Meta-Inductive Node Classification across Graphs

Zhihao Wen, Yuan Fang, Zemin Liu

School of Computing and Information Systems

Singapore Management University

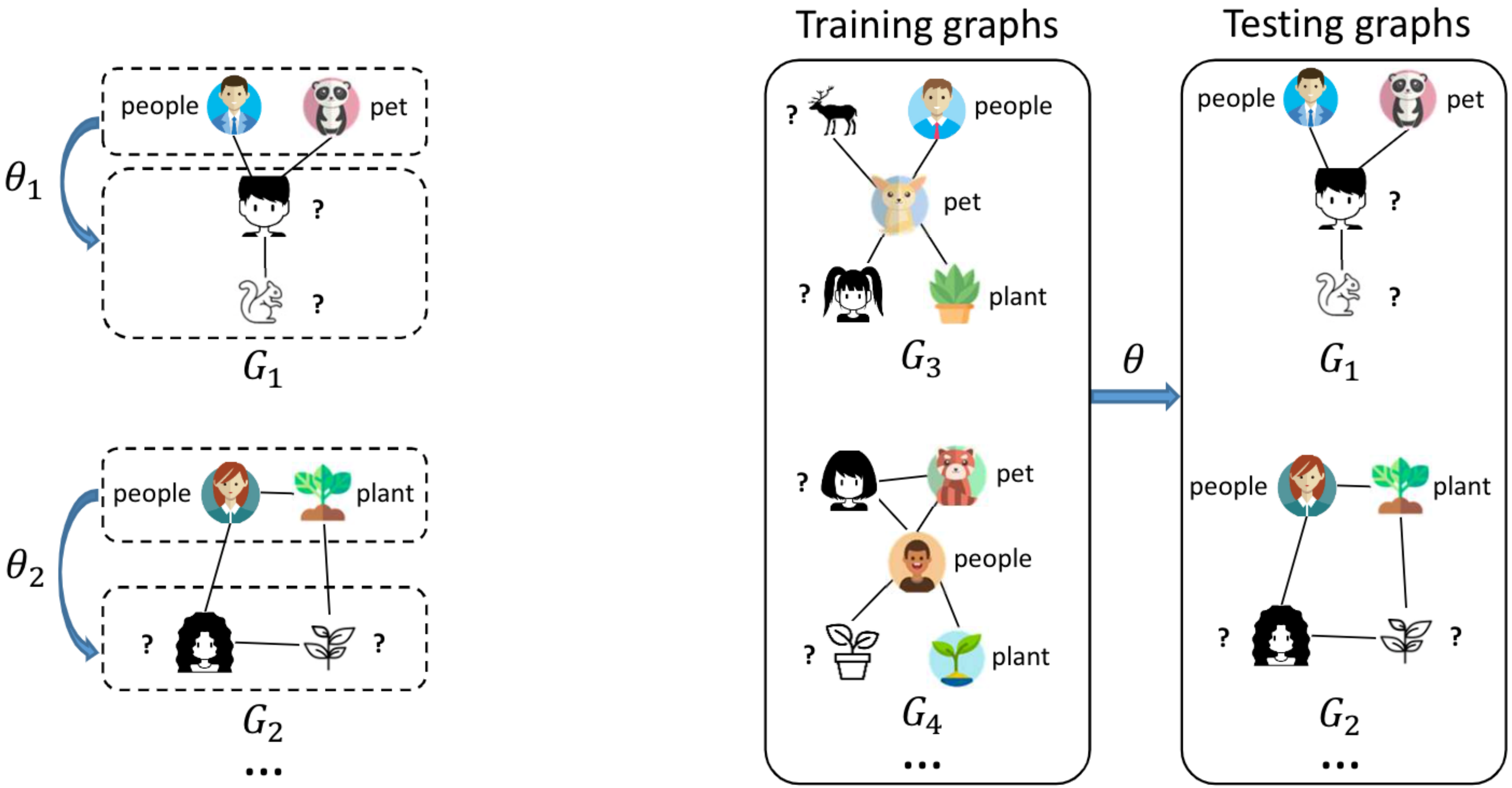
SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and
Information Systems

