KDD'24 Tutorial: Graph Intelligence with Large Language Models and Prompt Learning

Jia Li, Xiangguo Sun, Yuhan Li, Zhixun Li, Hong Cheng, Jeffrey Xu Yu

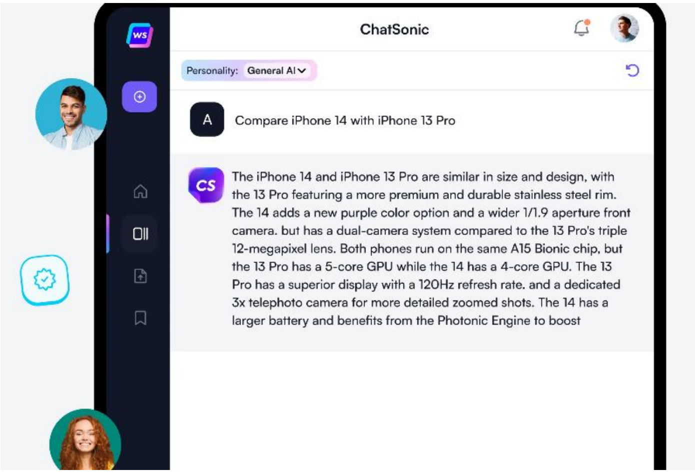
The Hong Kong University of Science and Technology (Guangzhou),
The Chinese University of Hong Kong

Outline

Time	Section	Presenter
10:00-10:15	Part A: Opening & Introduction	Hong Cheng
10:15-10:50	Part B: Uni-modal Pretraining	Zhixun Li
10:50-11:30	Part C: Multi-modal Pretraining	Yuhan Li
11:30-12:00	Coffee Break	-
12:00-12:45	Part D: Pretraining with Prompting	Xiangguo Sun
12:45-13:00	Q&A	-

Artificial General Intelligence (AGI)

- **Artificial General Intelligence (AGI) has achieved huge success in NLP and CV areas.**
  - e.g. Copilot, ChatGPT, Midjourney, etc



A Basic Workflow of AGI

- **Step 1: Pre–train a very large language model (LLM) via specific strategies.**
  - e.g. masked word prediction

KDD24 will **witness** many high-quality **outcomes**.

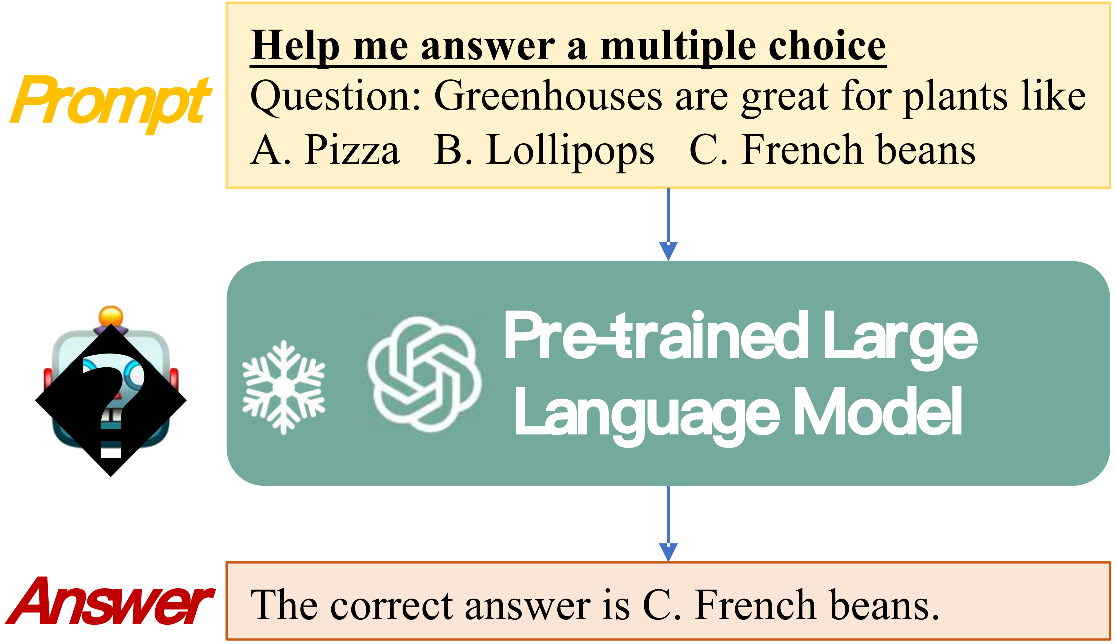
Pre-training Large Language Model via Masked Word Prediction Task

Calculating loss for updating parameters

KDD24 will **<Mask>** many high-quality **<Mask>.**

A Basic Workflow of AGI

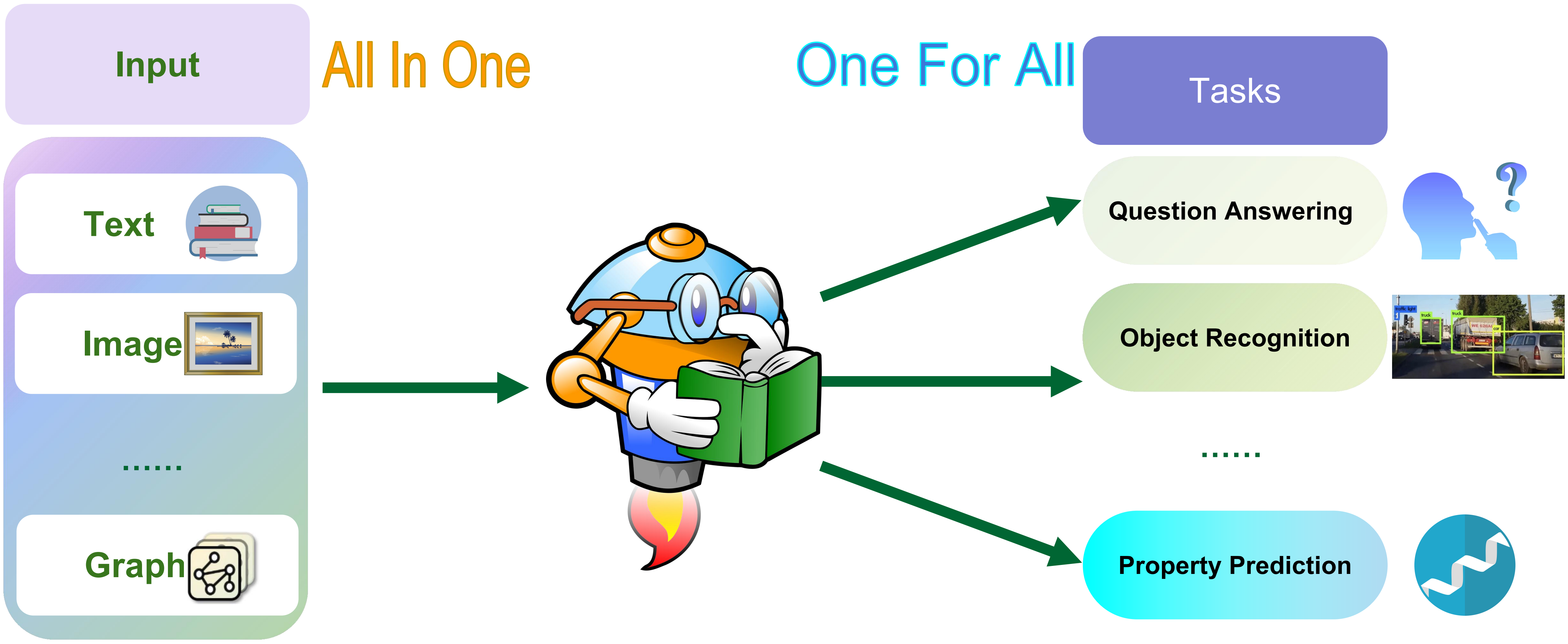
➢ Step 2: Prompting a pre–trained LLM



- A language prompt is a piece of text added to the beginning of an input text.
- The large language model can be pre-trained via next word prediction

Question-answer task is reformulated to word prediction task, which is consistent with the pre-training strategy, thus we do not need to tune LLM.

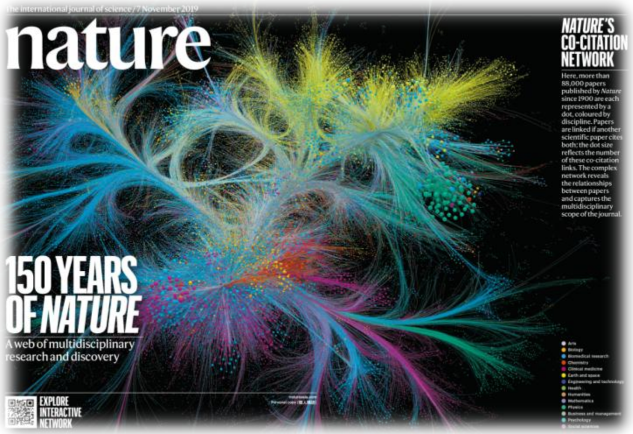
Graph AGI: All In One and One For All



Three Foundation Problems on Graph AGI

- Do we have any graph foundation model?
- How to preserve graph knowledge?
- How to use the knowledge for general tasks (or even domains)?

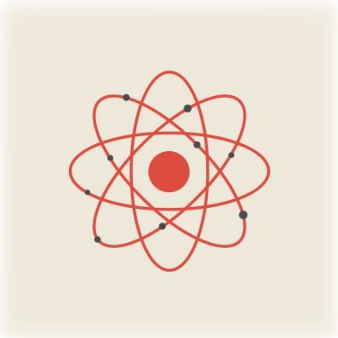
Graph



Citation Network

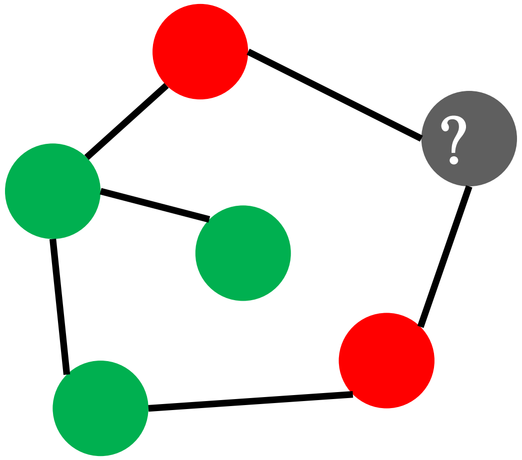
Social Network

Web Link

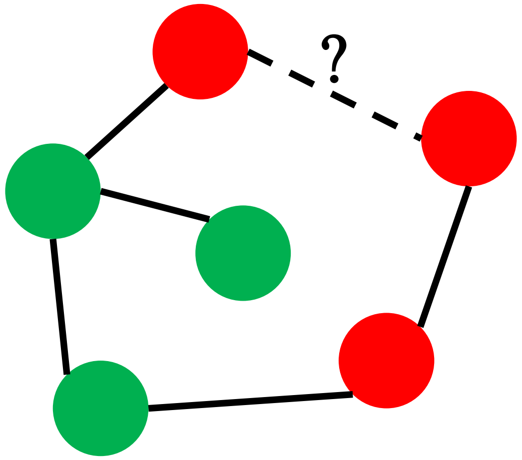


Molecular Graph

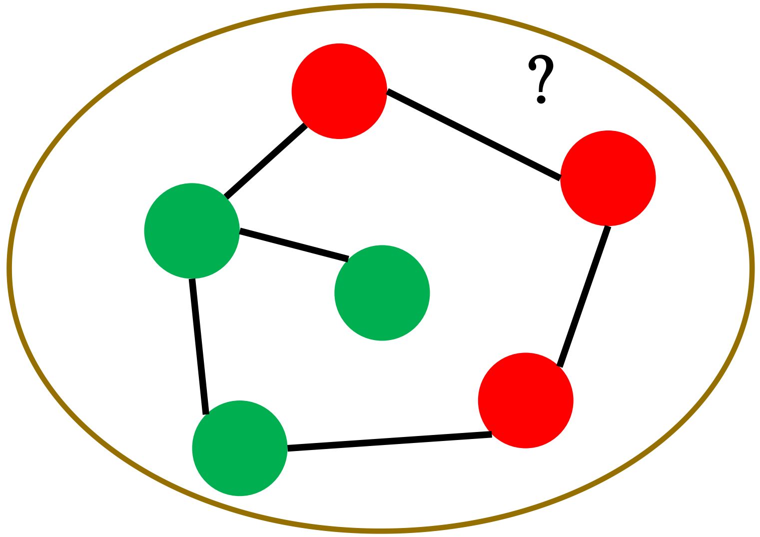
Basic Tasks in Graph



(a) Node Classification



(b) Link Prediction



(c) Graph Classification

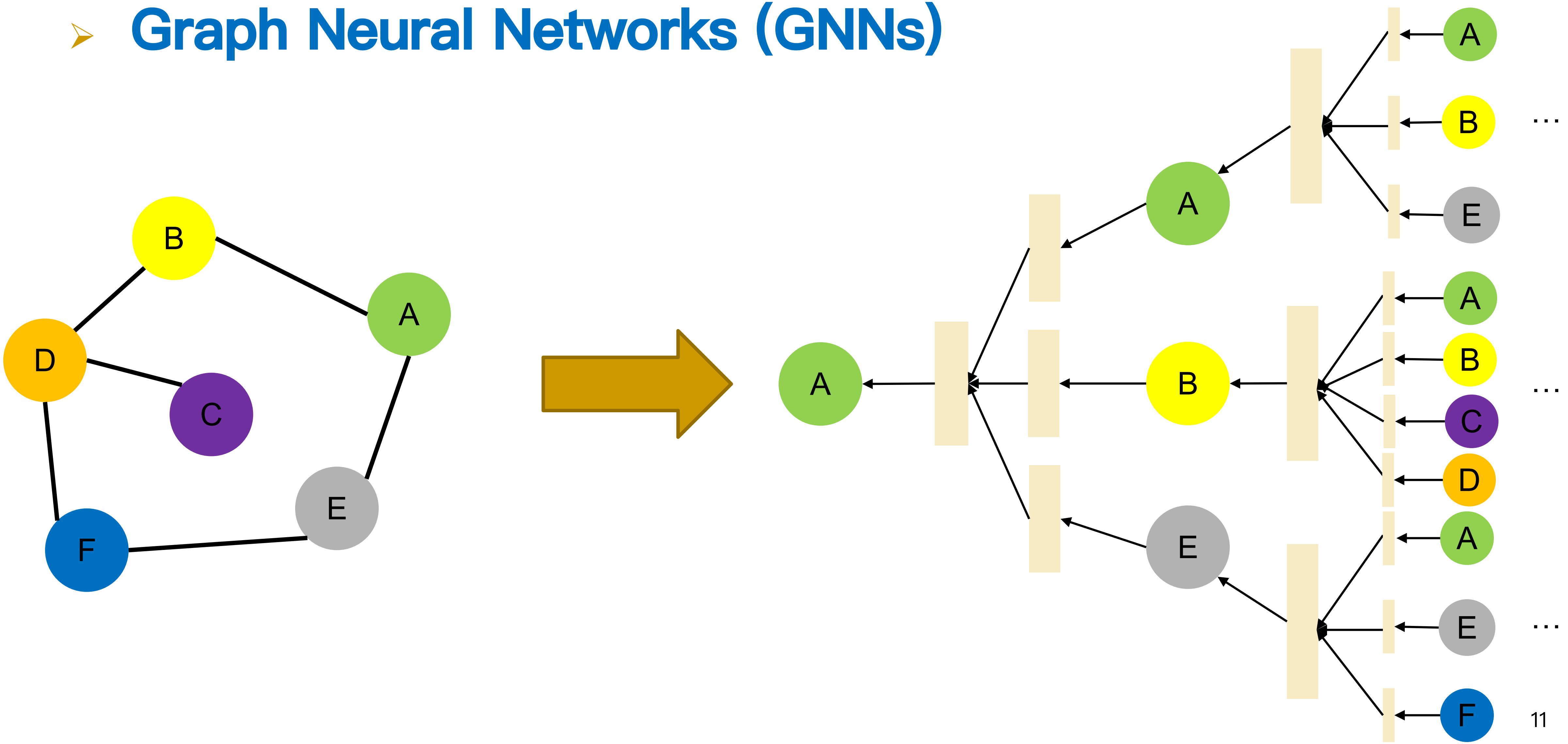
Current Graph Neural Networks

- Message-passing: GCN, GAT, etc.
- Transformer: Graph Transformer.
- From pair-wise to more general relations

We are still exploring more general graph model design

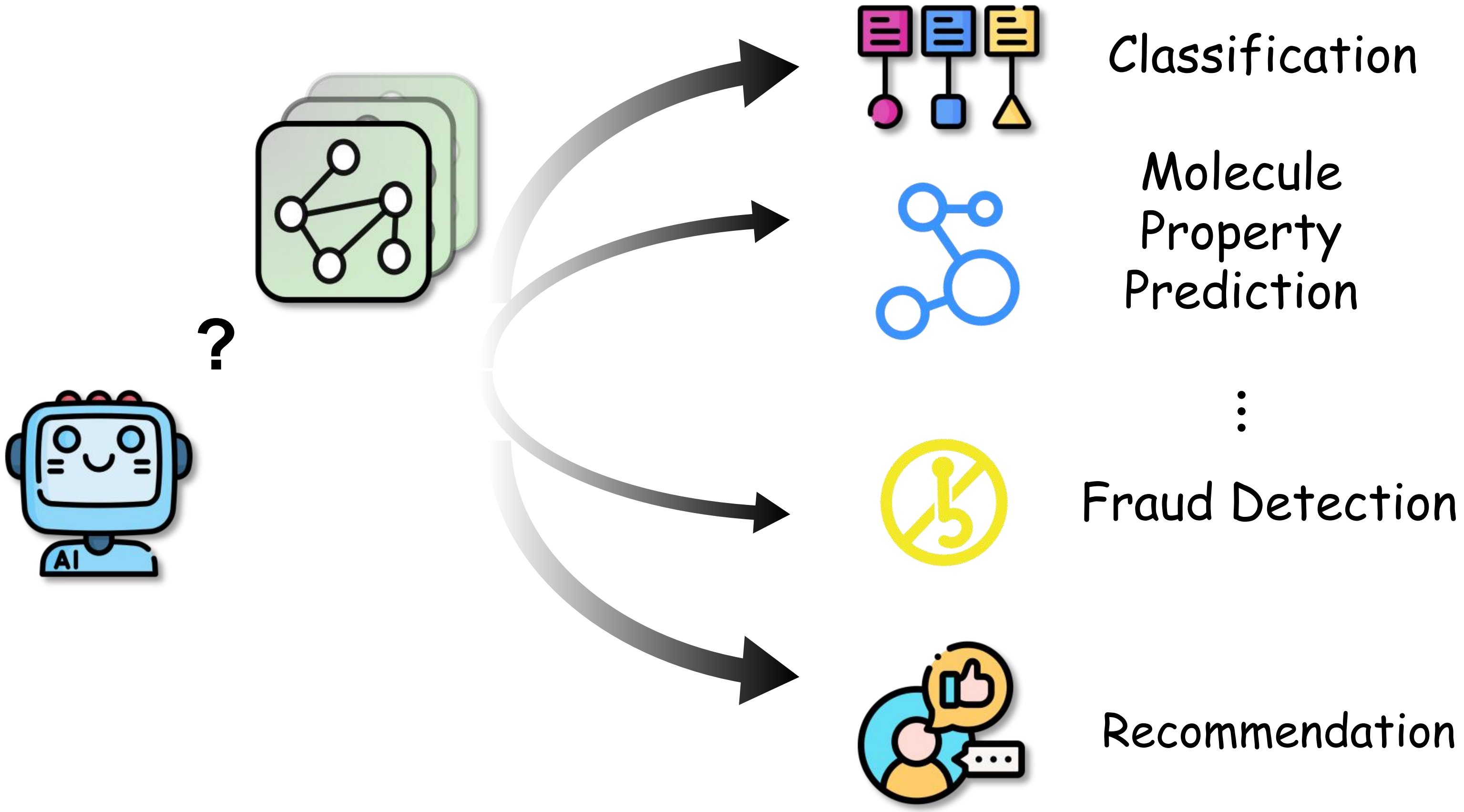
Most Popular Methods to Process Graph

➤ **Graph Neural Networks (GNNs)**



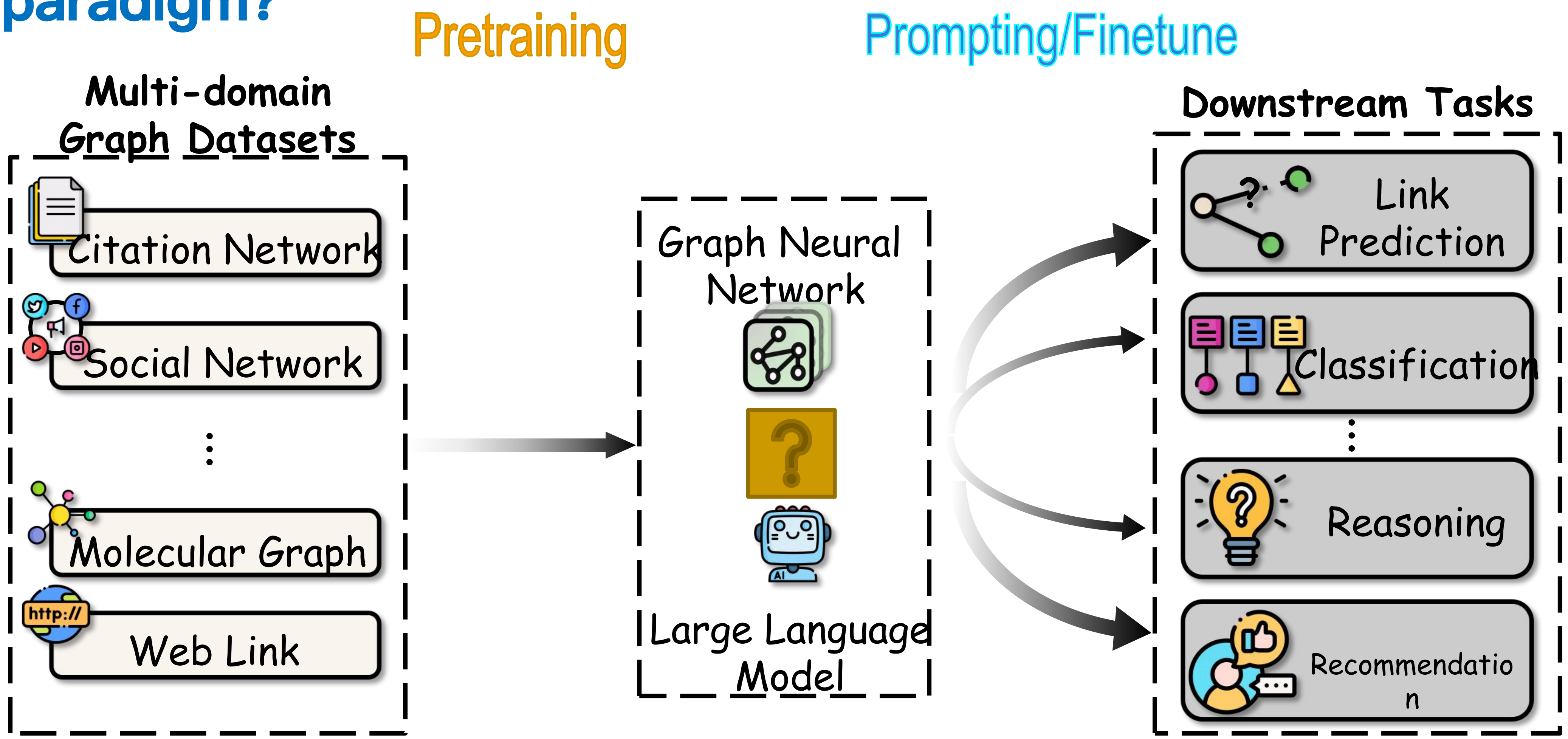
Graph

- **How can graph learning benefit from *All In One and One For All* paradigm?**



Graph

➢ **How can graph learning benefit from pretrain–prompt/finetune paradigm?**



Road Map

- **How to preserve graph knowledge?**
  - Uni-modal Pretraining
  - Multi-modal Pretraining
- **How to use the knowledge for general tasks (or even domains)?**
  - Pretraining with Prompting

Part B Uni-modal Graph Pre-training

Zhixun Li

Outline

- **Motivation**
- **Categorization of graph pre–training methods**
  - Supervised graph pre-training
  - Unsupervised graph pre-training
    - Predictive-based
    - Contrastive-based
    - Generative-based
- **Limitations**
  - Advanced graph pre-training

Motivation

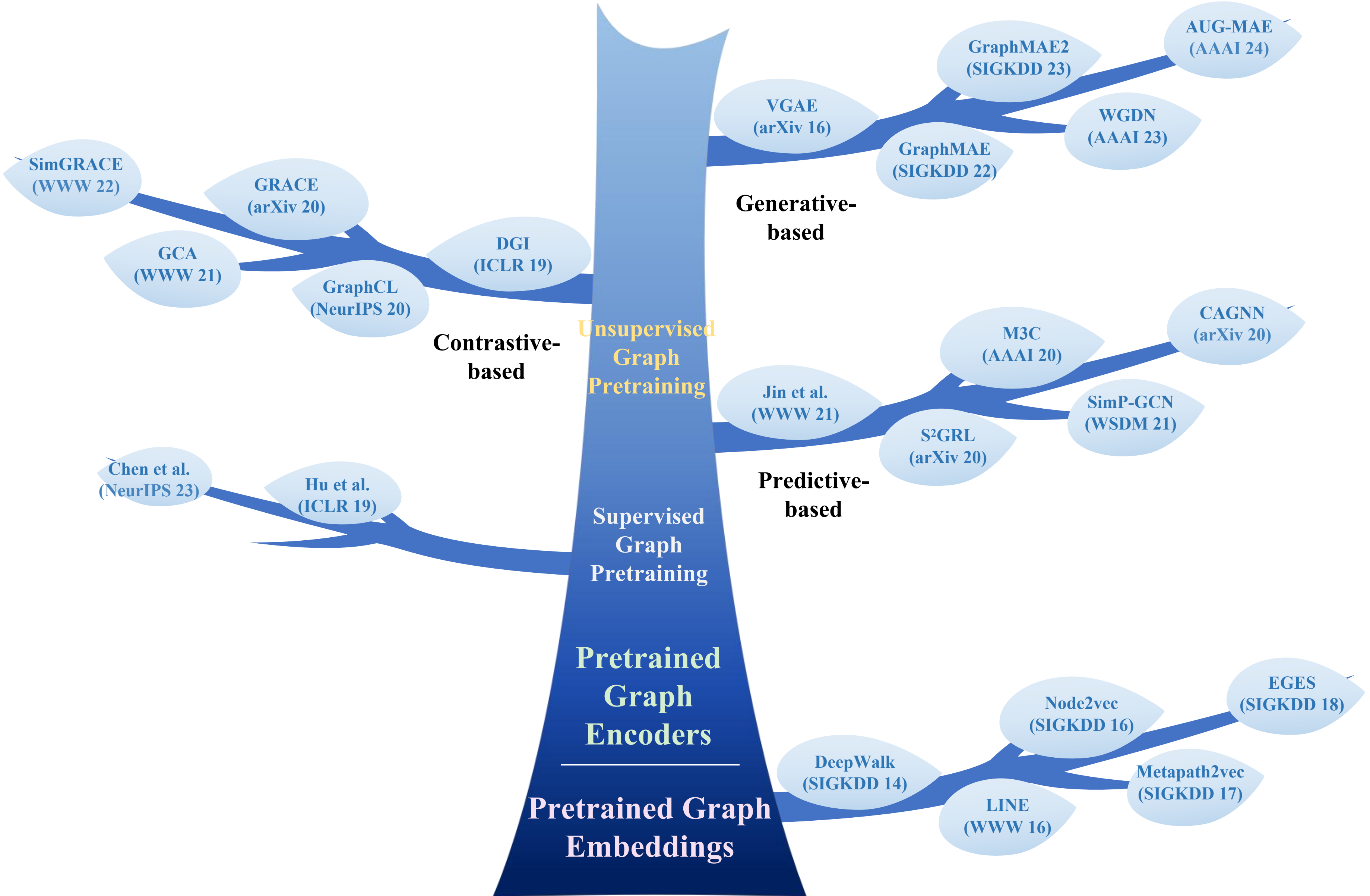
- **Scarce Labeled Data.**
  - Many applications of machine learning require a model to make accurate predictions on test examples that are distributionally different from training ones, while task-specific labels are scarce during training.
- **Out–of–distribution Generalization.**
  - Existing GNNs lack out-of-distribution generalization abilities so that their performance substantially degrades when there exist distribution shifts between training and testing graph data.

Categorization

- **First Generation: Pre–trained Graph Embeddings.**
  - Inspired by Skip-gram, the first generation pre-trained graph embedding methods aim to learn good graph embeddings for node clustering, link prediction and visualization.
- **Second Generation: Pre–trained Graph Encoders.**
  - With the emergence of expressive GNNs and Transformer, recent graph pre-training methods have embraced a transfer learning setting where the goal is to pre-train a generic encoder that can deal with different tasks.

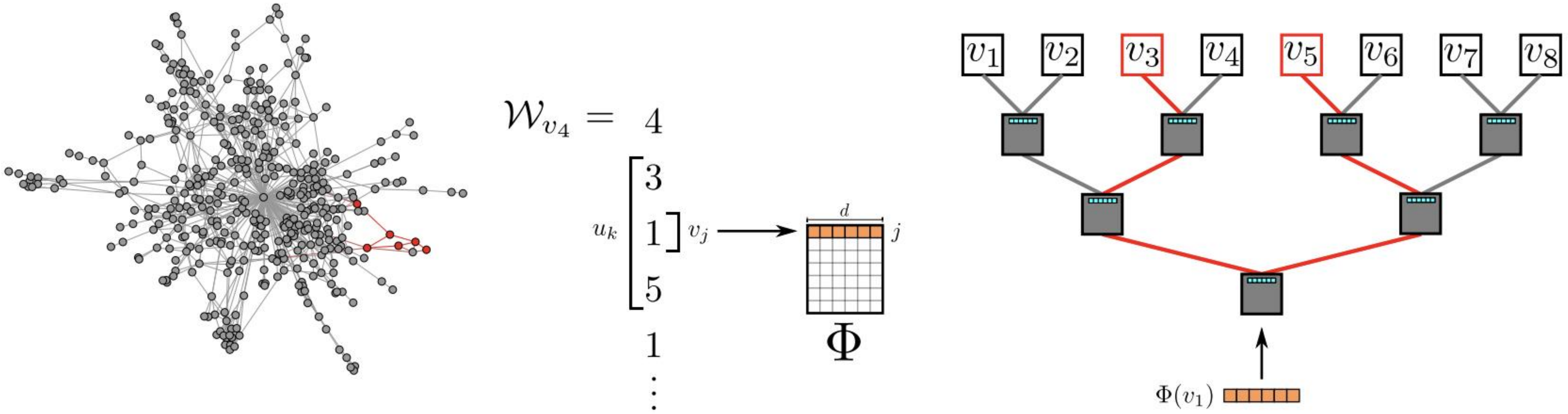
Xia, Jun, et al. "A survey of pretraining on graphs: Taxonomy, methods, and applications." arXiv preprint arXiv:2202.07893 (2022).

