Non-Homophilic Graph Pre-Training and Prompt Learning

Xingtong Yu[1*], Jie Zhang[2*], Yuan Fang[1†], Renhe Jiang[3†]

[1] Singapore Management University, Singapore
[2] National University of Singapore, Singapore
[3] The University of Tokyo, Japan

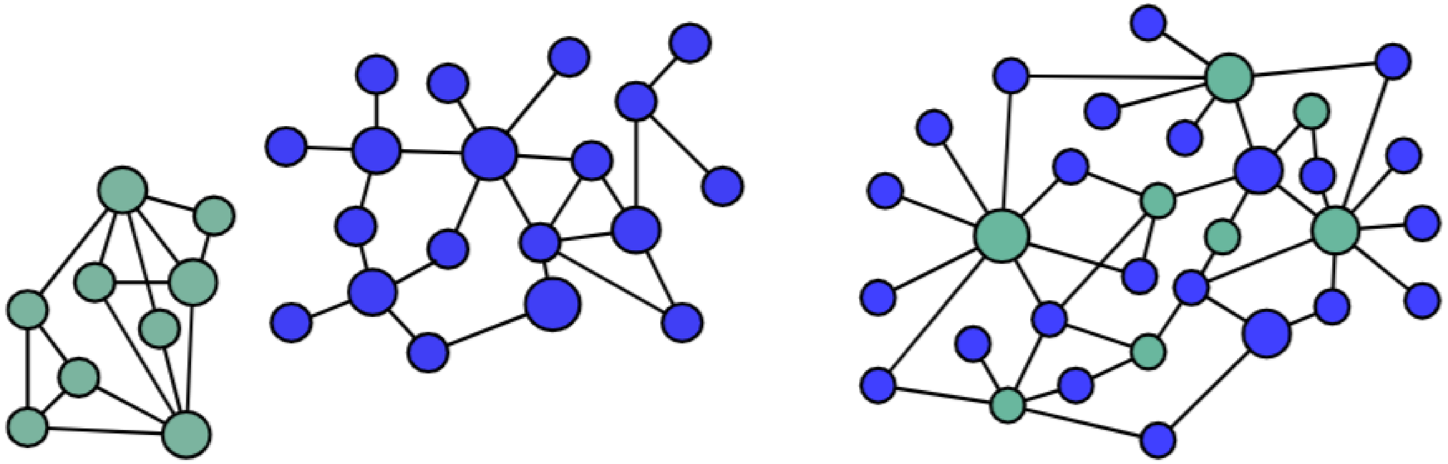
31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2025

* Co-first author
† Corresponding author

Outline

1. Motivation & Challenges
2. Theoretical Insights
3. Proposed Model: ProNoG
4. Experiments
5. Conclusions

Motivation & Challenges

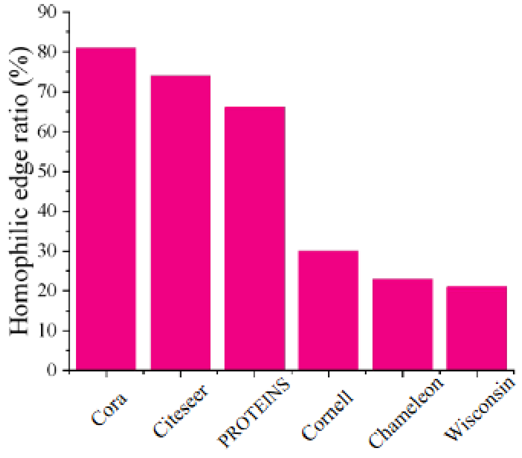


(a) Homophily (b) Heterophily

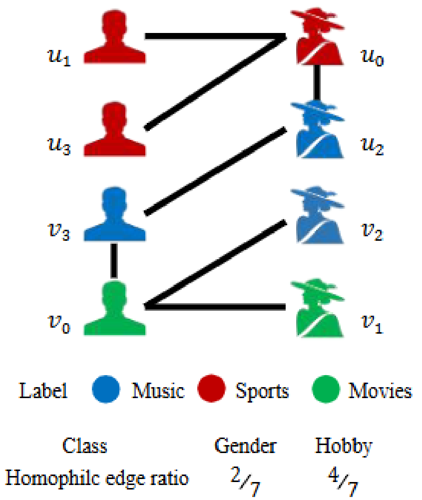
Real-world graphs are typically non-homophilic:

- They are neither strictly or uniformly homophilic;
- Mix both homophilic and heterophilic patterns.