Outline

- ❑ **Introduction**
- ❑ Methodology
- ❑ Experiment
- ❑ Conclusion & Future Work

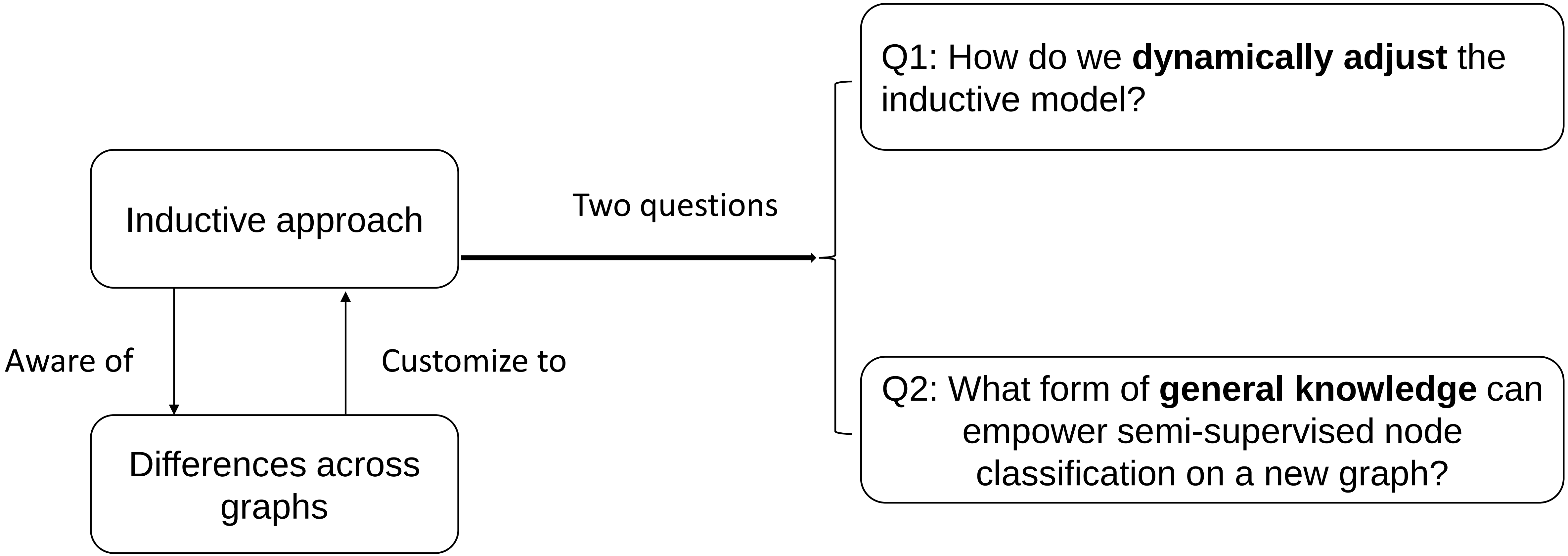
Semi-supervised node classification on graphs



(a) Transductive approach

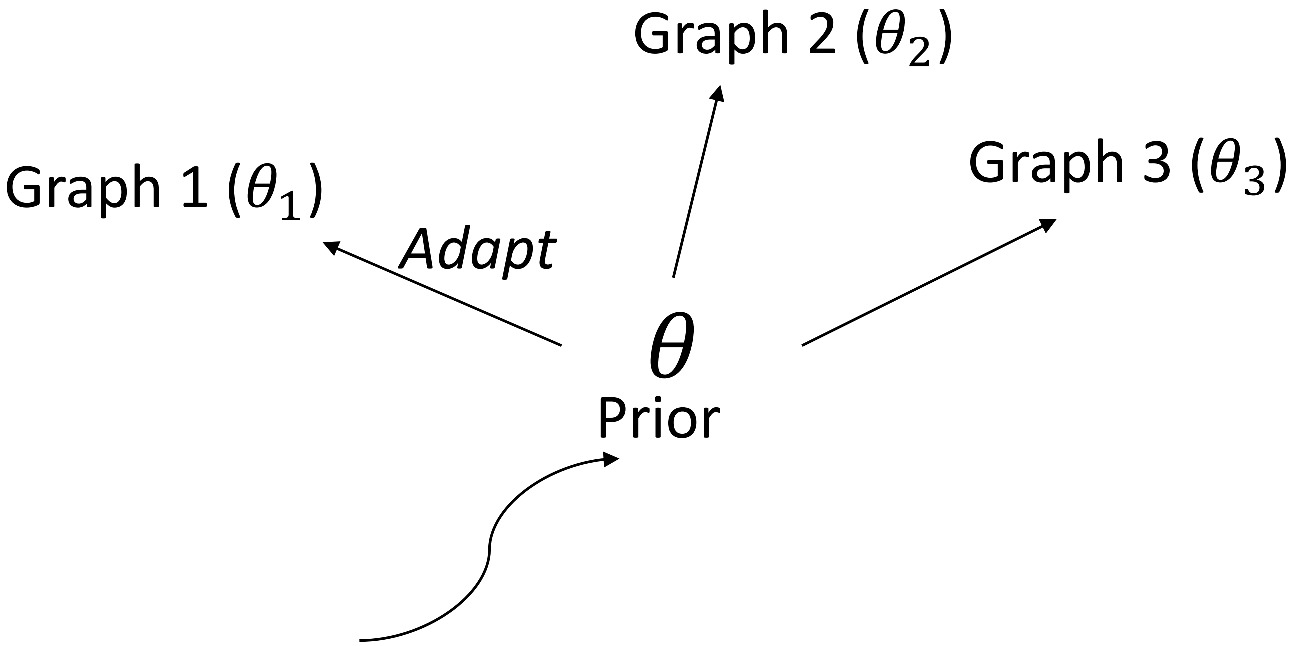
(b) Conventional inductive approach

Two challenges of inductive semi-supervised node classification (Q1 & Q2)