Categorization



Pre-trained Graph Embeddings

- ***DeepWalk*** **considers the node paths traversed by random walks over graphs as the sentences and leveraging Skip-Gram for learning latent node representations.**

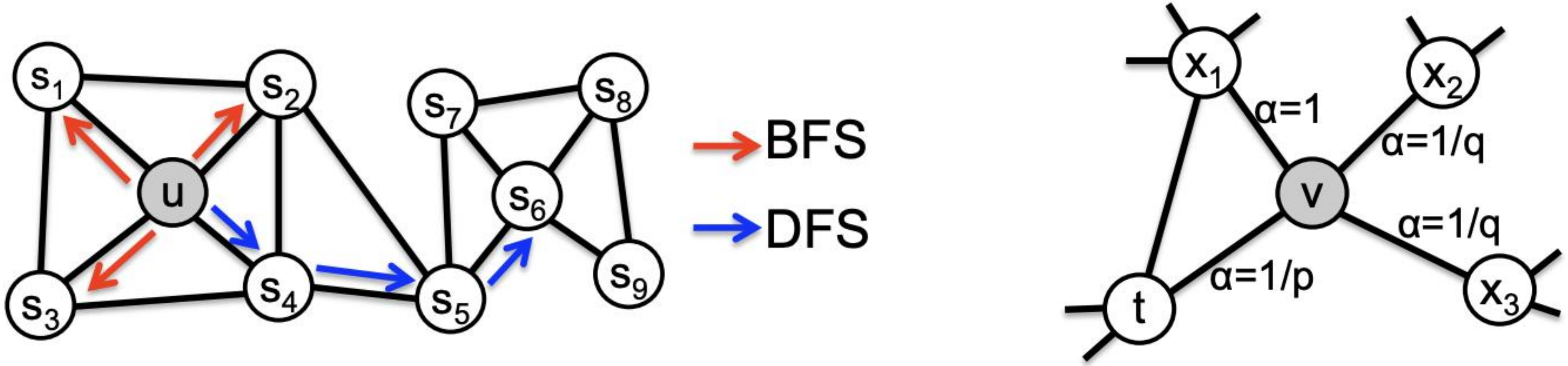


(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. SIGKDD 2014.

Pre-trained Graph Embeddings

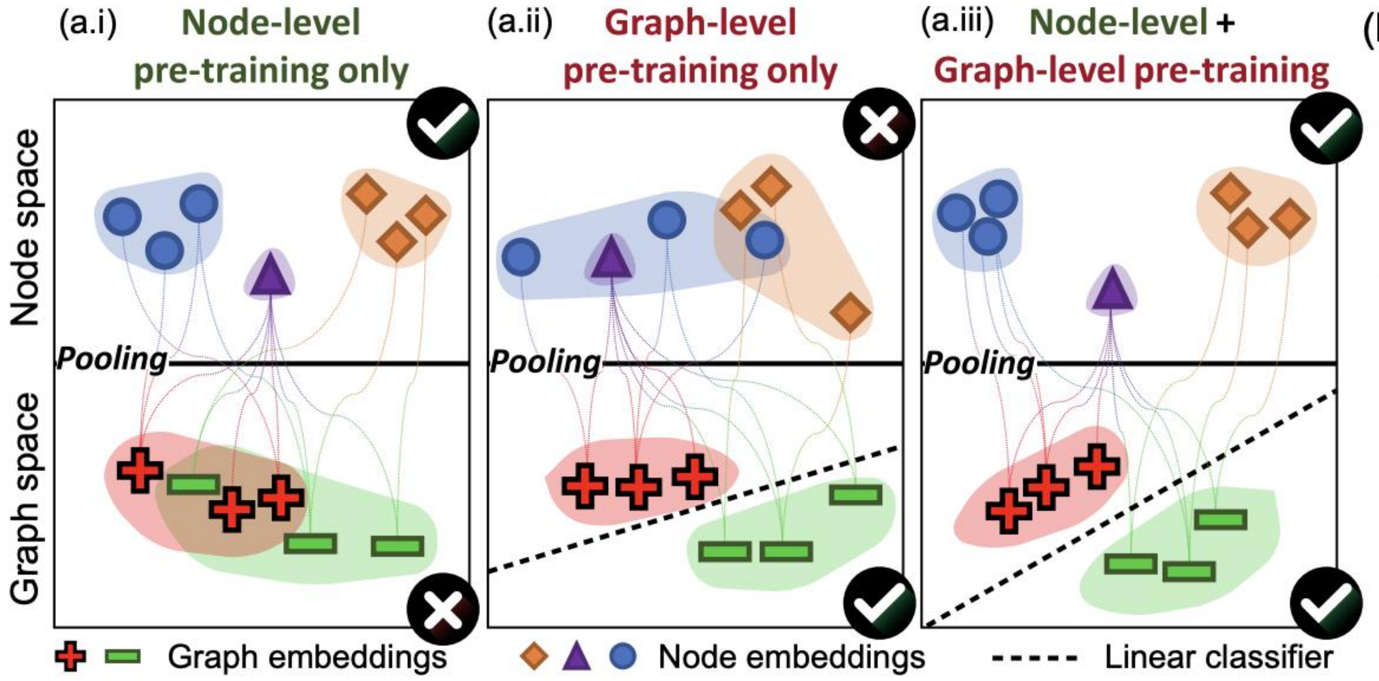
- ***Node2vec*** **learns a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes.**



Grover, Aditya, and Jure Leskovec. node2vec: Scalable feature learning for networks. SIGKDD 2016.

Supervised Graph Pre-training

➢ **Hu et al. pretrain GNNs by graph-level multi-task supervised pre-training to jointly predict a diverse set of supervised labels of individual graphs.**



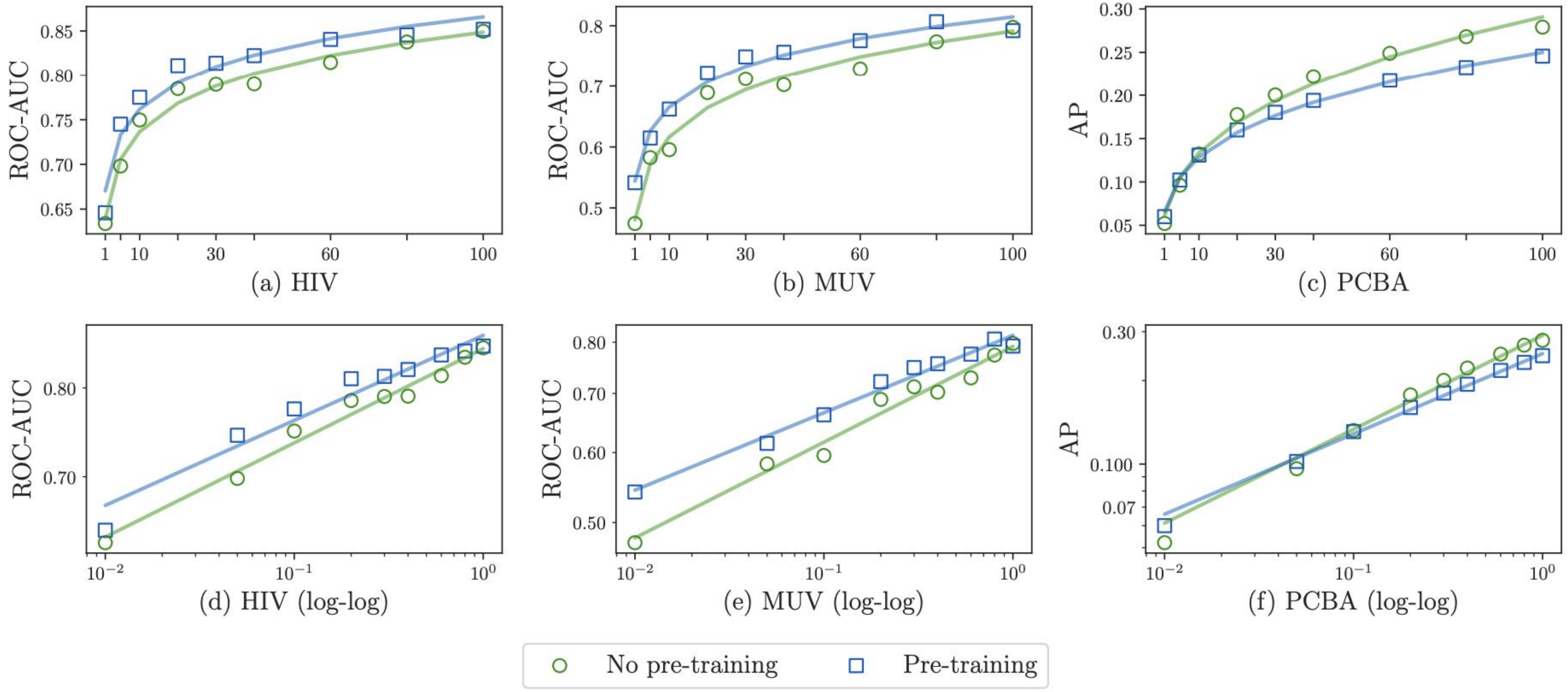
(b) Categorization of our pre-training methods

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

Hu, Weihua, et al. Strategies for pre-training graph neural networks. arXiv preprint arXiv:1905.12265 (2019).

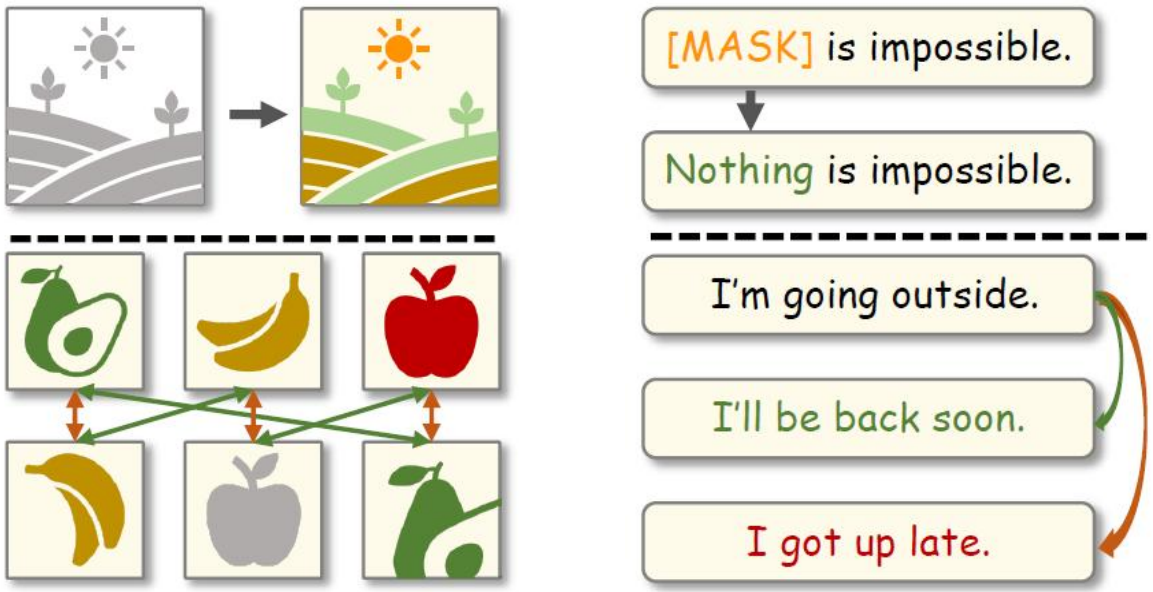
Supervised Graph Pre-training

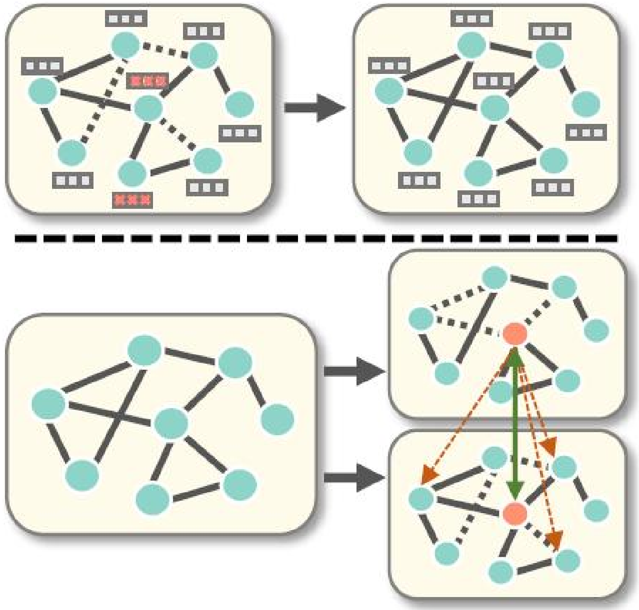
➢ **Influence of Pre-training on the Scaling Laws**



Chen, Dingshuo, et al. Uncovering neural scaling laws in molecular representation learning. NeurIPS 2023.

Unsupervised Graph Pre–training





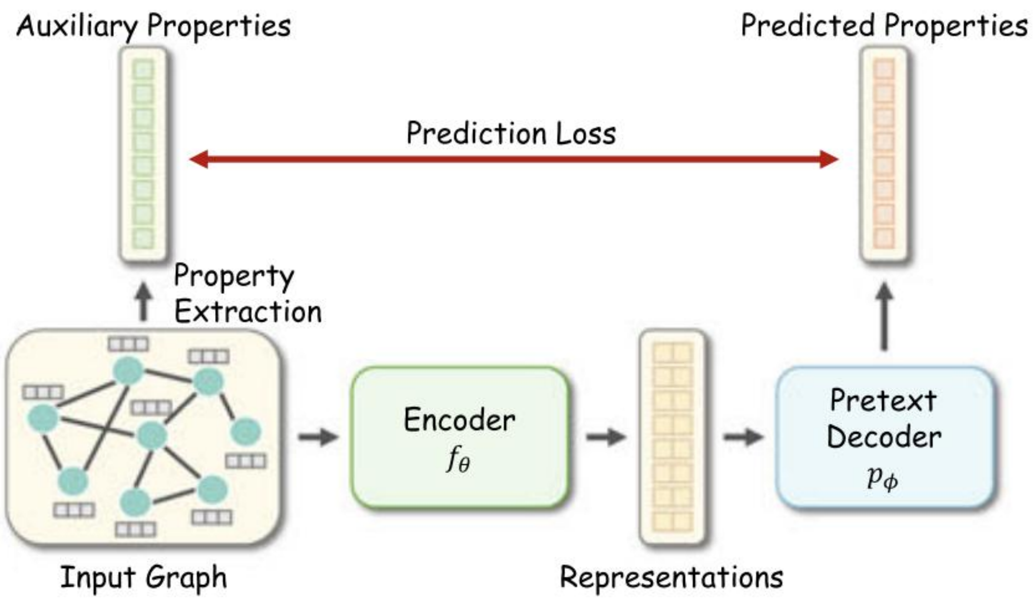
- Regular grid space
- Sample independency

- Non-Euclidean space
- Node dependency

Liu, Yixin, et al. Graph self-supervised learning: A survey. TKDE 2022.

Unsupervised Graph Pre-training

- **Predictive-based methods acquire supervision signals from the node-, link- and graph-level properties which *can be obtained from the graph freely*.**



Liu, Yixin, et al. Graph self-supervised learning: A survey. TKDE 2022.

Predictive-based

- Jin et al. first deepen understandings on *when*, *why*, and *which* strategies of self-supervised predictive-based work with GNNs by empirically studying numerous basic pretext tasks on graphs.

Table 3: Performance evaluation of using SSL for GNNs.

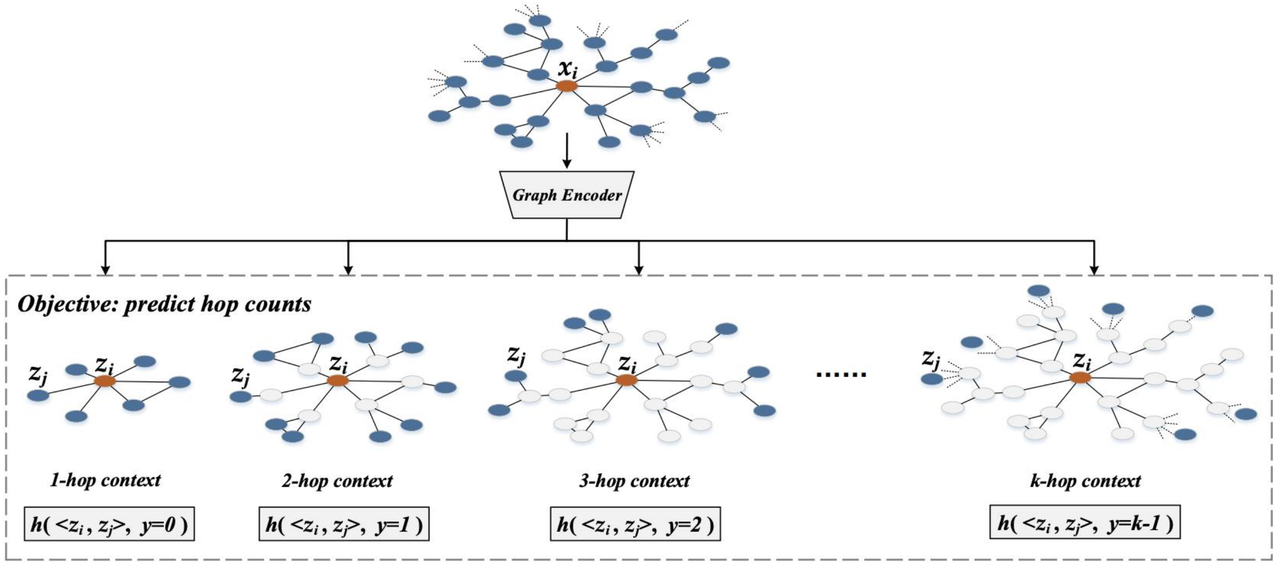
Model	Joint Training			Two-stage Training		
	Cora	Citeseer	Pubmed	Cora	Citeseer	Pubmed
GCN	81.32	71.53	79.28	81.32	71.53	79.28
GCN-DroppedGraph	81.03	71.29	79.28	81.03	71.29	79.26
GCN-PCA	81.74	70.38	78.83	81.74	70.38	78.83
NodeProperty	81.94	71.60	79.44	81.59	71.69	79.24
EdgeMask	81.69	71.51	78.90	81.44	71.57	79.33
PairwiseNodeDistance	83.11	71.90	80.05	82.39	72.02	79.57
Distance2Cluster	83.55	71.44	79.88	81.80	71.55	79.51
AttributeMask	81.47	70.57	78.88	81.31	70.40	78.72
PairwiseAttrSim	83.05	71.67	79.45	81.57	71.74	79.42

Jin, Wei, et al. Self-supervised learning on graphs: Deep insights and new direction. arXiv preprint arXiv:2006.10141 (2020).

Predictive–based | S²GRL

➢ S²GRL

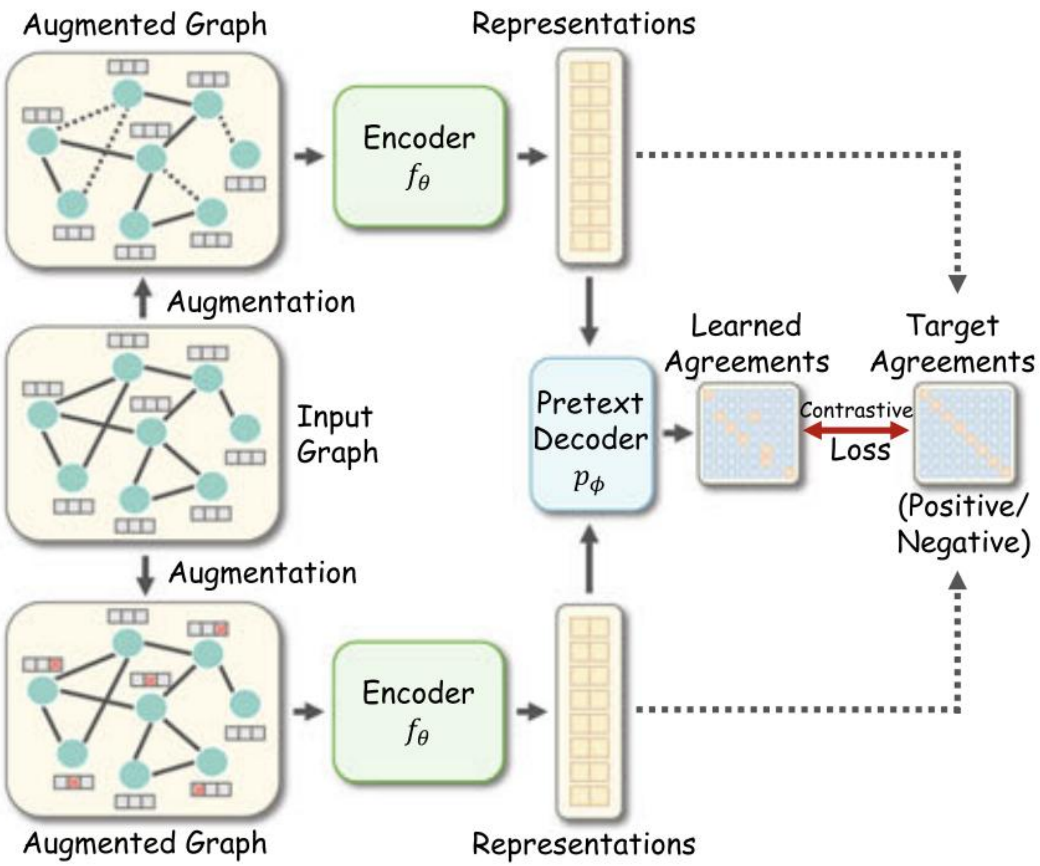
- Predicted property: ***shortest path***.
- They randomly select pairs of nodes in a graph and train a well-designed neural network to predict the contextual position of one node relative to the other.



Peng, Zhen, et al. Self-supervised graph representation learning via global context prediction. arXiv preprint arXiv:2003.01604 (2020).

Contrastive–based

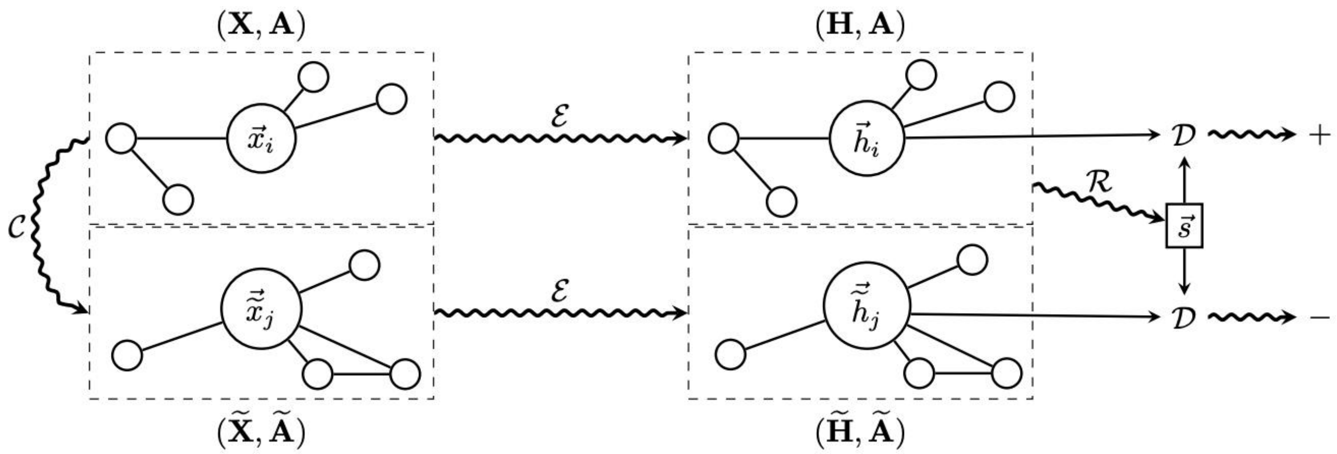
- **Motivation**
  - Contrastive-based methods are built on the idea of mutual information (MI) maximization, which learns by predicting the agreement between two augmented instances.
- **Components**
  - Graph Augmentations
  - Graph contrastive pretext tasks
  - Mutual information estimation



Liu, Yixin, et al. Graph self-supervised learning: A survey. TKDE 2022.

Contrastive–based | DGI

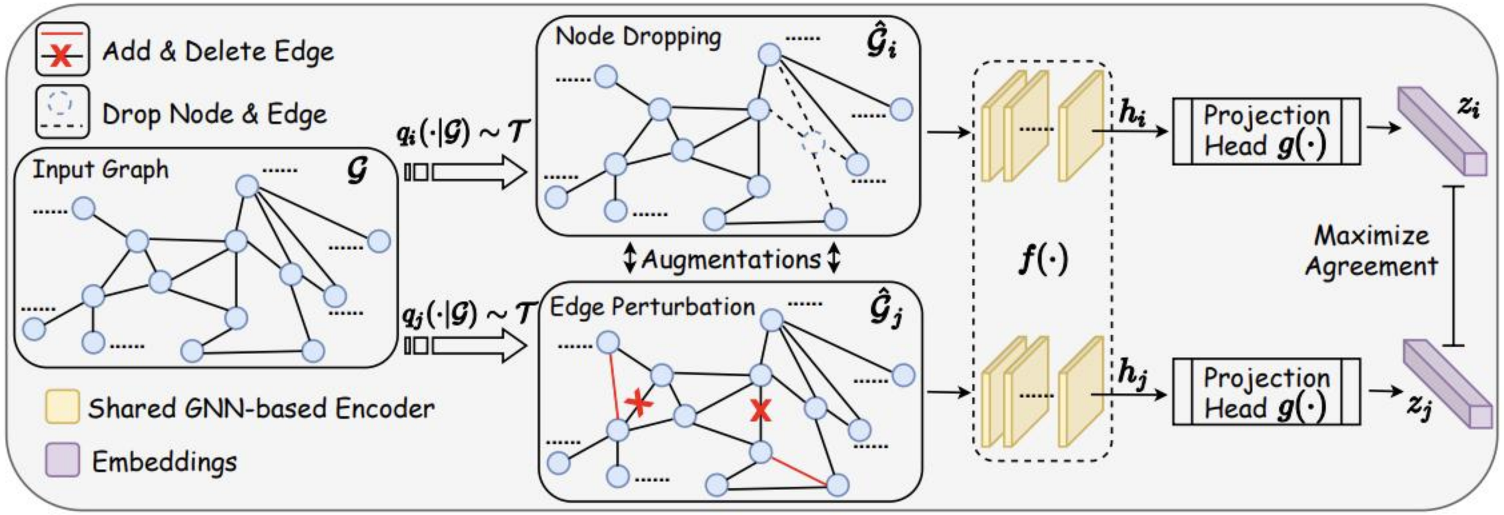
- **Motivation of DGI**
  - DGI relies on maximizing mutual information between patch representations and corresponding high-level summaries of graphs.
  - The learnt patch representations summarize subgraphs centered around nodes of interest, and can thus by reused for downstream node-wise learning tasks.



Veličković, Petar, et al. "Deep graph infomax." arXiv preprint arXiv:1809.10341 (2018).

Contrastive-based | GraphCL

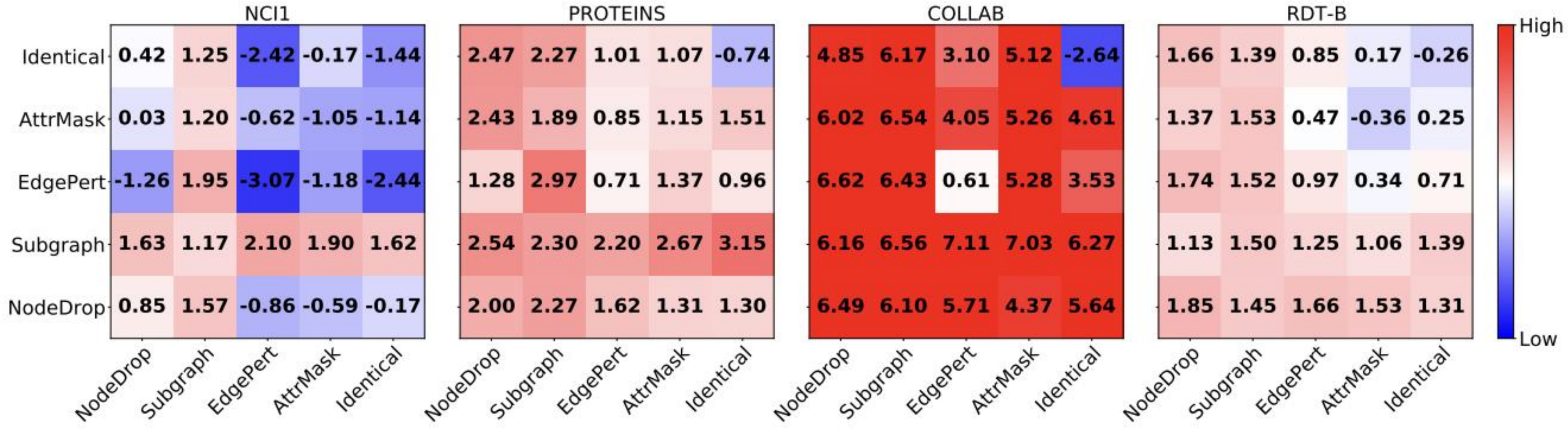
- You et al. first design four types of graph augmentations in graph contrastive learning. And they systematically study the impact of various combinations of graph augmentations on multiple datasets.



You, Yuning, et al. Graph contrastive learning with augmentations. NeurIPS 2020.

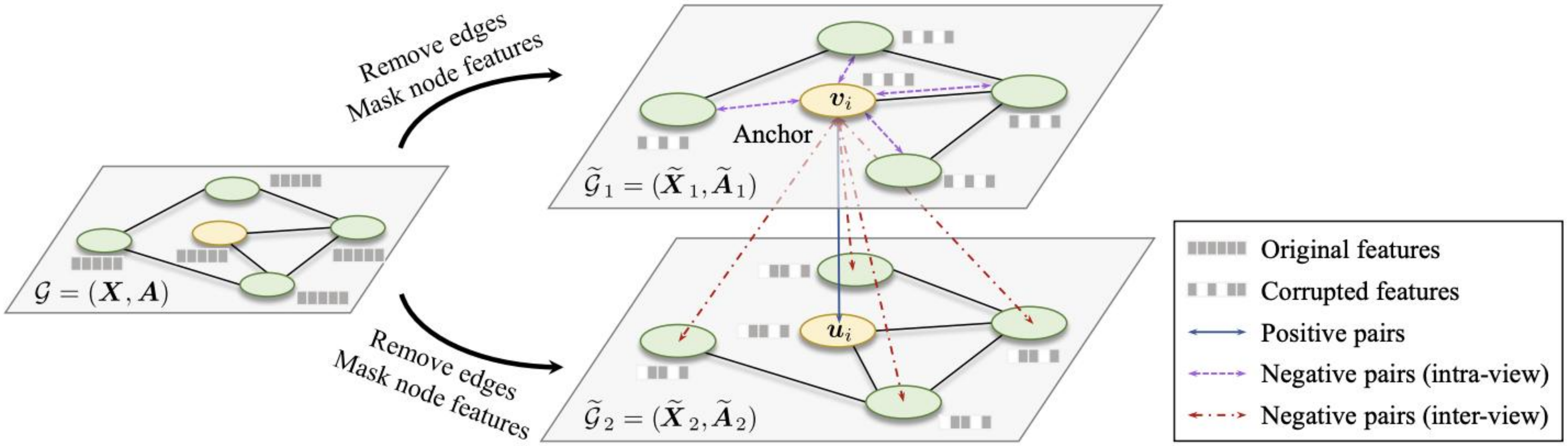
Contrastive-based | GraphCL

- **Graph data augmentation:**
  - NodeDrop, Subgraph, EdgePert, AttrMask
- **Observations**
  - Data augmentations are crucial in graph contrastive learning.
  - Composing different augmentations benefits more.
  - Edge perturbation benefits social networks but hurts some biochemical molecules.
  - Applying attribute masking achieves better performance in denser graphs.
  - Node dropping and subgraph are generally beneficial across datasets.



Contrastive-based | GRACE

➢ **Inspired by the success of self-supervised learning in CV, like SimCLR, Zhu et al. proposed GRACE for unsupervised graph representation learning by leveraging a contrastive objective at the node level.**



Zhu, Yanqiao, et al. Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131 (2020).

