Low-resource Learning on Graphs

Yuan FANG

School of Computing and Information Systems
Singapore Management University

RecSys'23 International Workshop on Deep Learning Practice for High-Dimensional Sparse Data

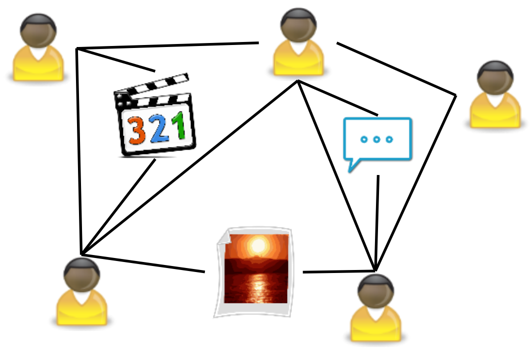
Sept. 18, 2023, Singapore

Outline

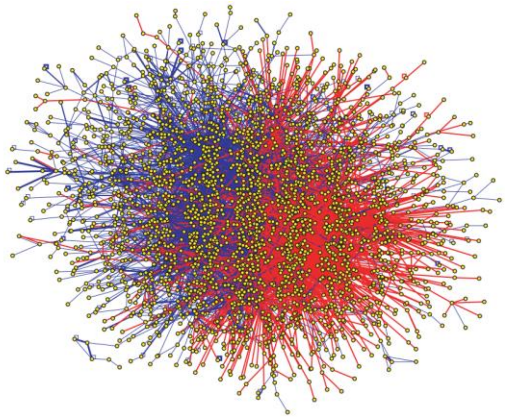
- **Introduction: Data, problems and methods**
- Structure-scarce learning on graphs
- Label-scarce learning on graphs
- Future directions and conclusion

Complex big data as graphs

Social networks

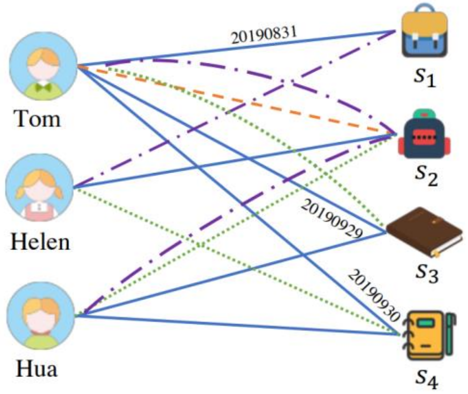


Biology



[Image from RVH05]

E-commerce



Knowledge graph

[Image from Microsoft]

[RVH05] Towards a proteome-scale map of the human protein–protein interaction network. J. Rual, *et al*. Nature: 437(7062), 2005.

Overview: Data, Problems and Methods

Data

Graphs/Networks

Heterogeneous graphs

User interaction graphs

Knowledge graph

Problems

Low-resource graph representation learning

- Structure scarcity
- Label scarcity

Methods

Meta-learning

Pre-training

Augmenting with external knowledge

Staff/student co-authors

Zemin Liu

Chenghao Liu

Zhihao Wen

Trung-Kien Nguyen

Yuanfu Lu

Wentao Zhang

Xingtong Yu

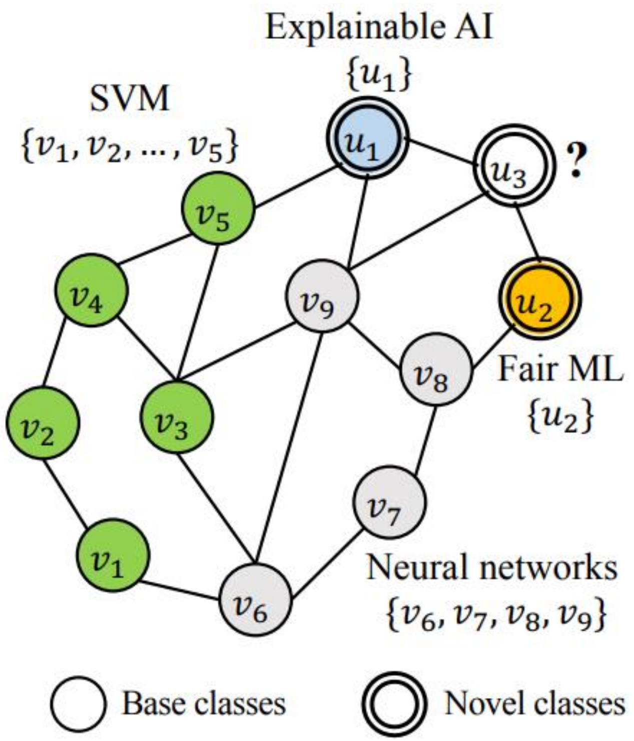
Main collaborators

Prof. Steven Hoi
Prof. Chuan Shi
Prof. Xinming Zhang
…

Low-resource problems on graphs

Label scarcity

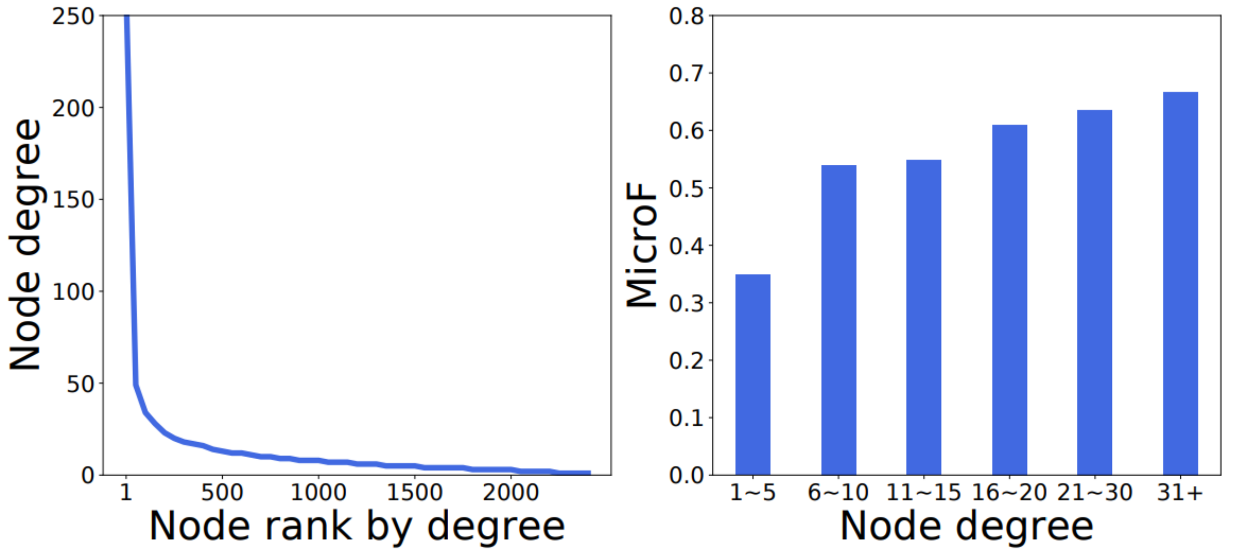
Novel classes emerge frequently with very few labelled data.



[Image from AAAI21a]

Structure scarcity

Graphs are characterized by structural information. Nodes with less structural contexts yield poor performance.



(a) Degree distribution (b) Classification performance

[Image from CIKM21]

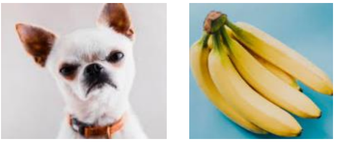
Low-resource method: Meta-learning

Supervised learning

Train

Cat Dog Banana Apple

Test



? ?

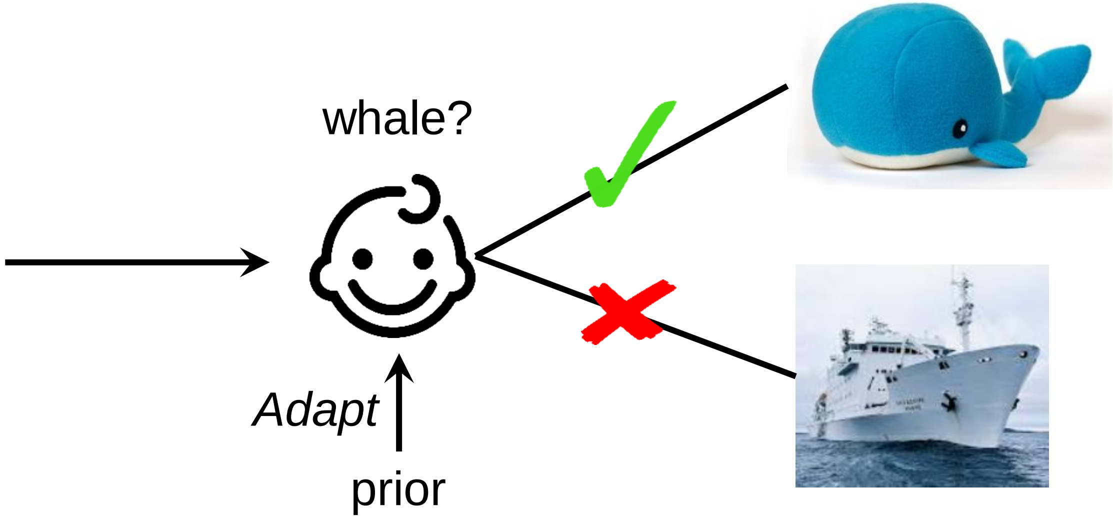
Learn a classifier

$f_\theta($  $) \rightarrow \text{dog}$

Need many, many labelled data!
Hard to deal with novel classes.

How humans learn?

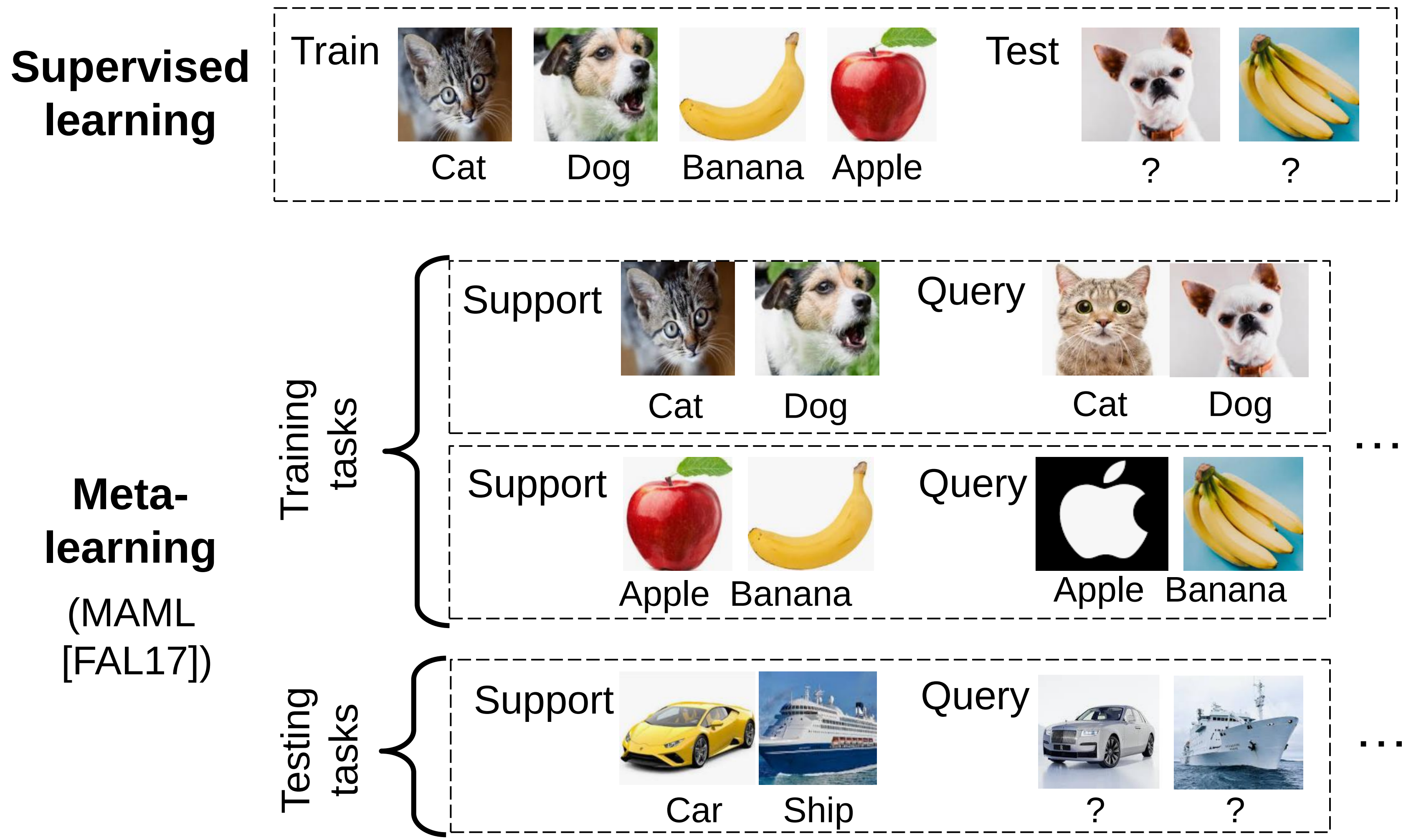
One example of toy whale



Even toddlers can learn novel classes very quickly with one/few examples…by generalizing from prior knowledge.