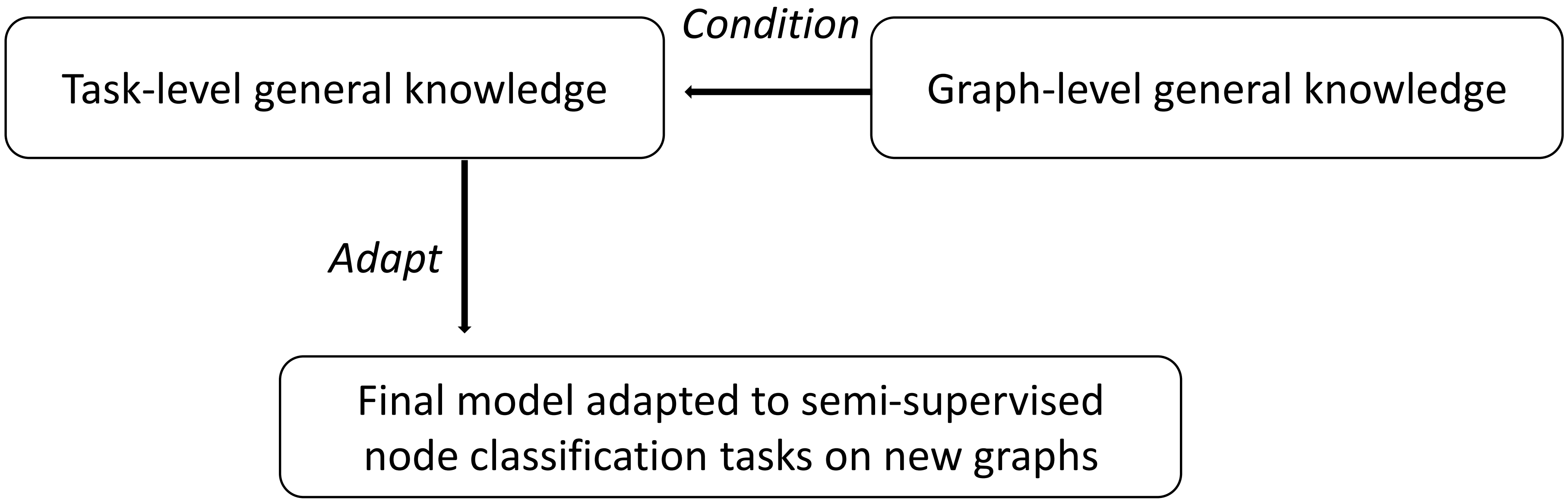
Q1: How to dynamically adjust the inductive model?

We propose a **learning-to-train** framework.



Learn a prior that can be adapted to semi-supervised node classification on different graphs (an instance of **meta-learning**)

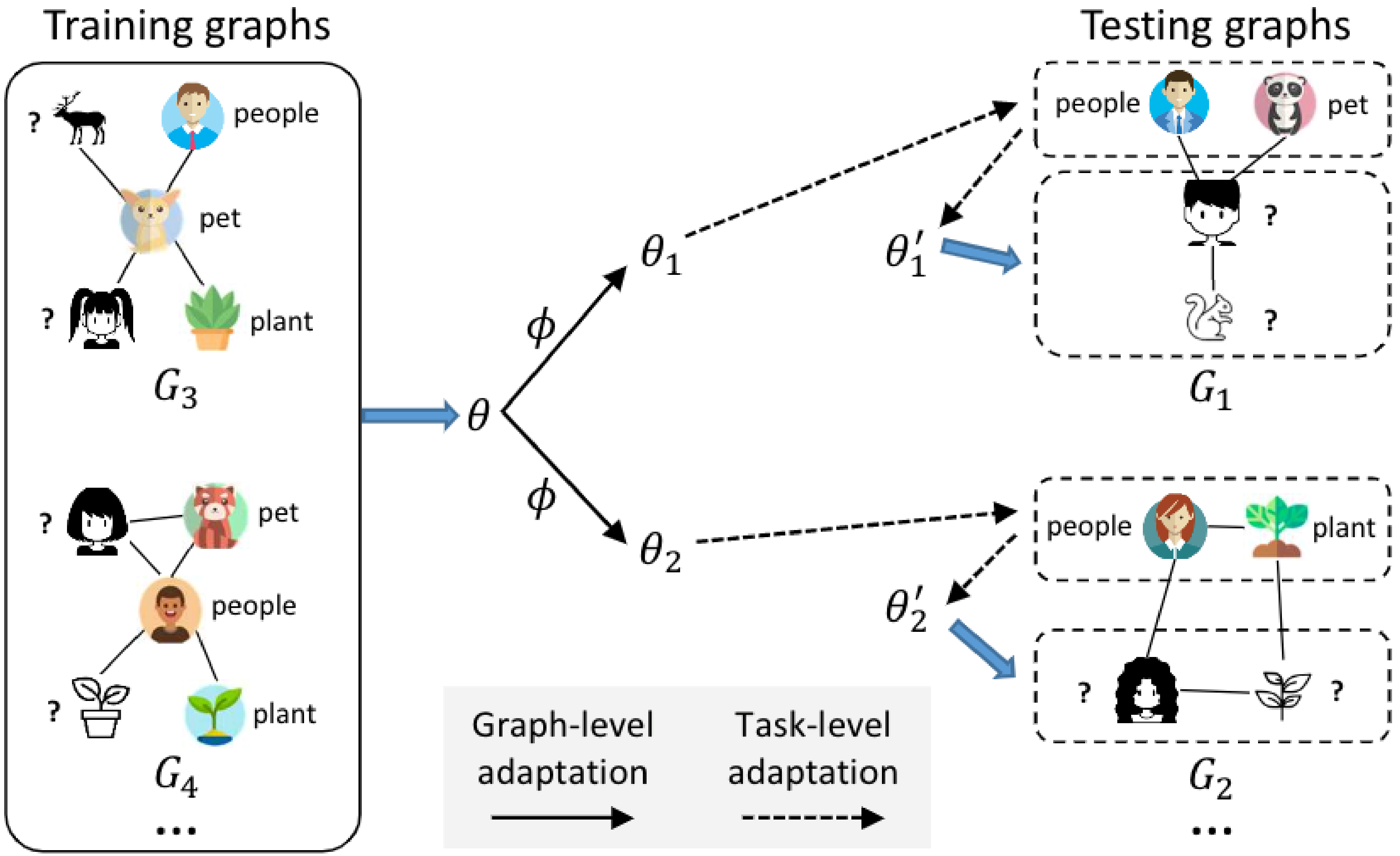
Q2: What form of general knowledge (i.e. prior)?