Contrastive-based | GRACE

- **Graph data augmentation**
  - GRACE firstly generates two graph views by randomly corrupting the original graph.
- **Learning objective**
  - Then, GRACE employs contrastive objective that enforces the encoded embeddings of each node in the two different views agree with each other and can be distinguished from embeddings of other nodes.

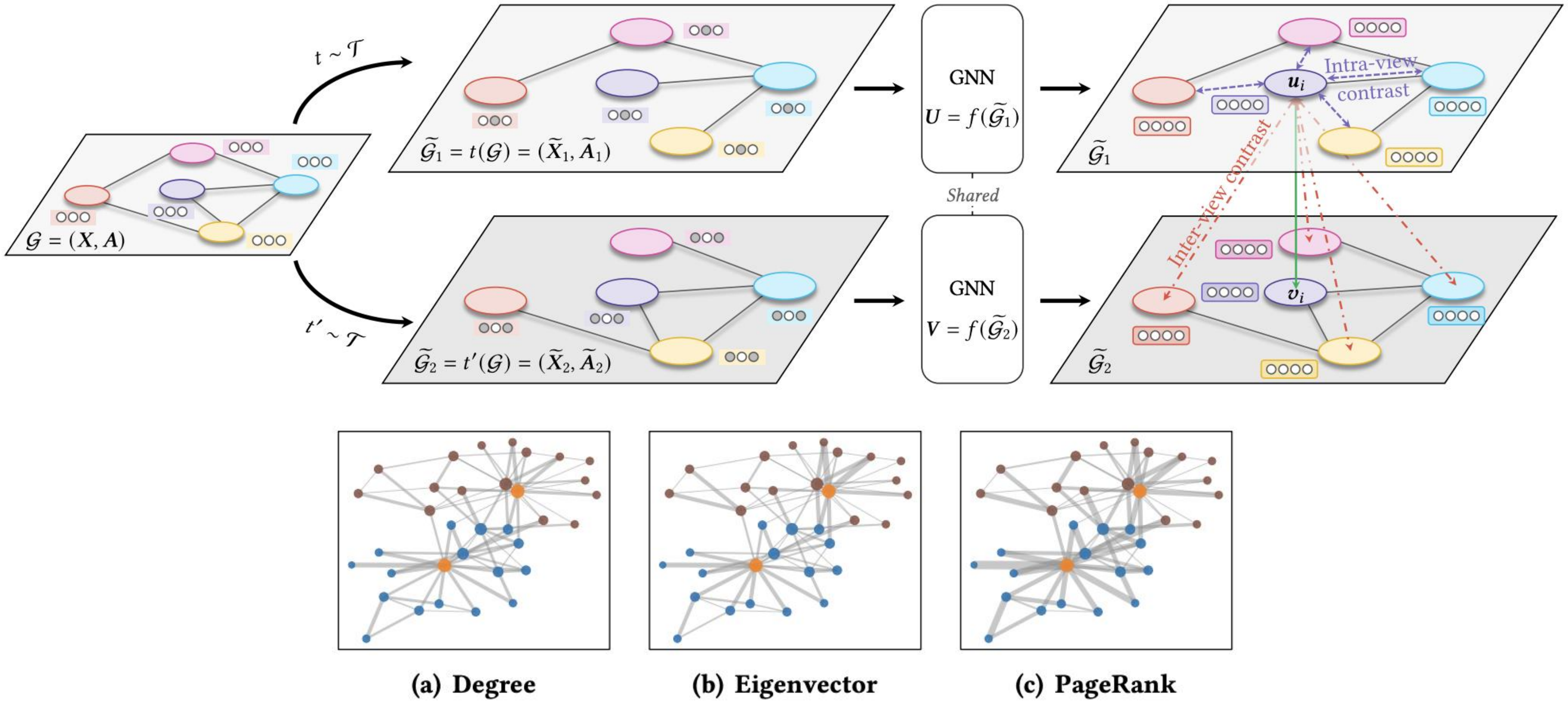
$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}{\underbrace{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}_{\text{the positive pair}} + \underbrace{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_k)/\tau}}_{\text{inter-view negative pairs}} + \underbrace{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} e^{\theta(\boldsymbol{u}_i, \boldsymbol{u}_k)/\tau}}_{\text{intra-view negative pairs}}},$$

$$\mathcal{J} = \frac{1}{2N} \sum_{i=1}^{N} \left[\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) + \ell(\boldsymbol{v}_i, \boldsymbol{u}_i) \right].$$

Contrastive-based | GCA

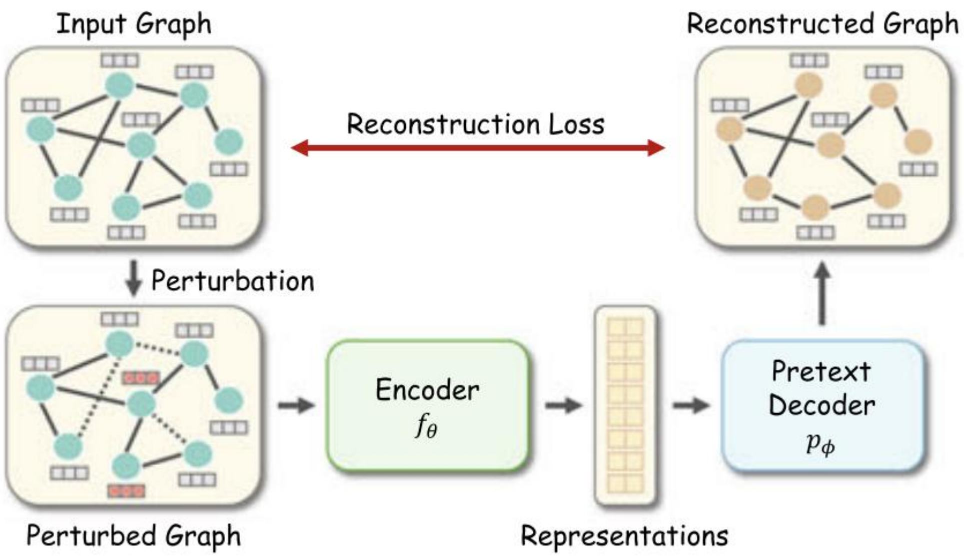
➢ Graph data augmentation

- Previous work ignores the discrepancy in the impact of nodes and edges when performing data augmentation.



Generative-based

- Generative-based methods inputs a perturbated graph. And in the pretext task, a generative decoder tries to recover the original graph from the representation, with a loss function aiming to minimize the difference between the reconstructed and original graph.



Liu, Yixin, et al. Graph self-supervised learning: A survey. TKDE 2022.

Generative–based | VGAE

Inference model

- VGAE tasks a simple inference model parameterized by a two-layer GCN

$$q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q(\mathbf{z}_i \mid \mathbf{X}, \mathbf{A}) \,, \quad \text{with} \quad q(\mathbf{z}_i \mid \mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{z}_i \mid \boldsymbol{\mu}_i, \operatorname{diag}(\boldsymbol{\sigma}_i^2)) \,.$$

Generative model

- The generative model of VGAE is given by an inner product between latent variables

$$p\left(\mathbf{A} \mid \mathbf{Z}\right) = \prod_{i=1}^{N} \prod_{j=1}^{N} p\left(A_{ij} \mid \mathbf{z}_i, \mathbf{z}_j\right) \,, \quad \text{with} \quad p\left(A_{ij} = 1 \mid \mathbf{z}_i, \mathbf{z}_j\right) = \sigma(\mathbf{z}_i^\top \mathbf{z}_j) \,,$$

Learning

- Optimize the variational lower bound w.r.t. the variational parameters

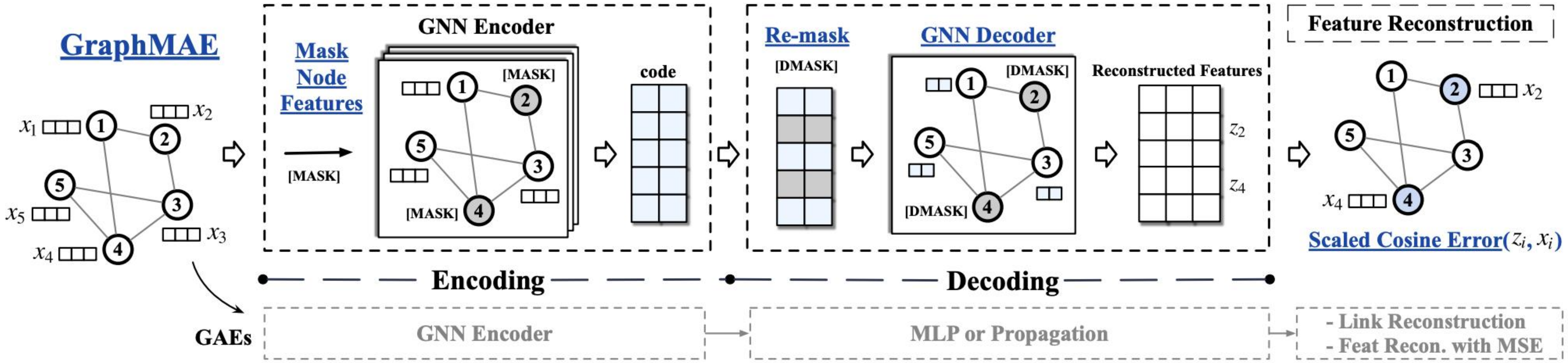
$$\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X},\mathbf{A})}\big[\log p\left(\mathbf{A} \mid \mathbf{Z}\right)\big] - \mathrm{KL}\big[q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) \,||\, p(\mathbf{Z})\big] \,,$$

Kipf, Thomas N., and Max Welling. "Variational graph auto-encoders." arXiv preprint arXiv:1611.07308 (2016).

Generative-based | GraphMAE

➢ Inspired by CV and NLP

- While contrastive SSL methods have experienced an emergence in graph learning, generative SSL has been gaining steadily increasing significant thanks to several groundbreaking practices, such as BERT and GPT in NLP as well as MAE in CV.



Hou, Zhenyu, et al. Graphmae: Self-supervised masked graph autoencoders. SIGKDD 2022.

Generative–based | GraphMAE

- **Objective**
  - Instead of reconstructing both features and structure, which unfortunately does not empower GAEs to produce significant progree, GraphMAE aims to reconstruct node features.
- **Weak Decoder**
  - Traditional GAEs employ either no neural decoders or a simple MLP for decoding with less expressiveness, causing the latent code to be nearly identical to input features. Therefore, GraphMAE utilizes re-mask decoding to process the latent code for decoding.
- **New Loss Function**
  - MSE could suffer from the issues of sensitivity and low selectivity. Therefore, GraphMAE leverages the cosine error as the criterion to reconstruct original node features.

Hou, Zhenyu, et al. Graphmae: Self-supervised masked graph autoencoders. SIGKDD 2022.

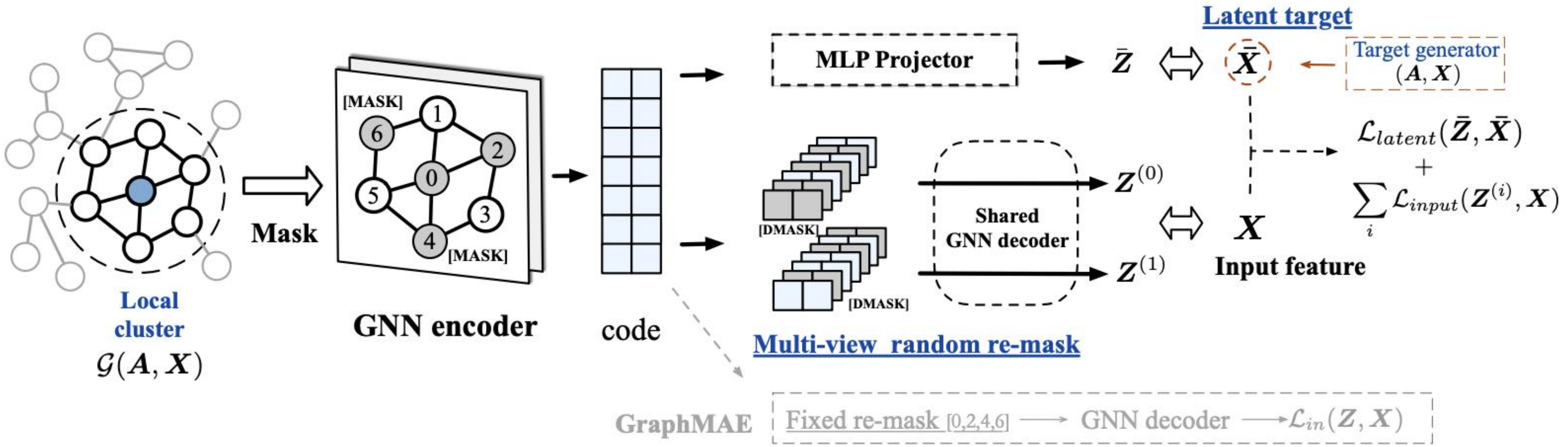
Generative–based | GraphMAE2

Limitation of GraphMAE

- The reconstruction of masked features fundamentally relies on the discriminability of the input node features.

Solution

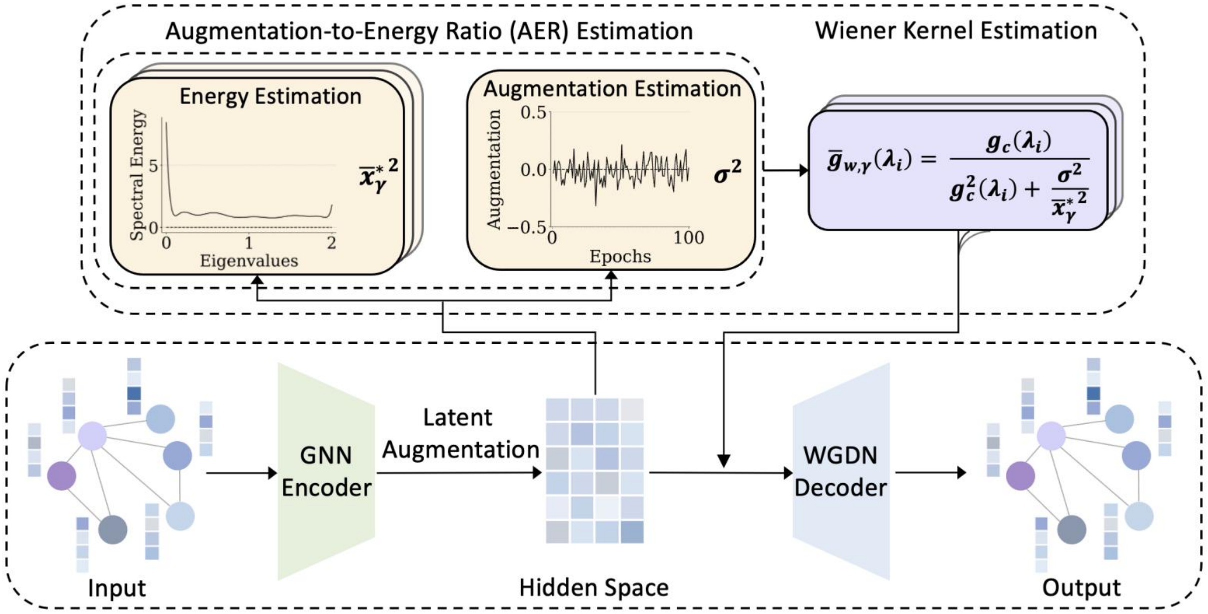
- Impose regularization on target reconstruction.



Generative–based | WGDN

Motivations

- Generative models weaponed with powerful decoder could achieve comparable or even better representation pwoer than contrastive models.
- A powerful decoder should at least remain effective against augmentations.



Cheng, Jiashun, et al. Wiener graph deconvolutional network improves graph self-supervised learning. AAAI 2023.

Limitations

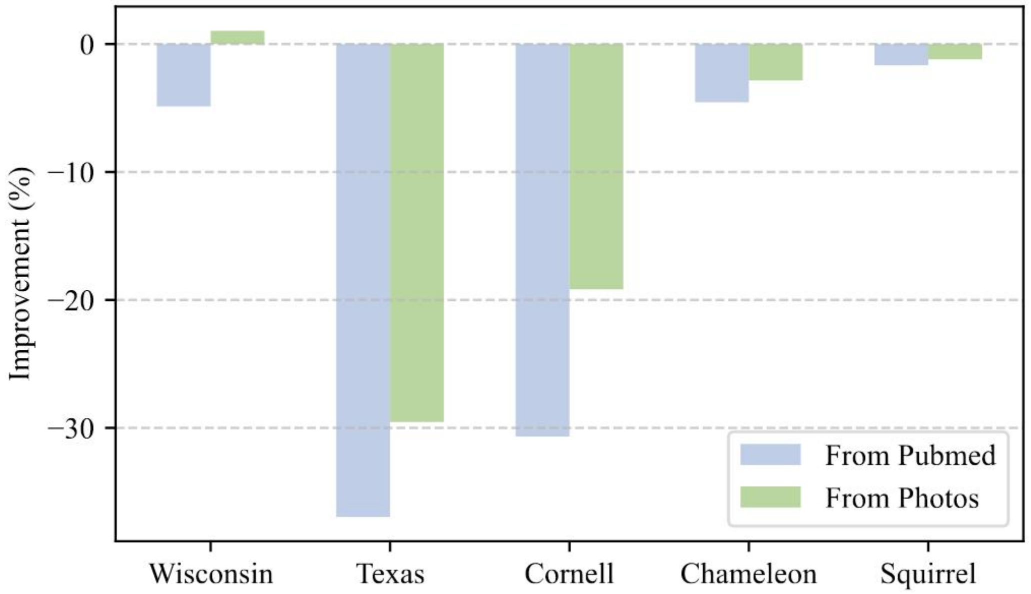
- **Hard to transfer**
  - Graph structure is extremely diverse. Graphs inherently exhibit diverse topologies and features, making it challenging to identify and leverage common patterns across different domains.
  - Features in one graph mighthave no direct counterpart in another, making it incredibly challenging to align these features in a meaningful way.

- **Not versatile**
  - Graph Neural Networks is hard to conduct multiple downstream tasks simultaneously.

Advanced Pre–training | GCOPE

Motivation

- Tranferring from a single source dataset does indeed negatively affect the target task. In order to overcome this obstacle, it is necessary to expand the scope of the source dataset so that it can offer valuable insights for the downstream task.



Advanced Pre-training | GCOPE

Aligning Graphs by Coordinators

- Feature Projection (singular value decomposition and attention mechanism).

$$\tilde{X}^{(i)} = \mathrm{Proj}(X^{(i)}) \in \mathbb{R}^{|\mathcal{V}^{(i)}| \times d_{\mathrm{P}}},$$

- Graph Coordinators

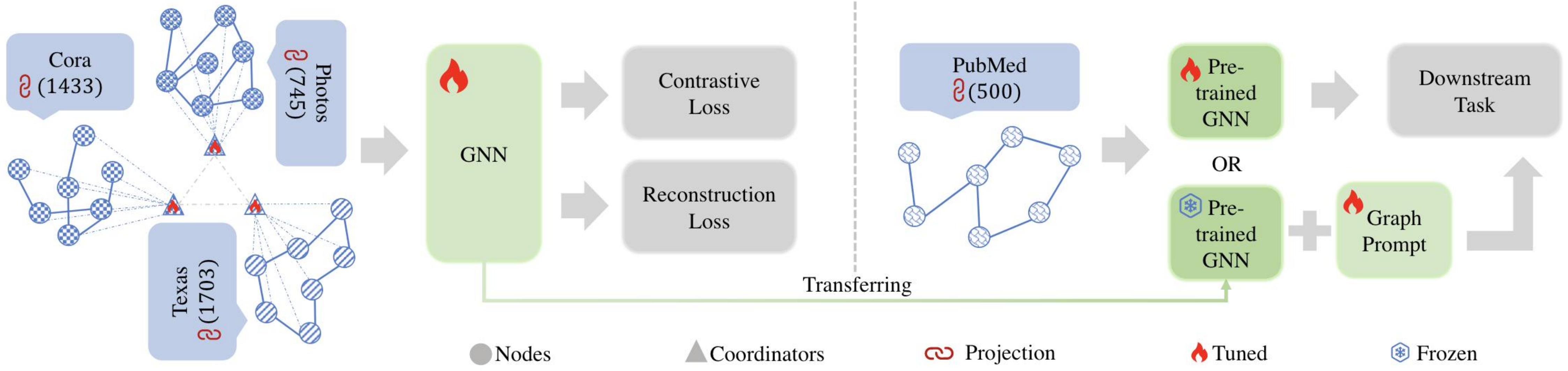
$$\tilde{A} = \begin{bmatrix} A_{\mathrm{diag}} & R_A^T \\ R_A & R_R \end{bmatrix}, \qquad R_A^{(i)}(j) = \begin{cases} 1 & \sum_1^i |\mathcal{V}^{(k)}| \leq j < \sum_1^{i+1} |\mathcal{V}^{(k)}| \\ 0 & \text{otherwise.} \end{cases}$$

- Learning Objective

$$\mathcal{L} = -\log \frac{\exp(\mathrm{sim}(h(\mathrm{PS}(\tilde{X}, \tilde{A}, a_i)), h(\mathrm{PS}(\tilde{X}, \tilde{A}, a_j))/\tau)}{\sum \exp(\mathrm{sim}(h(\mathrm{PS}(\tilde{X}, \tilde{A}, a_i)), h(\mathrm{NS}(\tilde{X}, \tilde{A}, a_j))/\tau)} + \|\tilde{X} - \hat{X}\|_2$$

Advanced Pre-training | GCOPE

➢ Overview of GCOPE



Zhao, Haihong, et al. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. SIGKDD 2024.

Advanced Pre-training | GCOPE

➢ Cross-domain transfer learning performance

Table 2: Cross-domain transfer learning performance (mean±std Acc/AUC/F1) on homophilic datasets (C-way-1-shot). IMP (%): the average improvement of GCOPE over the rest. GCL and Sim respectively represent GraphCL and SimGRACE.

Training schemes	Methods	Cora Acc	Cora AUC	Cora F1	Citeseer Acc	Citeseer AUC	Citeseer F1	Pubmed Acc	Pubmed AUC	Pubmed F1	Computers Acc	Computers AUC	Computers F1	Photos Acc	Photos AUC	Photos F1
supervised	GCN	0.3012±.06	0.6444±.04	0.2591±.04	0.4358±.09	0.7234±.07	0.3583±.10	0.4210±.01	0.6040±.06	0.3026±.04	0.2602±.07	0.6773±.02	0.2428±.04	0.4603±.04	0.8458±.01	0.4592±.04
	GAT	0.3646±.04	0.6769±.03	0.3108±.04	0.3695±.05	0.7232±.06	0.3305±.04	0.4209±.04	0.5710±.06	0.3227±.07	0.3482±.07	0.6878±.05	0.2397±.05	0.4742±.08	0.8213±.02	0.4498±.07
	BWGNN	0.2543±.05	0.5563±.03	0.1971±.02	0.3599±.07	0.6954±.05	0.3112±.06	0.3976±.03	0.4934±.03	0.2686±.04	0.2768±.05	0.6273±.03	0.1864±.03	0.4113±.04	0.7769±.00	0.3883±.01
	FAGCN	0.3819±.03	0.6818±.04	0.3009±.09	0.5219±.08	0.8042±.03	0.4667±.08	0.4522±.02	0.5622±.04	0.4275±.07	0.4651±.04	0.7762±.02	0.3009±.07	0.5937±.05	0.8847±.00	0.5346±.03
IP + finetuning	GCL+GCN	0.2507±.06	0.6350±.03	0.2240±.03	0.3140±.02	0.6661±.04	0.2397±.02	0.4217±.02	0.5257±.05	0.2896±.07	0.2856±.04	0.6467±.03	0.1653±.06	0.5533±.01	0.8661±.01	0.5217±.01
	GCL+FAGCN	0.3892±.05	0.7228±.03	0.3619±.05	0.4461±.02	0.7781±.01	0.4126±.02	0.4532±.02	0.5708±.03	0.4168±.04	0.4371±.06	0.7616±.01	0.3450±.02	0.6273±.01	0.8710±.01	0.5406±.03
	Sim+GCN	0.2492±.02	0.5765±.03	0.1567±.04	0.2950±.06	0.6203±.06	0.1812±.06	0.3980±.01	0.5067±.02	0.2805±.01	0.2666±.10	0.6286±.01	0.1603±.03	0.4290±.04	0.7645±.02	0.3955±.02
	Sim+FAGCN	0.3957±.03	0.7284±.02	0.3585±.01	0.5101±.03	0.7969±.01	0.4615±.04	0.4398±.01	0.5535±.01	0.4225±.02	0.4393±.01	0.7718±.02	0.3100±.02	0.5704±.02	0.8543±.02	0.4984±.01
GCOPE + finetuning	GCL+GCN	0.3368±.02	0.6971±.04	0.2967±.03	0.3701±.03	0.7066±.02	0.3265±.05	0.4443±.04	0.5888±.04	0.4242±.04	0.3439±.03	0.7023±.01	0.2976±.03	0.5635±.02	0.8733±.00	0.5480±.02
	GCL+FAGCN	0.4618±.03	0.7597±.05	0.4388±.05	0.5631±.03	0.8258±.02	0.4953±.04	0.4591±.01	0.5512±.01	0.4203±.03	0.4465±.01	0.7747±.00	0.3432±.03	0.6329±.02	0.8850±.00	0.5935±.03
	Sim+GCN	0.2525±.05	0.5744±.03	0.1722±.06	0.3475±.05	0.6527±.05	0.2704±.05	0.4116±.00	0.5166±.04	0.2994±.03	0.3230±.01	0.6994±.00	0.2515±.00	0.4772±.03	0.7851±.01	0.4277±.02
	Sim+FAGCN	0.3875±.04	0.7163±.03	0.3355±.08	0.5704±.04	0.8425±.01	0.5178±.04	0.4727±.03	0.5587±.03	0.5672±.03	0.4677±.04	0.7875±.01	0.3823±.02	0.5985±.02	0.8757±.02	0.5556±.05
IMP (%)		11.23%	5.23%	14.63%	13.81%	4.26%	16.59%	5.02%	0.99%	25.32%	13.79%	6.28%	30.70%	10.31%	2.30%	12.18%

Zhao, Haihong, et al. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. SIGKDD 2024.

Q&A

Part C Multi-modal Graph Pre-training with Large Language Models

Yuhan Li

Outline

- **Motivation**
- **Categorization of Combining LLM with Graph**
  - LLM as Enhancer
  - LLM as Predictor
  - LLM as Aligner
  - Others
- **Benchmarking GraphLLM**
- **Future Directions**

Motivation

- ## Large Language Models (LLMs)
  - Non-autoregressive.
    - Encoder-only LLMs.
    - Masked language modeling
  - Autoregressive.
    - Encoder-decoder LLMs.
    - Decoder-only LLMs.
    - Next token prediction
  - Applications:
    - NLP tasks -> machine translation, text classification.
    - Other modality tasks -> images, videos

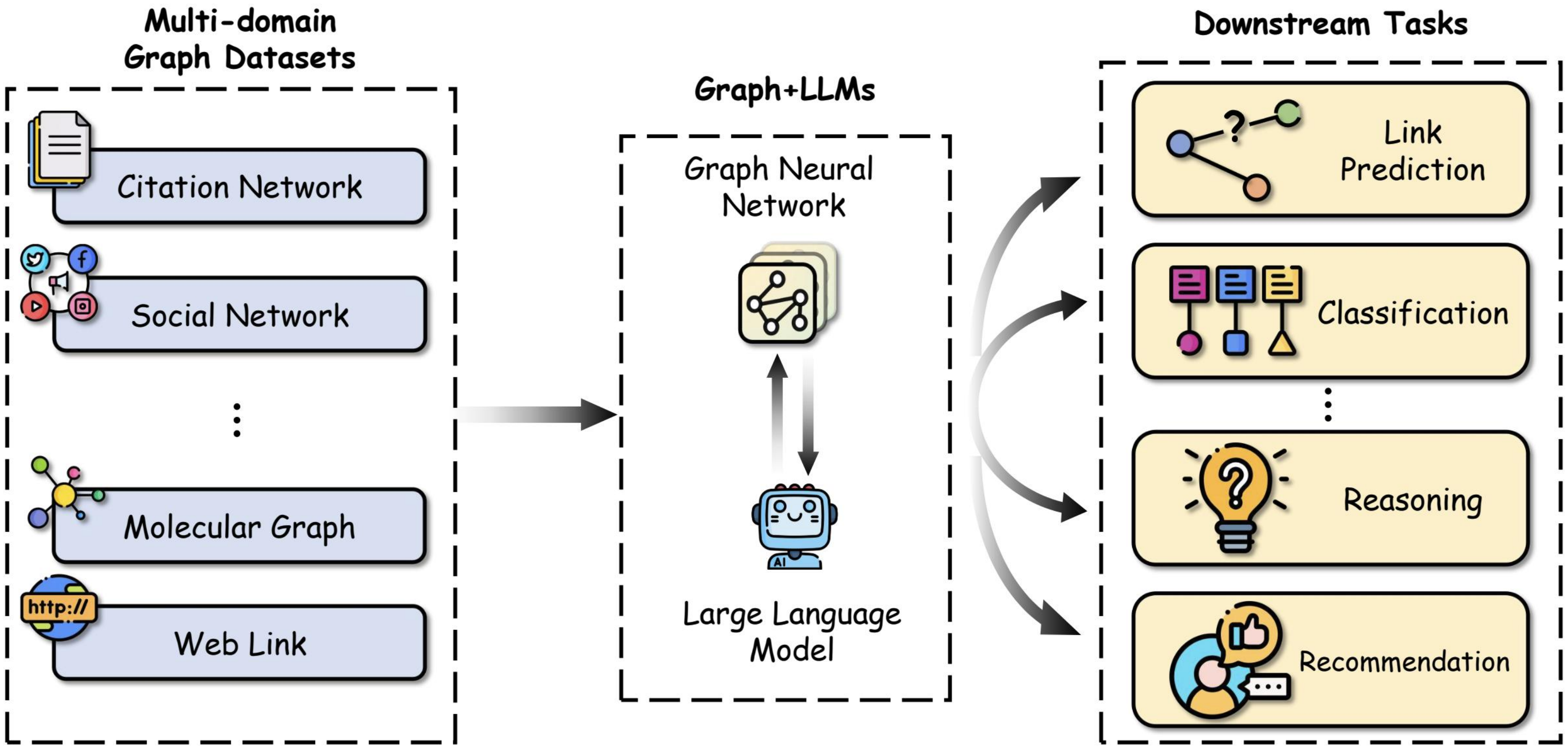
Motivation

- **Integrating LLMs with traditional GNNs can be mutually beneficial and enhance graph learning.**
  - GNNs -> constrained embeddings as node features
  - LLMs -> struggle to capture structural information
  - Combining GNNs with LLMs …

Li Y, Li Z, Wang P, et al. A Survey of Graph Meets Large Language Model: Progress and Future Directions. IJCAI 2024.