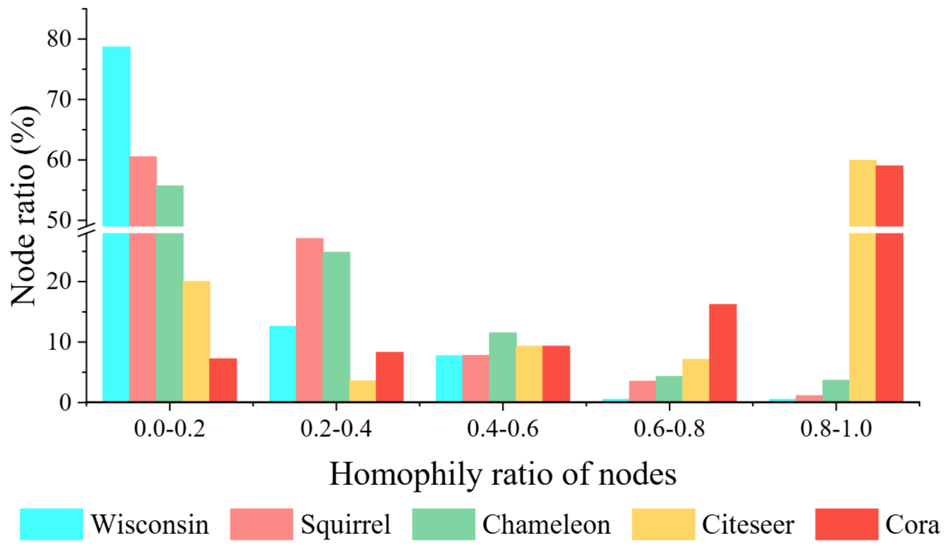
Motivation & Challenges



(a) Varying non-homophilic patterns across different graphs



(b) Dependence of homophiliy ratio on the target label



(c) Diverse non-homophilic patterns across nodes in the same graph

- *C1:* How do we pre-train a graph model irrespective of the graph's homophily characteristics?
- *C2:* How do we capture the fine-grained, node-specific non-homophilic characteristics?

Outline

1. Motivation & Challenges
2. Theoretical Insights
3. Proposed Model: ProNoG
4. Experiments
5. Conclusions

Theoretical Insights

Contrastive pre-training method loss function

$$\mathcal{L}_T = -\sum_{u \in V} \ln P(u, \mathcal{A}_u, \mathcal{B}_u), \tag{4}$$

$$P(u, \mathcal{A}_u, \mathcal{B}_u) \triangleq \frac{\sum_{a \in \mathcal{A}_u} \mathtt{sim}(\mathbf{h}_u, \mathbf{h}_a)}{\sum_{a \in \mathcal{A}_u} \mathtt{sim}(\mathbf{h}_u, \mathbf{h}_a) + \sum_{b \in \mathcal{B}_u} \mathtt{sim}(\mathbf{h}_u, \mathbf{h}_b)}, \tag{5}$$

Definition of homophily task

DEFINITION 1 (HOMOPHILY TASK). *On a graph $G = (V, E)$, a pre-training task $T = (\{\mathcal{A}_u : u \in V\}, \{\mathcal{B}_u : u \in V\})$ is a homophily task if and only if, $\forall u \in V, \forall a \in \mathcal{A}_u, \forall b \in \mathcal{B}_u, (u, a) \in E \wedge (u, b) \notin E$. A task that is not a homophily task is called a non-homophily task.* □

Table 6: Positive and negative samples for homophily and non-homophily methods.