Outline

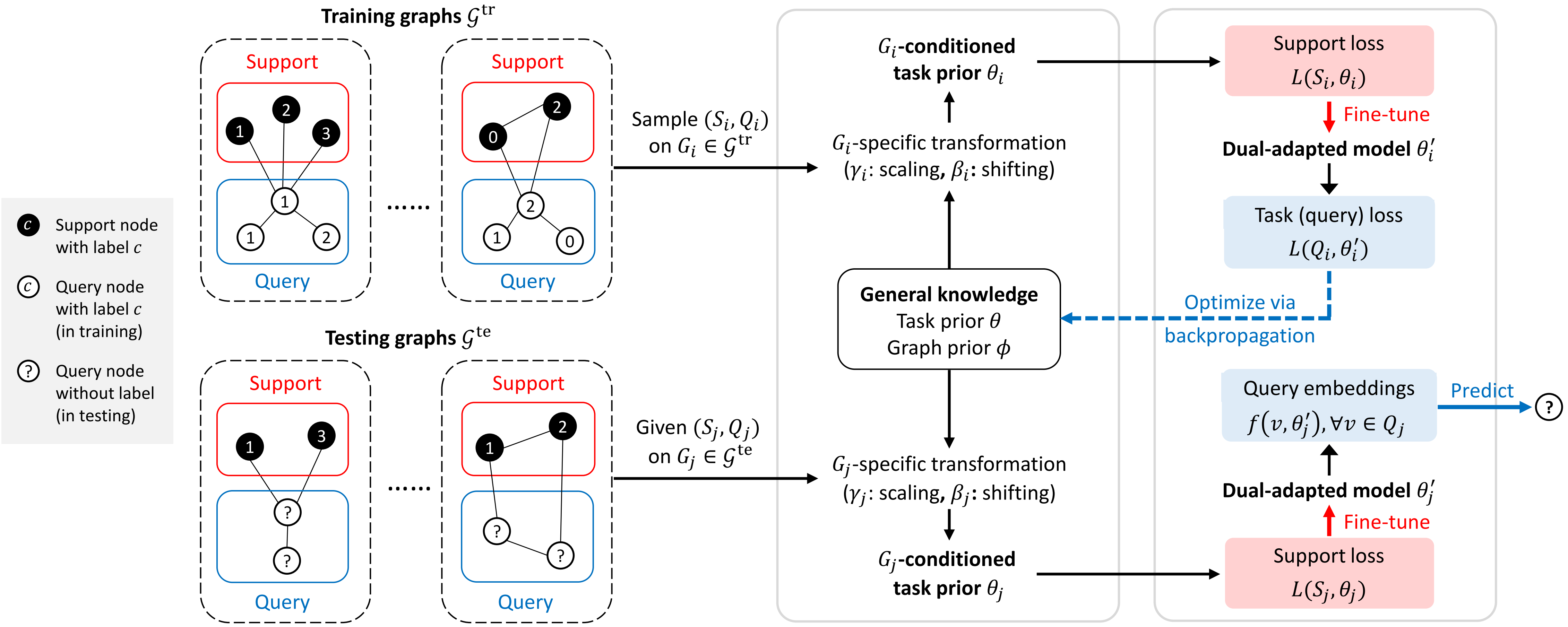
- ❑ Introduction & Challenge
- ❑ **Methodology**
- ❑ Experiment
- ❑ Conclusion & Future Work

MI-GNN: Meta-Inductive GNN



Our meta-inductive approach (MI-GNN)