Motivation

➢ **The integration of GNNs and LLMs across a myriad of domains**



Different roles played by LLMs

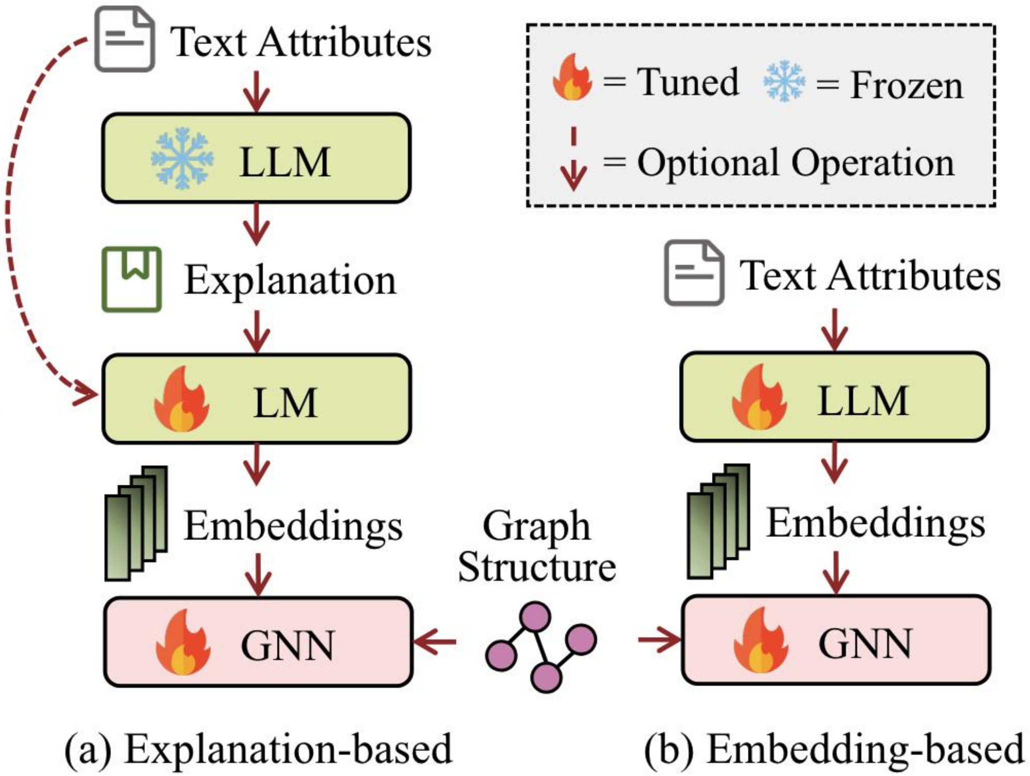
➢ LLM as enhancer

- Aiming to enhancing the quality of node embeddings with the help of powerful LLMs.
- Explanation-based enhancement.

$$\text{Enhancement:} \quad e_i = f_{\text{LLM}}(t_i, p), \ \mathbf{x}_i = f_{\text{LM}}(e_i, t_i),$$
$$\text{Graph Learning:} \quad \mathbf{H} = f_{\text{GNN}}(\mathbf{X}, \mathbf{A}),$$

- Embedding-based enhancement.

$$\text{Enhancement:} \quad \mathbf{x}_i = f_{\text{LLM}}(t_i),$$
$$\text{Graph Learning:} \quad \mathbf{H} = f_{\text{GNN}}(\mathbf{X}, \mathbf{A}).$$



Li Y, Li Z, Wang P, et al. A Survey of Graph Meets Large Language Model: Progress and Future Directions. IJCAI 2024.

LLM as enhancer | TAPE

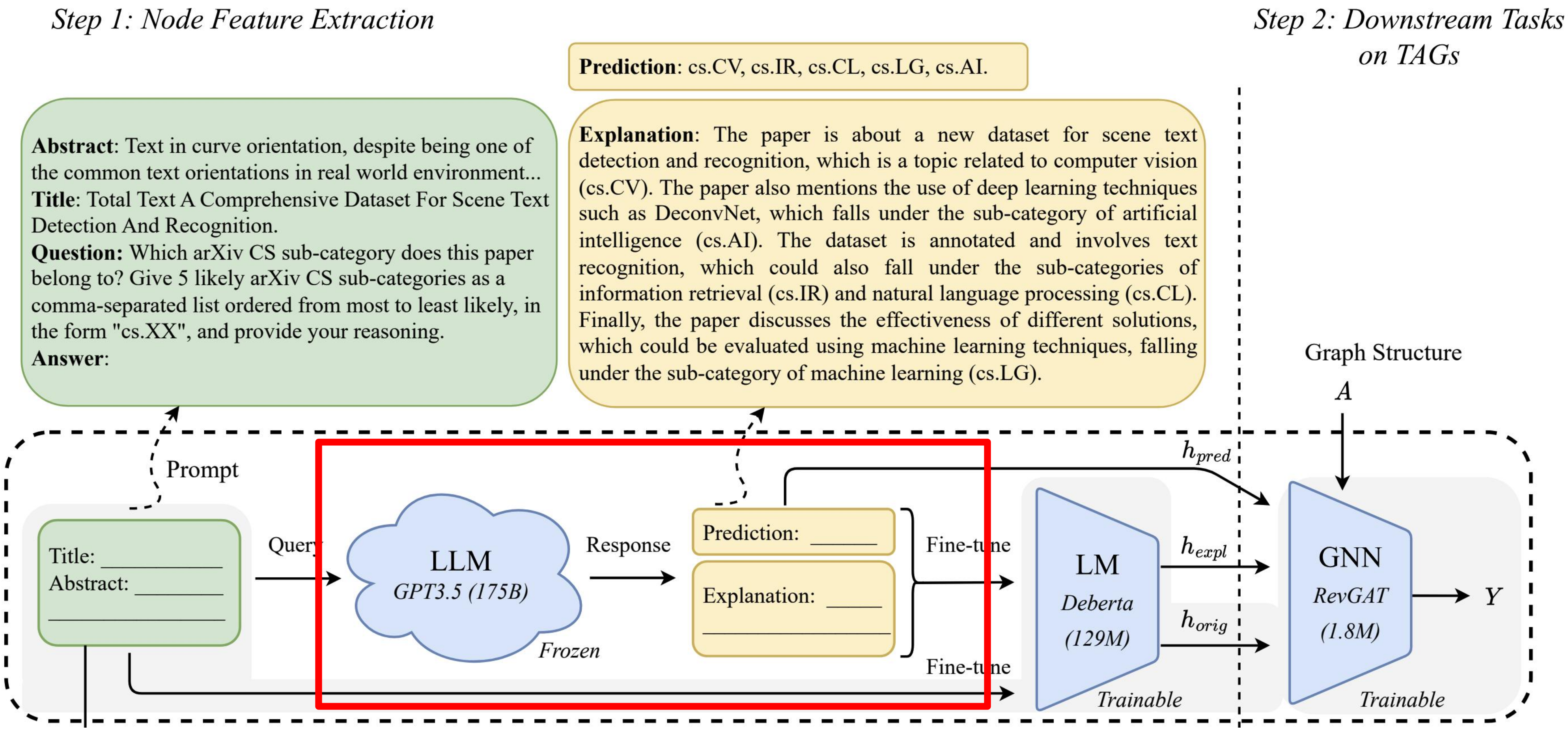
- ## Citation Networks
  - Text-rich graphs.
  - Each node represents a paper, and its corresponding textual description (e.g., title and abstract) is treated as the node's text attributes.
- ## Retrieval–Augmented
  - Leveraging LLMs to enhance more explanations for each node.

Abstract: [paper abstract]
Title: [paper title]
Question: [ask the model to predict one or more class labels of the paper, ordered from most to least likely, and provide explanations for its predictions]
Answer:

He X, Bresson X, Laurent T, et al. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. ICLR 2024.

LLM as enhancer | TAPE

➢ Pipeline



He X, Bresson X, Laurent T, et al. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. ICLR 2024.

LLM as enhancer | TAPE

Experimental results

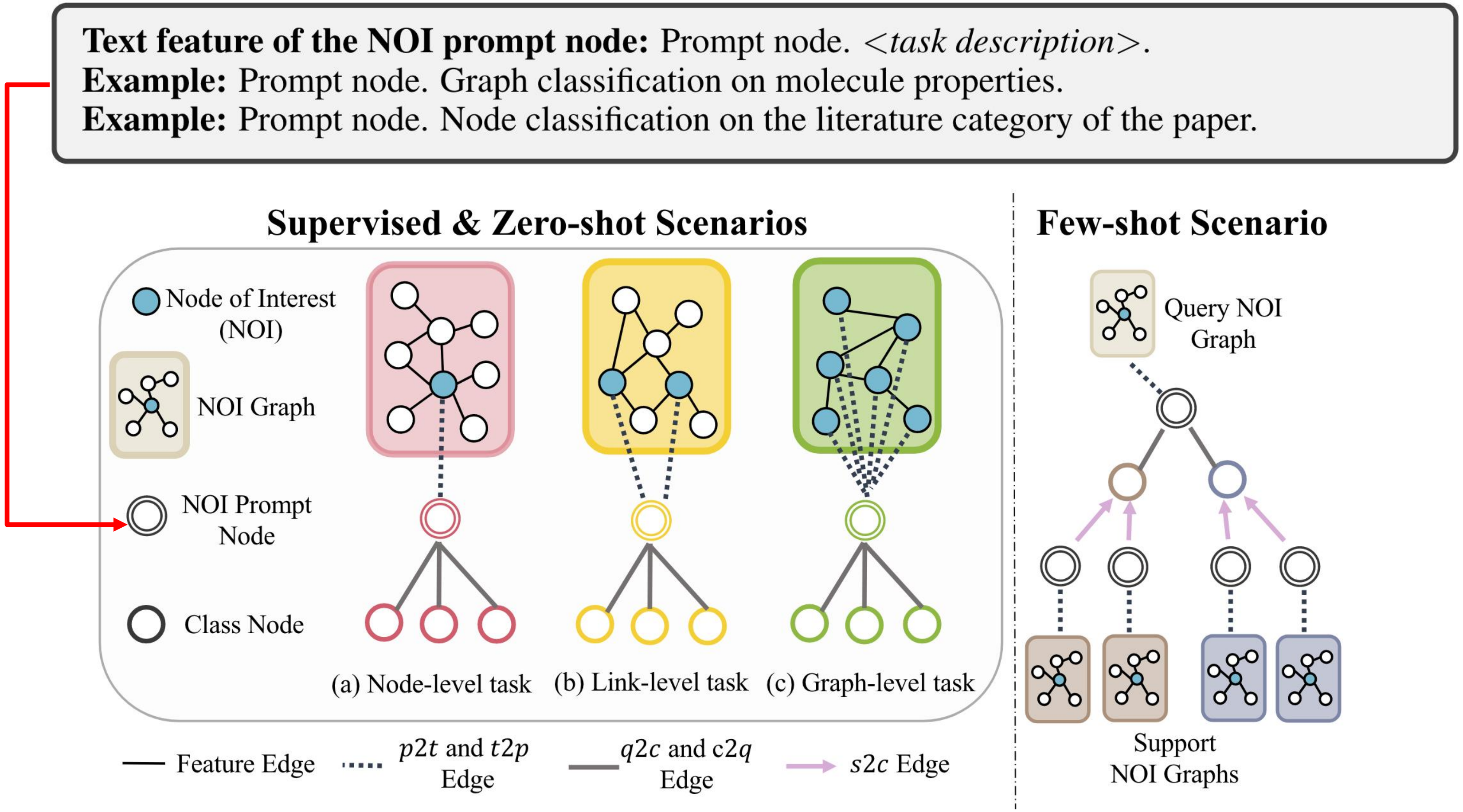
Dataset	Method	GNN			LM			Ours
		h_{shallow}	h_{GIANT}	$G \uparrow$	LLM	$\text{LM}_{\text{finetune}}$	$L \uparrow$	h_{TAPE}
Cora	MLP	0.6388 ± 0.0213	0.7196 ± 0.0000	37.41%	0.6769	0.7606 ± 0.0378	13.35%	0.8778 ± 0.0485
	GCN	0.8911 ± 0.0015	0.8423 ± 0.0053	2.33%	0.6769	0.7606 ± 0.0378	16.59%	0.9119 ± 0.0158
	SAGE	0.8824 ± 0.0009	0.8455 ± 0.0028	5.28%	0.6769	0.7606 ± 0.0378	18.13%	**0.9290 ± 0.0307**
	RevGAT	0.8911 ± 0.0000	0.8353 ± 0.0038	4.14%	0.6769	0.7606 ± 0.0378	18.04%	0.9280 ± 0.0275
PubMed	MLP	0.8635 ± 0.0032	0.8175 ± 0.0059	10.77%	0.9342	0.9494 ± 0.0046	0.75%	0.9565 ± 0.0060
	GCN	0.8031 ± 0.0425	0.8419 ± 0.0050	17.43%	0.9342	0.9494 ± 0.0046	-0.66%	0.9431 ± 0.0043
	SAGE	0.8881 ± 0.0002	0.8372 ± 0.0082	8.30%	0.9342	0.9494 ± 0.0046	1.31%	**0.9618 ± 0.0053**
	RevGAT	0.8850 ± 0.0005	0.8502 ± 0.0048	8.52%	0.9342	0.9494 ± 0.0046	1.15%	0.9604 ± 0.0047
ogbn-arxiv	MLP	0.5336 ± 0.0038	0.7308 ± 0.0006	42.19%	0.7350	0.7361 ± 0.0004	3.07%	0.7587 ± 0.0015
	GCN	0.7182 ± 0.0027	0.7329 ± 0.0010	4.71%	0.7350	0.7361 ± 0.0004	2.16%	0.7520 ± 0.0003
	SAGE	0.7171 ± 0.0017	0.7435 ± 0.0014	6.98%	0.7350	0.7361 ± 0.0004	4.22%	0.7672 ± 0.0007
	RevGAT	0.7083 ± 0.0017	0.7590 ± 0.0019	9.42%	0.7350	0.7361 ± 0.0004	5.28%	**0.7750 ± 0.0012**
ogbn-products	MLP	0.5385 ± 0.0017	0.6125 ± 0.0078	46.3%	0.7440	0.7297 ± 0.0023	7.96%	0.7878 ± 0.0082
	GCN	0.7052 ± 0.0051	0.6977 ± 0.0042	13.39%	0.7440	0.7297 ± 0.0023	9.58%	0.7996 ± 0.0041
	SAGE	0.6913 ± 0.0026	0.6869 ± 0.0119	17.71%	0.7440	0.7297 ± 0.0023	11.51%	0.8137 ± 0.0043
	RevGAT	0.6964 ± 0.0017	0.7189 ± 0.0030	18.24%	0.7440	0.7297 ± 0.0023	12.84%	**0.8234 ± 0.0036**
tape-arxiv23	MLP	0.6202 ± 0.0064	0.5574 ± 0.0032	35.20%	0.7356	0.7358 ± 0.0006	12.25%	0.8385 ± 0.0246
	GCN	0.6341 ± 0.0062	0.5672 ± 0.0061	27.42%	0.7356	0.7358 ± 0.0006	8.94%	0.8080 ± 0.0215
	SAGE	0.6430 ± 0.0037	0.5665 ± 0.0032	30.45%	0.7356	0.7358 ± 0.0006	12.28%	0.8388 ± 0.0264
	RevGAT	0.6563 ± 0.0062	0.5834 ± 0.0038	28.34%	0.7356	0.7358 ± 0.0006	12.64%	**0.8423 ± 0.0256**

He X, Bresson X, Laurent T, et al. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. ICLR 2024.

LLM as enhancer | OFA

➢ NOI (Node of Interest) prompt node

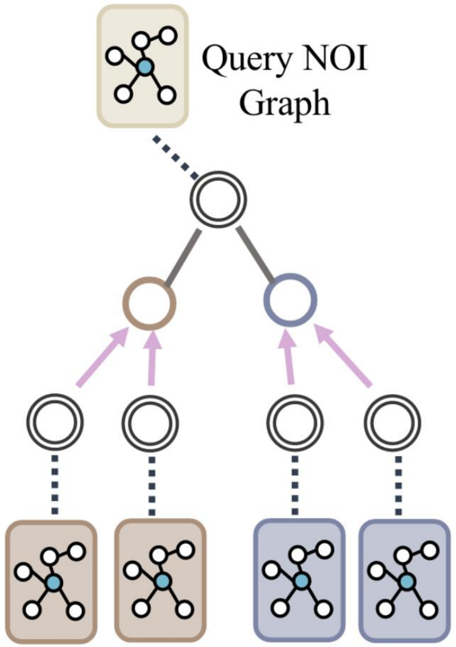
- Associated with a task prompt text, encoded by an LLM.



Liu H, Feng J, Kong L, et al. One for All: Towards Training One Graph Model for All Classification Tasks. ICLR 2024.

LLM as enhancer | OFA

- **Few/Zero-shot Ability**
  - "In-context Learning": It utilizes few-shot support examples by connecting the support NOI prompt nodes to the corresponding class nodes to provide exemplary information.



# Way	ogbn-arxiv-5-way (Transductive)				Cora-2-way (Transfer)		
Task	5-shot	3-shot	1-shot	0-shot	5-shot	1-shot	0-shot
GPN	50.53±3.07	48.32±3.80	38.58±1.61	-	63.83±2.86	56.09±2.08	-
TENT	60.83±7.45	56.03±8.90	45.62±10.70	-	58.97±2.40	54.33±2.10	-
GLITTER	56.00±4.40	57.44±4.90	47.12±2.73	-	-	-	-
TLP-BGRL	50.13±8.78	46.21±7.92	35.81±8.58	-	81.31±1.89	59.16±2.48	-
TLP-SURGL	77.89±6.46	74.19±7.55	61.75±10.07	-	92.49±1.02	81.52±2.09	-
Prodigy	61.09±5.85	58.64±5.84	48.23±6.18	-	-	-	
OFA-joint-lr	61.45±2.56	59.78±2.51	50.20±4.27	46.19±3.83	76.10±4.41	67.44±4.47	56.92±3.09

Liu H, Feng J, Kong L, et al. One for All: Towards Training One Graph Model for All Classification Tasks. ICLR 2024.

LLM as enhancer | ZeroG

Zero–shot Transferability in Graphs

- This trend of zero-shot capabilities in machine learning, particularly after the advent of foundation models such as LLMs, has demonstrated considerable advancements in the field of AI.
- NLP field: **generative** paradigm, such as LLaMA, GPT-series, …
- CV field: **retrieval** paradigm, such as CLIP, …
- In Graph field, zero-shot transfer is also important since:
  - 1. The emergence of new graphs.
  - 2. The difficulty of human labeling.
- We focus on cross-dataset zero-shot transferability in graphs.

Li Y, Wang P, Li Z, et al. ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs. SIGKDD 2024.

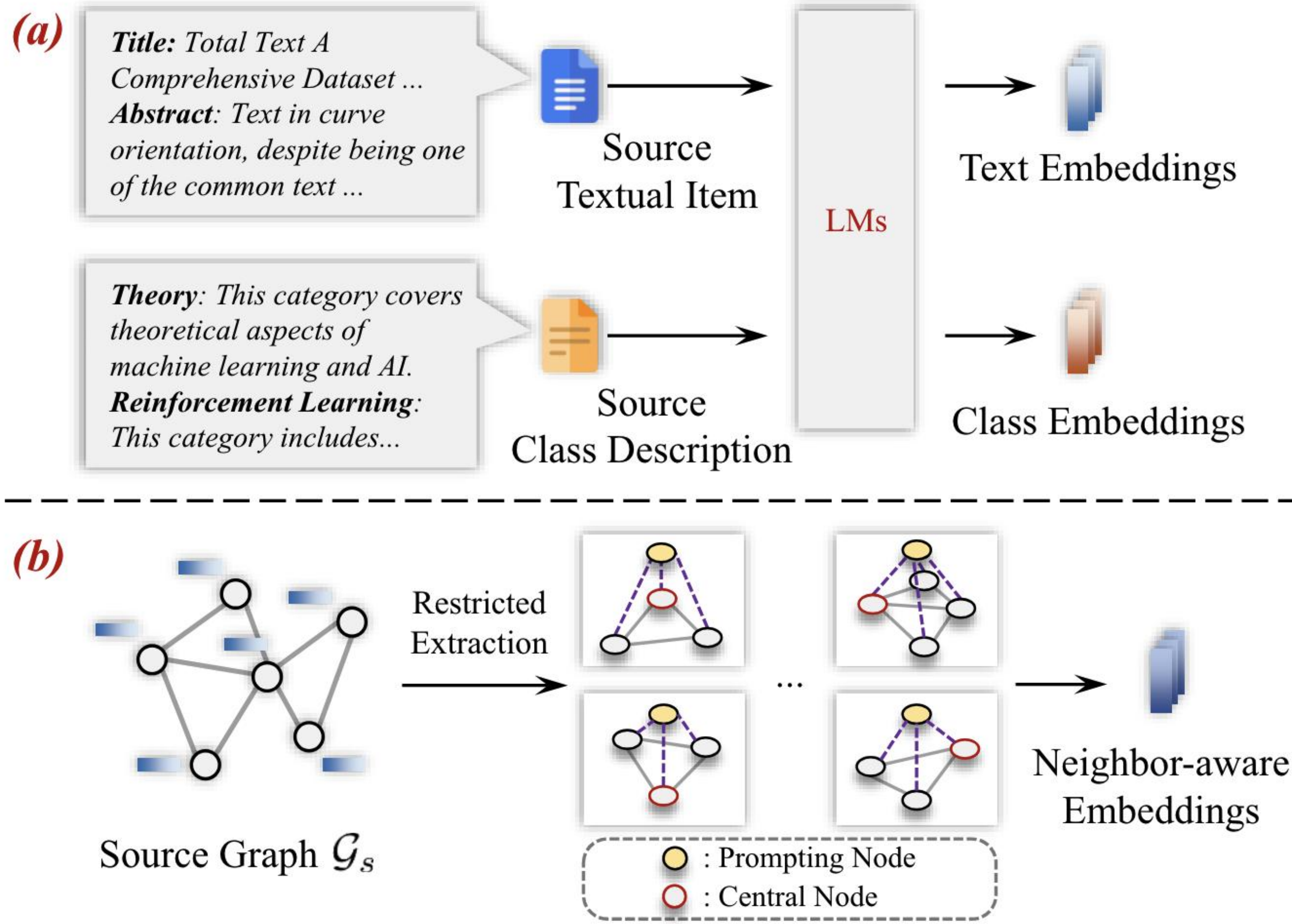
LLM as enhancer | ZeroG

- **Dimension Misalignment**
  - Shallow embedding: bag-of-words, skip-gram, TF-IDF, …
- **Mismatched Label Spaces**
  - GNN's classification head is fixed to the number of classes during pre-training.
- **Negative Transfer**
  - Fully adapting graph models to source graphs often causes overfitting.

Li Y, Wang P, Li Z, et al. ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs. SIGKDD 2024.

LLM as enhancer | ZeroG

- **Step1: Unified Graph Representation**
  - Use a unified pre-trained LLM to encode both node attributes and descriptions associated with classes.
  - A unified semantic space.
- **Step2: Prompt–based Subgraph Sampling**
  - Retricted Extraction
  - Prompting Node
  - Neighborhood Aggregation



Li Y, Wang P, Li Z, et al. ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs. SIGKDD 2024.

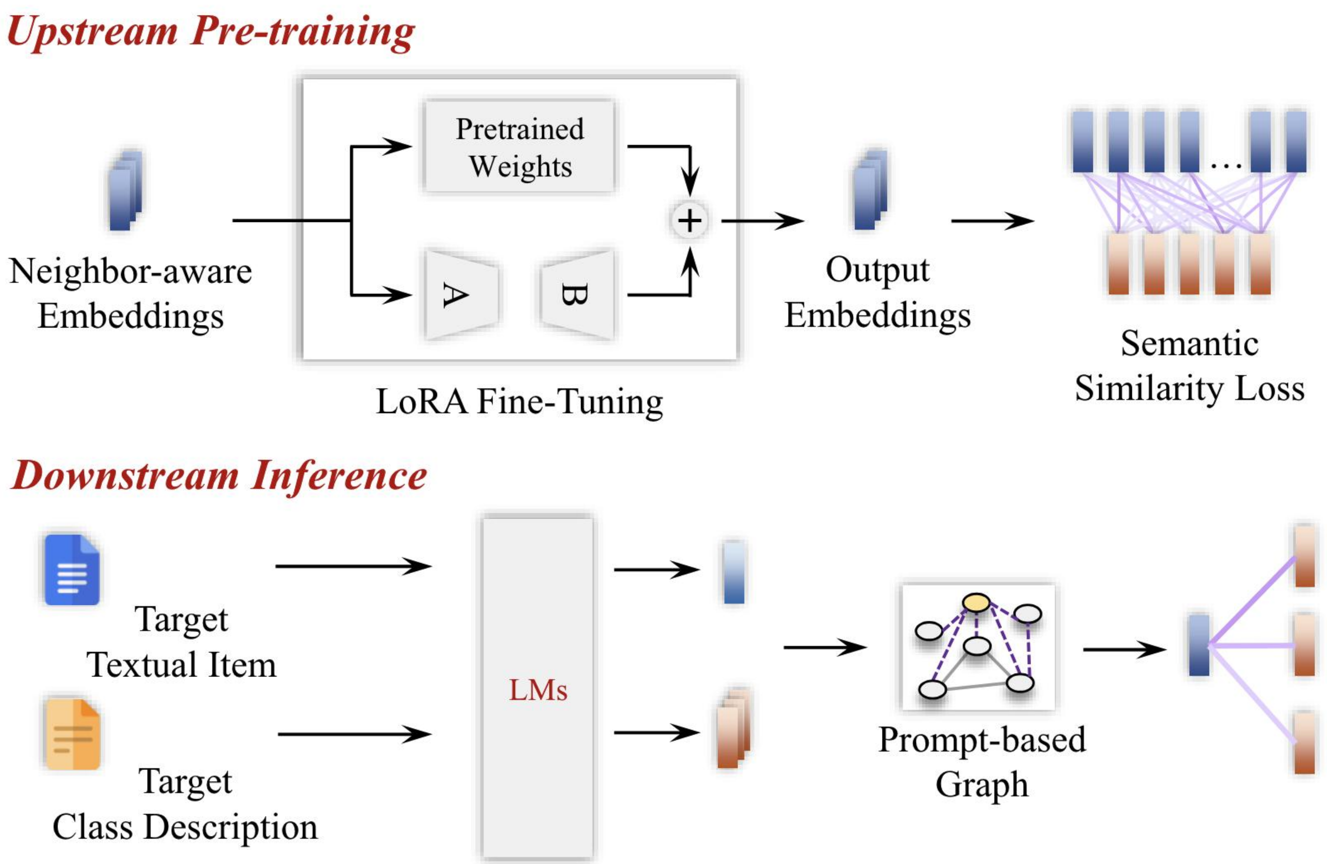
LLM as enhancer | ZeroG

- **Step3: Upstream Pre–training**
  - PEFT strategy: LoRA.
  - Cross-entropy loss

$$\mathcal{L}_{\text{pre}}(\Theta) = -\sum_{s \in \mathcal{T}_{\text{pre}}} \sum_{n \in N_s} \log \frac{\exp(\text{sim}(\mathbf{h}_n, \mathbf{h}_{y_n}))}{\sum_{c \in Y_s} \exp(\text{sim}(\mathbf{h}_n, \mathbf{h}_c))}$$

- **Downstream Inference**
  - The class that yields the highest similarity score is predicted to be the class of the node.

$$y' = \text{argmax}_i(\text{sim}(\mathbf{h}_n, \mathbf{h}_{c_i}) \mid i \in \{1, \dots, N\})$$



Li Y, Wang P, Li Z, et al. ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs. SIGKDD 2024.

LLM as enhancer | ZeroG

In-domain Transferability

Methods	$\mathcal{A}$	$\mathcal{S}$	Cora	Pubmed	Citeseer	P-Home	P-Tech
zero-shot settings							
DGI [56]	✓	✗	19.97	43.89	21.12	33.06	55.83
GraphCL [63]	✓	✗	26.22	43.73	20.59	37.44	62.63
GraphMAE [19]	✓	✗	34.79	48.23	34.62	37.04	73.37
BERT [12]	✗	✓	19.90	34.79	23.76	37.32	56.44
RoBERTa [34]	✗	✓	28.91	27.33	30.95	35.50	66.31
E5 [58]	✗	✓	39.70	41.93	45.89	57.56	59.17
Sent-BERT [40]	✗	✓	52.25	41.71	47.52	63.22	67.21
OFA [31]	✓	✓	27.07	37.87	37.92	32.86	71.03
ZEROG (ours)	✓	✓	**68.72**	**78.02**	**64.94**	**73.20**	**82.96**
semi-supervised settings							
GCN* [25]	-	-	81.50	79.00	70.30	73.85	93.28
GAT* [55]	-	-	83.00	79.00	72.50	73.46	88.89

Cross-domain Transferability

Test	Pre-training	OFA	In-D	ZEROG
Wiki-CS	Arxiv ∪ Cora ∪ Pubmed ∪ Citeseer	48.42	-	53.28
Wiki-CS	P-Home ∪ P-Tech	21.09	-	60.97
Cora	P-Home ∪ P-Tech	18.57	68.72	67.65(-1.07%)
Pubmed	P-Home ∪ P-Tech	31.89	78.02	69.12(-8.90%)
Citeseer	P-Home ∪ P-Tech	20.78	64.94	53.17(-11.77%)
P-Home	Arxiv ∪ Cora ∪ Pubmed ∪ Citeseer	35.73	73.20	71.45(-1.75%)
P-Tech	Arxiv ∪ Cora ∪ Pubmed ∪ Citeseer	62.10	82.96	83.20(+0.24%)

Ablation Study

Methods	Cora	Pubmed	Citeseer	P-Home	P-Tech
ZEROG	68.72	78.02	64.94	73.20	82.96
- (w/o p)	68.25(-0.47%)	76.49(-1.53%)	61.64(-3.30%)	70.46(-2.74%)	79.68(-7.18%)
- (w/o NA)	43.31(-25.43%)	47.21(-30.81%)	48.68(-16.26%)	60.26(-12.94%)	58.91(-27.95%)
- (w/o norm)	54.43(-13.39%)	39.25(-38.77%)	34.84(-30.10%)	41.26(-31.94%)	72.17(-14.69%)
- (w/o LoRA)	17.36(-51.36%)	46.49(-31.35%)	23.98(-40.96%)	39.77(-33.43%)	87.22(+4.26%)

Li Y, Wang P, Li Z, et al. ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs. SIGKDD 2024.