Pre-training task	Positive instances $\mathcal{A}_u$	Negative instances $\mathcal{B}_u$	Homophily task
Link prediction [26, 62, 64]	a node connected to node u	nodes disconnected to node u	Yes
DGI [48]	nodes in graph G	nodes in corrupted graph G'	No
GraphCL [60]	an augmented graph from graph G	augmented graphs from $G' \neq G$	No
GraphACL [55]	nodes with similar ego-subgraph to node u	nodes with dissimilar ego-subgraph to node u	No

Theoretical Insights

Theorems

THEOREM 1. *For a homophily task T, adding a homophily sample always results in a smaller loss than adding a non-homophily sample.*

THEOREM 2. *Consider a graph $G = (V, E)$ with a label mapping function $V \longrightarrow Y$, and let $y_v \in Y$ denote the label mapped to $v \in V$. Suppose the label mapping satisfies that*

$$\forall u, a, b \in V, y_u = y_a \wedge y_u \neq y_b \Rightarrow \mathtt{sim}(u, a) > \mathtt{sim}(u, b).$$

Let $\mathbb{E}_T$ denote the expected number of homophily samples for a homophily task T on the graph G. Then, $\mathbb{E}_T$ increases monotonically as the homophily ratio $\mathcal{H}(G)$ defined w.r.t. Y increases.

Insights

For non-homophilic graphs, especially those with low homophily ratio, non-homophily tasks are a better choice compared to homophily tasks when optimizing the training loss.

Outline

1. Motivation & Challenges
2. Theoretical Insights
3. Proposed Model: ProNoG
4. Experiments
5. Conclusions

Proposed Method: ProNoG

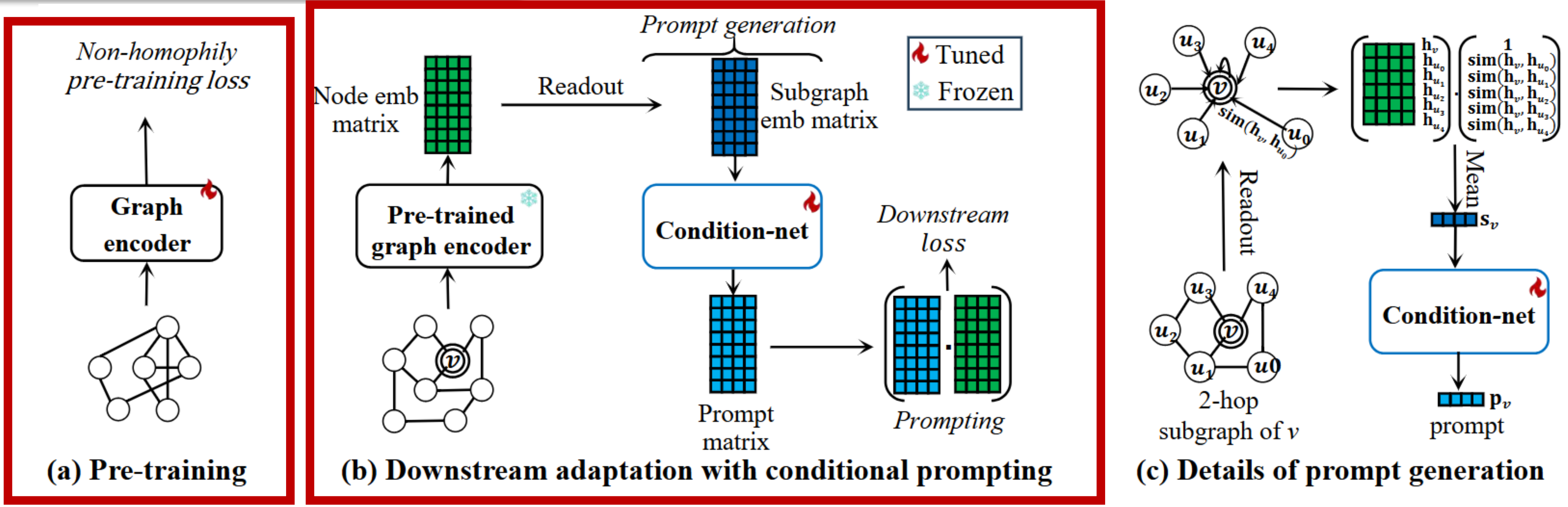


Figure 2: Overall framework of ProNoG.