Overall framework



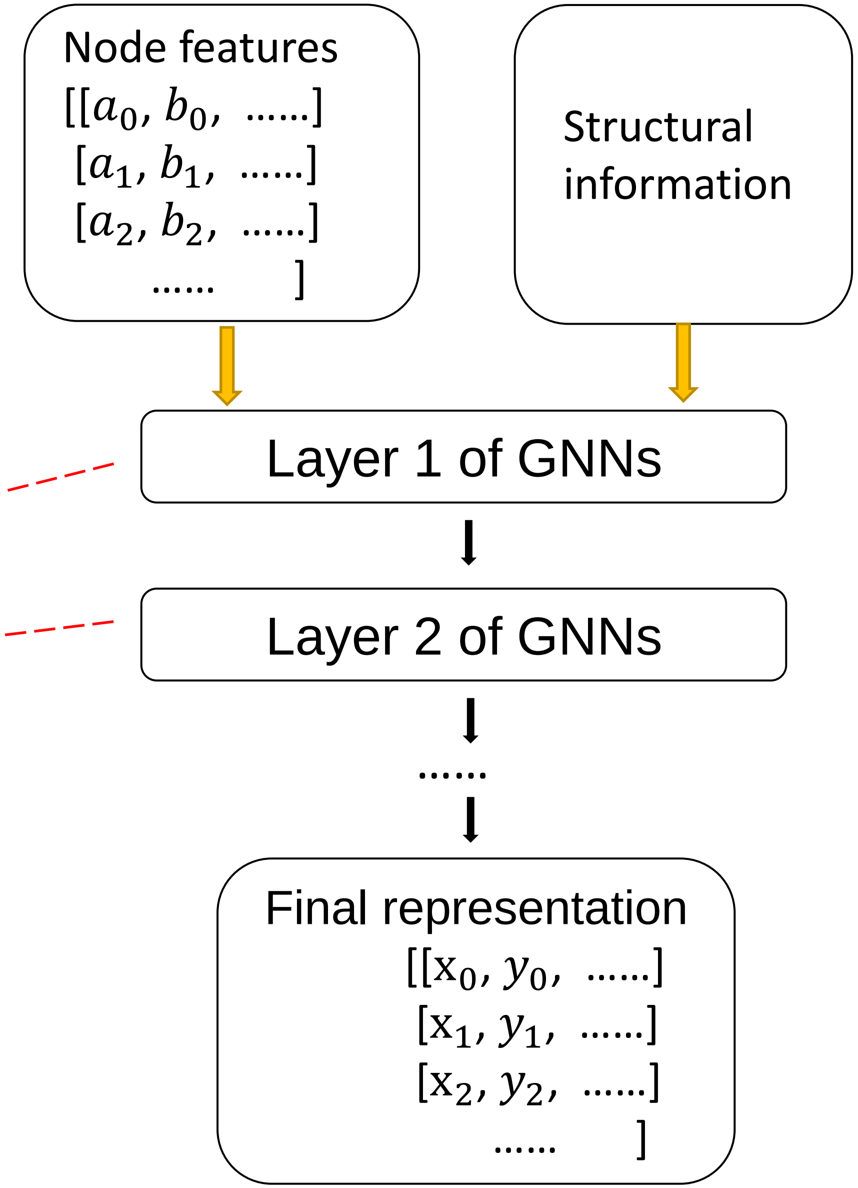
(a) Training/testing graphs and tasks (b) Graph-level adaptation (c) Task-level adaptation

Task prior

Our task prior $\boldsymbol{\theta}$ takes the form of

GNNs model, i.e.,

$$\theta = (W^1, W^2, \ldots \ldots)$$

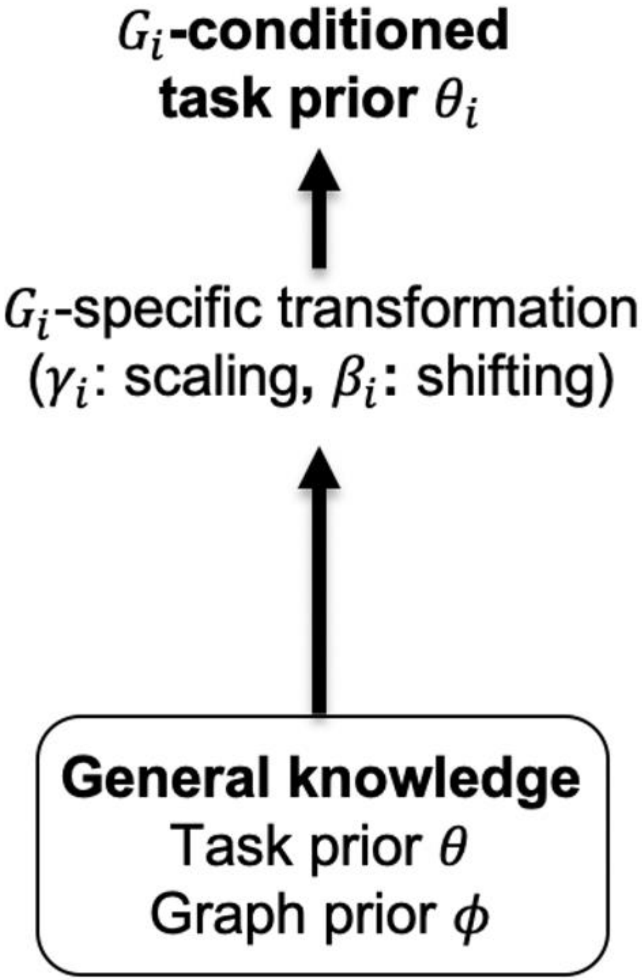


Graph-level adaptation

- Graph prior $\boldsymbol{\phi}$.
  - We employ a graph prior $\boldsymbol{\phi}$ to condition the task prior $\boldsymbol{\theta}$

