9.45	20.90 ± 3.32	57.57 ± 9.93
GraphCL	56.38 ± 7.24	55.40 ± 12.04	28.11 ± 4.00	59.22 ± 7.42
GraphPrompt	**64.42** ± 4.37	**59.21** ± 6.82	**31.45** ± 4.32	**61.63** ± 7.68

Experiments

Table 4: Study of parameter efficiency on node classification.

Methods	Flickr		PROTEINS		ENZYMES	
	Params	FLOPs	Params	FLOPs	Params	FLOPs
GIN	22,183	240,100	5,730	12,380	6,280	11,030
GPPT	4,096	4,582	1,536	1,659	1,536	1,659
GraphPrompt	96	96	96	96	96	96
GraphPrompt-ft	21,600	235,200	6,176	13,440	6,176	10,944

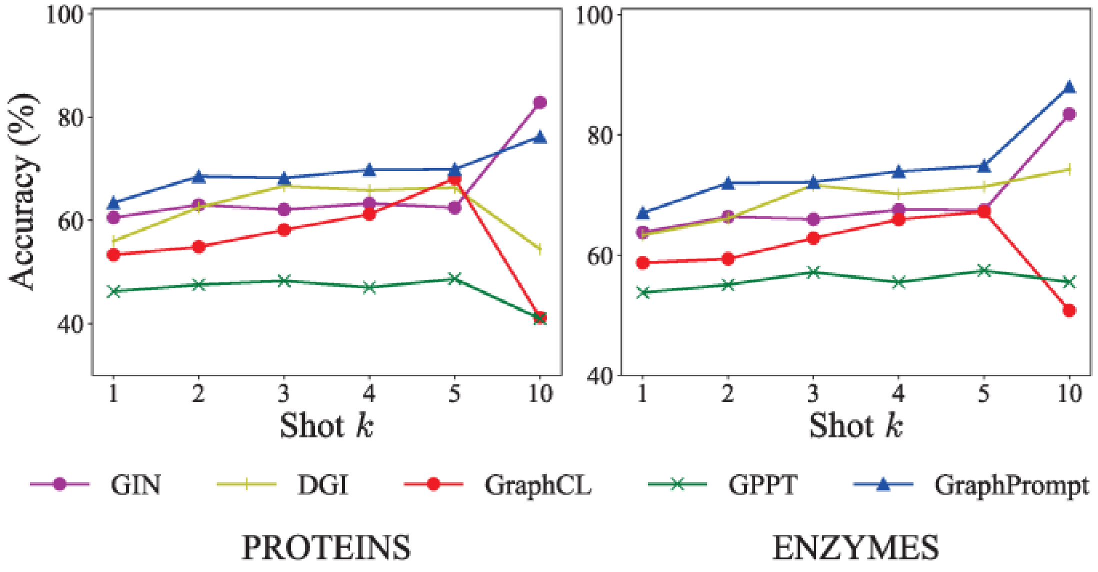


Figure 3: Impact of shots on few-shot node classification.

Experiments

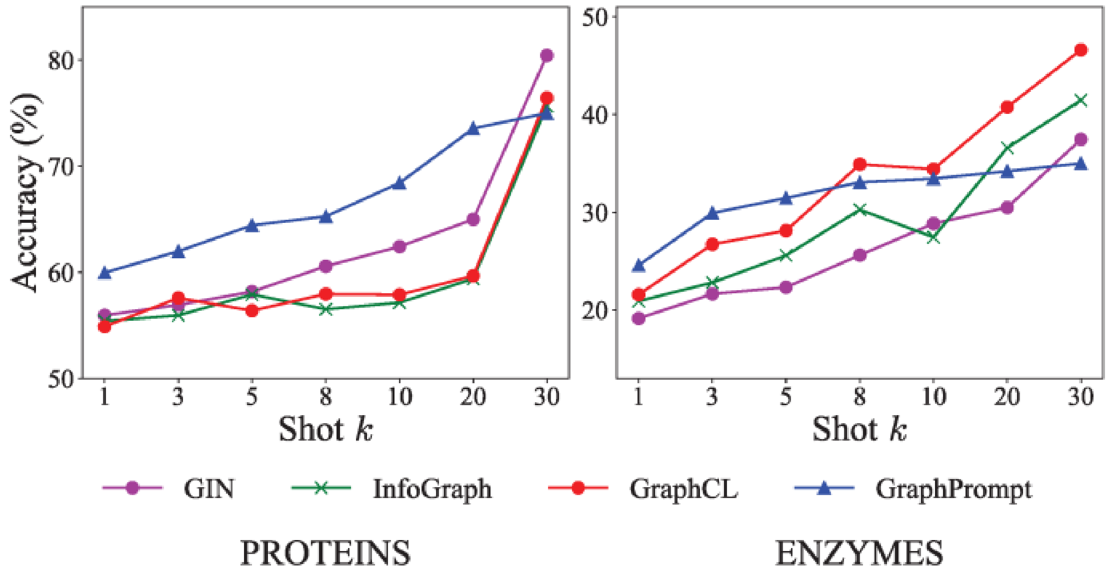


Figure 4: Impact of shots on few-shot graph classification.

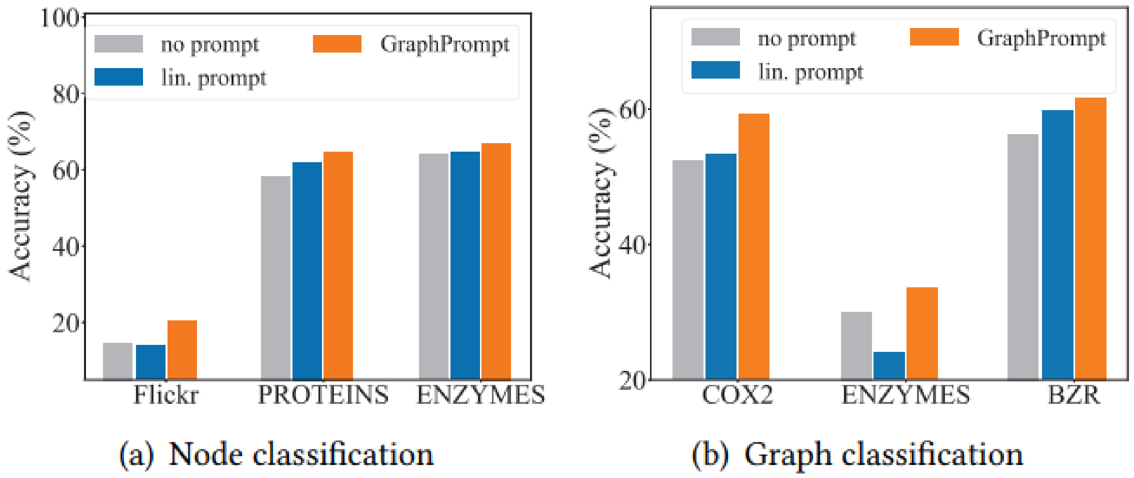


Figure 5: Ablation study.

Thanks