Different roles played by LLMs

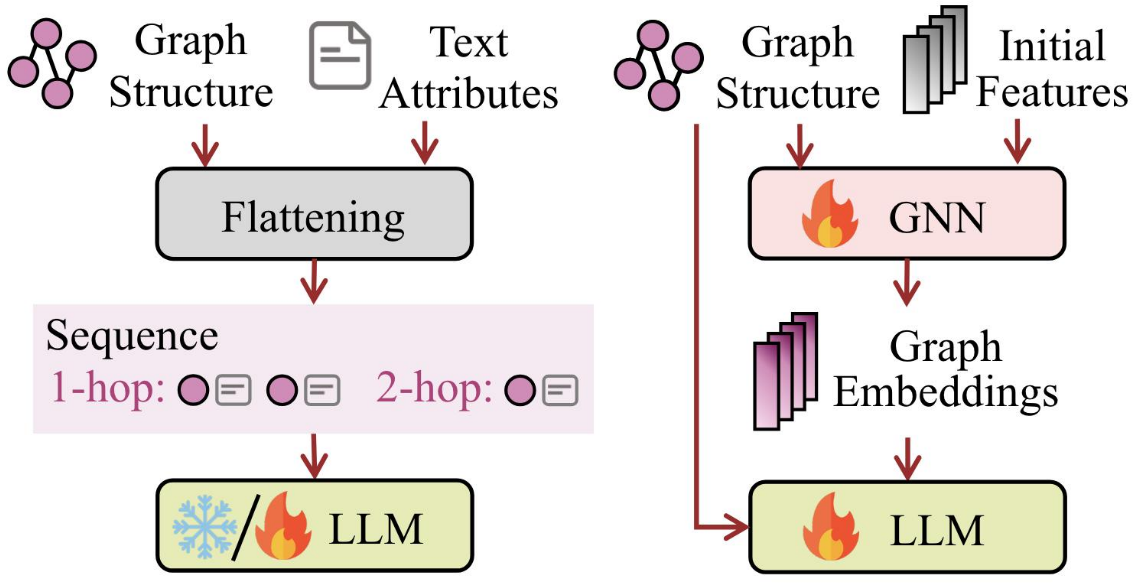
➢ LLM as predictor

- Aiming to utilize LLMs to make predictions for a wide range of graph-related tasks, within a unified generative paradigm.
- Flatten-based prediction.

$$\begin{aligned}\text{Graph Flattening:}\quad & G_{seq} = \texttt{Flat}(\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{J}),\\ \text{Prediction:}\quad & \tilde{Y} = \texttt{Parse}(f_{\text{LLM}}(G_{seq}, p)),\end{aligned}$$

- GNN-based prediction.

$$\begin{aligned}\text{Graph Learning:}\quad & \mathbf{H} = f_{\text{GNN}}(\mathbf{X}, \mathbf{A}),\\ \text{Prediction:}\quad & \tilde{Y} = \texttt{Parse}(f_{\text{LLM}}(\mathbf{H}, p)),\end{aligned}$$



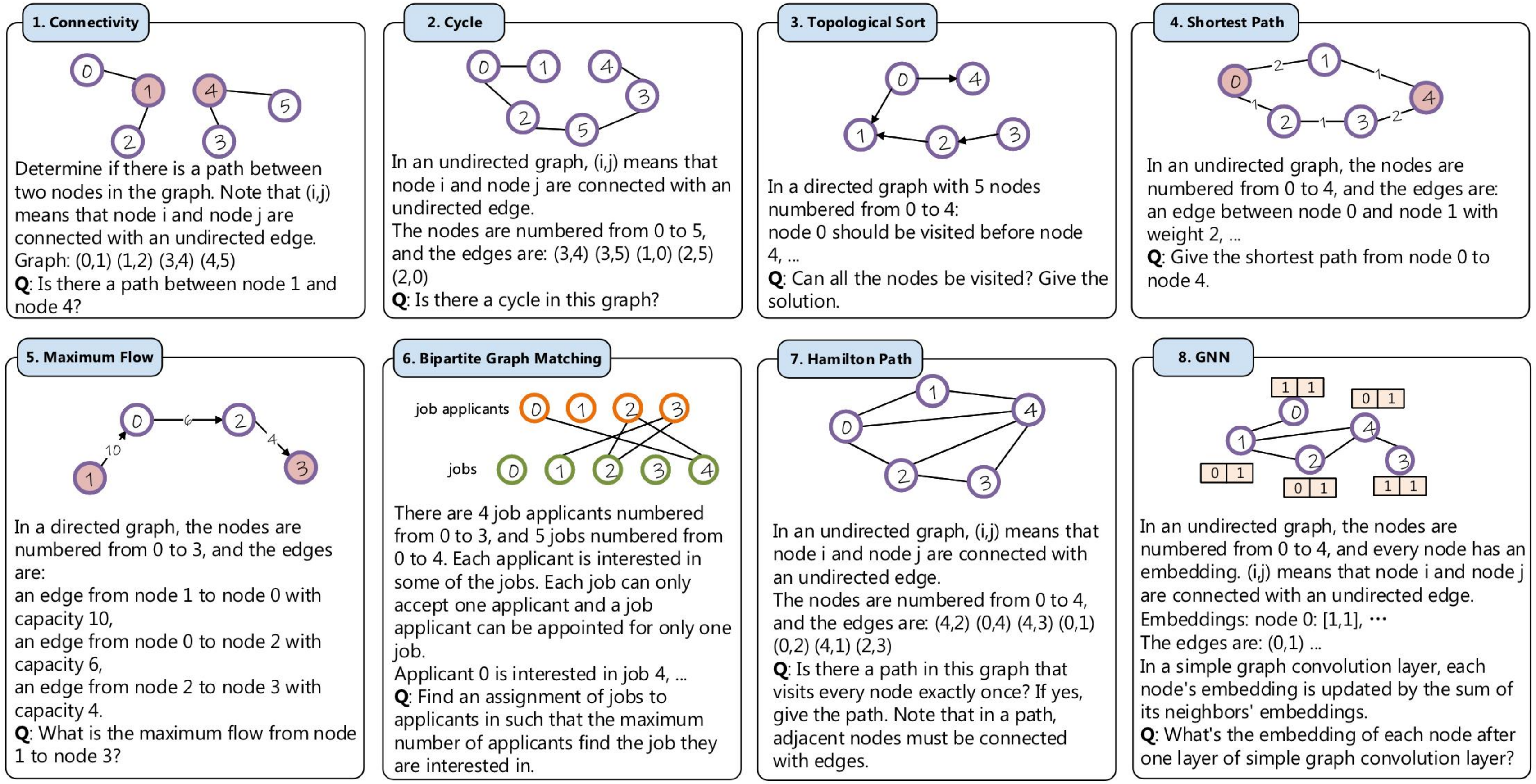
(a) Flatten-based (b) GNN-based

Li Y, Li Z, Wang P, et al. A Survey of Graph Meets Large Language Model: Progress and Future Directions. IJCAI 2024.

LLM as predictor | NLGraph

➢ NLGraph Benchmark

- ❑ Graph reasoning tasks, such as shortest path and cycle detection.



Wang H, Feng S, He T, et al. Can Language Models Solve Graph Problems in Natural Language? NeurIPS 2023 spotlight.

LLM as predictor | NLGraph

➢ Input & Output

- Shortest path, GPT-4:

In an undirected graph, the nodes are numbered from 0 to 5, and the edges are: An edge between node 2 and node 0 with weight 2, An edge between node 0 and node 4 with weight 1, An edge between node 1 and node 2 with weight 4, An edge between node 1 and node 3 with weight 4, An edge between node 4 and node 1 with weight 4, An edge between node 2 and node 3 with weight 1, An edge between node 2 and node 5 with weight 2, An edge between node 2 and node 4 with weight 3, An edge between node 5 and node 3 with weight 2, An edge between node 3 and node 4 with weight 3, An edge between node 4 and node 5 with weight 2.
Q: Give the shortest path from node 3 to node 0.

=>

All the paths from node 3 to node 0 are: 3,2,0 with a total weight of 1 + 2 = 3, 3,1,2,0 with a total weight of 4 + 4 + 2 = 10, 3,4,0 with a total weight of 3 + 1 = 4, 3,4,2,0 with a total weight of 3 + 3 + 2 = 8, 3,5,2,0 with a total weight of 2 + 2 + 2 = 6, 3,5,4,0 with a total weight of 2 + 2 + 1 = 5.
The weight of path 3,2,0 is the smallest, so the shortest path from node 3 to node 0 is 3,2,0 with a total weight of 3. ✔

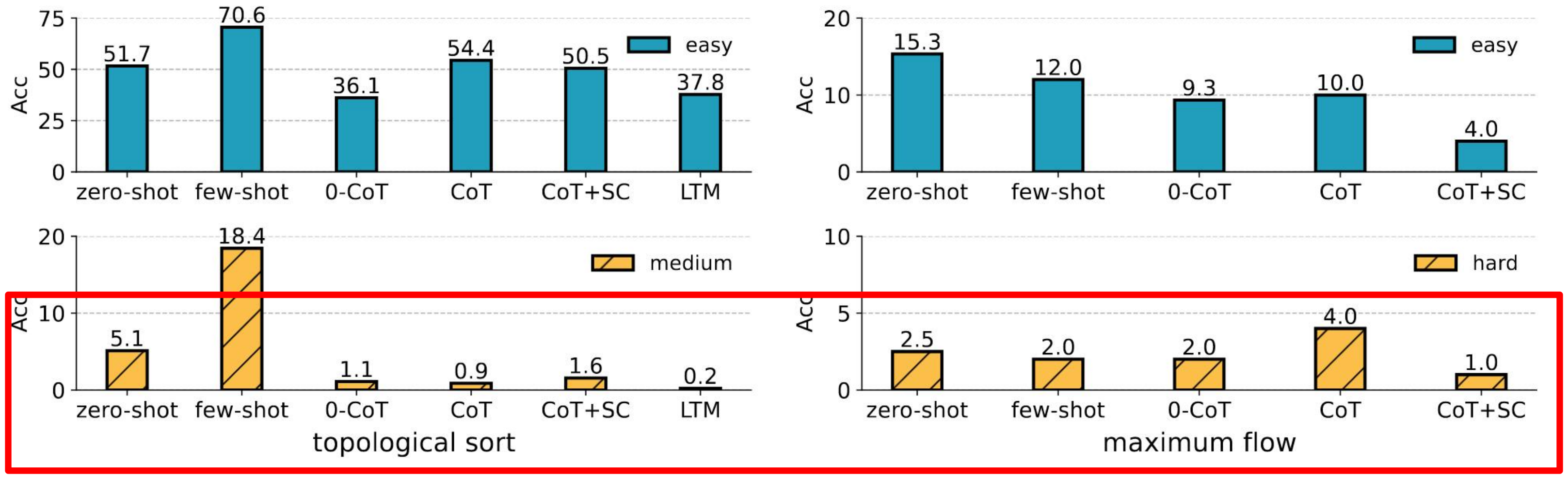
Wang H, Feng S, He T, et al. Can Language Models Solve Graph Problems in Natural Language? NeurIPS 2023 spotlight.

LLM as predictor | NLGraph

➢ Insights

- LLMs have (Preliminary) graph reasoning abilities.

Method	Connectivity				Cycle				Shortest Path				
	Easy	Medium	Hard	Avg.	Easy	Medium	Hard	Avg.	Easy	Hard	Easy (PC)	Hard (PC)	Avg.
RANDOM	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	6.07	6.69	14.73	13.81	17.81
ZERO-SHOT	83.81	72.75	63.38	71.31	50.00	50.00	50.00	50.00	29.40	21.00	46.00	26.76	30.79
FEW-SHOT	93.75	83.83	76.61	84.73	80.00	**70.00**	**61.00**	**70.33**	31.11	26.00	49.19	35.73	35.51
CoT	**94.32**	82.17	77.21	84.57	**84.67**	63.33	53.25	66.75	63.89	**29.50**	76.84	35.79	51.51
0-CoT	79.55	65.83	68.53	71.30	55.33	57.67	49.00	54.00	8.89	7.50	62.39	**43.95**	32.03
CoT+SC	93.18	**84.50**	**82.79**	**86.82**	82.00	63.67	53.50	66.39	**68.89**	29.00	**80.25**	38.47	**54.15**

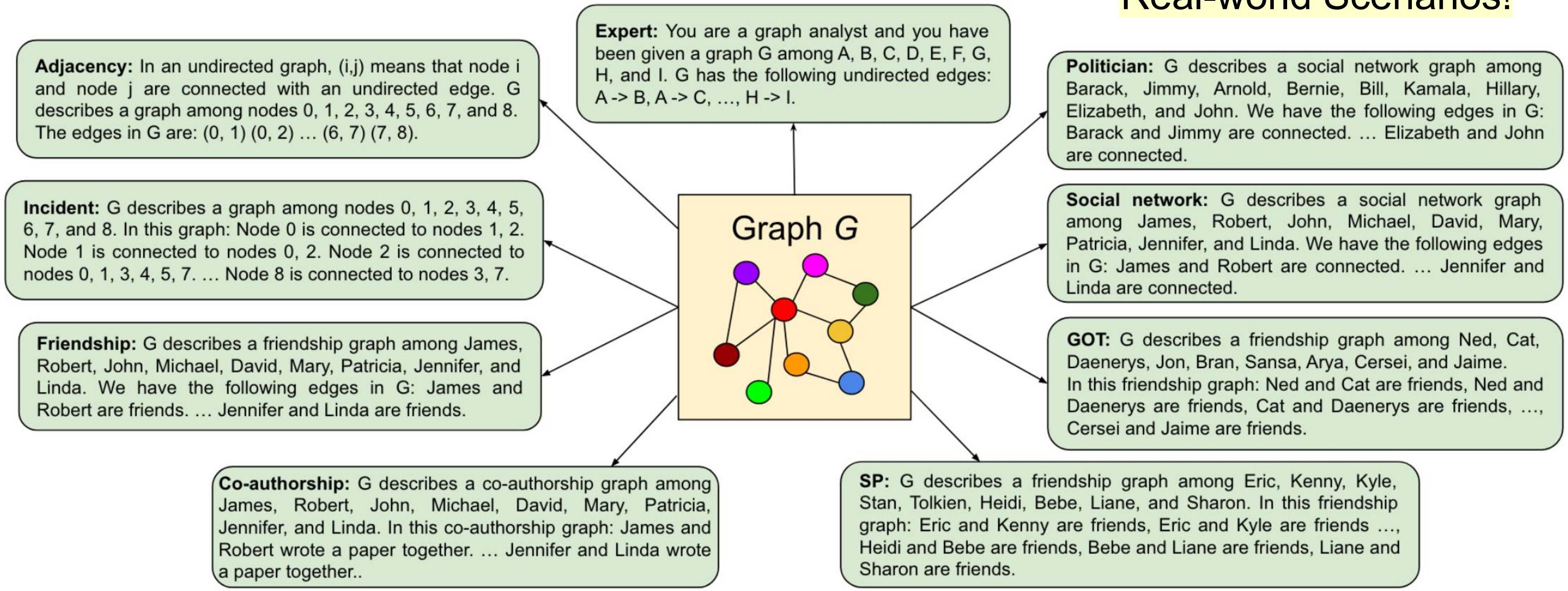


Wang H, Feng S, He T, et al. Can Language Models Solve Graph Problems in Natural Language? NeurIPS 2023 spotlight.

LLM as predictor | Talk like a graph

- ## GraphQA Benchmark
  - Graph reasoning tasks, similar to NLGraph.
- ## Encoding Strategies

Real-world Scenarios!



Fatemi B, Halcrow J, Perozzi B. Talk like a Graph: Encoding Graphs for Large Language Models? ICLR 2024.

LLM as predictor | Talk like a graph

➢ Insights

- Graph encoding functions have significant impact on LLM reasoning.

Method	Encoding	Edge Existence	Node degree	Node count	Edge count	Connected nodes	Cycle check
	Overall (μ/δ)	44.5 / 9.4	14.0/16.0	21.73 / 8.6	12.4 / 4.8	14.7 / 11.0	76.0 / 13.2
ZERO-SHOT	Adjacency	45.8	12.4	18.8	14.0	19.8	71.6
	Incident	39.6	25.0	15.6	10.6	53.8	68.8
	Co-authorship	44.0	13.8	22.0	11.4	7.6	70.8
	Friendship	46.6	11.2	23.0	10.2	4.0	**82.0**
	SP	46.4	9.0	22.4	15.0	6.2	80.4
	GOT	**49.0**	13.6	22.8	13.2	7.6	79.0
	Social network	43.2	16.0	22.8	10.8	8.2	81.2
	Politician	44.6	15.2	24.2	11.6	8.8	81.0
	Expert	41.2	10.0	24.0	14.8	16.4	69.6

- As a result, it becomes important to translate a given task into more contextually meaningful textual information when employing LLMs for inference.

Fatemi B, Halcrow J, Perozzi B. Talk like a Graph: Encoding Graphs for Large Language Models? ICLR 2024.

LLM as predictor | GraphWiz

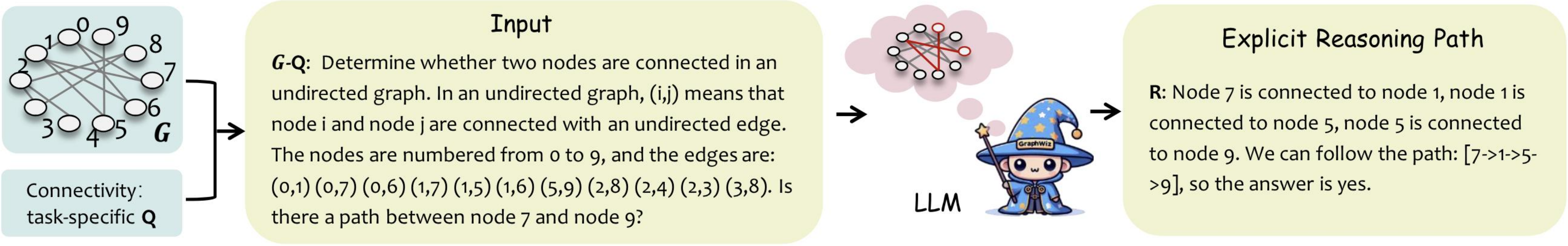
- **Motivation 1: Graph Size Limitation**
  - Support more nodes and edges
- **Motivation 2: Solve Graph Problems Explicitly**
  - CoT ability => Explicit Reasoning Path.
- **Motivation 3: Training, not only Inference**
  - Existing works only focus on inference close-sourced LLMs.
  - Can we train our own LLMs for graph reasoning?

Chen N, Li Y, Tang J, et al. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. SIGKDD 2024.

LLM as predictor | GraphWiz

Goal

We aim at leveraging instruction-tuning to build a powerful instruction-following LLM that can map textural descriptions of graphs and structures, and then solve different graph problems explicitly in natural language



➢ What we do

- Tackle the data challenge: GraphInstruct (G-Q-R).
- Explore training strategies: Mix-tasked Instruction Tuning and DPO.
- In-Depth Analysis: Data Amount, Transferability, GraphWiz limit, etc.

Chen N, Li Y, Tang J, et al. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. SIGKDD 2024.

LLM as predictor | GraphWiz

➢ GraphInstruct–Tasks

Strategy

1. **Diverse Distributions**: Node range and edge density
2. **Length Constraints**: No more than 4K
3. **Unique Instances**
4. **Scalable Graph Sizes**.

Initial **27k** graph problem (G-Q)

Problem	Definition	Time Complexity	Weighted?	Directed?	Node Range	Difficulty
Cycle Detection	Detect if a given graph $\mathcal{G}$ contains any cycles.	$O(\|\mathcal{V}\| + \|\mathcal{E}\|)$	✗	✗	[2, 100]	Easy
Connectivity	Assess if two nodes u and v in a given graph $\mathcal{G}$ are connected via a path.	$O(\|\mathcal{V}\| + \|\mathcal{E}\|)$	✗	✗	[2, 100]	Easy
Bipartite Graph Check	Judge if a given graph $\mathcal{G}$ is bipartite.	$O(\|\mathcal{V}\| + \|\mathcal{E}\|)$	✗	✓	[2, 100]	Easy
Topological Sort	Find a topological ordering of vertices in a directed acyclic graph $\mathcal{G}$.	$O(\|\mathcal{V}\| + \|\mathcal{E}\|)$	✗	✓	[2, 50]	Easy
Shortest Path	Compute the shortest path between two specific nodes u and v in a given graph $\mathcal{G}$.	$O(\|\mathcal{E}\| + \|\mathcal{V}\| \log \|\mathcal{V}\|)$	✓	✗	[2, 100]	Medium
Maximum Triangle Sum	Find the maximum sum of weights for any connected triplet of vertices in a given graph $\mathcal{G}$.	$O(\|\mathcal{V}\|^3)$	✓	✗	[2, 25]	Medium
Maximum Flow	Calculate the maximum flow from a source node s to a sink node t in a directed graph $\mathcal{G}$.	$O(\|\mathcal{V}\|^2 \sqrt{\|\mathcal{E}\|})$	✓	✓	[2, 50]	Medium
Hamilton Path	Determine if a given graph $\mathcal{G}$ has a Hamiltonian path that visits each vertex exactly once.	NP-Complete	✗	✗	[2, 50]	Hard
Subgraph Matching	Verify if there exists a subgraph in $\mathcal{G}$ that is isomorphic to a given graph $\mathcal{G}'$.	NP-Complete	✗	✓	[2, 30]	Hard

Chen N, Li Y, Tang J, et al. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. SIGKDD 2024.

LLM as predictor | GraphWiz

➢ GraphInstruct–Statistics

Datasets	Include Training Set?	Include CoTs?	Tasks	Node Scale	Edge Scale
NLGGraph	No	No	8	9-35	10-30
GraphQA	No	No	12	5-20	1-100
GraphInstruct	**Yes**	**Yes**	**9**	**2-100**	**5-500**

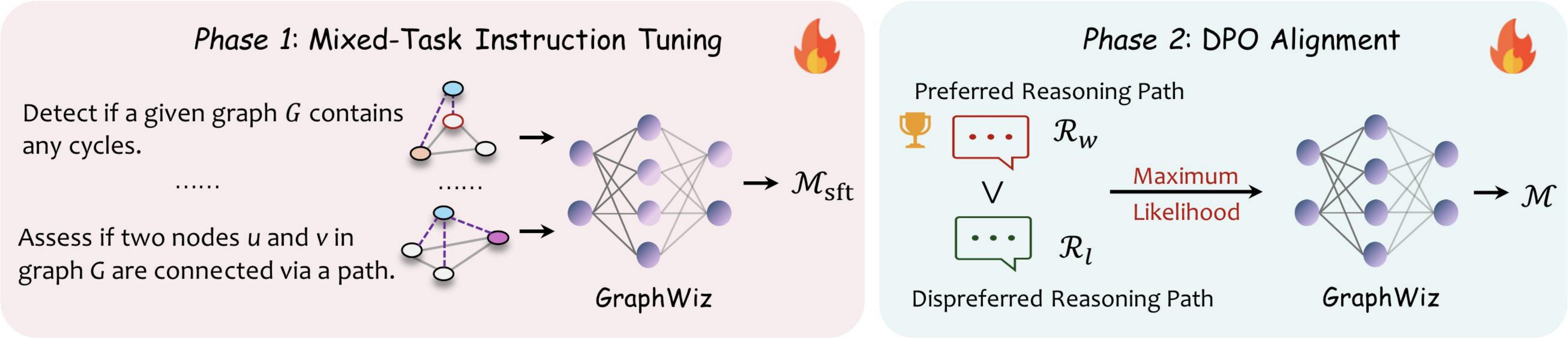
Tasks		*Easy*				*Medium*			*Hard*		Sum.
		cycle	connect	bipartite	topology	shortest	triangle	flow	hamilton	Subgraph	
Train	Total $\mathcal{G}$-$\mathcal{Q}$	3,717	2,687	2,013	902	1,392	2,756	405	2,097	1,435	17,158
	Total $\mathcal{V}$	84,980	79,853	58,860	10,146	23,204	14,714	4,333	33,284	7,847	315,051
	Total $\mathcal{R}$	13,122	10,001	9,324	4,481	5,859	13,483	747	8,454	6,274	72,785
Test	Total $\mathcal{G}$-$\mathcal{Q}$	400	400	400	400	400	400	400	400	400	3,600
	Total $\mathcal{V}$	19,570	19,500	19,515	9,449	19,449	4,990	10,024	9,732	6,594	118,823

Chen N, Li Y, Tang J, et al. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. SIGKDD 2024.

LLM as predictor | GraphWiz

➢ Training

- Two-phases training.



Chen N, Li Y, Tang J, et al. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. SIGKDD 2024.

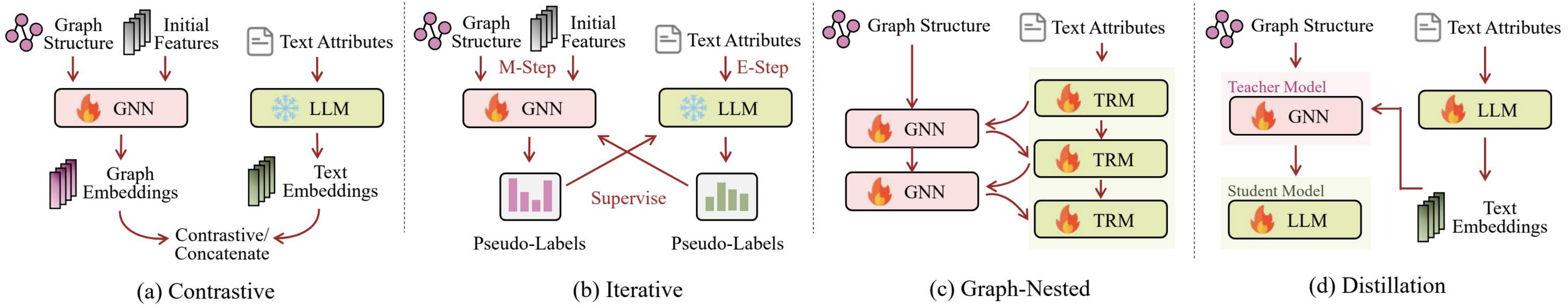
LLM as predictor | GraphWiz

Main Results of GraphWiz

Categories	Algorithms	*Easy*				*Medium*			*Hard*		Average
		cycle	connect	bipartite	topology	shortest	triangle	flow	hamilton	subgraph	
Closed-Source	GPT-4 (zero-shot)	38.75	17.00	65.25	5.00	9.25	5.75	3.25	59.25	45.50	27.67
	GPT-3.5 (2-shot)	51.25	43.75	70.75	4.50	3.50	17.25	8.50	54.25	43.00	32.97
	GPT-4 (2-shot)	52.50	62.75	74.25	25.25	18.25	31.00	7.75	75.75	46.75	43.81
Graph Neural Networks	GCN	84.00	74.00	82.00	-	5.75	6.75	9.25	-	68.00	-
	GIN	87.50	73.00	85.25	-	7.25	7.30	12.00	-	66.50	-
	GAT	87.50	79.25	85.25	-	7.25	7.50	12.50	-	66.25	-
Mistral-7B	Naive SFT	73.75	83.50	78.50	1.00	23.00	47.00	28.75	31.75	41.25	46.56
	GraphWiz	92.00	89.50	72.00	19.00	31.25	38.75	29.25	26.50	85.50	53.75
	GraphWiz-DPO	85.50	79.50	85.50	85.25	12.50	29.00	35.50	62.75	48.50	58.22
LLaMA 2-7B	Naive SFT	73.75	83.50	41.25	4.00	9.50	30.00	16.50	69.00	75.45	44.81
	GraphWiz	91.50	87.00	74.00	18.00	28.00	38.25	24.50	52.25	82.25	55.08
	GraphWiz-DPO	89.00	82.50	84.75	46.75	24.00	52.75	43.50	81.50	77.25	65.00
LLaMA 2-13B	Naive SFT	73.75	83.75	59.00	0.50	11.75	34.75	24.25	59.75	54.75	44.69
	GraphWiz	94.75	87.00	78.00	28.00	27.75	36.00	24.50	59.00	81.50	57.39
	GraphWiz-DPO	87.50	88.50	88.25	72.75	22.00	48.75	43.75	46.50	77.00	63.89

Chen N, Li Y, Tang J, et al. GraphWiz: An Instruction-Following Language Model for Graph Computational Problems. SIGKDD 2024.

Different roles played by LLMs



➢ LLM as aligner

- Aligning the embedding spaces of GNNs and LLMs is an effective way to integrate the graph modality with the text modality.
- Ensuring that each encoder's unique functionalities are preserved while coordinating their embedding spaces at a specific stage.

Li Y, Li Z, Wang P, et al. A Survey of Graph Meets Large Language Model: Progress and Future Directions. IJCAI 2024.

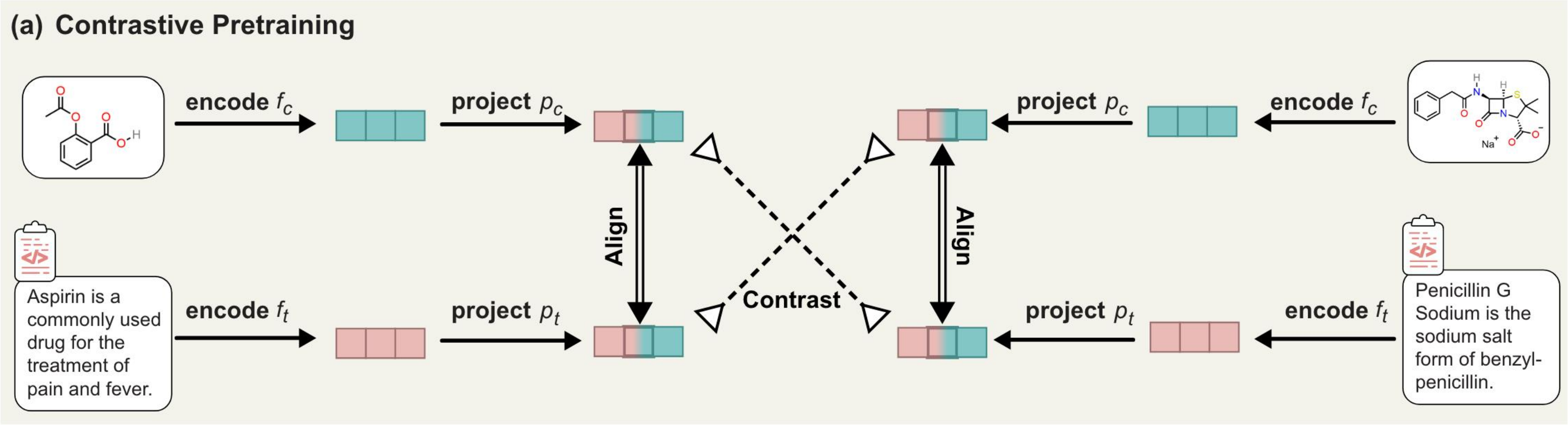
LLM as aligner | MoleculeSTM

Molecules

- Each molecule is corresponded with a description.

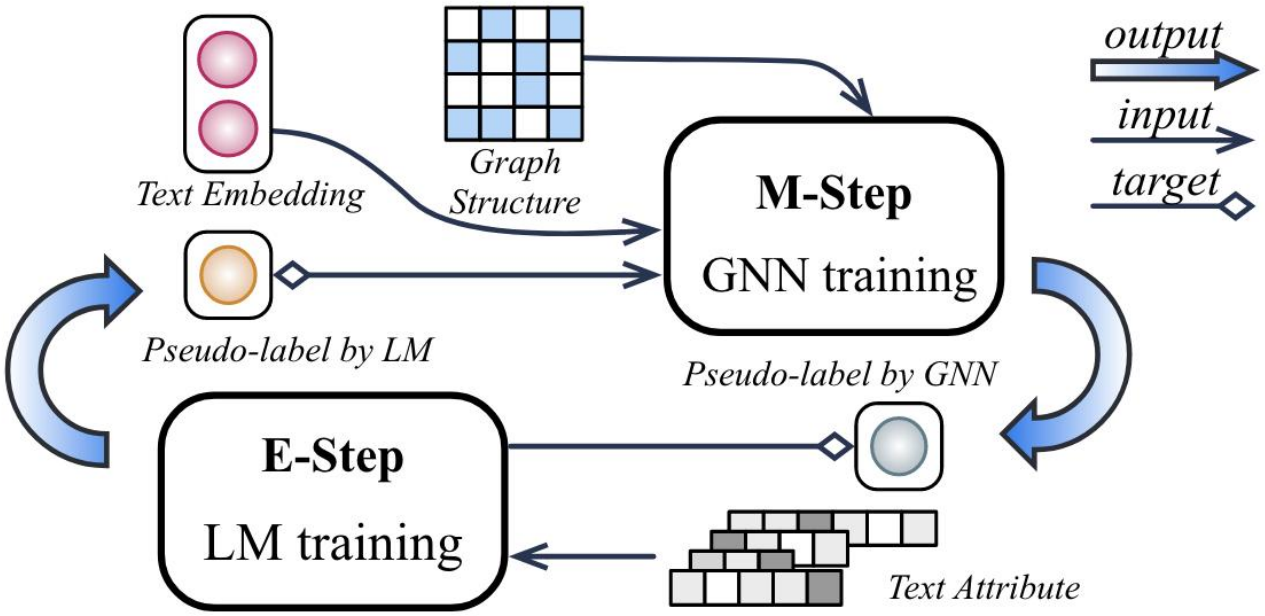
PubChemSTM-raw	PubChemSTM-extracted
SMILES: c1ccccc1	
Benzene is a colorless liquid with a sweet odor. It evaporates into the air very quickly and dissolves slightly in water.	*This molecule is* a colorless liquid with a sweet odor. It evaporates into the air very quickly and dissolves slightly in water.

Pipeline – Contrastive Learning



Liu S, Nie W, Wang C, et al. Multi-modal Molecule Structure-text Model for Text-based Editing and Retrieval. NMI 2023.

