The 39th Annual AAAI Conference on Artificial Intelligence

Unlocking the Potential of Black-box Pre-trained GNNs for Graph Few-shot Learning

Qiannan Zhang[1], Shichao Pei[2], Yuan Fang[3], Xiangliang Zhang[4]

[1]Cornell University
[2]University of Massachusetts Boston
[3]Singapore Management University
[4]University of Notre Dame

Motivation

- Graph few-shot learning deals with label scarcity on graphs.
- Pre-trained GNNs have promoted the advancement of graph few-shot learning by providing rich graph knowledge via large-scale label-free training.
- For economic and security considerations, pre-trained GNNs might be accessed only via Model-as-a-Service (Black-box).

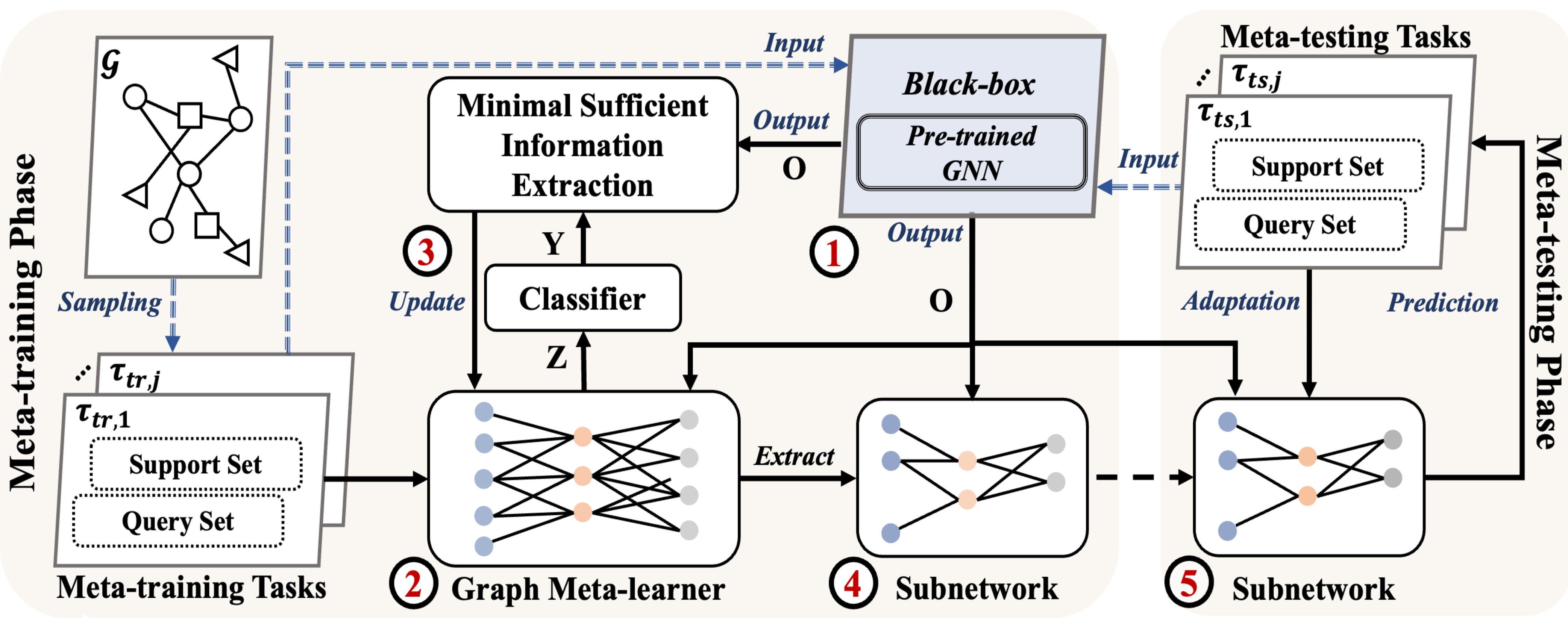
Challenges

- When applying pre-trained GNNs in few-shot learning:
  - ❖ *Task Gap:* Pre-training objectives may include task-irrelevant information for downstream few-shot tasks.
  - ❖ *Black-box Setting:* Conventional fine-tuning isn't applicable without access to internal gradients.
  - ❖ *Overfitting:* Limited meta-training tasks can lead to memorization and poor generalization

Key Idea

- Leverage a lightweight graph meta-learner, enabling:
  - ✓ Capturing just task-relevant knowledge needed for downstream node classification
  - ✓ Leveraging outputs from black-box pre-trained GNNs, no access to the gradients or parameters
  - ✓ Fast adapting to novel few-shot tasks with a compact subnetwork.