[Images from the Web]

Low-resource method: Meta-learning



Learn a classifier

$f_\theta(\ ) \rightarrow \text{dog}$

Need many, many labelled data!
Hard to deal with novel classes.

Learn a prior ϕ from the training tasks

Adapt

$g_\phi(\ \overset{\text{support}}{\text{Car}\ \ \text{Ship}}\ ) \rightarrow f_{\phi'}$

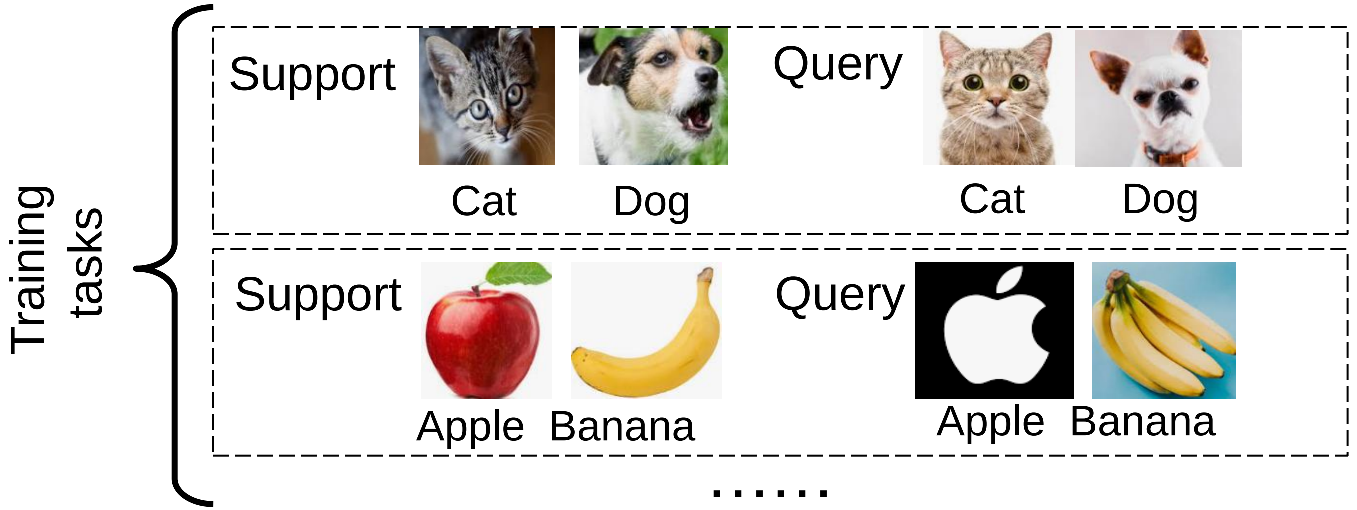
$f_{\phi'}(\ ) \rightarrow \text{car}$

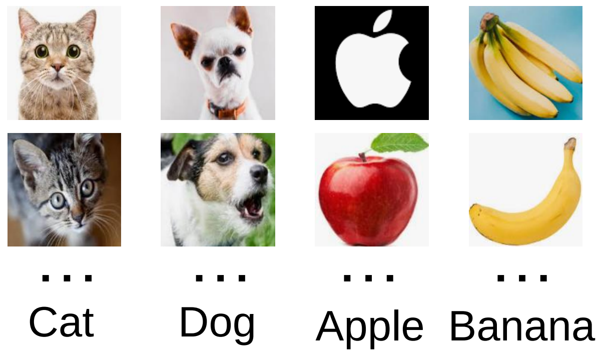
"Learn to learn"

[FAL17] Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. C. Finn *et al*. ICML 2017.

Low-resource method: Pre-training

Meta-learning

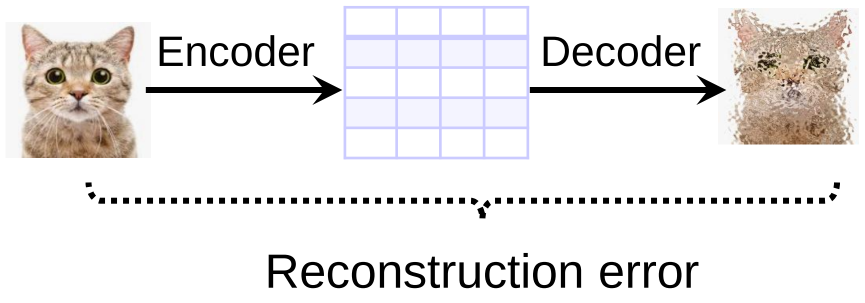




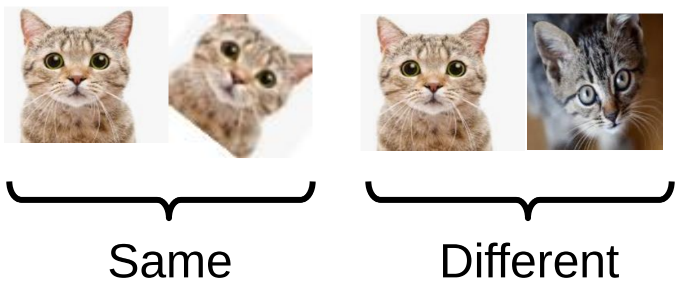
Still require many labels on these base classes to form training tasks

Self-supervised learning

Generative

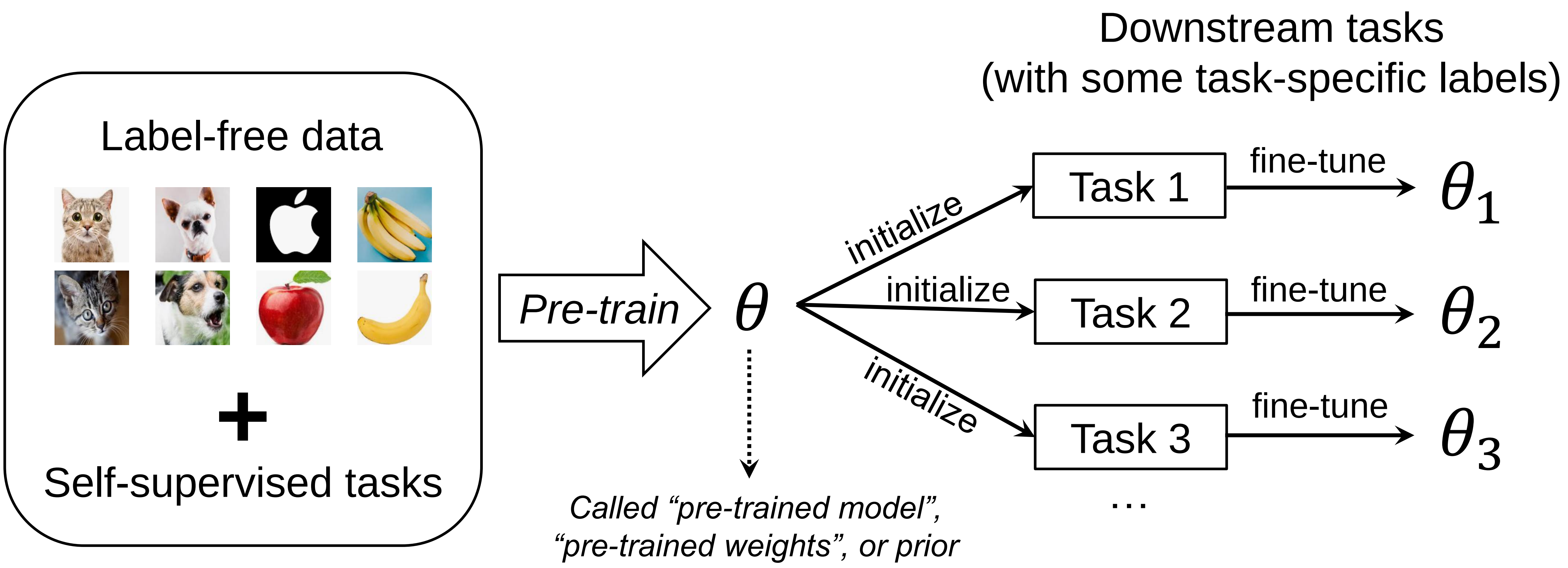


Contrastive



"Free" supervision, no annotation cost!

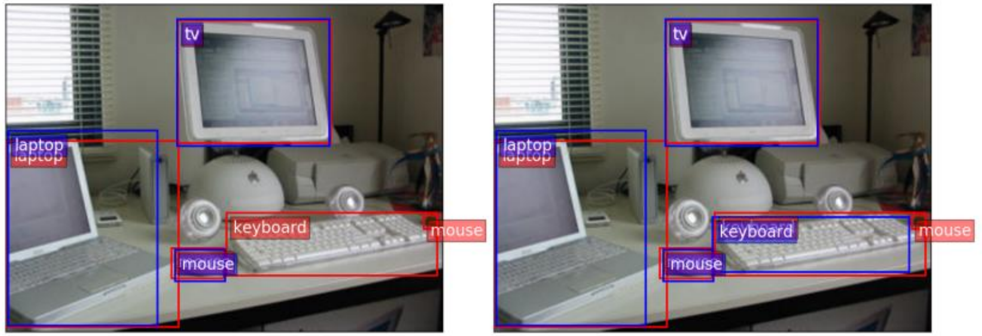
Low-resource method: Pre-training



Low-resource method: External knowledge

Object detection

(a) Office scene: FRCNN (left) fails to detect `keyboard`, but KG-CNet (right) does due to the presence of `laptop`.



External knowledge

Laptop, keyboard, and mouse often appear together.

(b) Outdoor scene: FRCNN (left) fails to detect `surfboard`, but KG-CNet (right) does due to the presence of `person`.

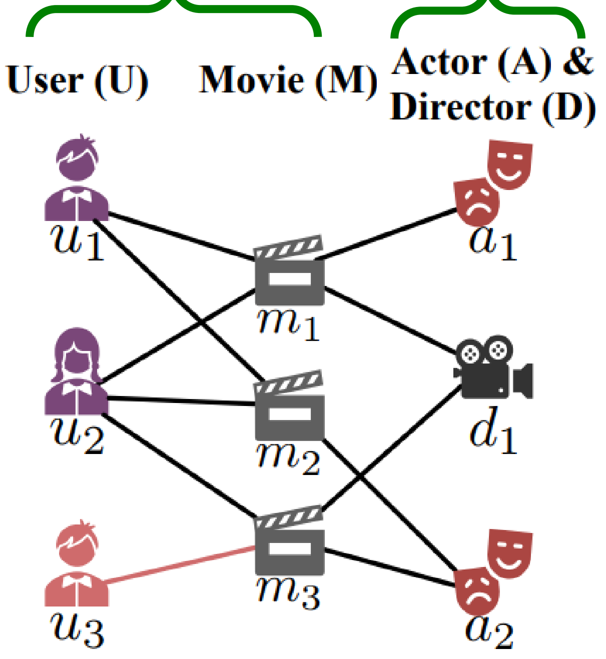
A person "standing" on a sea is usually on a surfboard

[Image from IJCAI17]

Recommendation

User interaction graph

Knowledge graph



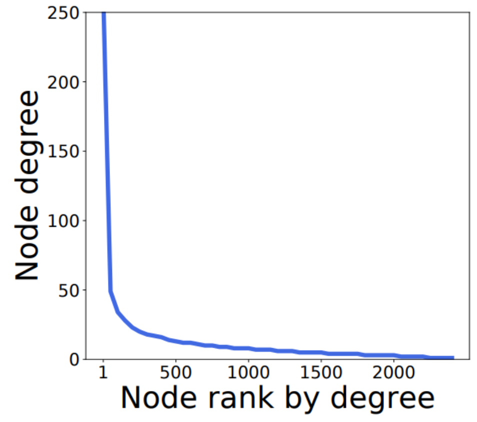
[Image from KDD20]

Outline

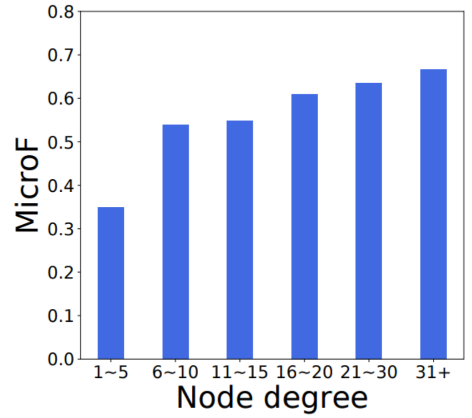
- Introduction: Data, problems and methods
- **Structure-scarce learning on graphs**
- Label-scarce learning on graphs
- Future directions and conclusion

meta-tail2vec: Meta-Learning of Tail Node Embeddings

- Tail nodes with very few links are ubiquitous
  - Newcomers
  - Existing less "active" nodes
- Tail nodes are not sufficiently modeled
  - Limited structural information
  - Existing methods regard all nodes uniformly using the same model
- **Problem:** Given the embedding vectors of nodes learned from a base embedding model, can we refine/improve the embeddings of the tail nodes?