Prompt generation

$$\mathbf{s}_v = \frac{1}{|S_v|} \sum_{u \in S_v} \mathbf{h}_u \cdot \mathtt{sim}(\mathbf{h}_u, \mathbf{h}_v),$$

$$\mathbf{p}_{t,v} = \mathtt{CondNet}(\mathbf{s}_v; \phi_t),$$

Prompt tuning

$$\tilde{\mathbf{h}}_{t,v} = \mathbf{p}_{t,v} \odot \mathbf{h}_v,$$

$$\mathcal{L}_{\text{down}}(\phi_t) = -\sum_{(x_i,y_i)\in\mathcal{D}_t} \ln \frac{\exp\left(\frac{1}{\tau}\text{sim}(\tilde{\mathbf{h}}_{t,x_i}, \bar{\mathbf{h}}_{t,y_i})\right)}{\sum_{c\in Y}\exp\left(\frac{1}{\tau}\text{sim}(\tilde{\mathbf{h}}_{t,x_i}, \bar{\mathbf{h}}_{t,c})\right)},$$

Outline

1. Motivation & Challenges
2. Theoretical Insights
3. Proposed Model: ProNoG
4. Experiments
5. Conclusions

Experiment

Table 2: Accuracy evaluation on few-shot node classification.

Methods	Wisconsin	Squirrel	Chameleon	Cornell	PROTEINS	ENZYMES	Citeseer	Cora
GCN	21.39 ± 6.56	20.00 ± 0.29	25.11 ± 4.19	21.81 ± 4.71	43.32 ± 9.35	48.08 ± 4.71	31.27 ± 4.53	28.57 ± 5.07
GAT	28.01 ± 5.40	21.55 ± 2.30	24.82 ± 4.35	23.03 ± 13.19	31.79 ± 20.11	35.32 ± 18.72	30.76 ± 5.40	28.40 ± 6.25
H2GCN	23.60 ± 4.64	21.90 ± 2.15	25.89 ± 4.96	32.77 ± 14.88	29.60 ± 6.99	37.27 ± 8.73	26.98 ± 6.25	34.58 ± 9.43
FAGCN	35.03 ± 17.92	20.91 ± 1.79	22.71 ± 3.74	28.67 ± 17.64	32.63 ± 9.94	35.87 ± 13.47	26.46 ± 6.34	28.28 ± 9.57
DGI	28.04 ± 6.47	20.00 ± 1.86	19.33 ± 4.57	32.54 ± 15.66	45.22 ± 11.09	48.05 ± 14.83	45.00 ± 9.19	54.11 ± 9.60
GraphCL	29.85 ± 8.46	21.42 ± 2.22	27.16 ± 4.31	24.69 ± 14.06	46.15 ± 10.94	48.88 ± 15.98	43.12 ± 9.61	51.96 ± 9.43
DSSL	28.46 ± 10.31	20.94 ± 1.88	<u>27.92</u> ± 3.93	20.36 ± 5.38	40.42 ± 10.08	<u>66.59</u> ± 19.28	39.86 ± 8.60	40.79 ± 7.31
GraphACL	<u>34.57</u> ± 10.46	<u>24.44</u> ± 3.94	26.72 ± 4.67	<u>33.17</u> ± 16.06	42.16 ± 13.50	47.57 ± 14.36	35.91 ± 7.87	46.65 ± 9.54
GPPT	27.39 ± 6.67	20.09 ± 0.91	24.53 ± 2.55	25.09 ± 2.92	35.15 ± 11.40	35.37 ± 9.37	21.45 ± 3.45	15.37 ± 4.51
GraphPrompt	31.48 ± 5.18	21.22 ± 1.80	25.36 ± 3.99	31.00 ± 13.88	<u>47.22</u> ± 11.05	53.54 ± 15.46	<u>45.34</u> ± 10.53	<u>54.25</u> ± 9.38
GraphPrompt+	31.54 ± 4.54	21.24 ± 1.82	25.73 ± 4.50	31.65 ± 14.48	46.08 ± 9.96	57.68 ± 13.12	45.23 ± 10.01	52.51 ± 9.73
ProNoG	**44.72** ± 11.93	**24.59** ± 3.41	**30.67** ± 3.73	**37.90** ± 9.31	**48.95** ± 10.85	**72.94** ± 20.23	**49.02** ± 10.66	**57.92** ± 11.50

Results are reported in percent. The best method is bolded and the runner-up is underlined.