Graph Meta-learning with Black-box Pre-trained GNNs (Meta-BP)



② Graph Meta-Learner (GML)

- Receives output representations (denoted as ***O***) from the black-box pre-trained GNN as the initial embeddings of GML
- Pre-computes *neighbor abstractions* with graph topology

$$\mathbf{h}_{\mathcal{N}_v} = \frac{1}{|\mathcal{N}_v|}\sum_{u\in\mathcal{N}_v}\mathbf{h}_u$$

- Fuses node features and neighbor abstractions by linear transformation

$$\mathbf{z}_v = \sigma([\mathbf{h}_v||\mathbf{h}_{\mathcal{N}_v}]\mathbf{W})$$

④ Meta-Learner Pruning

- Extracts a *sparse subnetwork* from GML based on *capacity ratio c*
  - ○ deals with overparameterization and overfitting the insufficient meta-training tasks
  - ○ inspired by the *Lottery Ticket Hypothesis* to learn a binary mask of GML parameters

$$\mathcal{L}_S = \min_{\mathbf{m}}\frac{1}{B}\sum_{j=1}^{B}\{\frac{1}{b}\sum_{i=1}^{b}(\mathcal{L}_{CE}(f_C(\mathrm{GML}(\mathbf{o}_i,\mathcal{G};\phi_j\odot\mathbf{m})),\hat{y}_i) - C_j)\}.$$

$$C_j = \mathcal{L}_{CE}(f_C(\mathrm{GML}(\mathbf{o}_i,\mathcal{G};\phi_j)),\hat{y}_i)$$

$$\text{subject to} \quad |\mathbf{m}^*| \le c\cdot|\phi|$$

③ Extracting Minimal Sufficient Information

- Outputs include both task-relevant and task-irrelevant information.
- *Information bottleneck:* learning node representations ***Z*** to be:
  - ○ a *minimal* knowledge (simplest mapping) from the pre-trained GNN
  - ○ yet *sufficient* to adapt well to meta-tasks

$$\mathcal{L}_I = \min_{\mathbf{Z}\sim\mathrm{GML}(\cdot)} I(\mathbf{O};\mathbf{Z}) - \beta I(\mathbf{Z};\mathbf{Y})$$

$$= \min_{\mathbf{Z}\sim\mathrm{GML}(\cdot)} I(\mathbf{O};\mathbf{Z}) - \beta H(\mathbf{Y}) + \beta H(\mathbf{Y}|\mathbf{Z})$$

constant classification loss

- *Neural estimation:* using neural networks to efficiently estimate *I(O, Z)* alongside meta-optimization

$$I(\mathbf{O};\mathbf{Z}) \approx \frac{1}{b}\sum_{i=1}^{b}T_\theta(\mathbf{o}_i,\mathbf{z}_i) - \log(\frac{1}{b}\sum_{i=1}^{b}e^{T_\theta(\mathbf{o}_i,\bar{\mathbf{z}}_i)})$$

meta-task samples shuffled order

Optimization

- $\mathcal{L}_\mathrm{I}$: Minimal sufficient *information extraction* during meta-optimization
- $\mathcal{L}_\mathrm{s}$: *Subnetwork* extraction during meta-optimization
- Total loss:

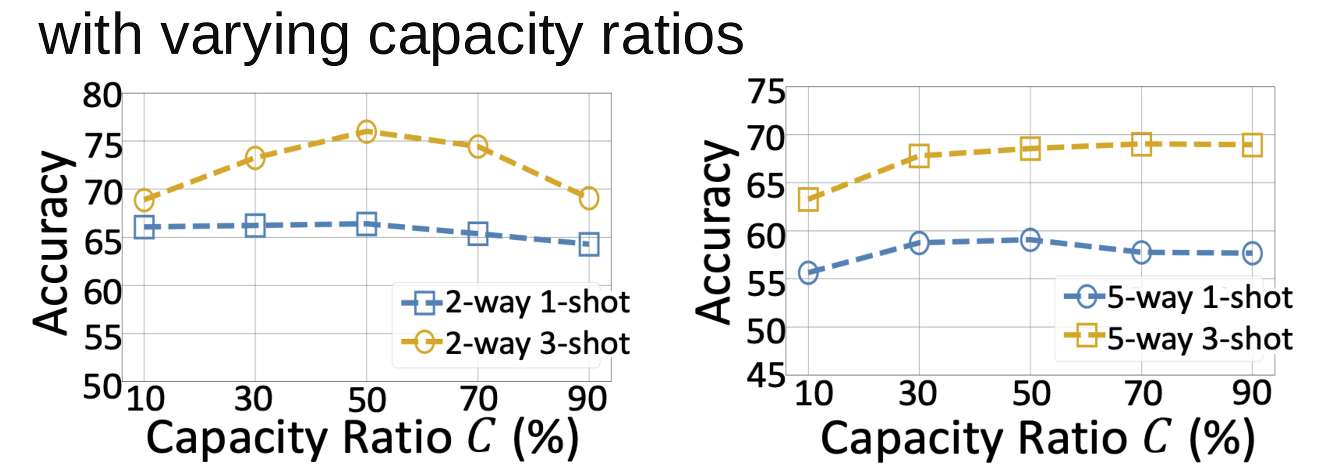
$$\mathcal{L}_{Meta} = \mathcal{L}_I + \alpha\mathcal{L}_S$$

Experiments

- Evaluated on four real-world graph datasets
- For node classification under various N-way K-shot settings

Methods	Cora	Computers	Cora-full	
	2-way	3-way	5-way	10-way
	1-shot			
GCN	55.21(±5.64)	37.33(±3.91)	43.75(±2.92)	31.26(±3.29)
GraphSage	58.33(±5.22)	39.98(±5.17)	44.26(±2.64)	32.54(±3.53)
GMI	60.25(±4.32)	62.28(±4.92)	56.81(±1.42)	40.98(±1.72)
DGI	61.56(±4.46)	64.52(±5.03)	56.52(±1.53)	40.36(±1.76)
PN	52.60(±5.23)	47.63(±5.23)	48.25(±1.80)	35.65(±1.98)
MAML	53.66(±4.92)	66.05(±5.16)	58.38(±2.01)	38.72(±2.19)
Meta-SGC	57.72(±5.99)	67.40(±6.79)	61.34(±4.53)	41.29(±4.06)
GPN	57.22(±4.20)	63.78(±5.27)	54.36(±2.29)	43.27(±1.92)
G-Meta	62.24(±4.93)	67.22(±5.61)	55.21(±2.15)	46.23(±1.79)
TLP	61.11(±3.14)	65.05(±5.40)	61.28(±2.41)	48.12(±1.53)
TENT	61.25(±5.15)	66.24(±5.24)	62.42(±2.16)	47.95(±1.88)
TEG	63.14(±4.43)	67.58(±5.11)	63.73(±2.08)	48.36(±2.03)
Meta-BP	**66.38**(±5.01)	**69.35**(±4.28)	**66.05**(±1.46)	**51.41**(±1.91)

Parameter sensitivity: training Meta-BP with varying capacity ratios



(a) Cora (b) OGBN-arxiv

Pretrained model impact: performance varies with different pre-trained backbones

Methods	Cora-full			
	5-way 1-shot	5-way 3-shot	10-way 1-shot	10-way 3-shot
Meta-BP-GMI	66.13(±1.48)	75.58(±1.23)	51.88(±1.44)	62.28(±1.86)
Meta-BP-BGRL	65.92(±1.62)	73.64(±1.42)	51.36(±1.65)	59.47(±1.94)
Meta-BP-DGI	66.05(±1.46)	72.98(±1.86)	51.41(±1.91)	57.79(±2.16)

Parameter efficiency: the number of parameters shows the extracted final subnetwork for adaptation is much smaller

Methods	Meta-PreGNN	Meta-BP
Computers	3-way	
#Total params	263.16 K	313.56 K
#Trainable Params in meta-training	263.16 K	50.40 k
#Params of GML	-	8.24 K
#Tunable params in meta-testing	263.16 K	2.47 K
Ratio of tunable params w.r.t. Meta-PreGNN	100%	0.94%

Weill Cornell Medicine
UMass Boston
Singapore Management University
University of Notre Dame