(a) Degree distribution



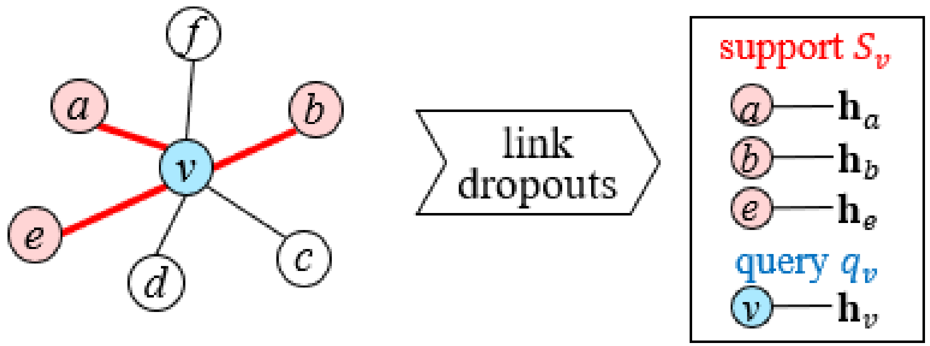
(b) Classification performance

meta-tail2vec: Meta-Learning of Tail Node Embeddings

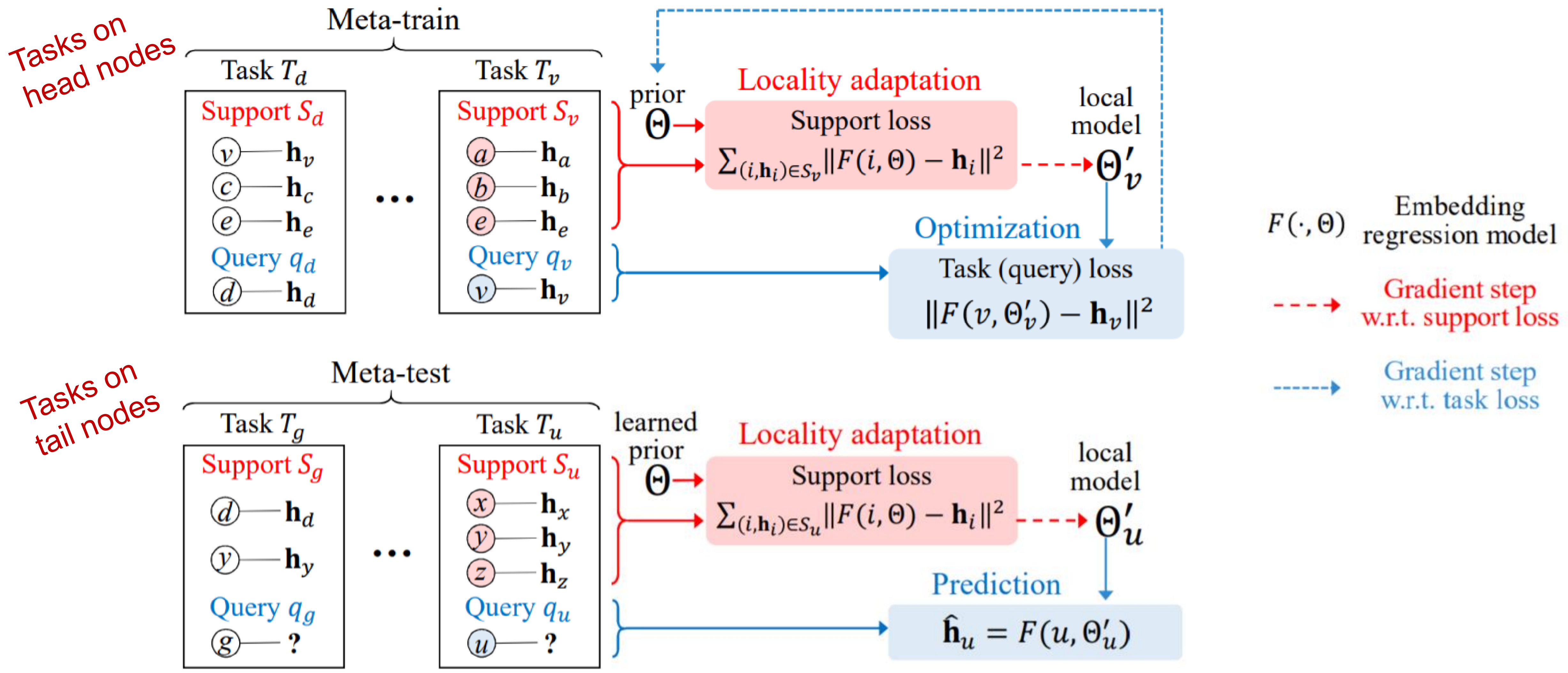
- Assumption: Head nodes have high-quality embeddings.
- Insights: Predict high-quality embeddings based on head nodes
  - Using a head node to simulate a mini-regression task
  - Perform link dropouts on head nodes to simulate tail nodes
  - Locality-aware tasks: support set sampled from neighboring nodes

- Meta-learning
  - Each task has a unique local context
  - Learn a prior from head node tasks
  - Adapt the prior to the tail node tasks

Mini-regression task on a head node



meta-tail2vec: Meta-Learning of Tail Node Embeddings

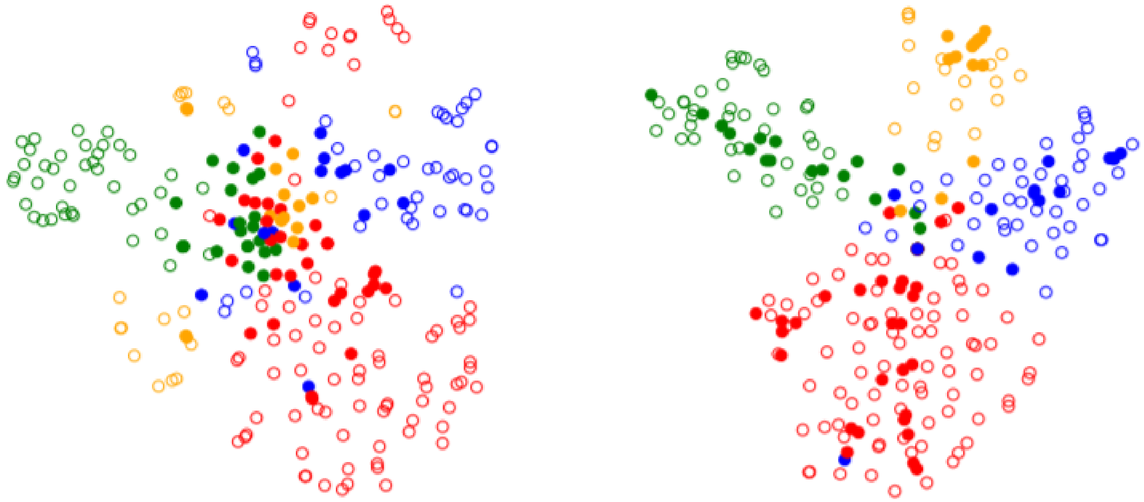


meta-tail2vec: Meta-Learning of Tail Node Embeddings

Visualization of base embeddings by SDNE, and their respective refinement by meta-tail2vec on the Email dataset.

Solid points - tail nodes
Hollow points - head nodes

Each color represents one class.



(a) Base embeddings (b) meta-tail2vec

Tail-GNN: End-to-end tail representation learning

- meta-tail2vec: Two-stage approach
  - Stage 1: Use any base model to generate node embeddings
  - Stage 2: Refine the tail node embeddings by meta-learning

- Tail-GNN: end-to-end approach [KDD21a]
  - Inspired by meta-tail2vec
  - Transfer knowledge from head to tail nodes
  - Perform link dropout on head nodes to simulate tail nodes
  - Adapt a global prior to individual nodes (but use a different meta-learning mechanism based on FiLM [PSV18])

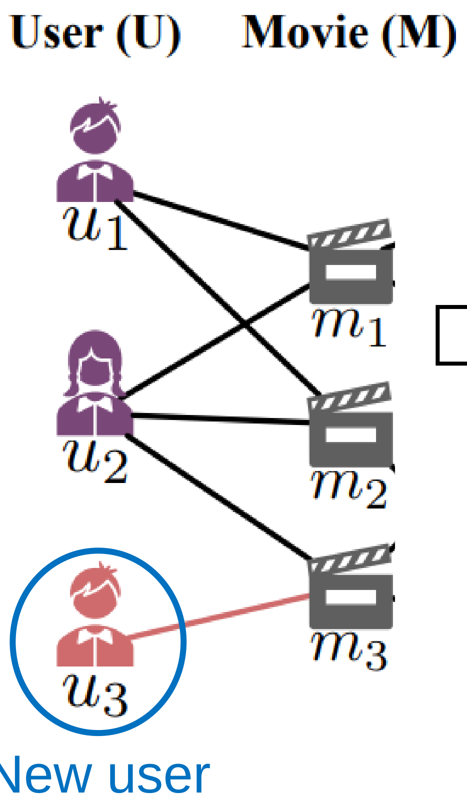
[PSV18] FiLM: Visual reasoning with a general conditioning layer. E. Perez *et al*. AAAI 2018.

MetaHIN: Cold-start recommendations

Collaborative filtering

[Image from the Web]

User-item graph

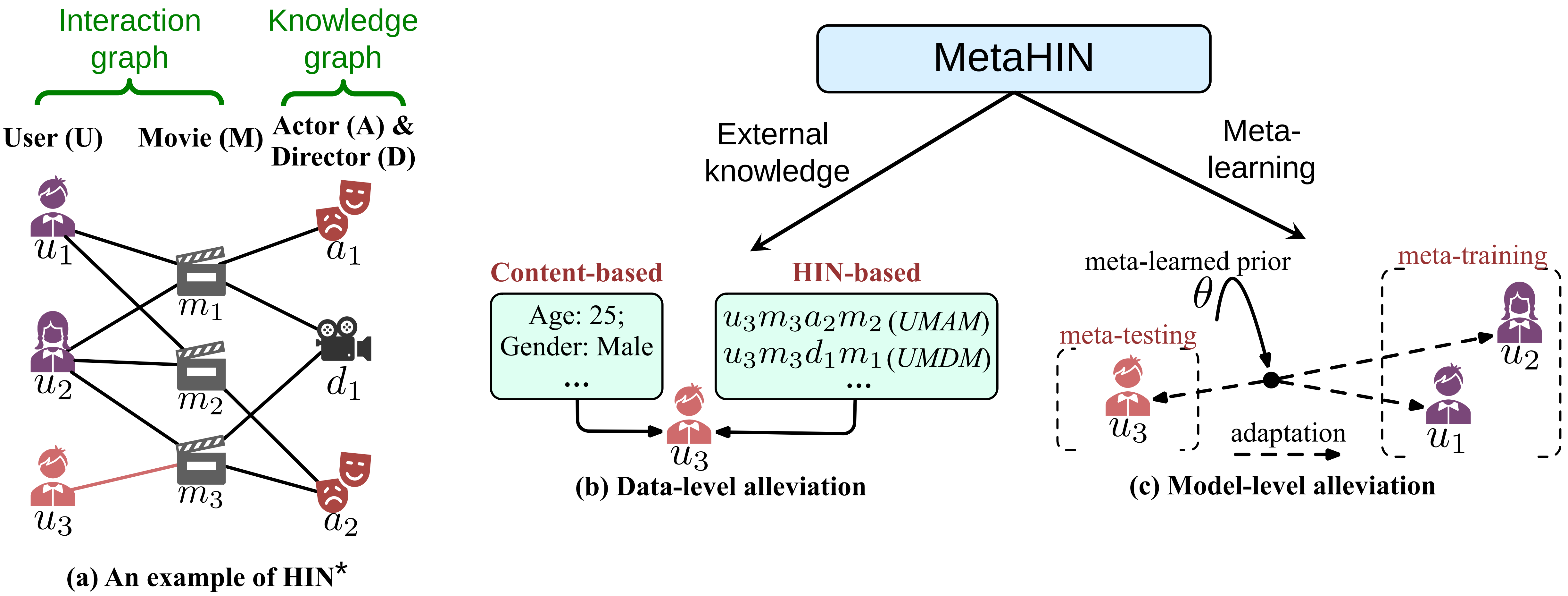


Problem: Cold-start recommendation

How about new users or items?

An instance of structure-scarce learning on the user-item interaction graph.