Experiment

Table 3: Accuracy evaluation on few-shot graph classification.

Methods	Wisconsin	Squirrel	Chameleon	Cornell	PROTEINS	ENZYMES	BZR	COX2
GCN	21.39 ± 6.56	11.77 ± 3.10	17.21 ± 4.80	26.36 ± 4.35	51.66 ± 10.87	19.30 ± 6.36	45.06 ± 16.30	43.84 ± 13.94
GAT	24.93 ± 7.59	20.70 ± 1.51	25.71 ± 3.32	22.66 ± 12.46	51.33 ± 11.02	20.24 ± 6.39	46.28 ± 15.26	51.72 ± 13.70
H2GCN	22.23 ± 6.38	20.69 ± 1.42	<u>26.76</u> ± 3.98	23.11 ± 11.78	53.81 ± 8.85	19.40 ± 5.57	50.28 ± 12.13	53.70 ± 11.73
FAGCN	23.81 ± 9.50	20.83 ± 1.43	25.93 ± 4.03	25.71 ± 13.12	55.45 ± 11.57	19.95 ± 5.94	50.93 ± 12.41	50.22 ± 11.50
DGI	<u>29.77</u> ± 6.22	20.50 ± 1.52	24.29 ± 4.33	18.60 ± 12.79	50.32 ± 13.47	21.57 ± 5.37	49.97 ± 12.63	54.84 ± 14.76
GraphCL	27.93 ± 5.27	<u>21.01</u> ± 1.86	26.45 ± 4.30	20.03 ± 10.05	54.81 ± 11.44	19.93 ± 5.65	50.50 ± 18.62	47.64 ± 22.42
DSSL	22.05 ± 3.90	20.74 ± 1.61	26.19 ± 3.72	18.38 ± 10.63	52.73 ± 10.98	**23.14** ± 6.71	49.04 ± 8.75	54.23 ± 14.17
GraphACL	22.98 ± 5.89	20.80 ± 1.28	26.28 ± 3.93	<u>26.50</u> ± 17.18	**56.11** ± 13.95	20.28 ± 5.60	49.24 ± 17.87	49.59 ± 23.93
GraphPrompt	28.34 ± 3.89	**21.22** ± 1.80	26.51 ± 4.67	24.06 ± 13.71	53.61 ± 8.90	21.85 ± 6.17	50.46 ± 11.46	<u>55.01</u> ± 15.23
GraphPrompt+	26.95 ± 7.42	20.80 ± 1.45	26.03 ± 4.17	25.31 ± 7.65	54.55 ± 12.61	21.85 ± 5.15	**53.26** ± 14.99	54.73 ± 14.58
ProNoG	**31.54** ± 5.30	20.92 ± 1.37	**28.50** ± 5.30	**27.17** ± 9.58	<u>56.11</u> ± 10.19	<u>22.55</u> ± 6.70	<u>51.62</u> ± 14.27	**56.46** ± 14.57