Graph-conditioned task prior

$$\theta_i = \tau(\theta, \mathrm{g}; \phi) = (\gamma_\mathrm{i} + \mathbf{1}) \circ \theta + \beta_\mathrm{i}$$



Graph conditioned transformation

- We use MLPs to generate the scaling and shifting vector γ_i & β_i

Both are vectors

$$\gamma_\mathrm{i} = MLP_\gamma(\mathrm{g}_i; \phi_\gamma)$$

$$\beta_\mathrm{i} = MLP_\beta(\mathrm{g}_i; \phi_\beta)$$

Learnable parameters of the two MLPs

g_i is a graph-level representation of graph i

Graph-conditioned task prior

$$\theta_i = \tau(\theta, \mathrm{g}; \phi) = (\gamma_\mathrm{i} + \mathbf{1}) \circ \theta + \beta_\mathrm{i}$$

Graph-level representation

$\phi = (\phi_\gamma, \phi_\beta)$

Vector of **ones**

Element-wise multiplication & addition

Task-level adaptation

G_i**-conditioned task prior** θ_i → Support loss $L(S_i, \theta_i)$ → Gradient steps → **Dual-adapted model** θ_i'

$$\theta_i' = \theta_i - \alpha \frac{\partial L(S_i, Q_i)}{\partial \theta_i}$$

task-level adaptation learning rate

cross-entropy classification loss of the support set S_i

Overall training objective

- After the dual adaptions on G_i, the goal is to **optimize the general knowledge** and the optimal $(\theta\ ,\phi)$ are given by

a hyper-parameter

$$\underset{\theta\ ,\phi}{\arg\min} \sum_{G_i \in \mathcal{G}^{tr}} \left(L(Q_i, \tau'(\theta, \mathrm{g}; \phi)) \right) + \lambda(\| \gamma_i \|_2 + \| \beta_i \|_2))$$

Task cross-entropy loss on query nodes Q_i

Dual-adapted model (θ_i')

L_2 regularization on scaling & shifting

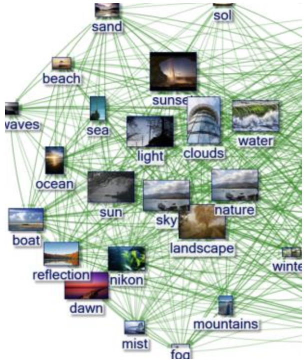
Outline

- ❑ Introduction & Challenge
- ❑ Methodology
- ❑ **Experiment**
- ❑ Conclusion & Future Work

Datasets

Dataset	Flickr	Yelp	Cuneiform	COX2	DHFR
# Graphs	800	800	267	467	756
# Edges (avg.)	13.1	43.5	20.1	44.8	44.5
# Nodes (avg.)	12.5	6.9	21.3	41.2	42.4
# Node features	500	300	3	3	3
# Node classes	7	10	7	8	9
Multi-label?	No	Yes	Yes	No	No

(All 5 datasets are from PyTorch Geometric Datasets: https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html)



https://www.smrfoundation.org/nodexl/automation/flickr-data-recipes/

https://www.yelp.com/dataset

Walker C B F. Cuneiform[M]

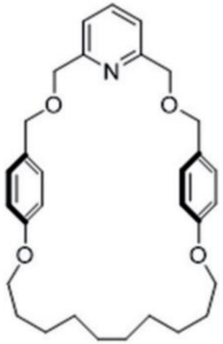
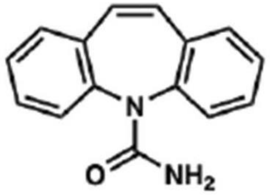


Figure: Fielden S D P, Leigh D A, Woltering S L. Molecular knots[J]



Service R F. Molecular CT scan could speed drug discovery [J]

Performance comparison with baselines

We report the average accuracy and micro-F1 with 95% confidence interval, in percent. In each column, the best result is bolded and the runner-up is underlined. Improvement by MI-GNN is calculated relative to the best baseline. ***/**/* denotes the difference between MI-GNN and the best baseline is statistically significant at the 0.01/0.05/0.1 level under the two-tail t-test.