MetaHIN: Cold-start recommendations

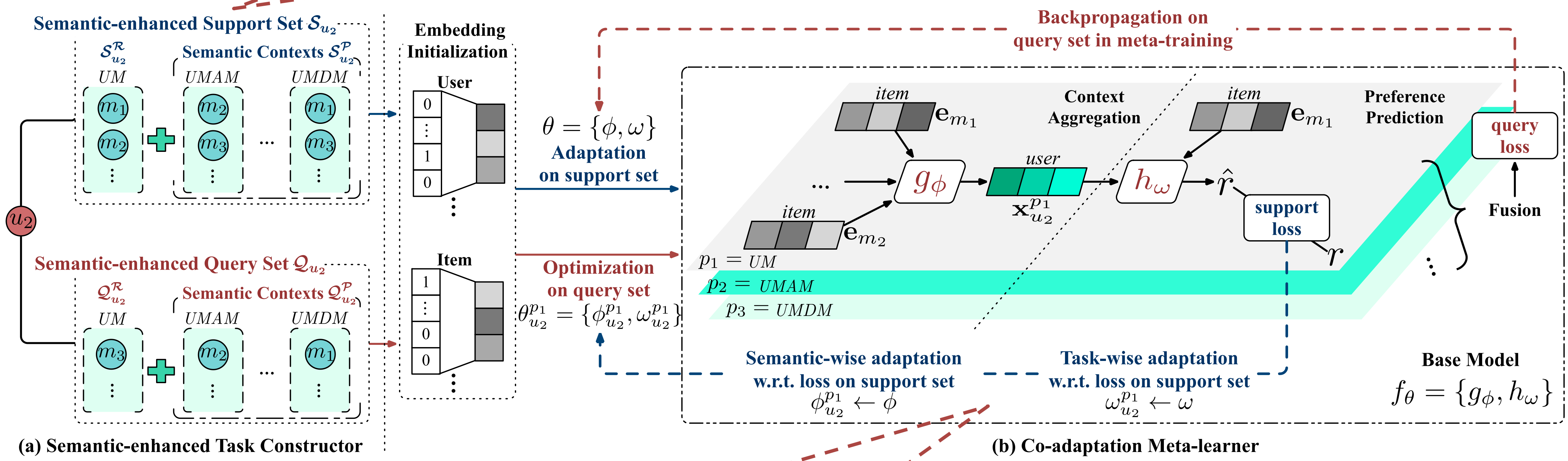


* HIN: Heterogeneous Information Network [SLZ17]

[SLZ17] A survey of heterogeneous information network analysis. C. Shi *et al*. TKDE: 29(1), 2017

MetaHIN: Cold-start recommendations

Task construction augmented with semantic contexts

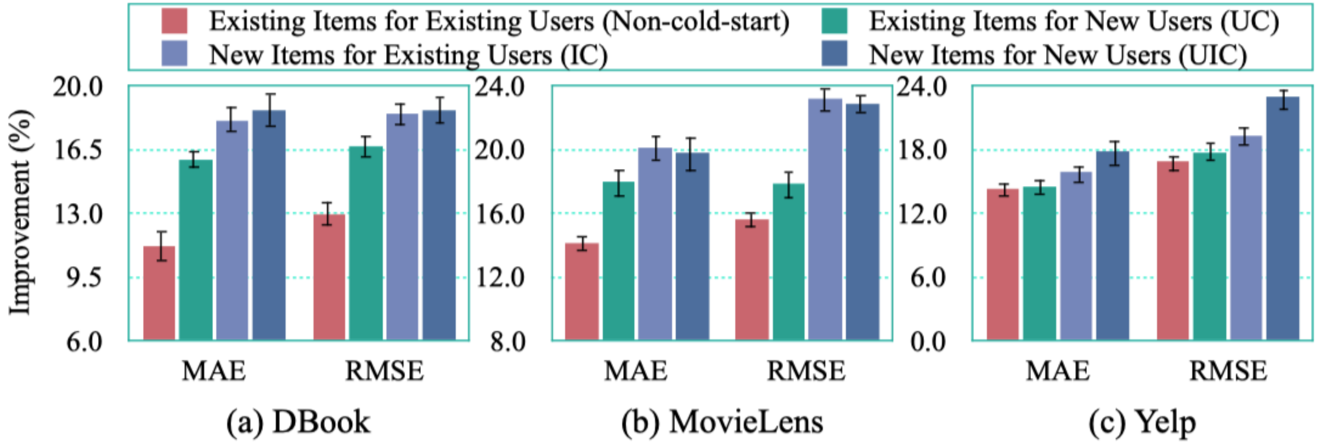


Semantic and task-wise adaptations

MetaHIN: Cold-start recommendations

Improvement of MetaHIN over SOTA in four code-start or non-cold-start scenarios

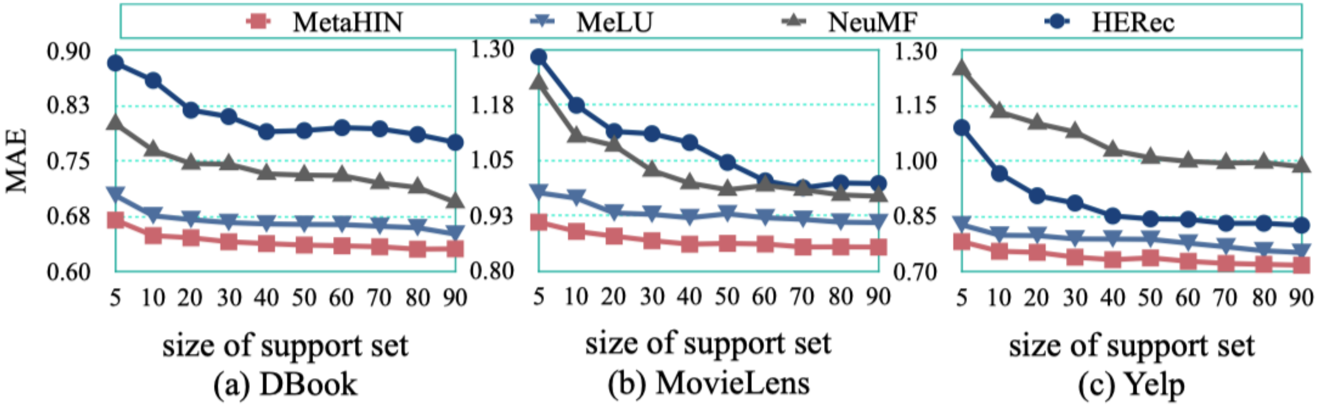
UIC > UC ~ IC > Non-cold-start



(a) Improvement in different scenarios

Impact of size of support set on MetaHIN and SOTA

Larger support, better performance; MetaHIN is robust: On small support sets, its performance is the least impacted.



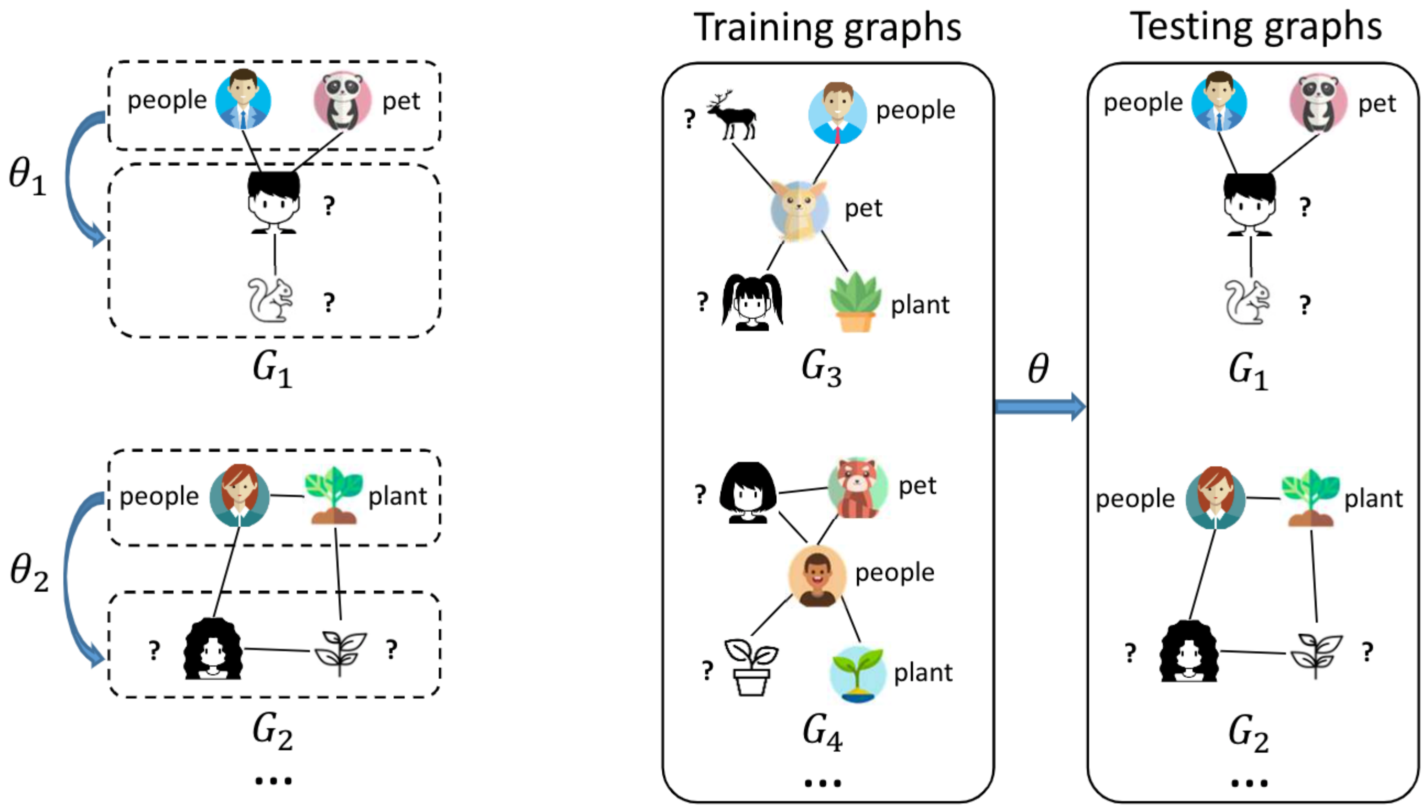
(b) Impact of size of support set

Outline

- Introduction: Data, problems and methods
- Structure-scarce learning on graphs
- **Label-scarce learning on graphs**
- Future directions and conclusion

MI-GNN: Meta-inductive, cross-graph GNNs

- Semi-supervised learning
  - A classic paradigm for learning with insufficient labelled data
  - Exploits the intrinsic structures between labelled and unlabelled data



(a) Transductive approach (e.g. label propagation [ZGL03])

Only able to utilize unlabelled nodes in a single graph.

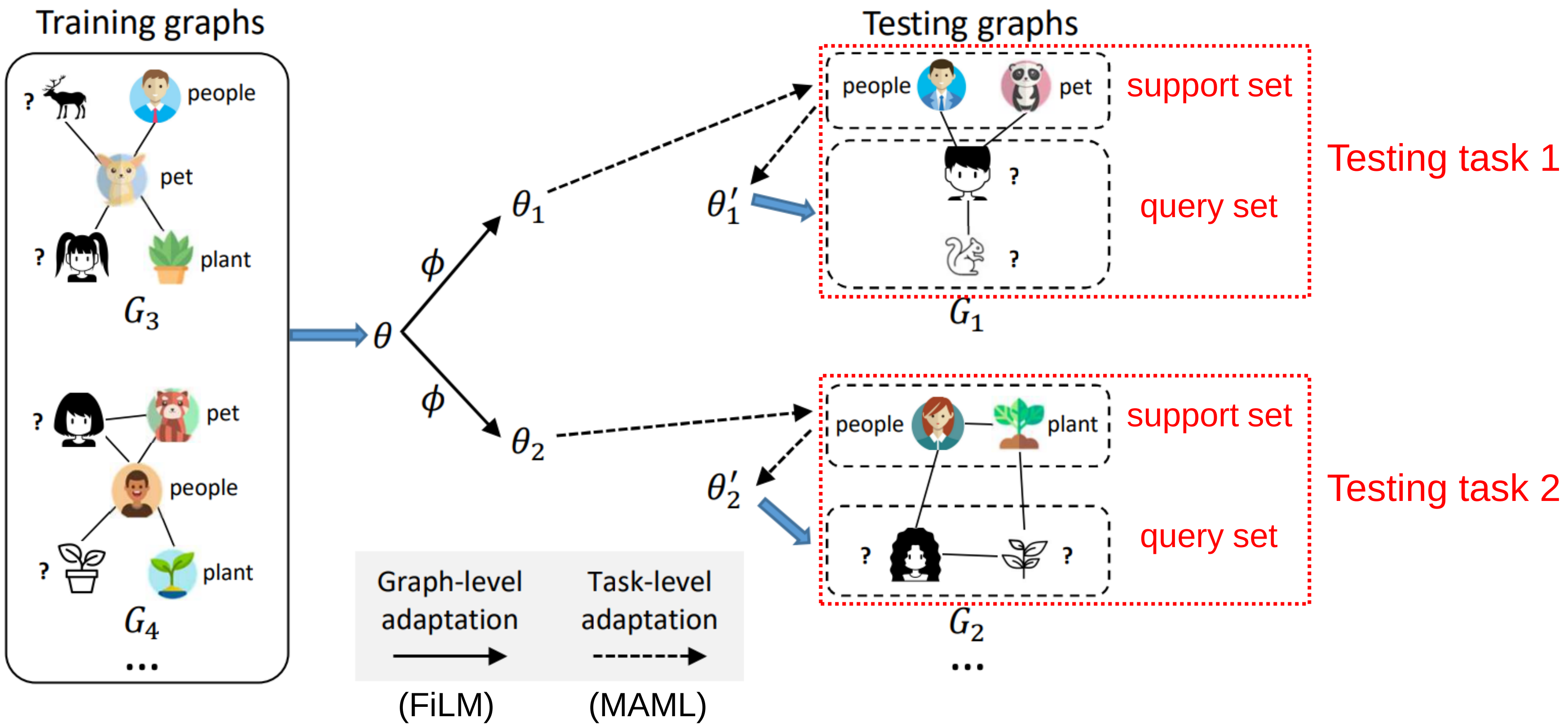
(b) Conventional inductive approach (e.g. most modern GNNs)

One-model-fits-all; ignores graph/task differences.