LLM as aligner | GLEM



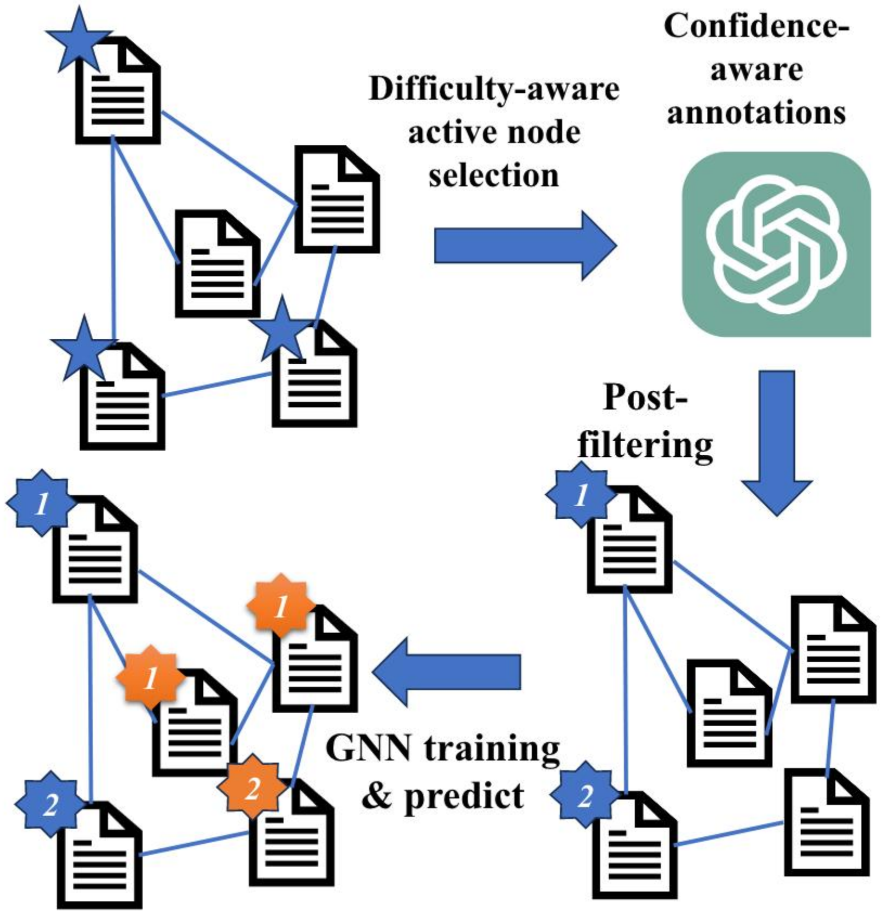
- **EM Framework**
  - E-step: LM optimization.
  - M-step: GNN optimization.
  - Iterative generate pseudo-labels and update both LM and GNN.

Zhao J, Qu M, Li C, et al. Learning on Large-scale Text-attributed Graphs via Variational Inference. ICLR 2023.

Others: LLM as Labeller | LLM-GNN

- **Pipeline**
  - Step1: Active node selection
  - Step2: Annotation
  - Step3: Post-filtering
  - Step4: GNN training/inference
- **Reliable Annotation**

	Cora		ogbn-products		WikiCS	
Prompt Strategy	Acc (%)	Cost	Acc (%)	Cost	Acc (%)	Cost
Vanilla (zero-shot)	68.33 ± 6.55	1	75.33 ± 4.99	1	68.33 ± 1.89	1
Vanilla (one-shot)	69.67 ± 7.72	2.2	78.67 ± 4.50	1.8	72.00 ± 3.56	2.4
TopK (zero-shot)	68.00 ± 6.38	1.1	74.00 ± 5.10	1.2	72.00 ± 2.16	1.1
Most Voting (zero-shot)	68.00 ± 7.35	1.1	75.33 ± 4.99	1.1	69.00 ± 2.16	1.1
Hybrid (zero-shot)	67.33 ± 6.80	1.5	73.67 ± 5.25	1.4	71.00 ± 2.83	1.4
Hybrid (one-shot)	70.33 ± 6.24	2.9	75.67 ± 6.13	2.3	73.67 ± 2.62	2.9



Chen Z, Mao H, Wen H, et al. Label-free Node Classification on Graphs with Large Language Models (LLMS). ICLR 2024.

Benchmarking GraphLLM

Role	Method	Predictor	GNN	PLM/LLM	Techniques Used		Learning Scenarios		Venue	Code
					Fine-tune	Prompt	Supervised	Zero-shot		
Enhancer	GIANT [9]	GNN	GraphSAGE, etc.	BERT	✗	✗	✓	✗	ICLR'22	Link
	TAPE [13]	GNN	RevGAT	ChatGPT	✗	✓	✓	✗	ICLR'24	Link
	OFA [26]	GNN	R-GCN	Sentence-BERT	✗	✓	✓	✓	ICLR'24	Link
	ENGINE [54]	GNN	GraphSAGE	LLaMA-2	✓	✓	✓	✗	IJCAI'24	Link
	ZeroG [25]	GNN	SGC	Sentence-BERT	✓	✓	✗	✓	SIGKDD'24	Link
Predictor	InstructGLM [50]	LLM	-	FLAN-T5/LLaMA-v1	✓	✓	✓	✗	EACL'24	Link
	GraphText [53]	LLM	-	ChatGPT/GPT-4	✓	✓	✓	✗	Arxiv	Link
	GraphAdapter [17]	LLM	GraphSAGE	LLaMA-2	✓	✓	✓	✗	WWW'24	Link
	GraphGPT [40]	LLM	GT	Vicuna	✓	✓	✓	✓	SIGIR'24	Link
	LLaGA [6]	LLM	-	Vicuna/LLaMA-2	✓	✓	✓	✗	ICML'24	Link
Aligner	GLEM [52]	GNN/LLM	GraphSAGE, etc.	RoBERTa	✓	✗	✓	✗	ICLR'23	Link
	PATTON [21]	LLM	GT	BERT/SciBERT	✓	✗	✓	✗	ACL'23	Link

Motivation

- 1. The use of different datasets, data processing approaches, and data splitting strategies in previous GraphLLM works.
- 2. The lack of benchmarks for zero-shot graph learning has led to limited exploration in this area.
- 3. Each method's computation and memory costs often overlooked.

Li Y, Wang P, Zhu X, et al. GLBench: A Comprehensive Benchmark for Graph with Large Language Models. Arxiv 2024.

Benchmarking GraphLLM | GLBench

➢ Comparison with existing benchmarks

Benchmark	#Datasets (Node-level)	#Domains	Text	#Models (GraphLLM)	Model Type	Supervision Scenario
Sen et al. [38]	2 (2)	1	✗	8 (0)	Classical	Supervised
Shchur et al. [39]	8 (8)	2	✗	8 (0)	GNN	Supervised
OGB [15]	14 (5)	3	✗	20 (0)	GNN	Supervised
CS-TAG [47]	8 (6)	2	✓	16 (2)	GNN, PLM, Enhancer	Supervised
GLBench	7 (7)	3	✓	18 (12)	GNN, PLM, GraphLLM	Supervised and Zero-shot

➢ Datasets

Dataset	# Nodes	# Edges	Avg. # Deg	Avg. # Tok	# Classes	# Train	Node Text	Domain
Cora	2,708	5,429	4.01	186.53	7	5.17%	Paper content	Citation
Citeseer	3,186	4,277	2.68	213.16	6	3.77%	Paper content	Citation
Pubmed	19,717	44,338	4.50	468.56	3	0.30%	Paper content	Citation
Ogbn-arxiv	169,343	1,166,243	13.77	243.19	40	53.70%	Paper content	Citation
WikiCS	11,701	216,123	36.94	642.04	10	4.96%	Entity description	Web link
Reddit	33,434	198,448	11.87	203.84	2	10.00%	User's post	Social
Instagram	11,339	144,010	25.40	59.25	2	10.00%	User's profile	Social

Li Y, Wang P, Zhu X, et al. GLBench: A Comprehensive Benchmark for Graph with Large Language Models. Arxiv 2024.

Benchmarking GraphLLM | GLBench

- **Supervised Scenario**
  - Effectiveness
  - LLM-as-predictor
  - LLM-as-enhancer
  - LLM-as-aligner
  - Scaling law

Model	Cora		Citeseer		Pubmed		Ogbn-arxiv		WikiCS		Reddit		Instagram	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
GCN [23]	**82.11**	**80.65**	69.84	65.49	79.10	79.19	72.24	51.22	80.35	77.63	63.19	62.49	65.75	58.75
GAT [43]	80.31	79.00	68.78	62.37	76.93	76.75	71.85	52.38	79.73	77.40	61.97	61.78	65.38	58.60
GraphSAGE [10]	79.88	79.35	68.23	63.10	76.79	76.91	71.88	52.14	79.87	77.05	58.51	58.41	65.12	55.85
Sent-BERT (22M) [36]	69.73	67.59	68.39	64.97	65.93	67.33	72.82	53.43	77.07	75.11	57.31	57.09	63.07	56.68
BERT (110M) [22]	69.71	67.53	67.77	64.10	63.69	64.93	72.29	53.30	78.55	75.74	58.41	58.33	63.75	57.30
RoBERTa (355M) [30]	69.68	67.33	68.19	64.90	71.25	72.19	72.94	52.70	78.67	76.16	57.17	57.10	63.57	56.87
GIANT [9]	81.04	<u>80.13</u>	65.82	62.31	76.89	76.05	72.04	50.81	80.48	78.67	64.67	64.64	66.01	56.11
TAPE [13]	80.95	79.79	66.06	61.84	79.87	79.30	72.99	51.43	<u>82.33</u>	<u>80.49</u>	60.73	60.50	65.85	50.49
OFA [26]	75.24	74.20	**73.04**	**68.98**	75.61	75.60	73.23	57.38	77.34	74.97	<u>64.86</u>	<u>64.95</u>	60.85	55.44
ENGINE [54]	<u>81.54</u>	79.82	72.15	<u>67.65</u>	74.74	75.21	<u>75.01</u>	57.55	81.19	79.08	63.20	59.34	**67.62**	**59.22**
InstructGLM [50]	69.10	65.74	51.87	50.65	71.26	71.81	39.09	24.65	45.73	42.70	55.78	53.24	57.94	54.87
GraphText [53]	76.21	74.51	59.43	56.43	74.64	75.11	49.47	24.76	67.35	64.55	61.86	61.46	62.64	54.00
GraphAdapter [17]	72.85	70.66	69.57	66.21	72.75	73.19	74.45	56.04	70.85	66.49	61.21	61.13	<u>67.40</u>	<u>58.40</u>
LLaGA [6]	74.42	72.50	55.73	54.83	52.46	68.82	72.78	53.86	73.88	70.90	**67.19**	**67.18**	62.94	54.62
$GLEM_{GNN}$ [52]	**82.11**	80.00	<u>71.16</u>	67.62	<u>81.72</u>	<u>81.48</u>	**76.43**	**58.07**	**82.40**	**80.54**	59.60	59.41	66.10	54.92
$GLEM_{LLM}$ [52]	73.79	72.00	68.78	65.32	79.18	79.25	74.03	<u>58.01</u>	80.23	78.30	57.97	57.56	65.00	54.50
PATTON [21]	70.50	67.97	63.60	61.12	**84.28**	**83.22**	70.74	49.69	80.81	77.72	59.43	57.85	64.27	57.48

Li Y, Wang P, Zhu X, et al. GLBench: A Comprehensive Benchmark for Graph with Large Language Models. Arxiv 2024.

Benchmarking GraphLLM | GLBench

Category	Model	$\mathcal{A}$	$\mathcal{S}$	Cora		Citeseer		Pubmed		WikiCS		Instagram	
				Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Graph SSL	DGI [44]	✓	✗	17.50	12.44	21.67	13.53	44.88	38.72	9.03	6.13	**63.64**	50.13
	GraphMAE [14]	✓	✗	27.08	23.66	15.24	14.44	22.03	15.65	10.74	6.69	53.56	<u>52.18</u>
LLMs	LLaMA3 (70B) [42]	✗	✓	<u>67.99</u>	<u>68.05</u>	51.44	49.98	77.00	64.18	**73.64**	**72.62**	38.23	36.41
	GPT-3.5-turbo [35]	✗	✓	65.67	63.22	50.58	49.34	75.99	69.90	68.75	66.56	49.39	49.67
	GPT-4o [1]	✗	✓	**68.62**	**68.49**	<u>53.55</u>	<u>52.42</u>	77.96	71.79	<u>71.52</u>	<u>70.06</u>	42.02	40.96
	DeepSeek-chat [3]	✗	✓	65.62	65.77	50.35	48.32	**79.23**	<u>74.30</u>	70.77	69.91	40.58	39.27
Training-free	**Emb w/ NA**	✓	✓	63.59	58.23	51.75	49.51	74.66	73.15	52.30	48.40	45.52	45.14
Enhancer	OFA [26]	✓	✓	23.11	23.30	32.45	28.67	46.60	35.04	34.27	33.72	53.63	51.10
	ZeroG [25]	✓	✓	62.52	57.53	**58.92**	**54.58**	<u>79.08</u>	**77.94**	60.46	57.24	<u>56.13</u>	**52.50**
Predictor	GraphGPT [40]	✓	✓	24.90	7.98	13.95	13.89	39.85	20.07	38.02	29.46	43.94	43.49

Zero-shot Scenario

- LLMs
- Semantics/Structures?
- Even a simple baseline can outperform existing GraphLLM methods.

Li Y, Wang P, Zhu X, et al. GLBench: A Comprehensive Benchmark for Graph with Large Language Models. Arxiv 2024.

Future Directions

- **Dealing with non–Text–Attributed–Graph.**
- **Dealing with Data Leakage.**
  - Especially for citation networks.
- **Improving Transferability.**
  - Transfer across datasets/domains/tasks.
- **Improving Explainability.**
  - Generate user-friendly explanations for graph reasoning, classification, etc.
- **Improving Efficiency.**
  - Especially for LLM-as-predictor methods.
  - PEFT.
- **Analysis and improvement of expressive ability.**

Q&A

Time	Section	Presenter
10:00-10:15	Part A: Opening & Introduction	Hong Cheng
10:15-10:50	Part B: Uni-modal Pretraining	Zhixun Li
10:50-11:30	Part C: Multi-modal Pretraining	Yuhan Li
11:30-12:00	Coffee Break	-
12:00-12:45	Part D: Pretraining with Prompting	Xiangguo Sun
12:45-13:00	Q&A	-

Part D Graph Pre-training with Prompting

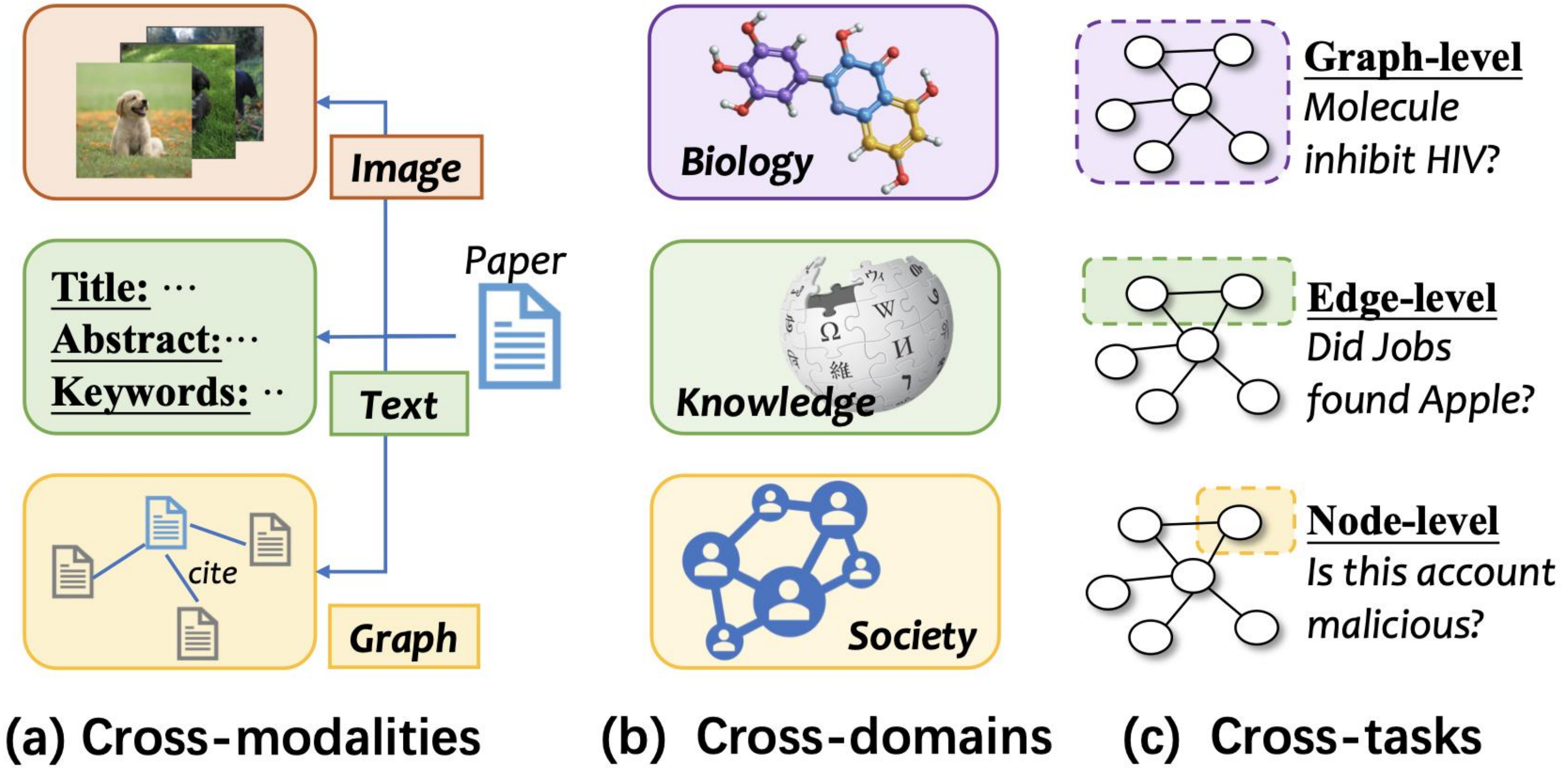
Xiangguo SUN

Graph Prompting

- What and why graph prompt
- A Basic workflow of graph prompt
- Graph prompt in multi-task settings
- Graph prompt in cross-domain settings
- Applications and open-source tools
- Prompt with LLMs and graphs

Graph AGI Still in the Early Stage

- **Why hard?**
  - Cross-modalities, cross-domains, cross-tasks
  - Social disputes: counterfactual outcomes, energy cost, etc.



Fine–tune v.s. Prompt

- ## Fine–tune
  - Need to tune the large pre-trained model (inefficient)
  - Do not change data
  - Limited task generalization
- ## Prompt
  - Frozen the large pre-trained model (efficient)
  - Has the capability of reformulating data
  - More general cross tasks

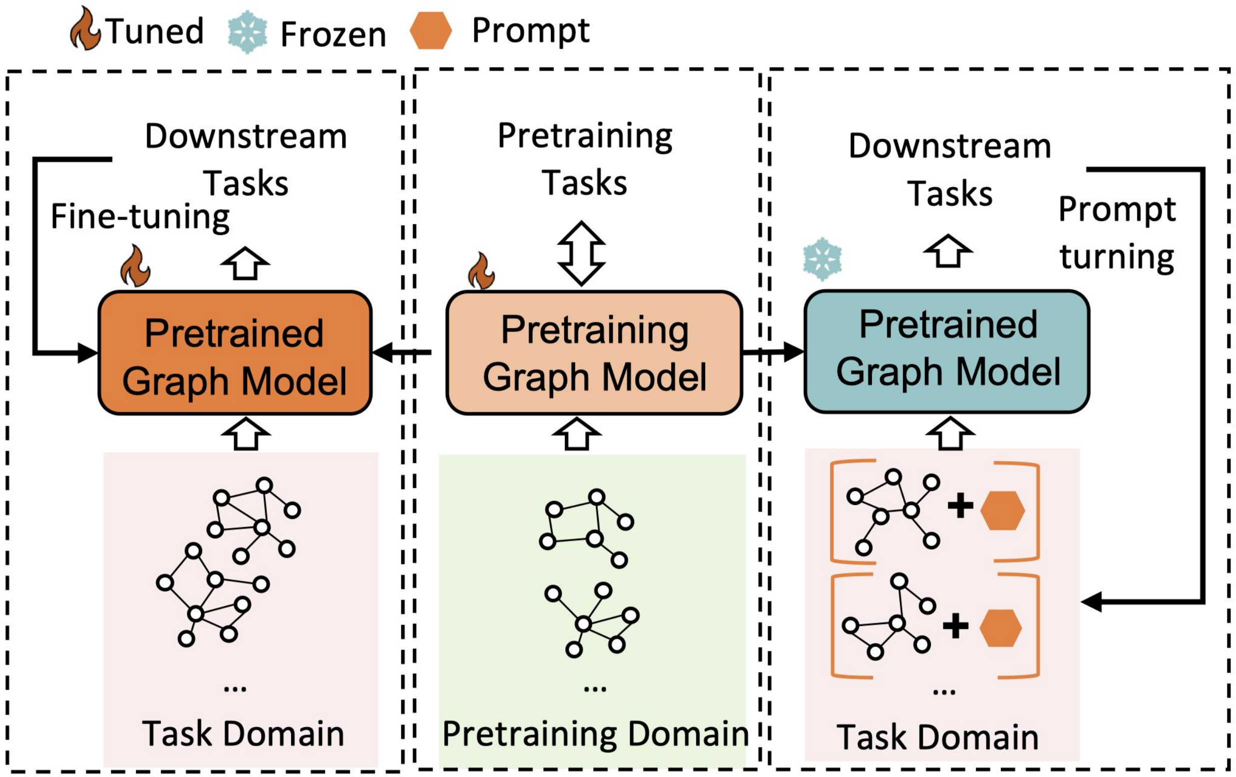
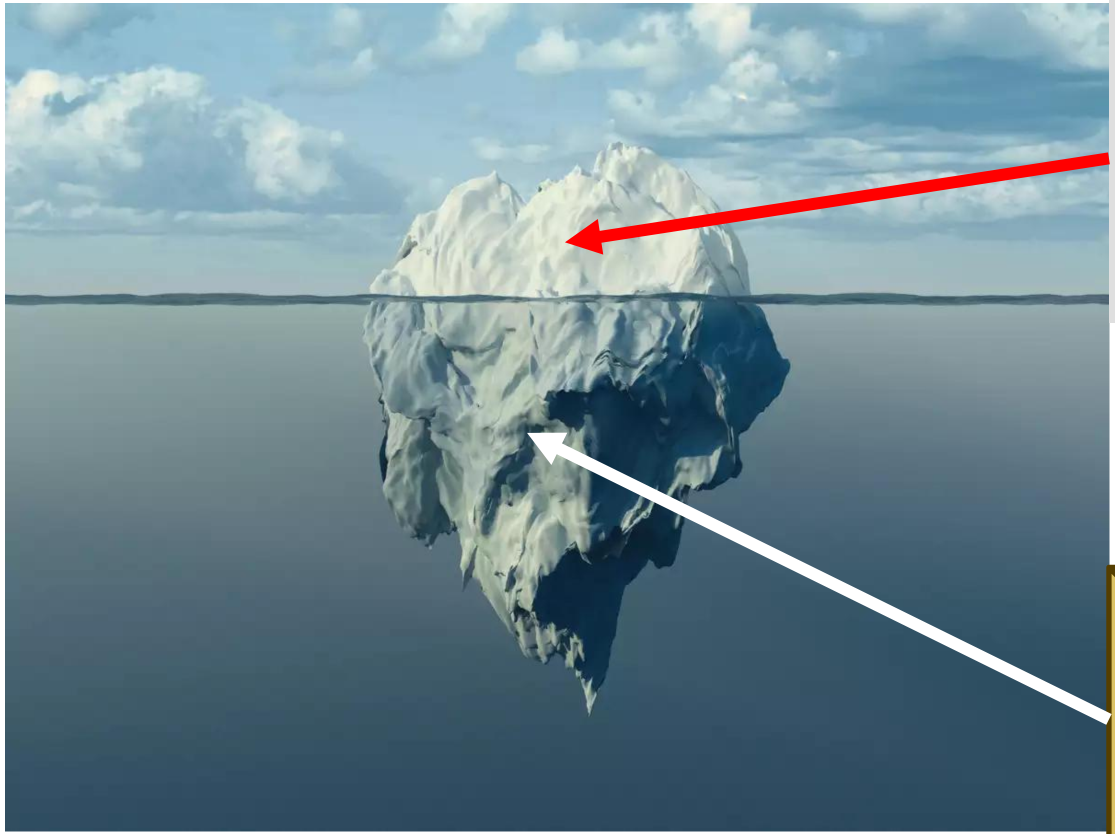


Figure 1: Fine-tuning, Pre-training, and Prompting.

Artificial General Intelligence (AGI)



- Large Language Models
- Training Tricks
- Computing Capability
- …

Data

Prompt: Promising for Graph AGI

- **A promising approach to reformulate data.**
  - Which is helpful for cross-domains demand.
- **Widely used in other modalities (NLP and CV)**
  - Which is promising for cross-modalities.

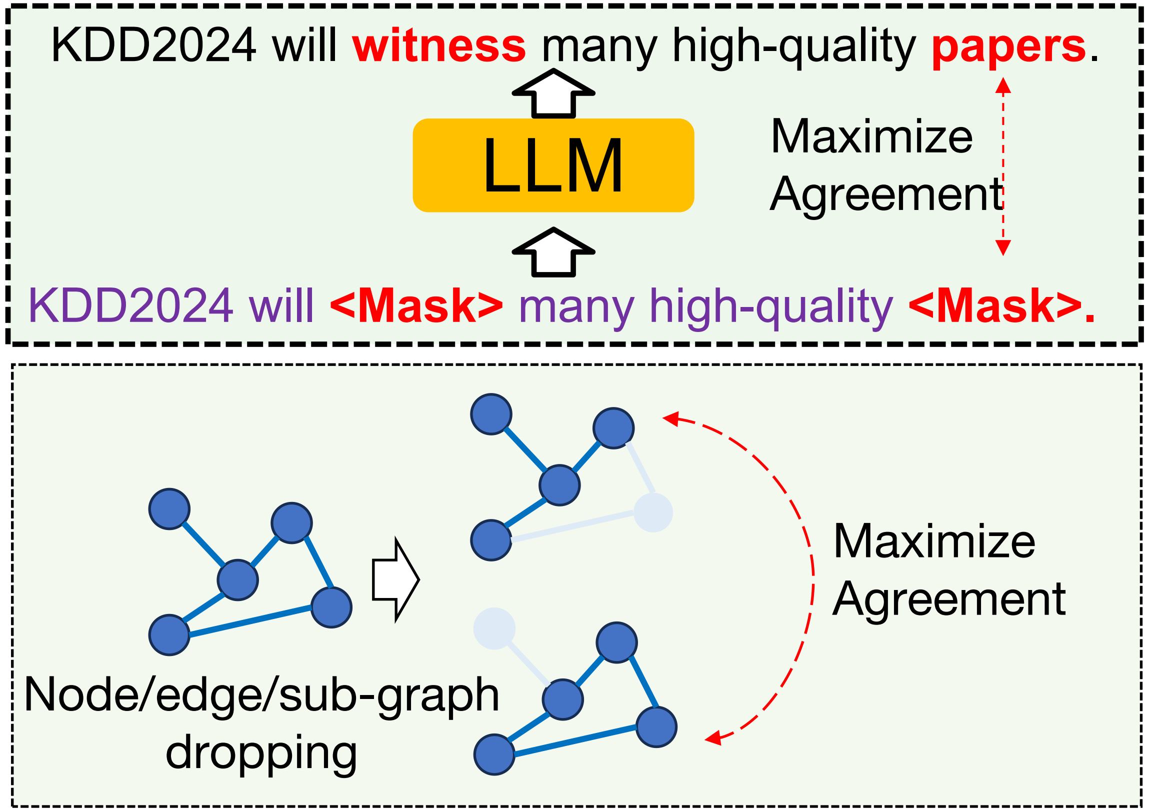
Prompt: Promising for Graph AGI

- **Reformulate downstream tasks to the pre-training task.**
  - Which is promising for cross-tasks.
- **No need to change the large foundation model again.**
  - Which is more efficient than fine-tuning

How to develop a prompting framework to graphs like language model?

Motivation

➢ Similar insights between LLM and GNN pre-training



Pre-training in LLM: Masked word prediction

Pre-training in graph models: contrastive learning.

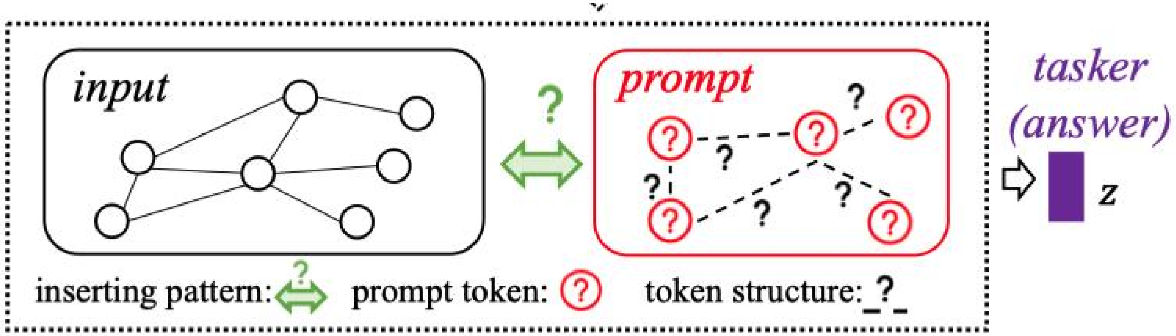
Aligning two graph views is very similar to predicting some vacant "masks" on graphs.

Challenge 1

- **Designing the graph prompt is more intractable than language prompts**
  - NLP prompts are usually some preset tokens, whereas the graph prompt needs to know how to organize these tokens and how to insert the prompt into the original graph.