		Flickr		Yelp		Cuneiform		COX2		DHFR	
		Accuracy	Micro-F1	Accuracy	Micro-F1	Accuracy	Micro-F1	Accuracy	Micro-F1	Accuracy	Micro-F1
Transductive	DeepWalk	39.88±2.42	30.01±1.21	63.27±2.73	57.11±6.29	74.61±0.60	27.05±2.11	37.68±0.73	26.16±1.08	33.14±0.18	29.93±0.58
	Transduct-GNN	13.61±1.22	10.71±1.20	24.87±15.4	23.85±14.6	49.63±0.95	34.00±1.15	13.23±0.17	9.73±0.22	11.21±0.33	8.65±0.22
Inductive	Planetoid	14.78±8.75	8.72±3.07	53.12±2.38	46.29±3.55	53.14±5.49	30.22±5.83	11.81±7.41	10.58±8.79	17.35±11.1	9.62±9.63
	Induct-GNN	40.48±1.69	29.67±1.77	65.95±0.56	56.61±1.81	74.89±0.35	18.03±0.93	53.71±0.92	41.56±1.90	45.23±0.62	29.38±6.07
	K-NN	34.11±1.76	26.39±1.39	61.70±0.90	57.35±1.42	70.36±0.27	35.66±0.84	33.16±0.95	32.84±1.00	36.32±0.89	27.12±1.20
	AGF	40.58±1.61	28.99±2.09	65.96±0.54	56.64±1.83	74.89±0.37	18.00±0.94	53.97±0.79	42.00±1.62	44.85±0.56	29.08±5.96
Meta learning	GFL	30.24±0.68	29.51±0.69	61.62±0.97	58.88±2.03	63.72±0.37	38.30±0.84	29.25±0.73	25.53±0.94	30.24±0.68	29.51±0.69
	Meta-GNN	39.66±0.92	30.02±2.49	66.24±0.84	56.20±1.81	75.12±0.33	19.21±1.25	53.24±0.77	37.36±3.02	**45.61**±0.65	28.34±4.46
	MI-GNN (improv.)	**44.45**±2.18 (+9.53%) **	**33.79**±1.87 (+12.57%) **	**67.92**±0.69 (+2.54%) ***	**60.20**±2.23 (+2.23%) ***	**81.48**±0.47 (+8.47%) ***	**43.32**±1.49 (+13.10%) ***	**57.27**±0.80 (+6.11%) ***	**44.66**±2.01 (+6.34%) *	45.19±0.70 (-0.92%)	**49.93**±1.62 (+66.82%) ***

Alternative GNN architectures

Performance comparison of L2T-GNN and a few key baselines, whilst using **GCN** and **GraphSAGE** as the GNN architecture. We report the average accuracy with **95% confidence interval**.

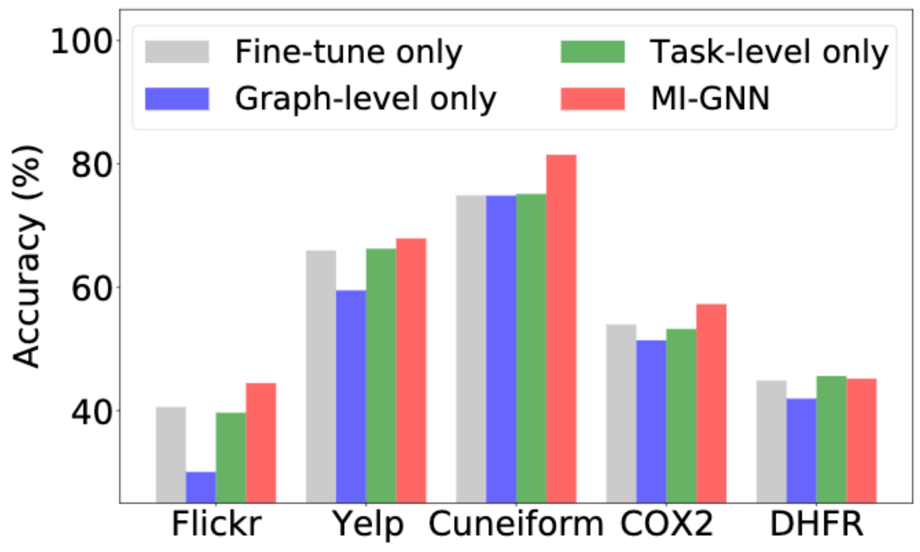
	GCN as the GNN Architecture				
	Flickr	Yelp	Cuneiform	COX2	DHFR
Transduct-GNN	14.89±0.94	50.92±0.95	49.40±2.27	11.89±0.63	10.89±0.43
Induct-GNN	12.08±3.98	55.04±1.77	71.65±0.46	86.06±2.78	90.31±1.03
AGF	11.94±2.45	53.66±3.04	71.66±0.46	86.32±3.08	89.64±1.00
Meta-GNN	22.51±3.05	54.80±1.86	72.24±0.88	86.92±3.66	90.26±0.91
MI-GNN	**29.91**±6.85	**57.22**±1.79	**75.36**±2.07	**86.97**±2.94	**91.39**±0.51

	GraphSAGE as the GNN Architecture				
	Flickr	Yelp	Cuneiform	COX2	DHFR
Transduct-GNN	14.97±1.96	50.14±1.19	50.59±1.37	12.78±0.65	11.19±0.75
Induct-GNN	7.31±1.57	56.48±1.73	84.46±2.68	85.28±1.78	88.65±4.79
AGF	7.45±1.31	56.70±2.04	84.66±2.73	85.21±1.85	88.21±4.45
Meta-GNN	33.88±2.91	61.80±1.81	84.46±2.44	86.05±2.80	88.17±4.71
MI-GNN	**42.37**±3.87	**69.23**±1.18	**91.09**±2.51	**93.24**±0.80	**93.89**±0.83