[ZGL03] Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. X Zhu *et al*. ICML 2003.

MI-GNN: Meta-inductive, cross-graph GNNs

- Using meta-learning to dynamically adapt the inductive model to take care of both **graph-level** and **task-level** differences



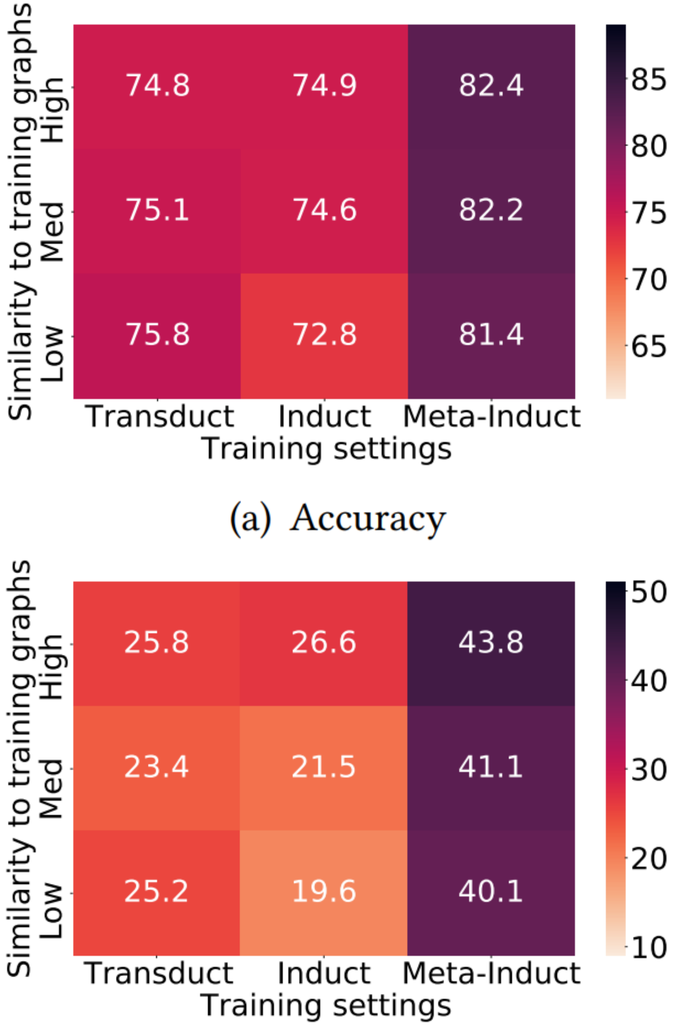
MI-GNN: Meta-inductive, cross-graph GNNs

Performance w.r.t. similarity to training graphs

<u>Transductive</u>: Minimal change in performance as no training graphs needed.

<u>Inductive</u>: Significant drop in performance when the testing graphs have low similarity.

<u>Meta-Inductive</u>: Robust, with only small decrease in performance when the testing graphs have low similarity.



(a) Accuracy

(b) Micro-F1

RALE: Few-shot learning on graphs

- **Problem:** Few-shot node classification

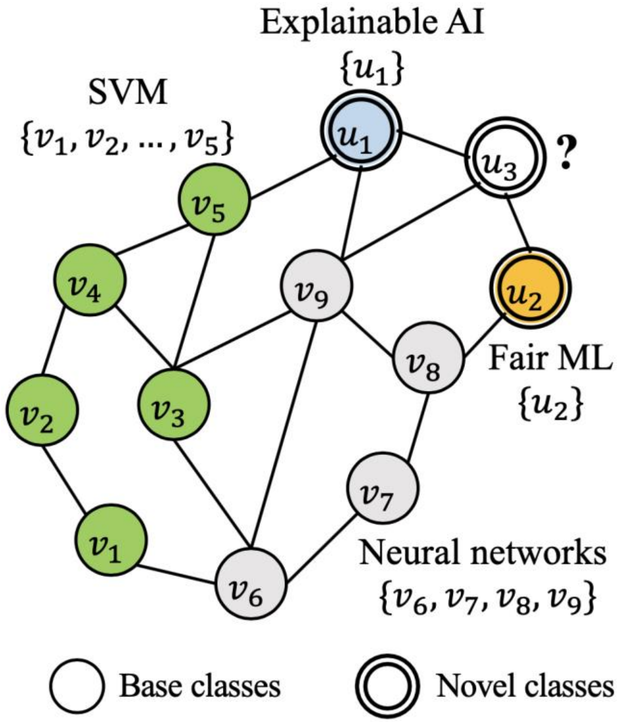
Support/query are randomly distributed in traditional meta-learning. How to capture their structural dependencies on a graph?

Base classes (sufficient labels)

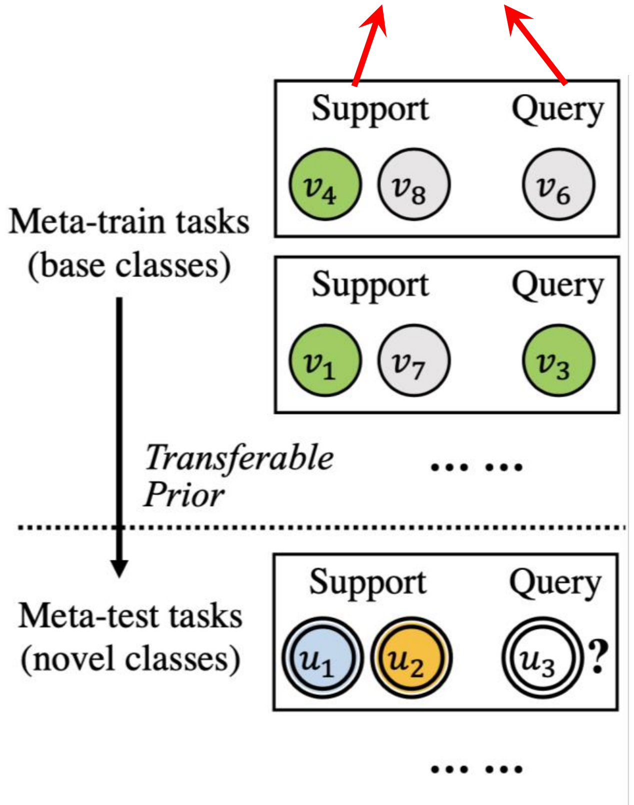
SVM
Neural networks

Novel classes (a few labels/class)

Explainable AI
Fair ML



(a) Base and novel classes on graph



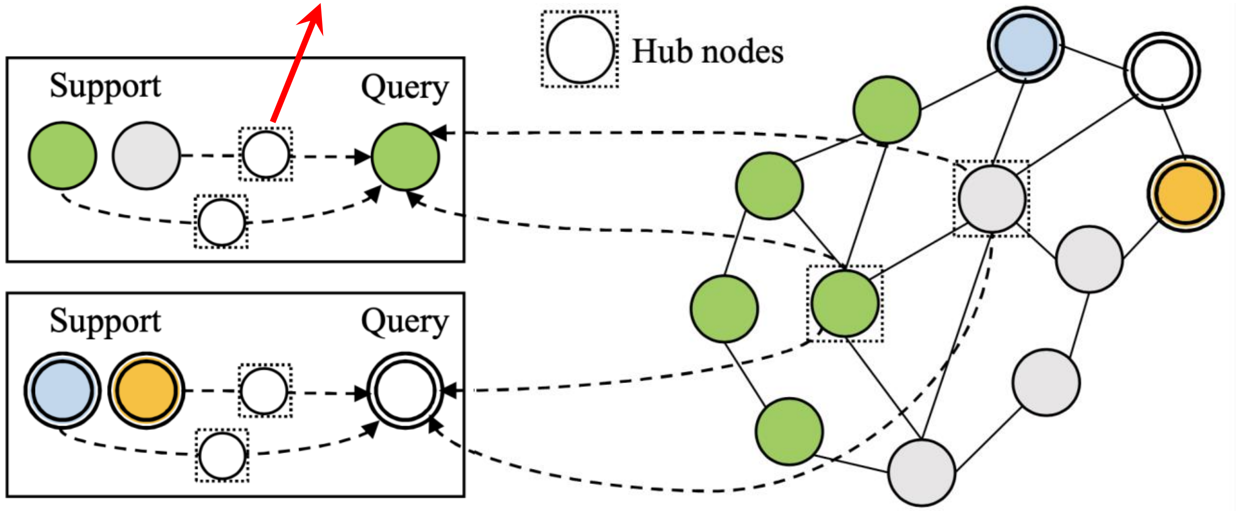
(b) Few-shot node classification

RALE: Few-shot learning on graphs

- Two challenges… How to
  - Capture long-ranged dependencies between nodes in a task?
  - Align dependencies across tasks to converge on a common prior?

- Insights: Use **hub** nodes
  - Within task: Define relative locations between support and query nodes
  - Globally: Define absolute locations of tasks on a graph

Hub nodes: Structurally important nodes, e.g., high degree or PageRank [PBM99]



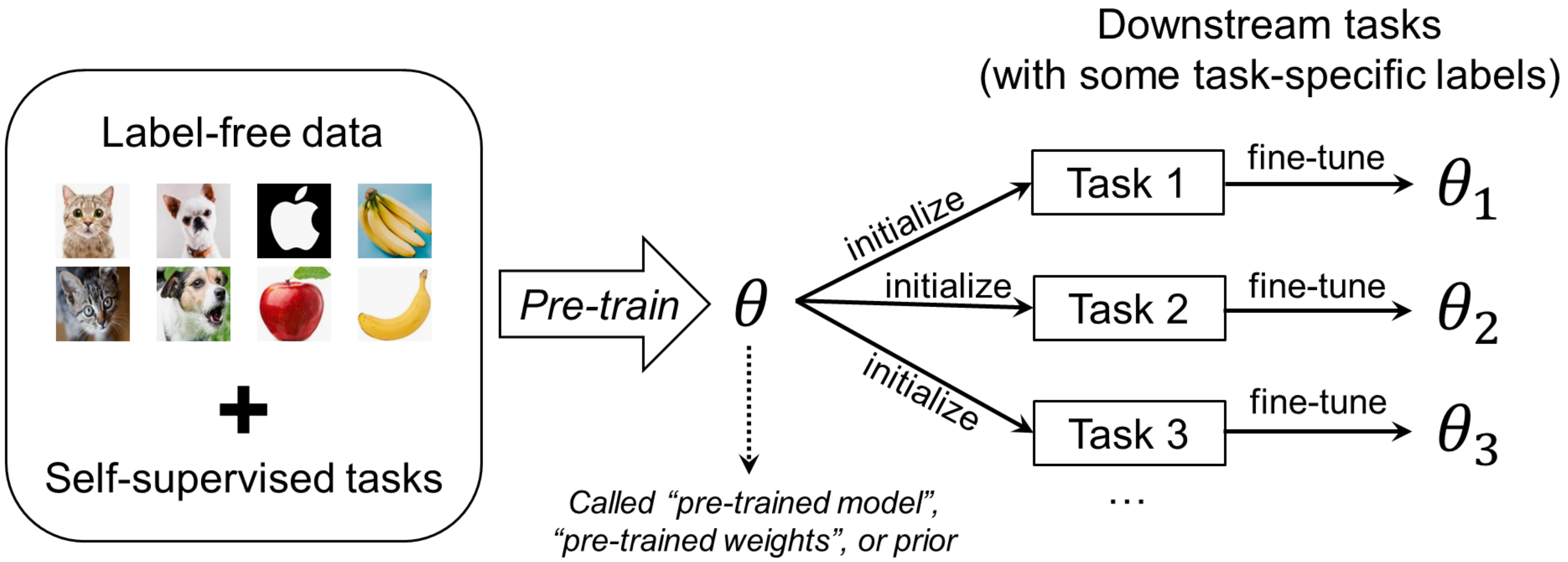
(a) Task-level relative location (b) Graph-level absolute location

[PBM99] The PageRank Citation Ranking: Bringing Order to the Web. L. Page, *et al*. WWW 1999.