Experiment

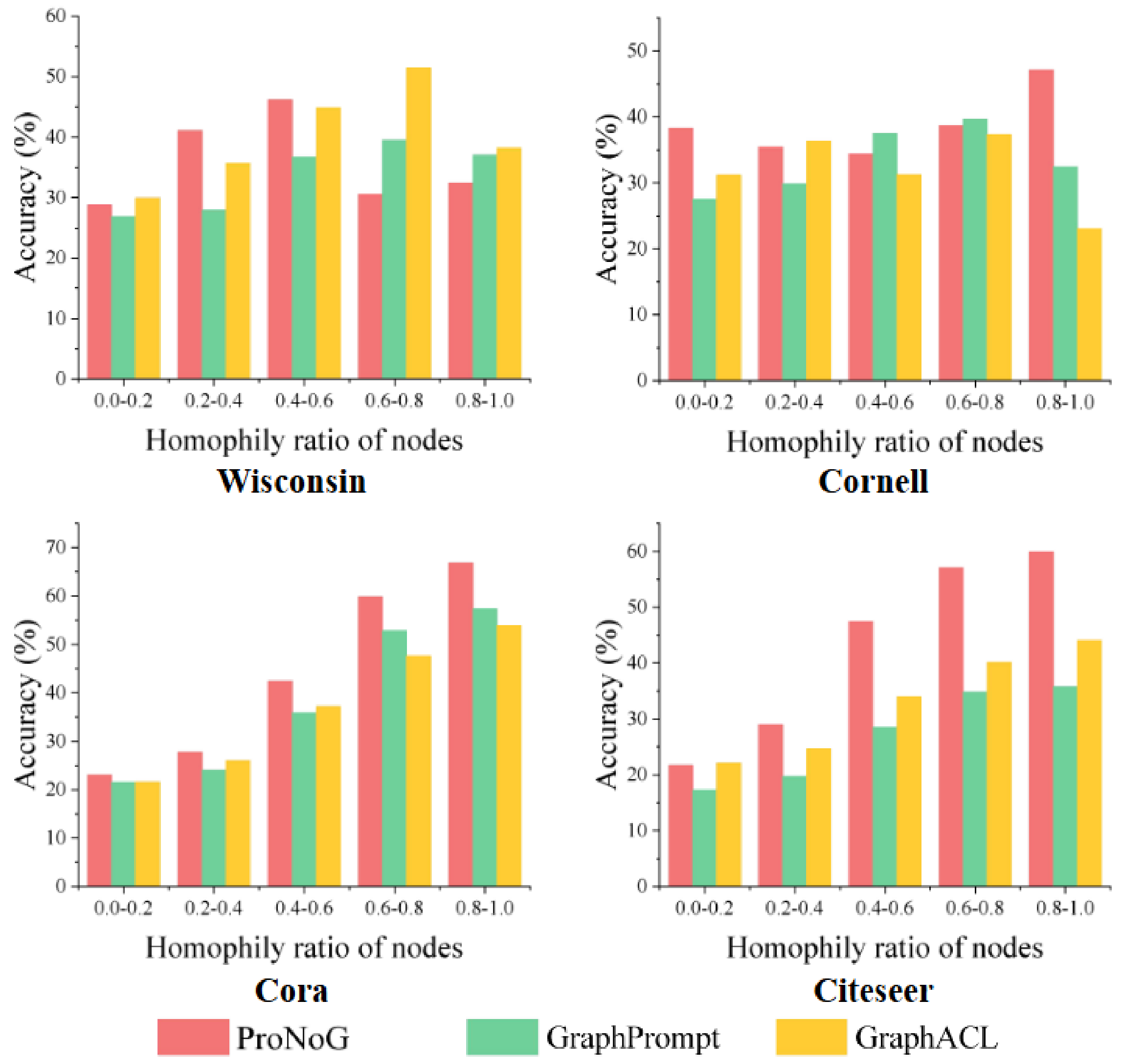


Figure 4: Results on different node patterns.