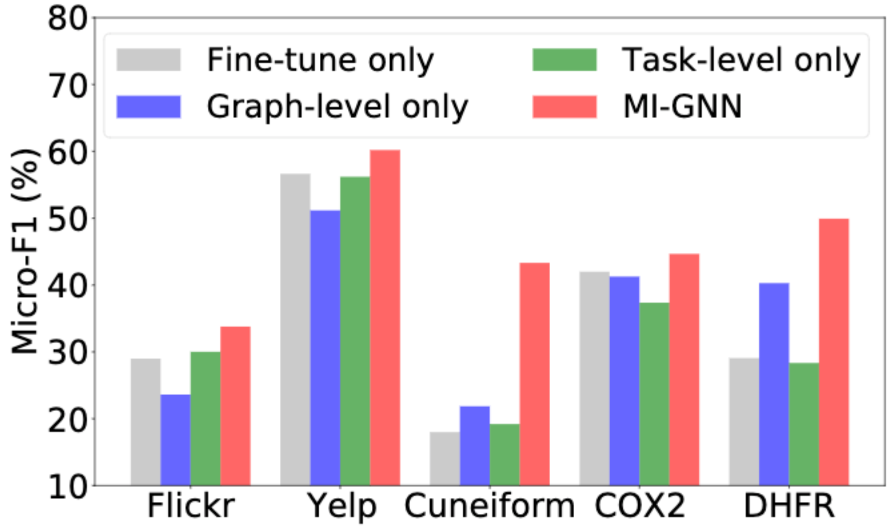
Ablation Study

Ablated versions of **MI-GNN**:

- **Fine-tune only**
  Neither graph- nor task-level adaptations, but a simple fine-tuning step.
- **Graph-level only**
  Remove the task-level adaptation from MI-GNN.
- **Task-level only**
  Remove graph-level adaptation from MI-GNN.



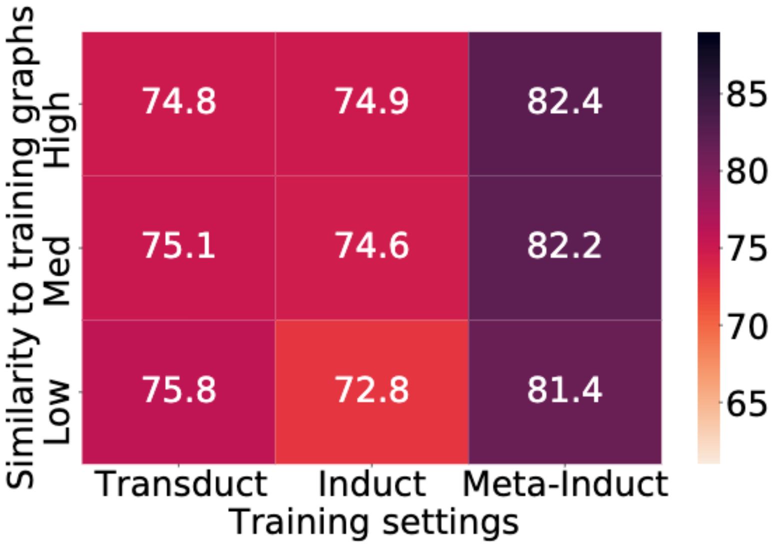
(a) Accuracy



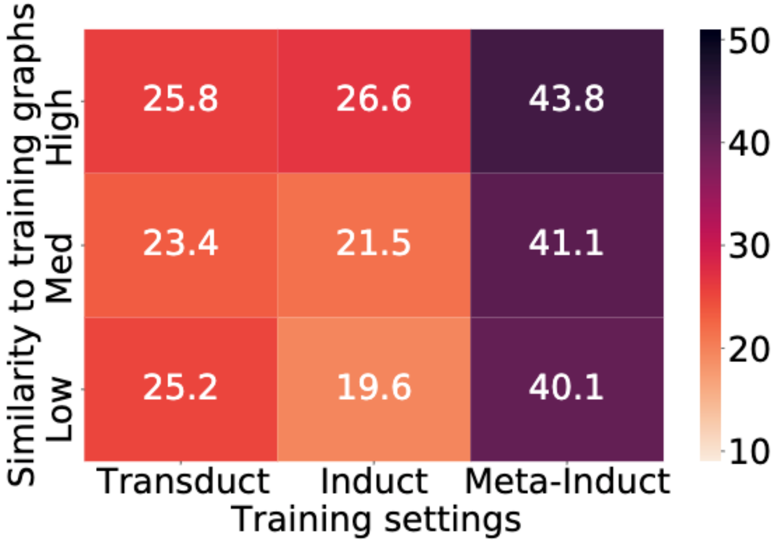
(b) Micro-F1

Performance case study

- Transductive methods are **not influenced** by similarity between testing and training graphs.
- Performance of inductive models **correlates** to such similarity.
- Our meta-inductive approach is **robust** due to the **dual-adaptation**.



(a) Accuracy



(b) Micro-F1

Outline

- ❑ Introduction & Challenge
- ❑ Methodology
- ❑ Experiment
- ❑ **Conclusion & Future Work**

Conclusion & Future Work

- Conclusion:
  - Studied the problem of inductive semi-supervised node classification across graphs.
  - Proposed a novel framework called MI-GNN, containing a dual adaptation mechanism at both the graph and task levels.
  - Conduct extensive experiments on five real-world graph collections.
- Future work:
  - Consider the **node level** adaptation, additionally
  - Apply to **heterogeneous** network

THANK YOU FOR YOUR ATTENTION

Paper, code, data…

www.yfang.site

SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and
Information Systems