Pre-training

- Limitation of meta-learning
  - Need enough base class labels to construct the meta-training tasks.
  - What if we don't have sufficient labels for meta-training?

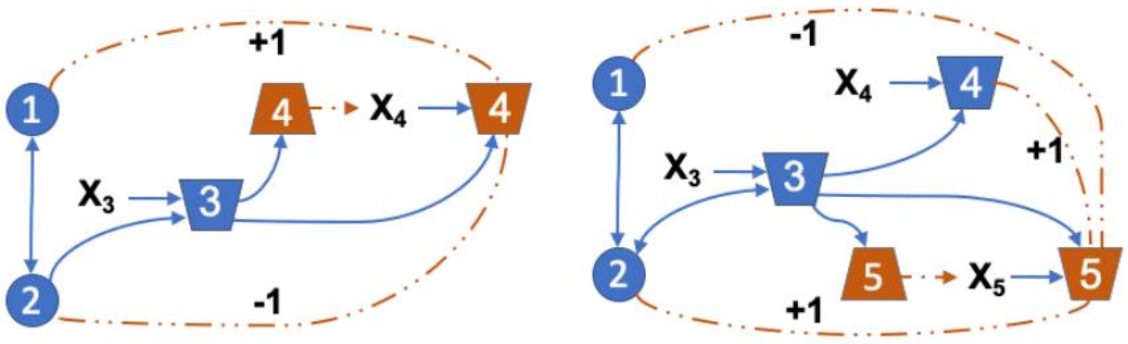
- Pre-training



Pre-training on graphs

- Key: Design self-supervised pre-training tasks on graphs
- Major strategies: **Generative** and **contrastive**

Generative

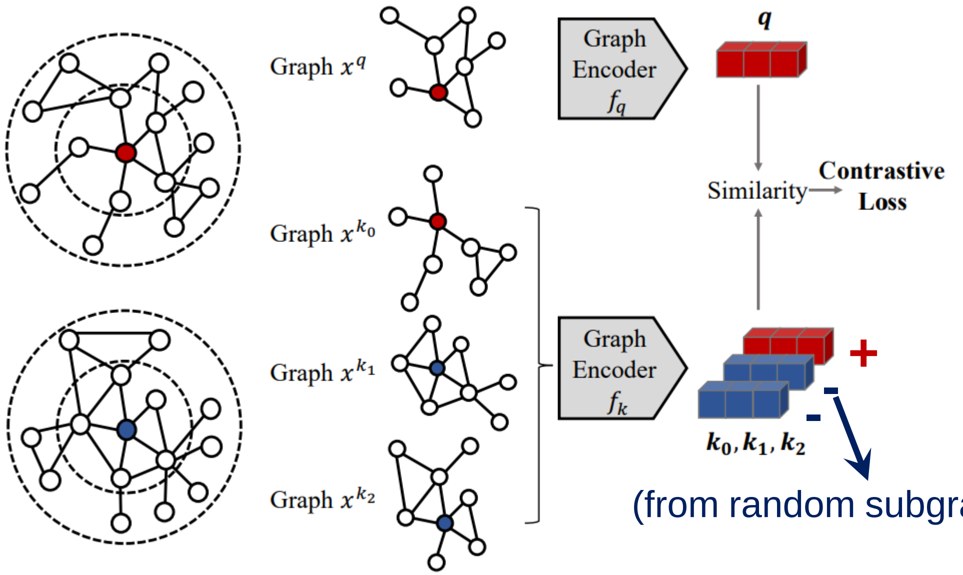


(d) Generate attributes and masked edges for node 4

(e) Generate attributes and masked edges for node 5

[Image from HDW20]

Contrastive

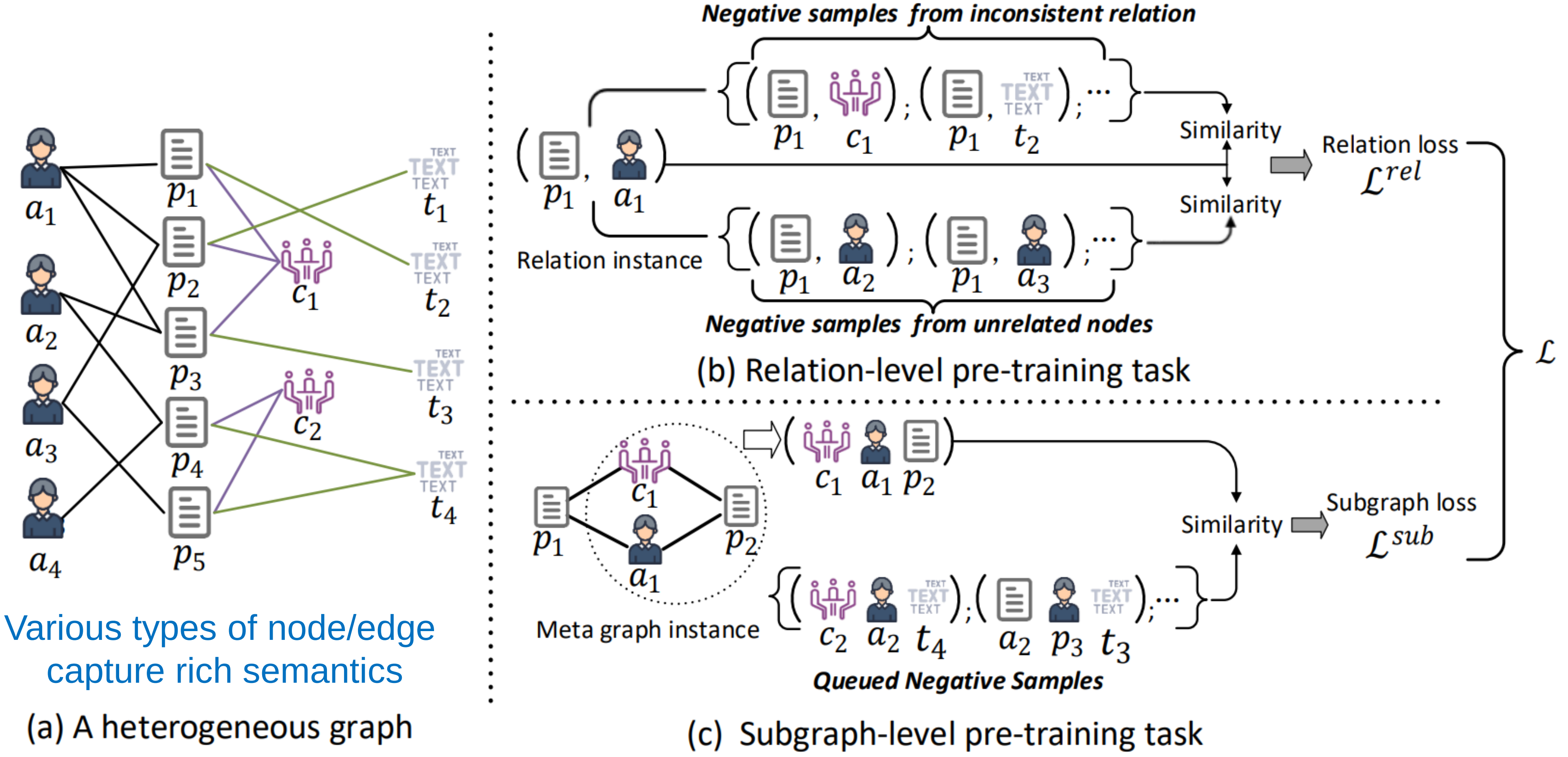


[Image from QCD20]

[HDW20] GPT-GNN: Generative Pre-Training of Graph Neural Networks. Z. Hu *et al*. KDD 2020
[QCD20] GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. J. Qiu *et al*. KDD 2020

Pre-training on heterogeneous graphs

- Pre-training tasks to capture relation- and subgraph-level semantics



(a) A heterogeneous graph

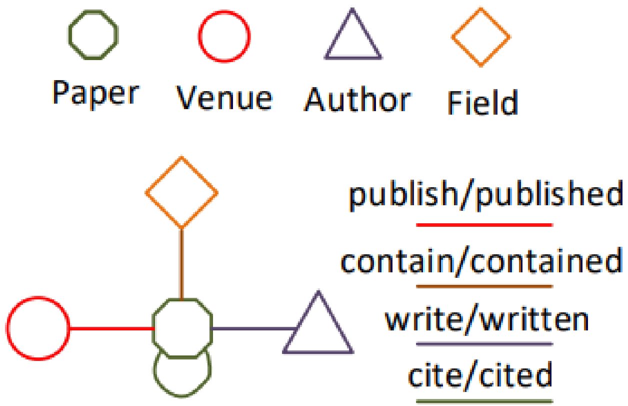
(b) Relation-level pre-training task

(c) Subgraph-level pre-training task

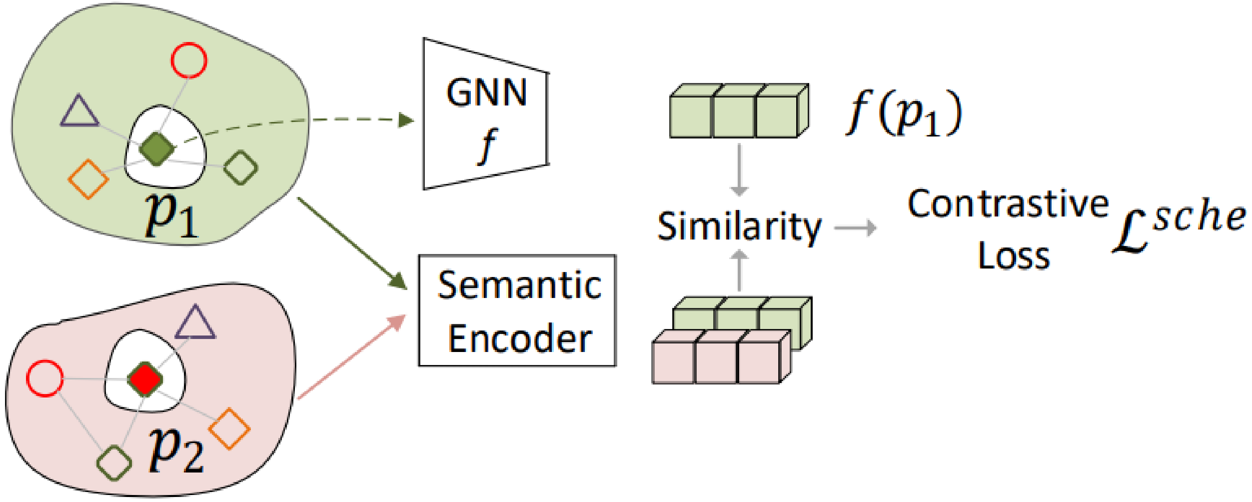
Pre-training on heterogeneous graphs

- Pre-training tasks to capture schema-level semantics

Schema

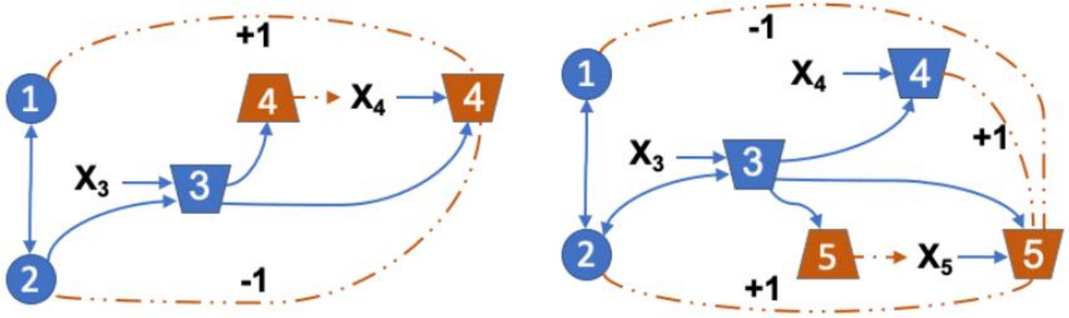


Schema-level task



Problem with pre-training approaches

- The gap between pre-training and downstream objectives



(d) Generate attributes and masked edges for node 4

(e) Generate attributes and masked edges for node 5

- And the fine-tuning step..
  - Can be expensive for large pre-trained models
  - may overfit if there are very few labels from downstream tasks

Bridging the gap: Learning to pre-train

- Pre-training is not aware of the fine-tuning step
- Learning to pre-train
  - Simulate the fine-tuning step within pre-training
  - Use meta-learning to adapt to the simulated task

Pre-training data: $\mathcal{D}^{pre} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_N\} \xrightarrow{\text{Pre-train}} \theta$

Meta-task: Construct a meta-task for a graph $\mathcal{T}_\mathcal{G} = (\mathcal{S}_\mathcal{G}, \mathcal{Q}_\mathcal{G})$

Fine-tune w.r.t. the loss on $\mathcal{S}_\mathcal{G} \rightarrow \theta'$

Update θ w.r.t. the loss on $\mathcal{Q}_\mathcal{G}$ with θ'