KDD2023 will witness many high-quality papers. **I feel so [MASK]**

A language prompt



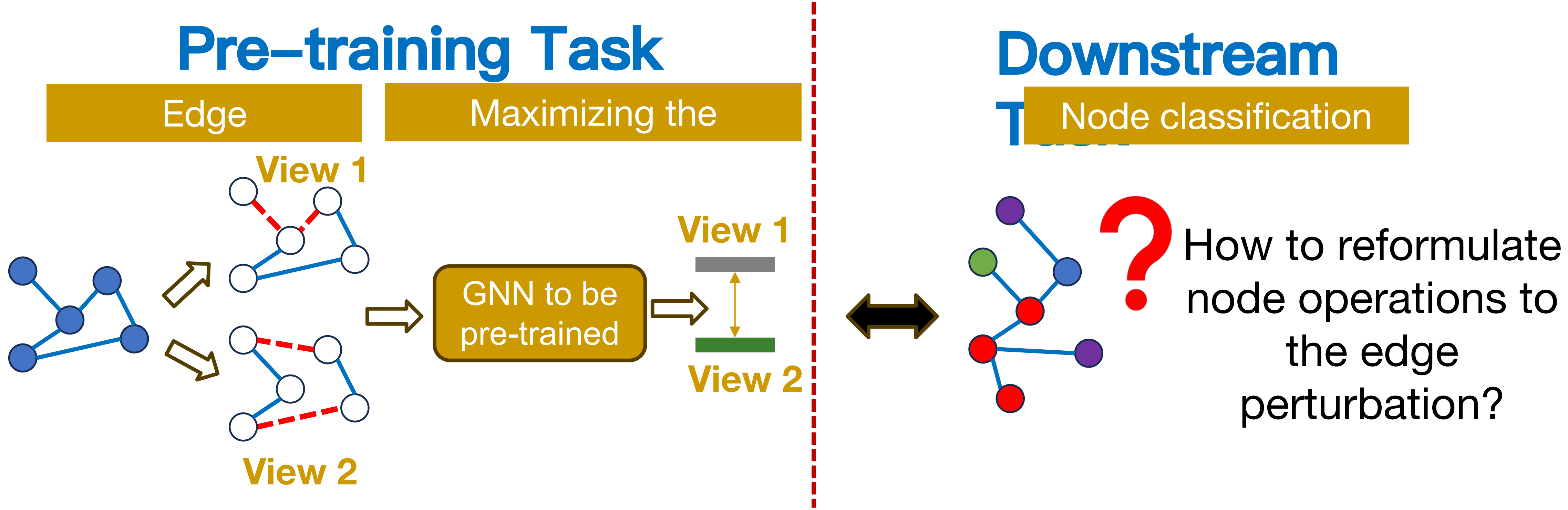
insert the prompt to the input graph

language prompt

graph prompt

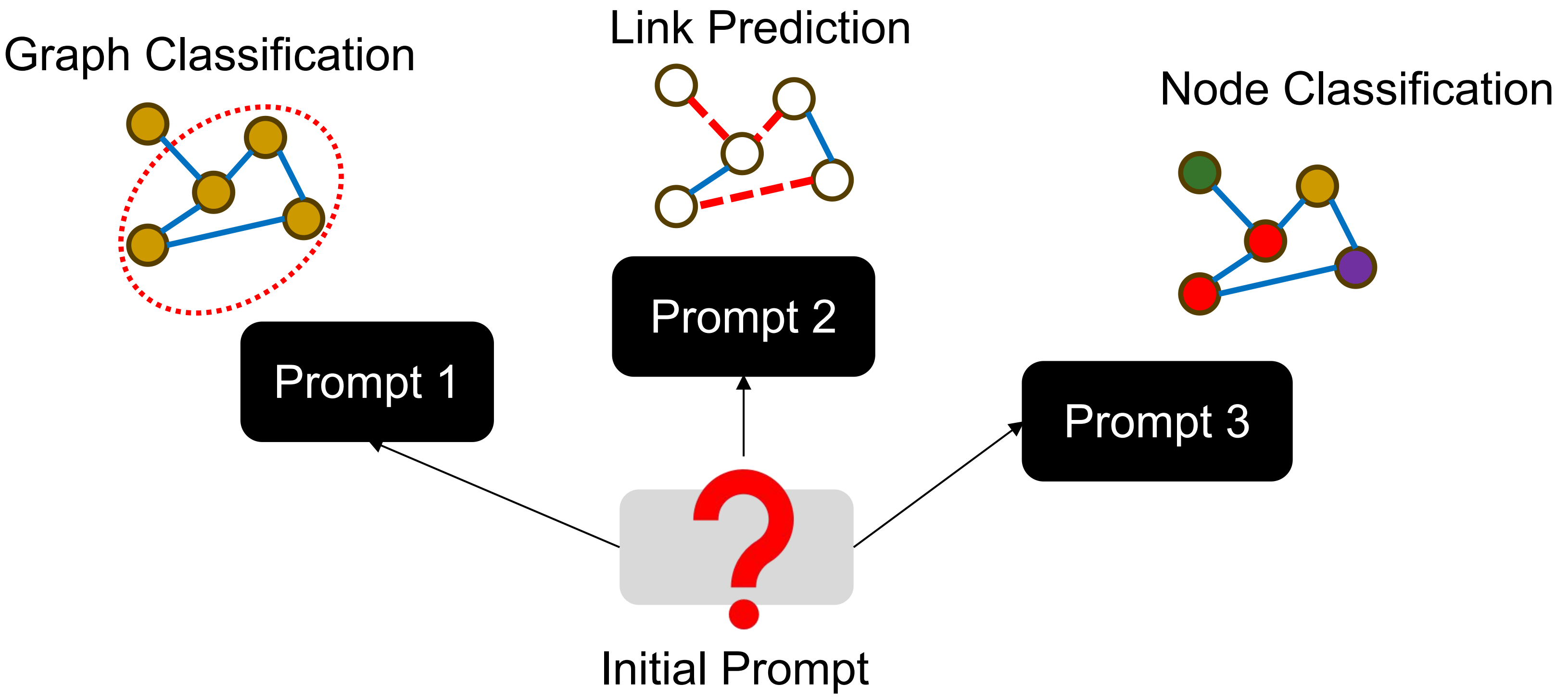
Challenge 2

- **Reconciling downstream problems to the pre–training task is more difficult in graph domains**
  - Graph tasks with node level, edge level, and graph level are far diversified.



Challenge 3

- **Learning reliable prompts is more difficult in the multi-task setting**
  - Hand-crafted prompts are usually task-bounded, which is far from sufficient for multiple tasks.



Revisit Language Prompt

- **Soft-Prompt and Hand-crafted Prompt**
  - Hand-crafted prompts are manually designed phrases.
  - Soft-prompts are learnable word latent vectors
- **Make graph prompt learnable (soft-prompt for graphs)**
  - Hand-crafted: not clear what should they look like.
  - Soft-prompts: learnable on graph and are more easily to achieve.

KDD2023 will witness many high-quality papers. I feel so [MASK]

A language prompt

Hand-crafted prompt

KDD2023 will witness many high-quality papers [MASK]

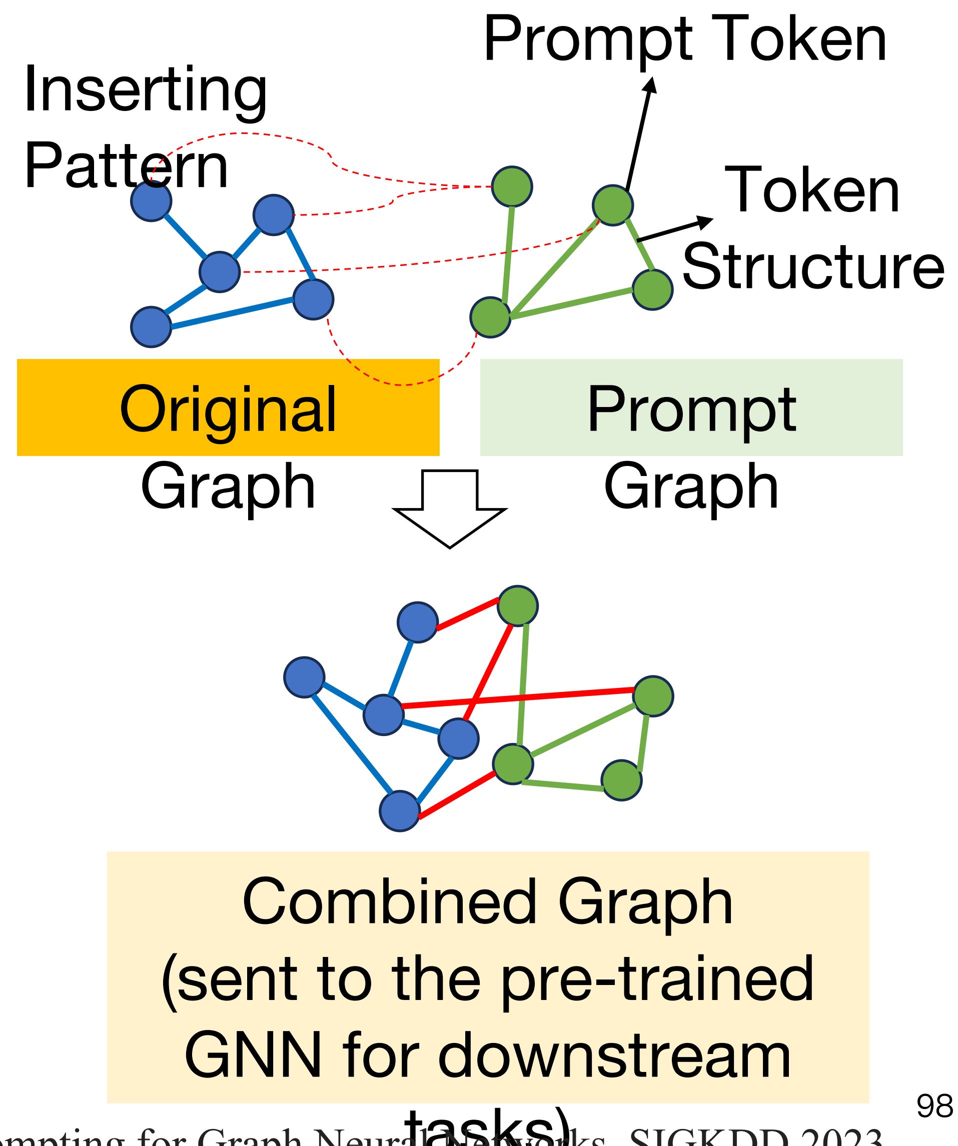
A language prompt

soft prompt

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD 2023

Unified Soft–Prompt for Graphs

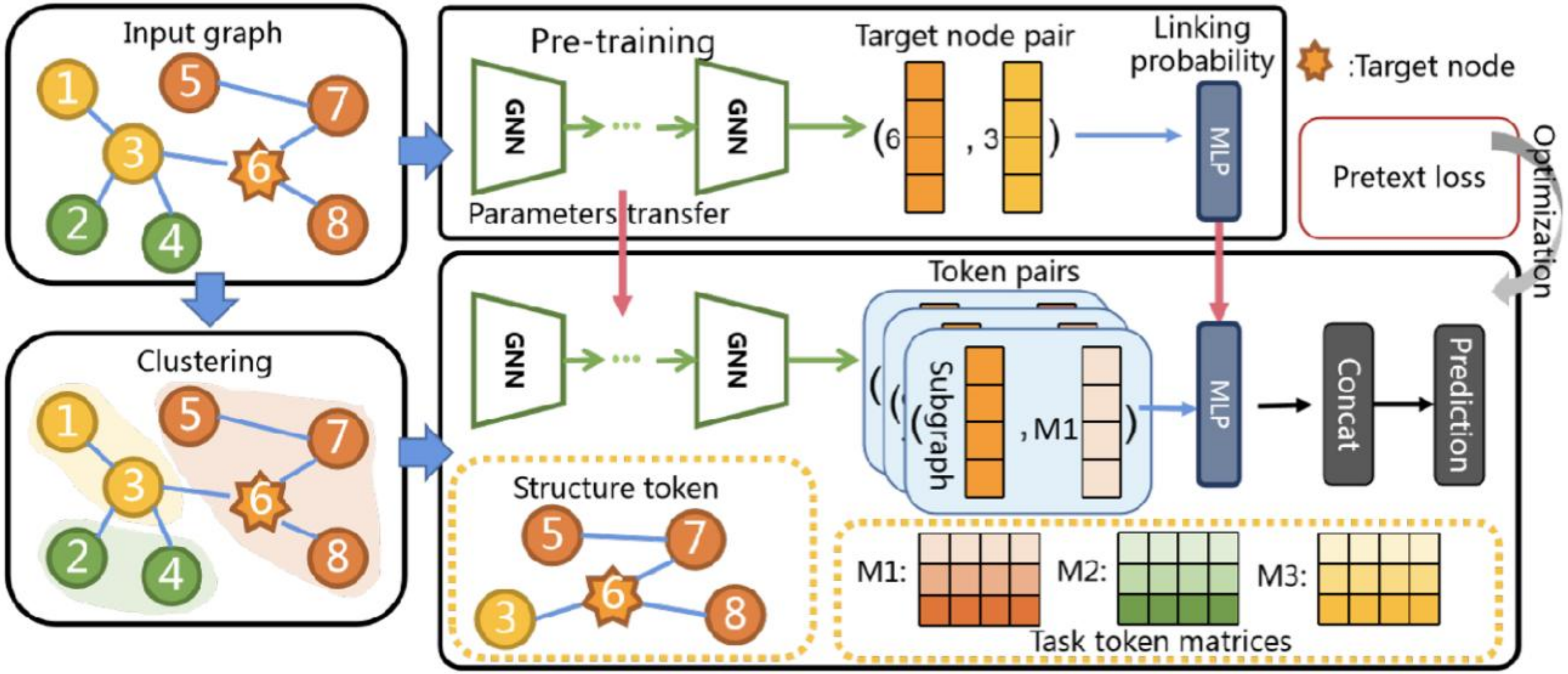
- **Prompt Token**
  - Vectorized information with the same size as node features.
- **Token Structure**
  - Inner connections among different tokens.
- **Inserting Pattern**
  - Cross links between prompt tokens and the original graph.



Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD 2023

Reformulating Downstream Tasks

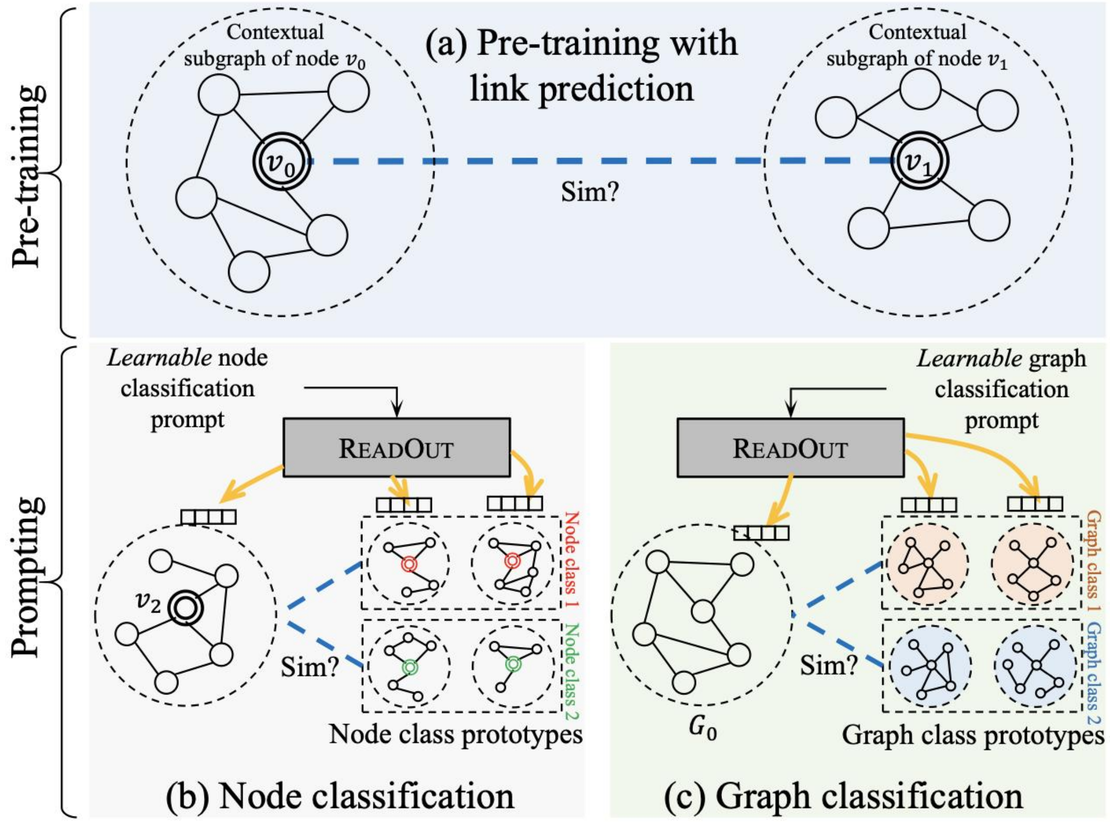
- **Node-level to edge-level**



Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, Xin Wang. GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks. In KDD'2022

Reformulating Downstream Tasks

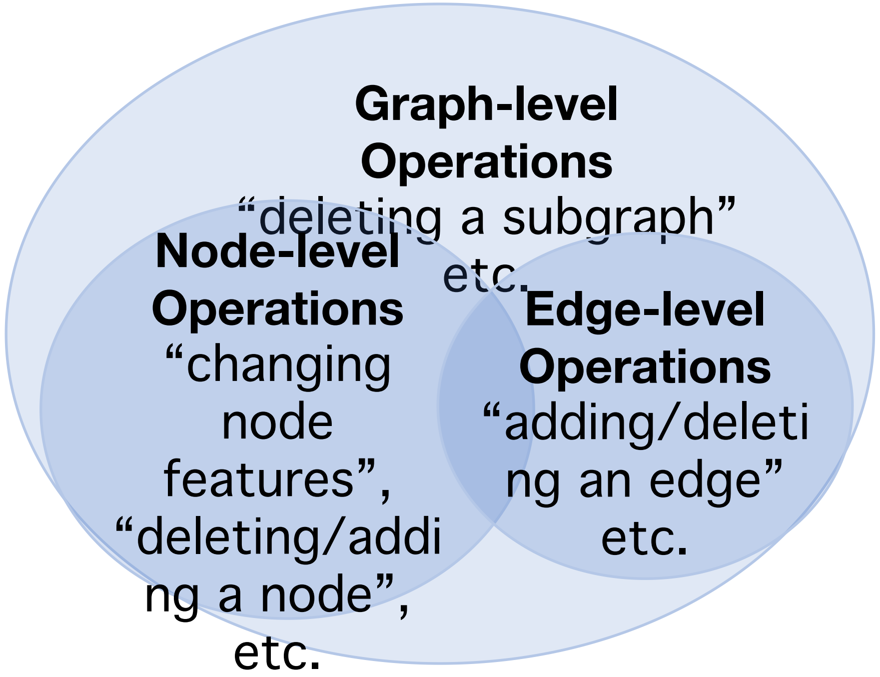
➢ **Reformulating downstream tasks to link-level tasks**



Zemin Liu, Xingtong Yu, Yuan Fang, Xinming Zhang. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In WWW'2023

Reformulating Downstream Tasks

- **Reformulating downstream tasks to graph-level tasks**
  - Node/edge-level operations can be treated as some special cases at the graph-level operations.
    - e.g, "deleting a subgraph" is the higher-level operation of "deleting nodes and edges".

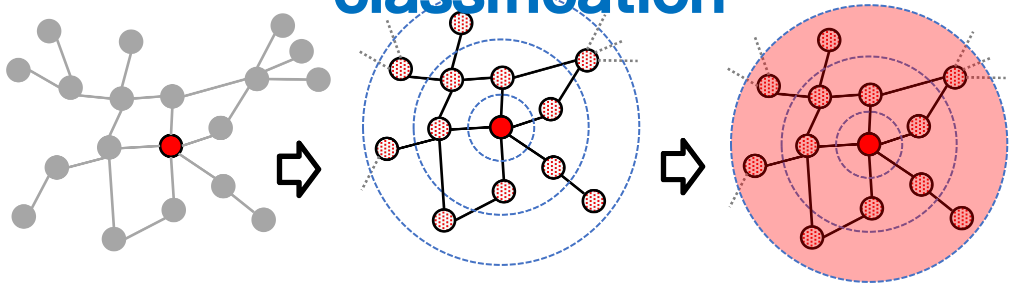


Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD 2023

Reformulating Downstream Tasks

- **Reformulating downstream tasks by induced graphs**
  - Node tasks to graph tasks.
  - Edge tasks to graph tasks.

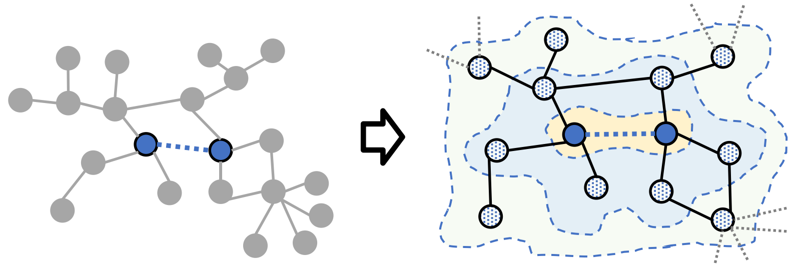
Reformulating node classification to graph classification



Finding a k-hop ego-net for the target node

Assigning the node label to the graph label

Reformulating link prediction to graph classification



Graph label is positive if the node pair has an edge and vice versa.

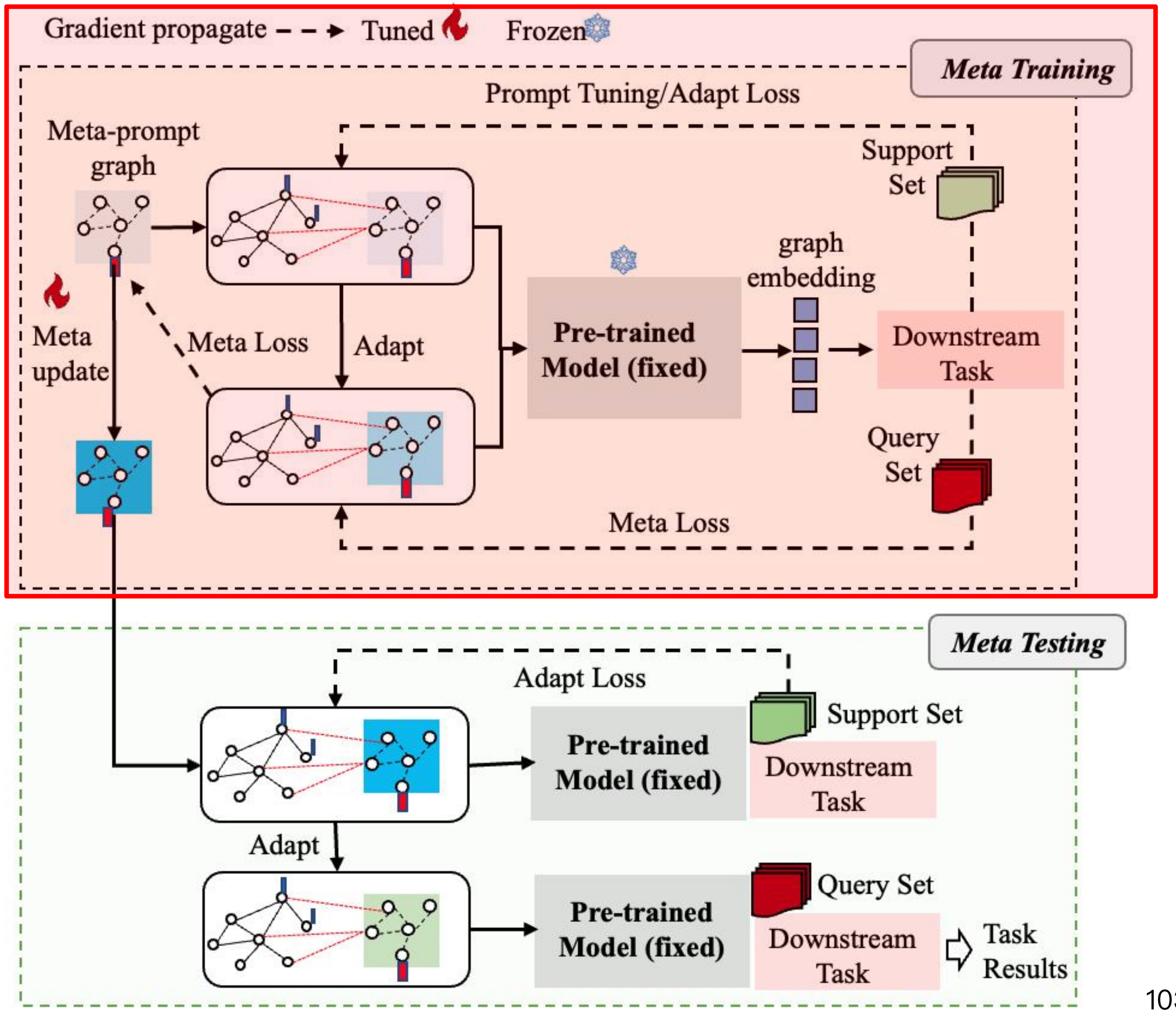
Extending a node pair to their k-hop neighbours

Assigning the graph label according to node pair connection

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD 2023

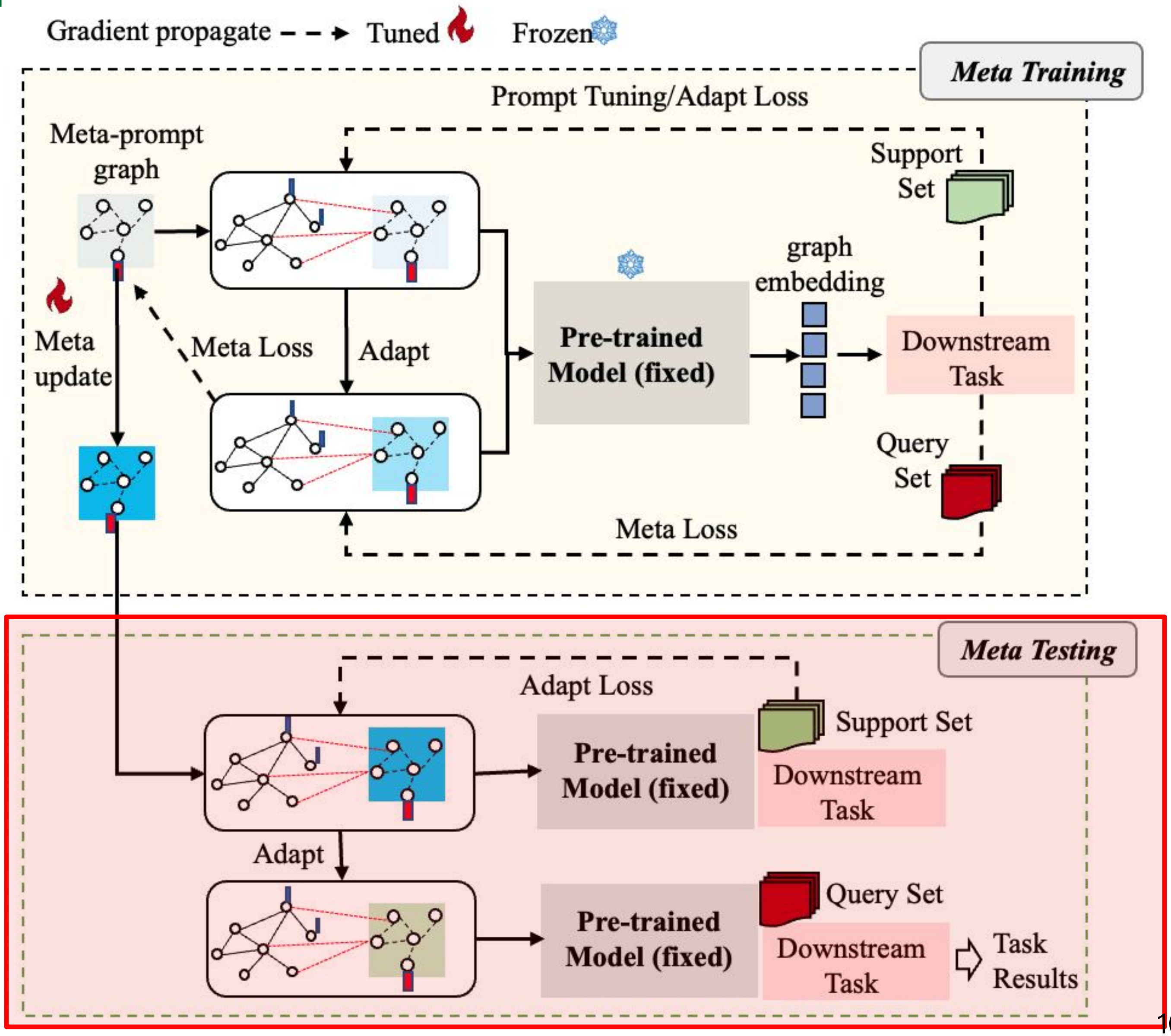
Multi-task Prompting via Meta Learning

Phase 1: Meta Training on Source Tasks



Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD 2023

Multi-task Prompting via Meta Learning

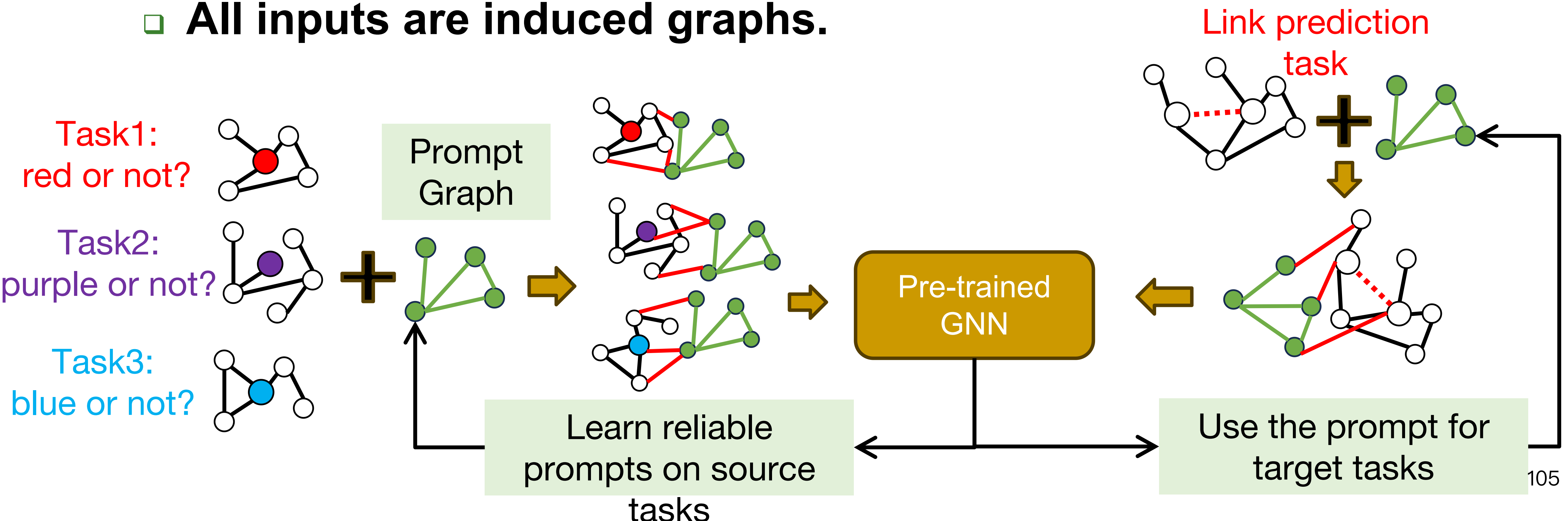


Phase 2: Meta Testing on the Target Task

Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD 2023

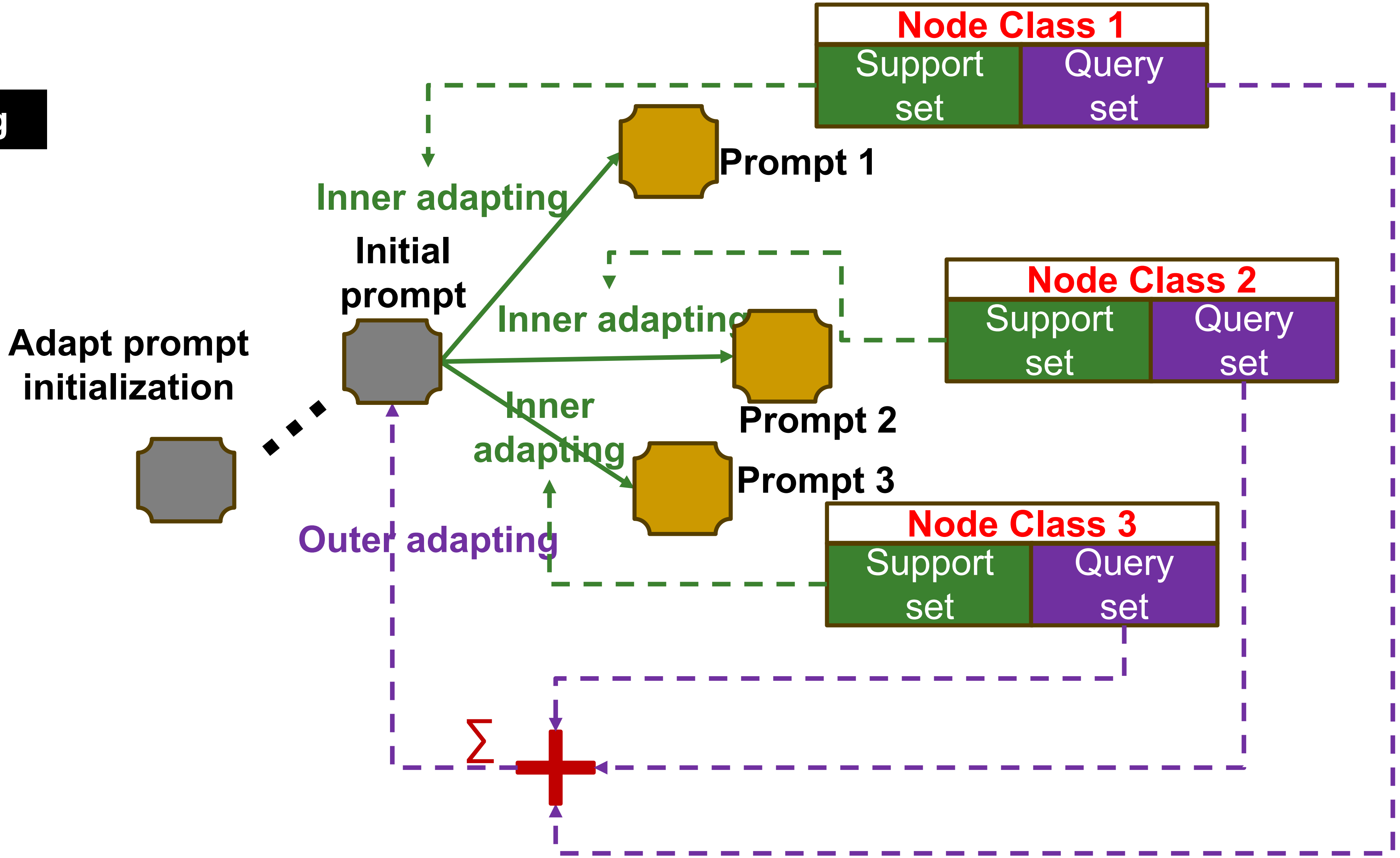
Multi–task Prompting via Meta Learning

- **An Example:**
  - **Target task:** link prediction.
  - **Source tasks:** Node binary classification tasks.
    - Each task corresponds to one node class.
  - **All inputs are induced graphs.**