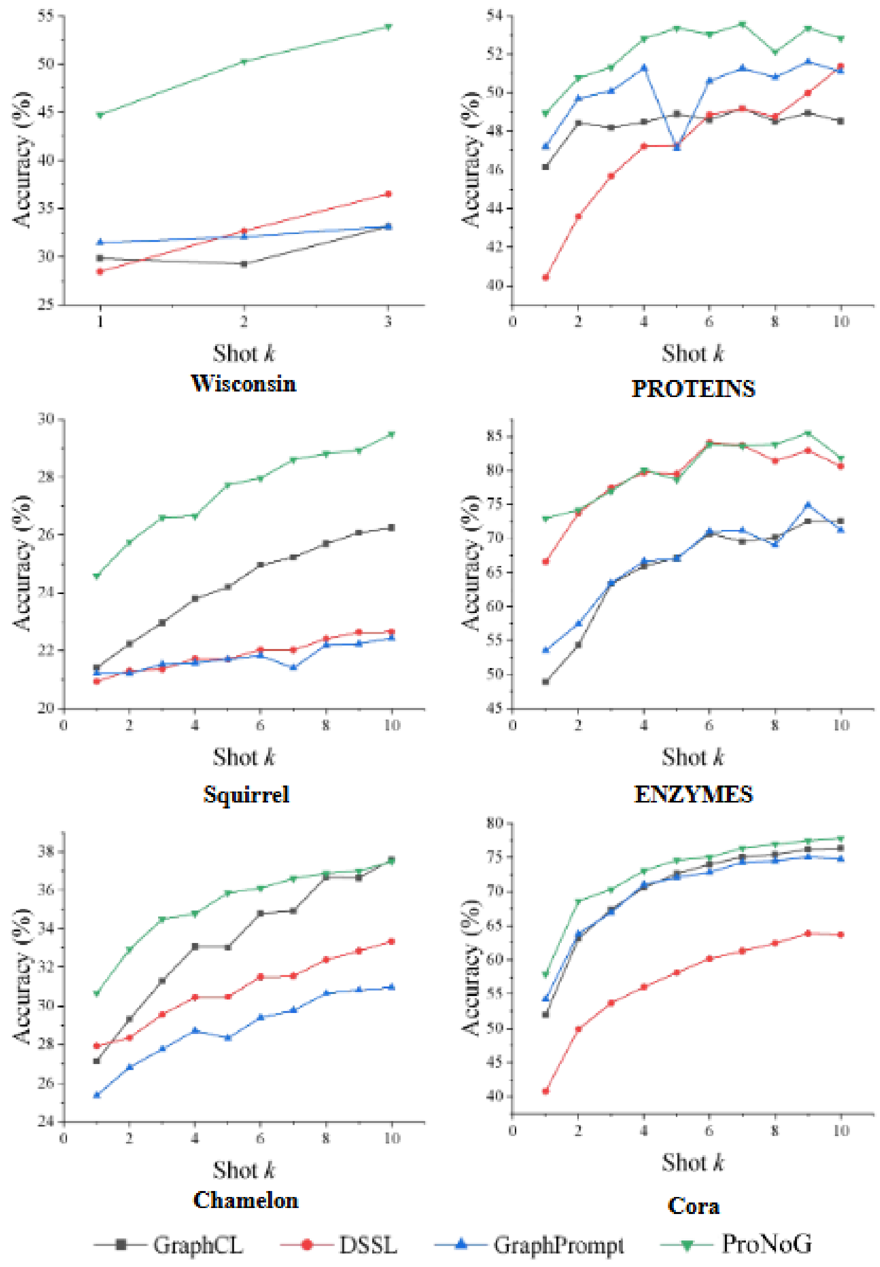


Figure 3: Impacts of different shots on node classification.

Conclusions

- We explored pre-training and prompt learning on non-homophilic graphs.
- We revisited graph pre-training on non-homophilic graphs, providing theoretical insights into the choice of pre-training tasks.
- For downstream adaptation, we proposed a condition-net to generate a series of prompts conditioned on node-specific non-homophilic patterns.
- We conducted extensive experiments showing that ProNoG significantly outperforms diverse state-of-the-art baselines.

Thank you! Questions?

- ProNoG paper & github repo:

GCoT: Chain-of-Thought Prompt Learning for Graphs

Xingtong Yu, Jie Zhang, Yuan Fang, Renhe Jiang

https://arxiv.org/abs/2408.12594

ProNoG

We provide the code (in pytorch) and datasets for our paper "Non-Homophilic Graph Pre-Training and Prompt Learning", which is accepted by SIGKDD 2025.

https://github.com/Jaygagaga/ProNoG