Simulate the fine-tuning step on a downstream task during pre-training

- But not a fundamental solution… Simulated task ≠ actual task

Bridging the gap: Pre-train, prompt

- **Problem:** Gap between pre-training and downstream tasks
- **Prompt** [LYF23]**:** an alternative to "pre-train, fine-tune"
  - Originated in NLP, an instruction to reformulate the original task to unify with pre-trained model (e.g., masked language modeling)

Task: Sentiment classification **+ Prompt**

"I missed the bus today." **"I felt so ___"**

Ask pre-trained model to fill in the blank

happy +

unlucky −

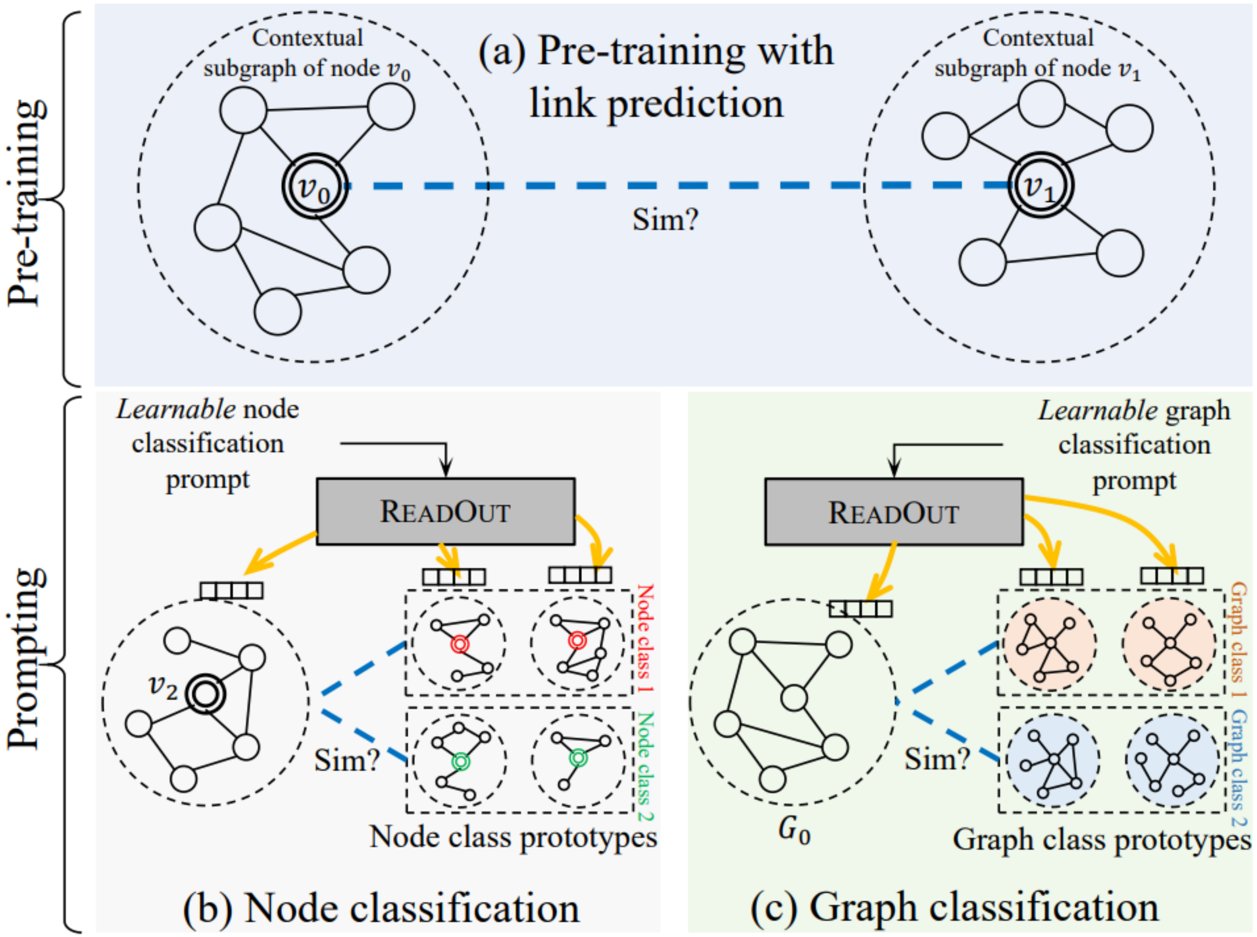
Zero-shot: Handcrafted (prompt engineering)
Few-shot: Learnable word vectors (prompt tuning)

[LYF23] Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. P. Liu, *et al*. ACM Computing Surveys: 55(9), 2023.

GraphPrompt: Pre-train, prompt on graphs

- Two challenges
  - How to unify various pre-training and downstream tasks on graph?
  - How to design prompts on graph?

- Insights
  - A **unified task template** based on subgraph similarity computation
  - Use a **learnable prompt** to guide graph readout for different tasks



GraphPrompt: Pre-train, prompt on graphs

Unified task template

Link prediction

Triplet (v, a, b), s.t. v is linked to a, but not b:

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

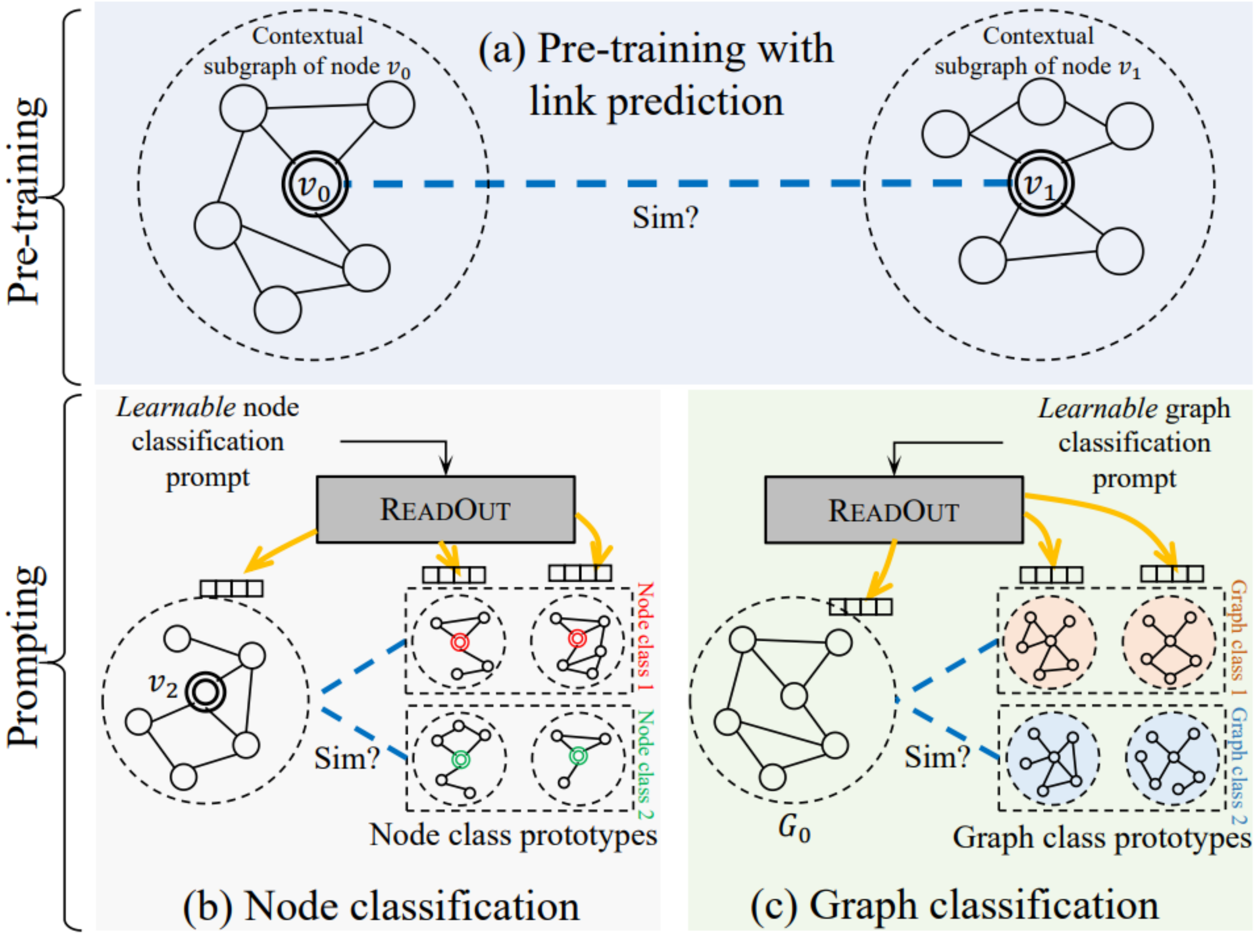
Node classification

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

Graph classification

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

All tasks converted to subgraph similarity computation!



$\mathbf{s}_x$: (sub)graph embedding of x (x is a node or graph)

$\tilde{\mathbf{s}}_c$: class c's prototype (a virtual subgraph, by aggregates all subgraph embeddings in the class)

GraphPrompt: Pre-train, prompt on graphs

Prompt design

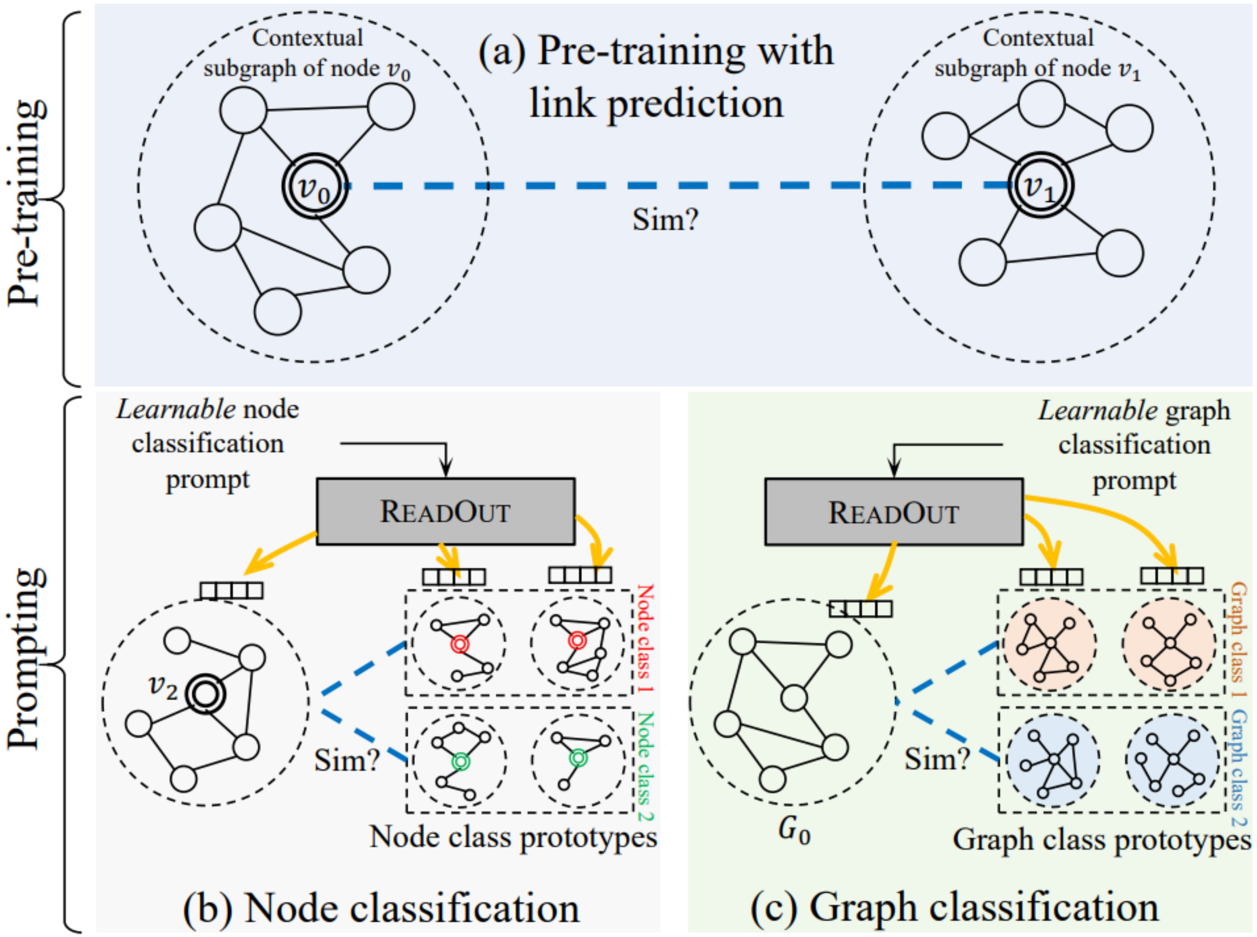
Different downstream tasks require different subgraph readout
→ Use task-specific learnable prompts

Prompt vector

$$\mathbf{s}_{t,x} = \textsc{ReadOut}(\{\mathbf{p}_t \odot \mathbf{h}_v : v \in V(S_x)\})$$

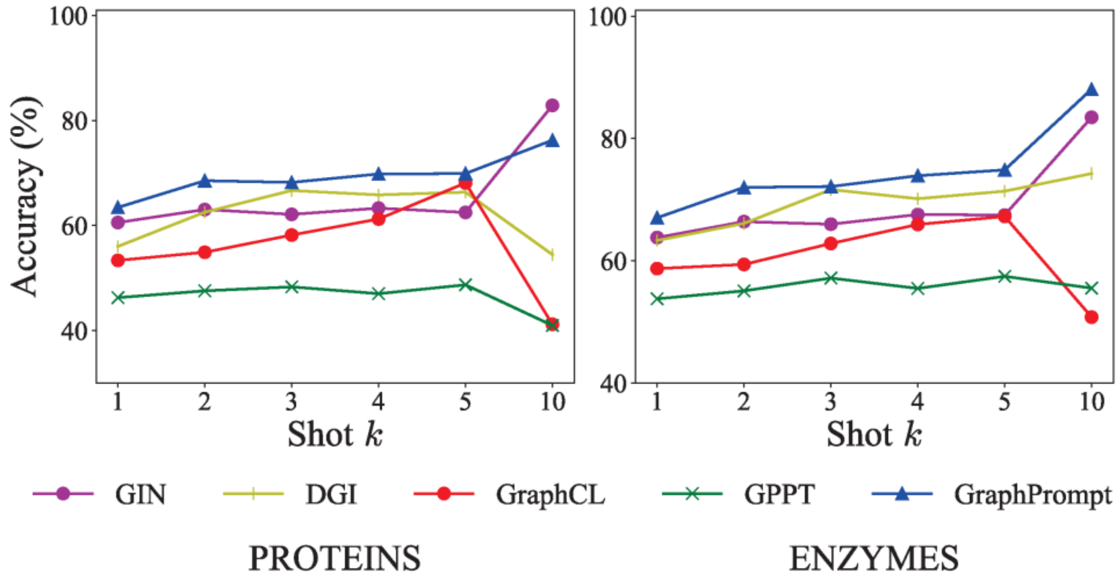
Prompt matrix

$$\mathbf{s}_{t,x} = \textsc{ReadOut}(\{\mathbf{P}_t \mathbf{h}_v : v \in V(S_x)\})$$

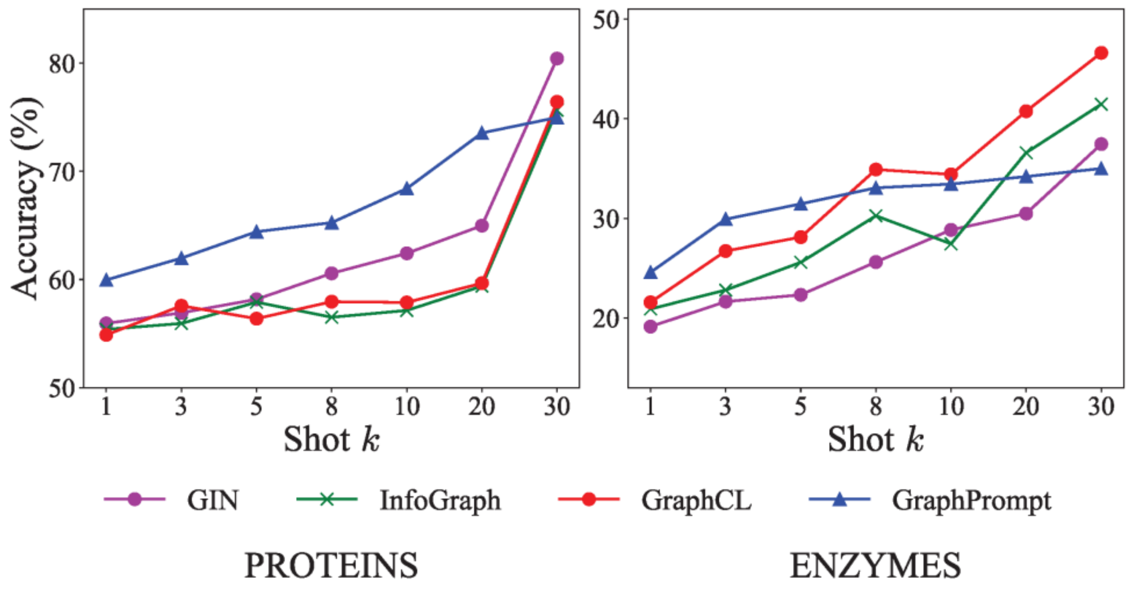


$\mathbf{s}_{t,x}$: (sub)graph embedding of x for a task t
$\mathbf{h}_v$: node v's embedding vector
$\mathbf{p}_t$ or $\mathbf{P}_t$: learnable prompt vector or matrix for task t

GraphPrompt: Pre-train, prompt on graphs



Impact of shots on few-shot node classification.



Impact of shots on few-shot graph classification.

<u>Few-shot:</u> Significantly better

<u>10-shot:</u> Still competitive
(as graphs are small – 10 shots are a lot)

<u>Few-shot:</u> Significantly better

<u>On ENZYMES:</u> worse performance on ≥20 shots
(only 600 graphs – 20 shots/class ~ 20% labels)

GraphPrompt: Pre-train, prompt on graphs

Comparison of parameter efficiency

Significantly fewer parameters/FLOPs than:

- *Supervised model (GIN [XHL19]),*
- *"Pretrain, fine-tune" model (GraphPrompt-ft),*
- *Existing prompt model (GPPT [SZH22])*

Methods	Flickr	
	Params	FLOPs
GIN	22,183	240,100
GPPT	4,096	4,582
GraphPrompt	96	96
GraphPrompt-ft	21,600	235,200

Methods	PROTEINS		ENZYMES	
	Params	FLOPs	Params	FLOPs
GIN	5,730	12,380	6,280	11,030
GPPT	1,536	1,659	1,536	1,659
GraphPrompt	96	96	96	96
GraphPrompt-ft	6,176	13,440	6,176	10,944

[XHL19] How Powerful are Graph Neural Networks? K. Xu *et al*. ICLR 2019
[SZH22] GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks. M. Sun *et al*. KDD 2022

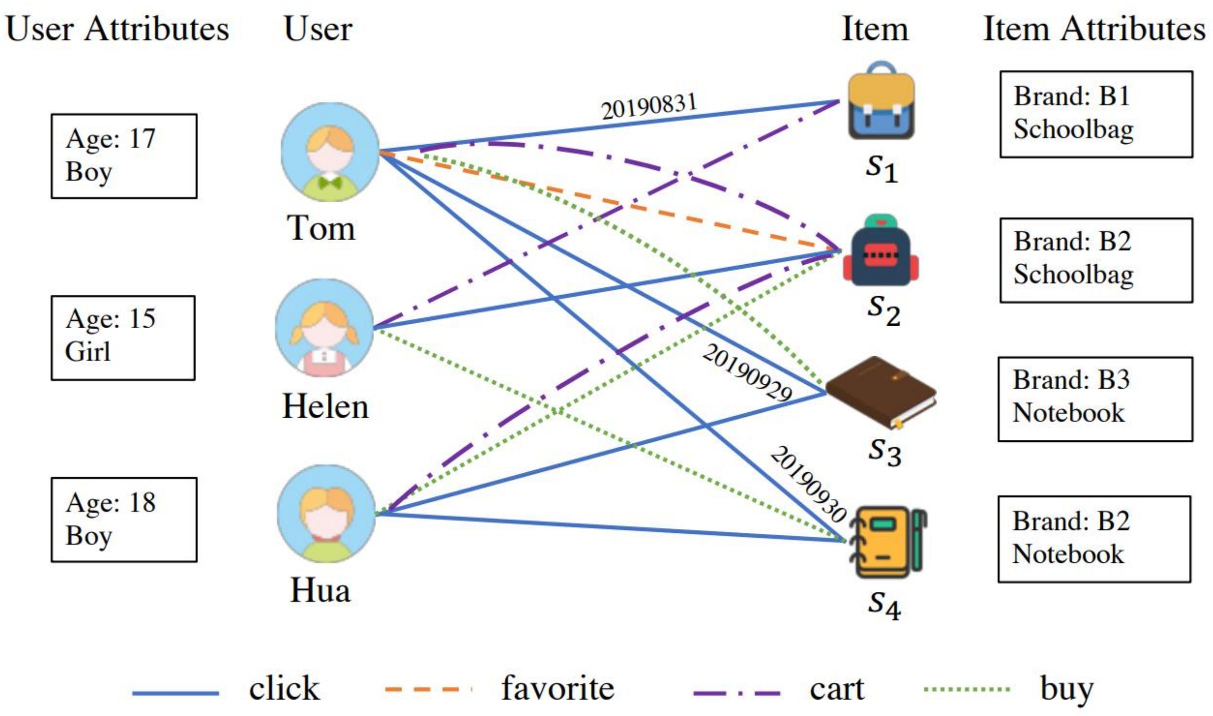
Outline

- Introduction: Data, problems and methods
- Structure-scarce learning on graphs
- Label-scarce learning on graphs
- **Future directions and conclusion**

Future directions

- Prompt on complex graphs
  - Heterogeneous graphs?
  - Dynamic graphs?
- Multi-modal graph learning?
  - Text on graphs? [SIGIR23]
  - Image on graphs?
  - Leveraging big models in other forms of data

Temporal Heterogeneous Interaction Graph



[Image from JYF20]

[JYF20] Temporal Heterogeneous Interaction Graph Embedding For Next-Item Recommendation. Y. Ji, *et al*. ECML-PKDD 2020.

Take-home messages

- Low-resource learning on graphs: structure, label
- Learning and transferring/using prior is the key
- Prompt is a promising paradigm

References of our work

- [IJCAI17] Y. Fang, K. Kuan, J. Lin, C. Tan and V. Chandrasekhar. *Object Detection Meets Knowledge Graphs.*
- [CIKM20] Z. Liu^, W. Zhang^, Y. Fang, X. Zhang and S. C. H. Hoi. *Towards Locality-Aware Meta-Learning of Tail Node Embeddings on Networks.*
- [KDD20] Y. Lu, Y. Fang and C. Shi. *Meta-learning on Heterogeneous Information Networks for Cold-start Recommendation.*
- [KDD21a] Z. Liu, K. Nguyen and Y. Fang. *Tail-GNN: Tail-Node Graph Neural Networks.*
- [KDD21b] X. Jiang, T. Jia, C. Shi, Y. Fang, Z. Lin and H. Wang. *Pre-training on Large-Scale Heterogeneous Graph.*
- [AAAI21a] Z. Liu, Y. Fang, C. Liu and S. C. H. Hoi. *Relative and Absolute Location Embedding for Few-Shot Node Classification on Graph.*
- [AAAI21b] Y. Lu, X. Jiang, Y. Fang and C. Shi. *Learning to Pre-train Graph Neural Networks.*
- [SIGIR21] Z. Wen, Y. Fang and Z. Liu. *Meta-Inductive Node Classification across Graphs.*
- [CIKM21] X. Jiang, Y. Lu, Y. Fang and C. Shi. *Contrastive Pre-training of GNNs on Heterogeneous Graphs.*
- [WWW23] Z. Liu^, X. Yu^, Y. Fang and X. Zhang. *GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks.* (Accepted)
- [SIGIR23] Z. Wen and Y. Fang. *Augmenting Low-Resource Text Classification with Graph-Grounded Pre-training and Prompting.* (Accepted)

^ Co-first authors with equal contribution.

[Additional references of others' work are given on individual slides.]

Acknowledgement of funding sources

- *One-shot learning: A crucial learning paradigm towards human-like learning.* National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-001).

- *Learning with less data*. Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funds (Grant No. A20H6b0151).

- *Universal pre-training of graph neural networks*. Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Proposal ID: T2EP20122-0041).

Hiring PhD Students

https://www.yfang.site/hiring

- **Why Join Us?**
  - We offer full scholarships to PhD students covering tuition fees and monthly stipends, with additional conference travel support.
  - Through the research journey with the team, you will have a great opportunity for self-development and generating high-quality publications at top-tier conferences and journals. Our team has published extensively and consistently in top venues such as KDD, SIGIR, AAAI, TKDE and more.
  - Overall, SMU is ranked **No. 40 globally (No. 16 in Asia)** in the broad AI category and No. 84 globally (No. 15 in Asia) in the overall CS discipline according to CSRankings; **No. 104 globally in QS 2022 Subject Ranking** (Computer Science and Information Systems).

- **Requirements for PhD Student**
  - Obtained/expected to obtain Bachelor or master's degree in Computer Science or related fields at a reputable university
  - Excellent academic performance with strong foundation in computing and other STEM subjects
  - Prior research experience or publications in data mining and machine learning a strong plus
  - Strong programming/implementation skills required; working knowledge of deep learning stack a plus
  - Intellectual curiosity to explore the unknown
  - Good communication and teamwork skills

Thank you

Questions?

Email: yfang@smu.edu.sg

Full publications, codes and data are available at
http://www.yfang.site/