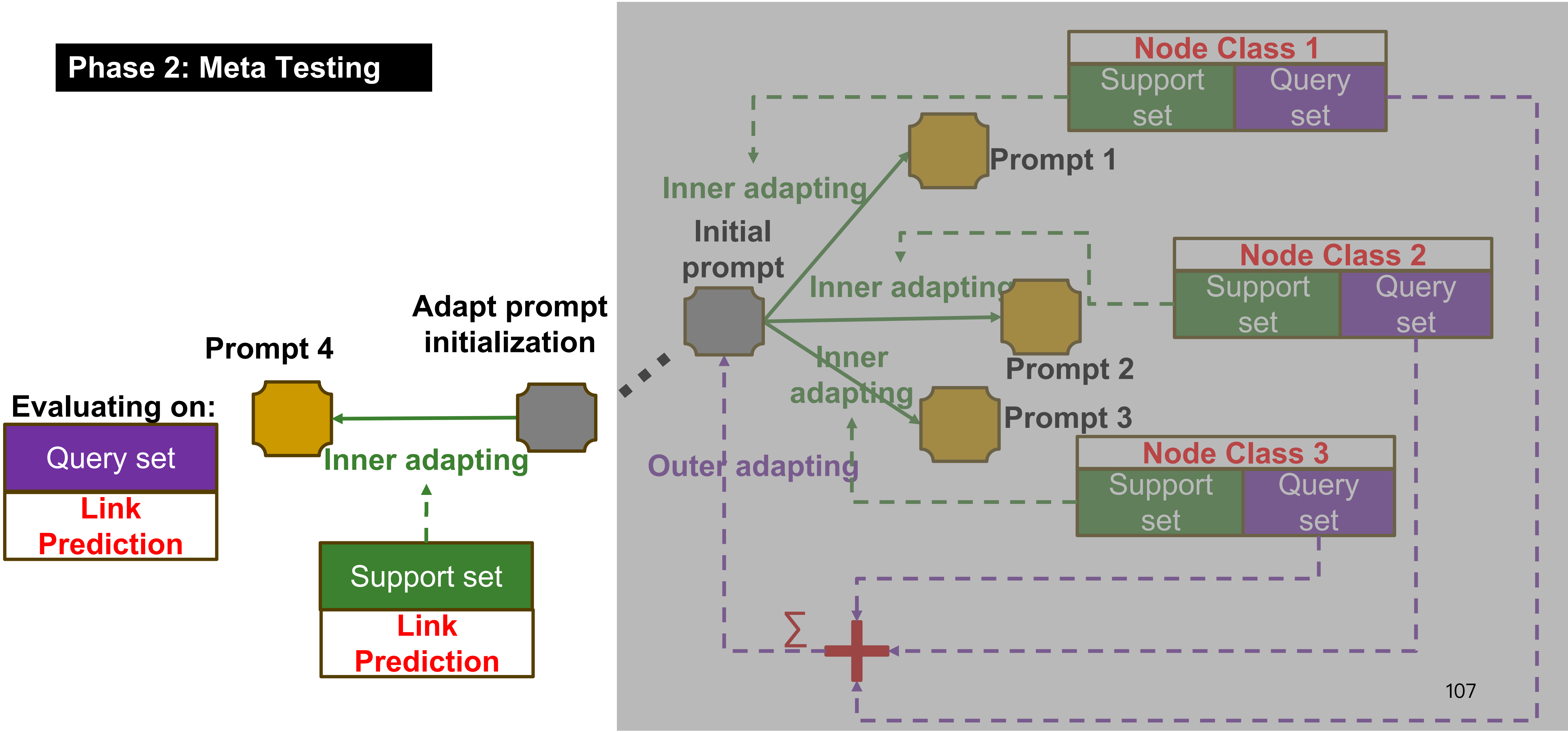
Multi–task Prompting via Meta Learning

Phase 1: Meta Training



Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, Jihong Guan. All in One: Multi-task Prompting for Graph Neural Networks. SIGKDD2023

Multi-task Prompting via Meta Learning



Why It Works?

- The nature of prompting is to manipulate the input data to match the pretext.
- The flexibility of data operations is the bottleneck of prompting performance.

Fang et al. [1] proved that we can always learn an appropriate prompt token $p*$ making the following equation stand:

$$\varphi^*(\boldsymbol{A}, \boldsymbol{X} + p^*) = \varphi^*(g(\boldsymbol{A}, \boldsymbol{X})) + O_{p\varphi}$$

- φ^*: pre-trained model
- p^*: a prompt token
- $\boldsymbol{A}, \boldsymbol{X}$: adjacent matrix and feature matrix
- $g(.)$: graph manipulation (e.g. “changing node features”, “adding or removing edges/subgraphs” etc)

This means we can learn an appropriate token applied to the original graph to imitate any graph manipulation.

[1] Taoran Fang, et al. Prompt Tuning for Graph Neural Networks. arXiv preprint arXiv:2209.15240 (2022).

Why It Works?

The error bound $O_{p\varphi}$ is related to: (1) some non-linear layers of the model (unchangeable), and (2) the quality of the learned prompt (changeable), which is promising to be further narrowed down by a more advanced prompt scheme.

$$\varphi^*(\underbrace{A,\ X + p^*}) = \varphi^*(g(A, X)) + O_{p\varphi}$$

$$\Downarrow$$

$$\varphi^*(\psi(\mathcal{G}, \mathcal{G}_p^*)) = \varphi^*(g(A, X)) + O_{p\varphi}^*$$

We extend the standalone token p^* to a prompt graph $\mathcal{G}_p^*$ that has multiple prompt tokens with learnable inner structures and more advanced inserting pattern ψ to the original graph $\mathcal{G}$

We can empirically demonstrate: $O_{p\varphi}^* \leq O_{p\varphi}$
That means our method supports more flexible transformations on graphs to match various pre-training strategies.

Experiments

➢ Multi-Task Performance with Few-shot Learning Setting

Node classification

Table 2: Node-level performance (%) with 100-shot setting. IMP (%): the average improvement of prompt over the rest.

Training schemes	Methods	Cora			CiteSeer			Reddit			Amazon			Pubmed		
		Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
supervised	GAT	74.45	73.21	82.97	83.00	83.20	89.33	55.64	62.03	65.38	79.00	73.42	97.81	75.00	77.56	79.72
	GCN	77.55	77.45	83.71	88.00	81.79	94.79	54.38	52.47	56.82	95.36	93.99	96.23	53.64	66.67	69.89
	GT	74.25	75.21	82.04	86.33	85.62	90.13	61.50	61.38	65.56	85.50	86.01	93.01	51.50	67.34	71.91
pre-train + fine-tune	GraphCL+GAT	76.05	76.78	81.96	87.64	88.40	89.93	57.37	66.42	67.43	78.67	72.26	95.65	76.03	77.05	80.02
	GraphCL+GCN	78.75	79.13	84.90	87.49	89.36	90.25	55.00	65.52	74.65	96.00	95.92	98.33	69.37	70.00	74.74
	GraphCL+GT	73.80	74.12	82.77	88.50	88.92	91.25	63.50	66.06	68.04	94.39	93.62	96.97	75.00	78.45	75.05
	SimGRACE+GAT	76.85	77.48	83.37	90.50	91.00	91.56	56.59	65.47	67.77	84.50	84.73	89.69	72.50	68.21	81.97
	SimGRACE+GCN	77.20	76.39	83.13	83.50	84.21	93.22	58.00	55.81	56.93	95.00	94.50	98.03	77.50	75.71	87.53
	SimGRACE+GT	77.40	78.11	82.95	87.50	87.05	91.85	66.00	69.95	70.03	79.00	73.42	97.58	70.50	73.30	74.22
prompt	GraphCL+GAT	76.50	77.26	82.99	88.00	90.52	91.82	57.84	67.02	75.33	80.01	75.62	97.96	77.50	78.26	83.02
	GraphCL+GCN	79.20	79.62	85.29	88.50	91.59	91.43	56.00	68.57	78.82	96.50	96.37	98.70	72.50	72.64	79.57
	GraphCL+GT	75.00	76.00	83.36	91.00	91.00	93.29	65.50	66.08	68.86	95.50	95.43	97.56	76.50	79.11	76.00
	SimGRACE+GAT	76.95	78.51	83.55	93.00	93.14	92.44	57.63	66.64	69.43	95.50	95.43	97.56	73.00	74.04	81.89
	SimGRACE+GCN	77.85	76.57	83.79	90.00	89.47	94.87	59.50	55.97	59.46	95.00	95.24	98.42	78.00	78.22	87.66
	SimGRACE+GT	78.75	79.53	85.03	91.00	91.26	95.62	69.50	71.43	70.75	86.00	83.72	98.24	73.00	73.79	76.64
IMP (%)		1.47	1.94	1.10	3.81	5.25	2.05	3.97	5.04	6.98	4.49	5.84	2.24	8.81	4.55	4.62
Reported Acc of GPPT (Label Ratio 50%)		77.16	–	–	65.81	–	–	92.13	–	–	86.80	–	–	72.23	–	–
appr. Label Ratio of our 100-shot setting			~ 25%			~ 18%			~ 1.7%			~ 7.3%			~ 1.5%	

Experiments

➢ **Multi-Task Performance with Few-shot Learning Settings**

Edge classification

Table 12: Edge-level performance (%) with 100-shot setting. IMP (%): the average improvement of prompt over the rest.

Training schemes	Methods	Cora			CiteSeer			Reddit			Amazon			Pubmed		
		Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
supervised	GAT	84.30	83.35	85.43	68.63	82.79	89.98	93.50	93.03	94.48	85.00	82.67	88.78	80.05	77.07	79.26
	GCN	83.85	84.90	85.90	66.67	81.01	89.62	83.50	84.51	91.43	89.00	89.81	98.85	79.00	77.73	80.19
	GT	85.95	86.01	87.25	69.70	83.03	82.46	95.50	94.52	96.89	94.00	93.62	99.34	74.50	65.77	85.19
pre-train + fine-tune	GraphCL+GAT	85.64	85.97	87.22	72.67	82.85	92.98	94.00	93.75	98.43	86.50	86.96	84.47	85.54	83.92	91.78
	GraphCL+GCN	86.36	85.82	86.39	70.67	81.82	90.00	94.00	93.94	97.04	86.50	84.92	98.41	80.00	78.05	85.21
	GraphCL+GT	85.79	86.27	87.51	86.01	85.38	88.58	96.67	95.38	97.65	96.50	97.42	98.12	85.50	87.11	81.68
	SimGRACE+GAT	86.85	86.80	88.12	85.33	85.26	90.04	95.50	95.54	97.11	87.50	86.34	88.65	80.01	81.03	86.89
	SimGRACE+GCN	85.62	85.38	87.83	89.33	86.34	95.10	88.00	87.88	94.49	98.45	97.57	98.29	80.50	82.58	91.22
	SimGRACE+GT	86.35	87.03	88.47	86.00	89.52	90.42	97.50	95.54	96.92	96.50	96.45	99.09	81.00	79.57	85.69
prompt	GraphCL+GAT	86.85	86.88	87.92	76.67	83.00	96.22	95.36	94.50	98.65	88.50	86.00	87.15	86.50	84.75	92.61
	GraphCL+GCN	86.87	86.80	87.79	76.67	82.37	93.54	95.50	95.52	97.75	86.96	85.63	98.66	81.50	78.61	86.11
	GraphCL+GT	87.02	86.90	87.97	86.67	88.00	91.10	97.03	95.94	98.62	98.50	98.48	98.53	86.50	87.78	82.21
	SimGRACE+GAT	87.37	87.33	88.37	91.33	92.30	95.18	95.72	96.69	97.64	95.50	95.38	98.89	80.50	82.03	87.86
	SimGRACE+GCN	86.85	86.80	88.67	93.47	97.69	97.08	88.00	88.12	95.10	98.50	98.52	98.55	81.00	83.76	91.41
	SimGRACE+GT	87.30	87.24	88.74	95.33	96.52	94.46	98.00	98.02	99.38	98.50	98.52	99.10	82.50	80.45	87.61
	IMP(%)	1.65	1.48	1.28	12.26	6.84	5.21	1.94	2.29	1.88	3.63	3.44	2.03	2.98	4.64	3.21

Experiments

➢ Multi-Task Performance with Few-shot Learning Settings

Graph classification

Table 13: Graph-level performance (%) with 100-shot setting. IMP (%): the average improvement of prompt over the rest.

Training schemes	Methods	Cora Acc	Cora F1	Cora AUC	CiteSeer Acc	CiteSeer F1	CiteSeer AUC	Reddit Acc	Reddit F1	Reddit AUC	Amazon Acc	Amazon F1	Amazon AUC	Pubmed Acc	Pubmed F1	Pubmed AUC
supervised	GAT	84.40	86.44	87.60	86.50	84.75	91.75	79.50	79.76	82.11	93.05	94.04	93.95	69.86	72.30	66.92
	GCN	83.95	86.01	88.64	85.00	82.56	93.33	64.00	70.00	78.60	91.20	91.27	94.33	61.30	59.97	66.29
	GT	85.85	85.90	89.59	77.50	75.85	89.72	69.62	68.01	66.32	90.33	91.39	94.39	60.30	60.88	67.62
pre-train + fine-tune	GraphCL+GAT	85.50	85.54	89.31	83.00	85.47	92.13	72.03	72.82	83.23	92.15	92.18	94.78	85.50	85.50	86.33
	GraphCL+GCN	85.50	85.59	87.94	86.50	84.57	94.56	71.00	71.90	80.33	93.58	93.55	94.93	78.75	77.29	89.40
	GraphCL+GT	85.95	85.05	87.92	84.50	81.87	88.36	69.63	70.06	81.35	91.68	91.55	94.78	86.85	86.93	88.91
	SimGRACE+GAT	86.04	86.33	88.55	83.50	85.84	90.09	81.32	81.64	88.61	93.58	93.57	93.91	87.33	86.70	88.02
	SimGRACE+GCN	85.95	86.05	89.33	84.50	86.46	91.60	80.50	81.52	89.11	90.73	90.52	94.85	85.26	84.64	86.99
	SimGRACE+GT	86.40	86.47	89.64	81.00	81.54	89.81	69.50	70.97	77.11	92.63	92.56	94.04	85.95	86.05	89.37
prompt	GraphCL+GAT	86.40	86.47	89.46	86.50	89.93	92.24	73.36	73.32	84.77	94.08	94.02	94.20	85.95	85.97	87.17
	GraphCL+GCN	85.95	86.01	88.95	87.00	85.87	95.35	72.50	72.91	81.37	94.05	94.05	94.98	84.60	84.43	88.96
	GraphCL+GT	86.05	85.17	88.93	85.50	85.28	88.60	72.63	70.97	82.39	92.63	92.64	94.82	87.03	86.96	89.10
	SimGRACE+GAT	86.67	86.36	89.51	87.50	88.37	91.47	82.62	83.33	89.41	93.35	94.66	94.61	87.75	87.69	88.88
	SimGRACE+GCN	86.85	86.90	89.95	85.00	85.85	91.95	81.00	82.24	89.43	93.95	92.06	93.89	85.50	85.54	87.30
	SimGRACE+GT	86.85	86.87	89.75	87.50	86.63	90.85	76.50	80.82	86.84	94.05	94.06	94.96	86.40	86.50	89.74
	IMP(%)	1.12	0.43	0.79	3.52	4.54	0.53	4.69	4.31	6.13	1.72	1.39	0.14	10.66	10.[illegible]7	9.16

From Multi-task to Multi-domain

➢ Domain transfer on graphs via prompt

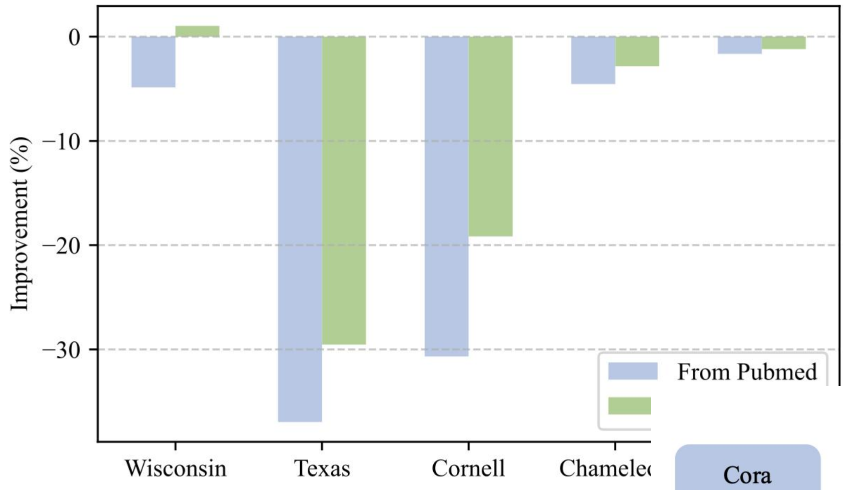


Figure 1: Negative transfer phenomenon in th
cross-domain transfer setting. Sources (Pubm
are two homophilic datasets. Targets (Wiscon
nell, Chameleon, and Squirrel) are five heter

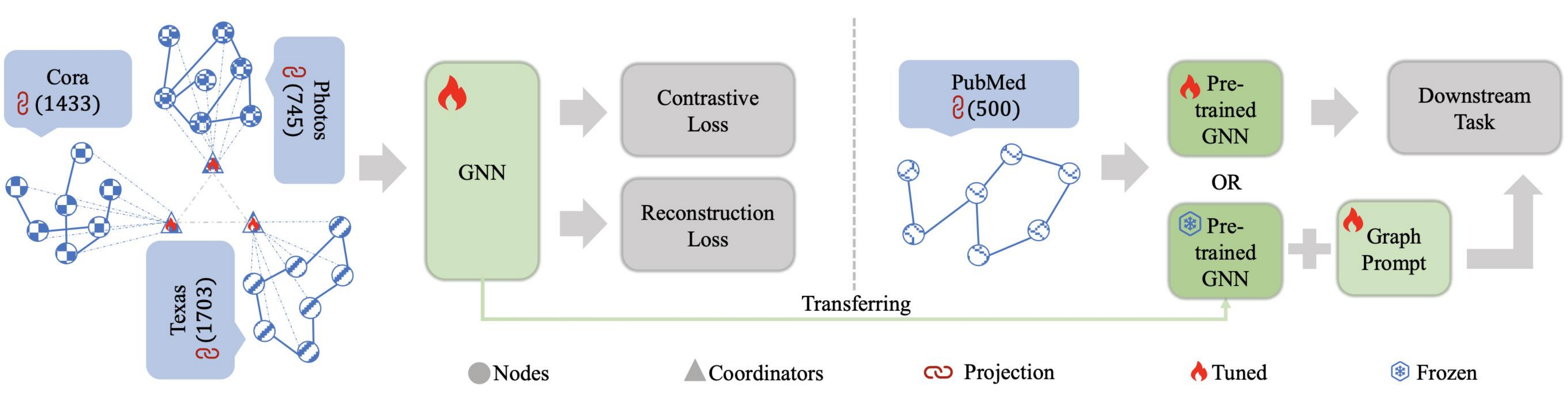


Figure 2: Overview of our proposed GCOPE method. The left part is our pretraining stage and the right part transferring stage.

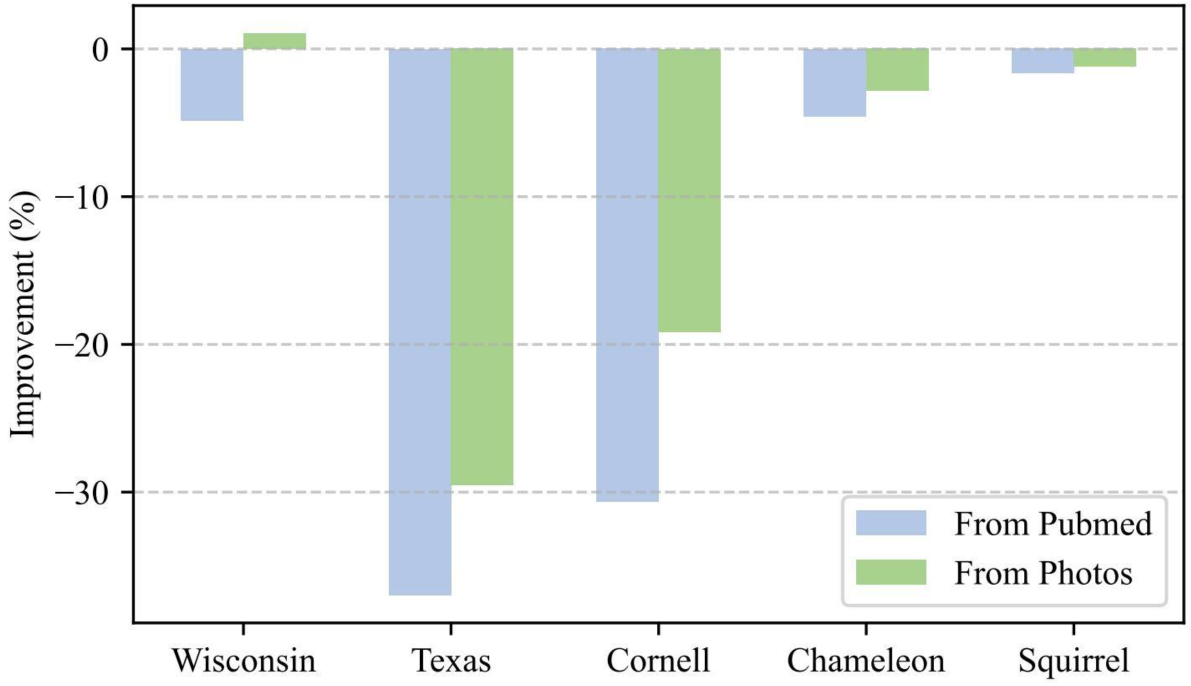
Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, Jia Li. All in One and One for All: A Simple yet Effective Method towards Cross-domain Graph Pretraining. SIGKDD 2024.

Cross-domain Graph Pre-training

- **Cross-domain ability is one of the key innovations in AGI (e.g., NLP and CV)**
  - Which pre-trains one foundation model using various contexts, absorbing cross-domain knowledge (‘All in One’).
  - Then, generalizes learned knowledge to a wide spectrum of downstream domains (‘One for All’).
- **Hard to replicate the success in the graph field remains.**
  - Which faces the negative transfer phenomenon.

Negative Transfer Phenomenon

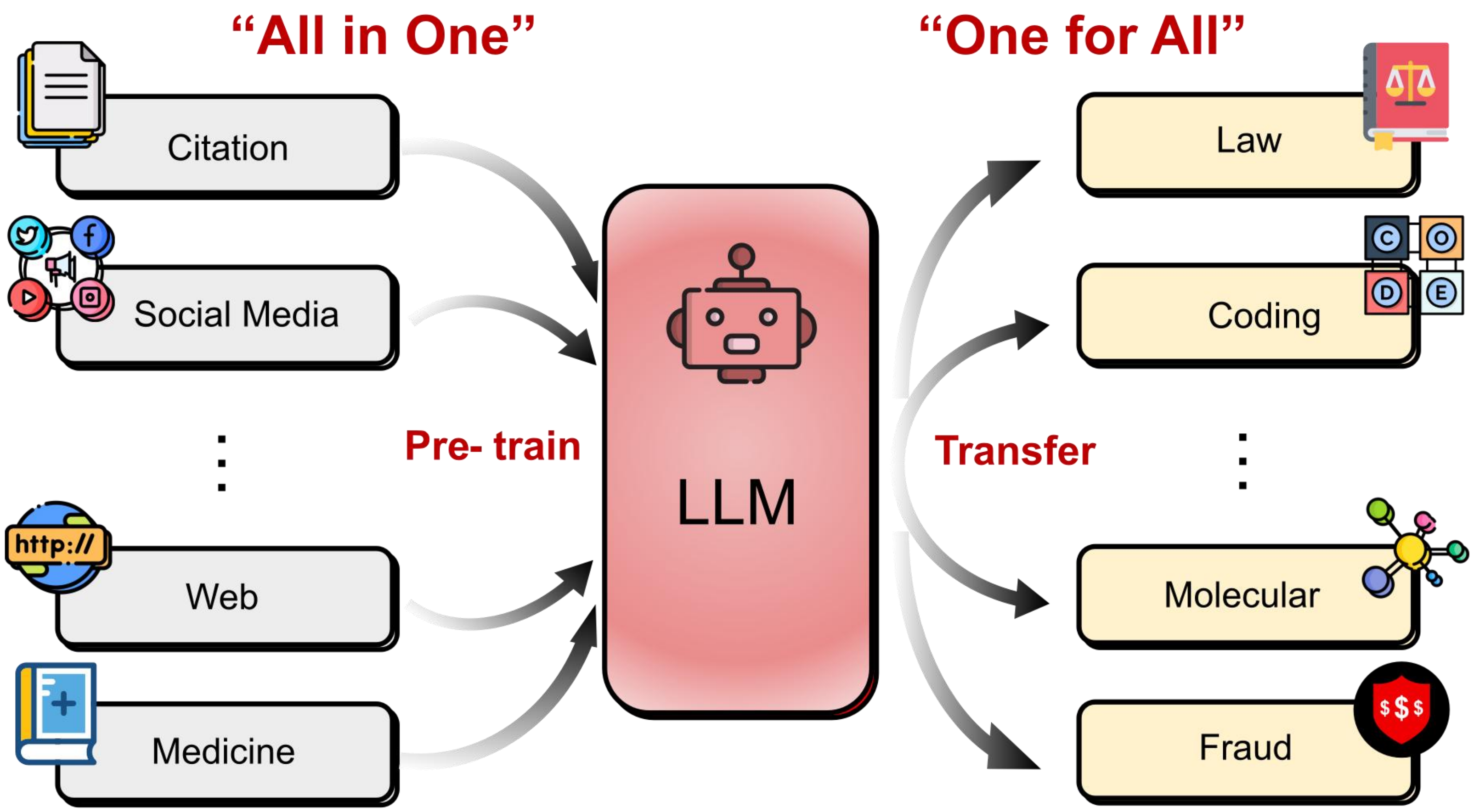
- **Homophilic Domain**
  - Source domain.
  - Pre-train on Pubmed or Photos.
- **Heterophilic Domain**
  - Target domain.
  - Transfer to Wisconsin, Texas, Cornell, Chameleon, or Squirrel.
- **Negative Transfer**
  - Hard to transfer across various domains via traditional pre-training approaches.



Negative transfer phenomenon in the single-source cross-domain transfer setting which is the traditional way to achieve transferring.

Motivation

➢ **Follow the pre–training paradigm in LLM**



Introducing the 'All in One and One for All' paradigm into the graph field like LLM

Challenge 1

- **Identifying and leveraging commonalities across domains is more intractable than LLMs during the pre-training phase**
  - The cross-domain training samples in NLP are all in text format, whereas the samples in graph fields are in **diverse structural patterns**, which is **particularly** observed between **homophilic** graphs (a pair of nodes are intended to be similar if they are connected) and **heterophilic** graphs (connected nodes depart from each other).

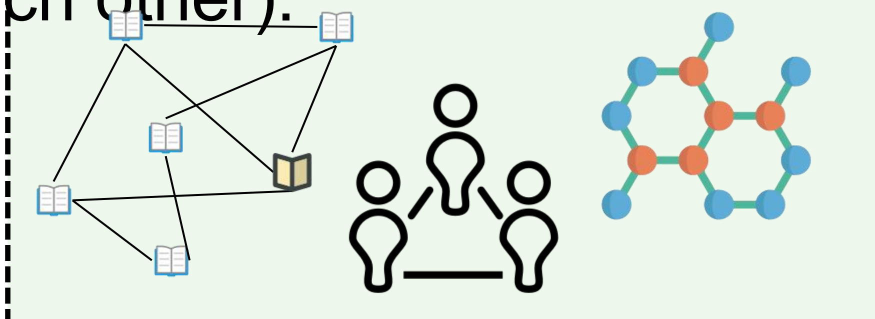
Just finished an amazing hike with stunning views!

The Eiffel Tower is a famous landmark in Paris, France.

⋮

The patient presents with symptoms consistent with a mild upper respiratory infection.

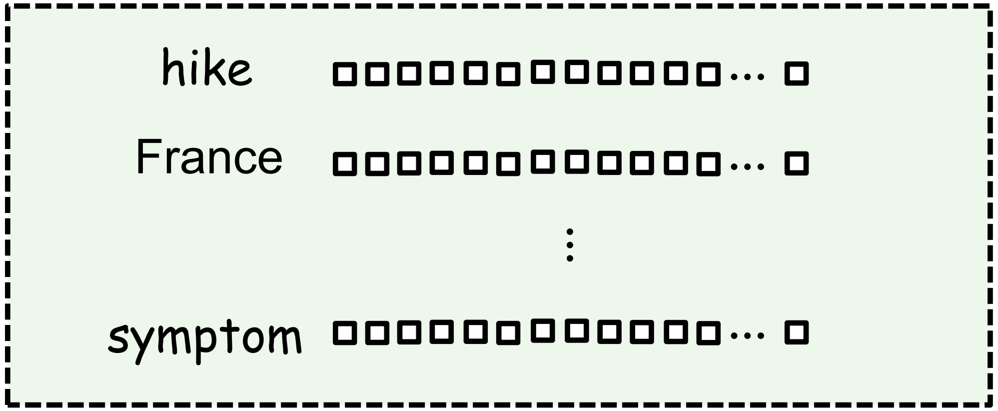
Text samples



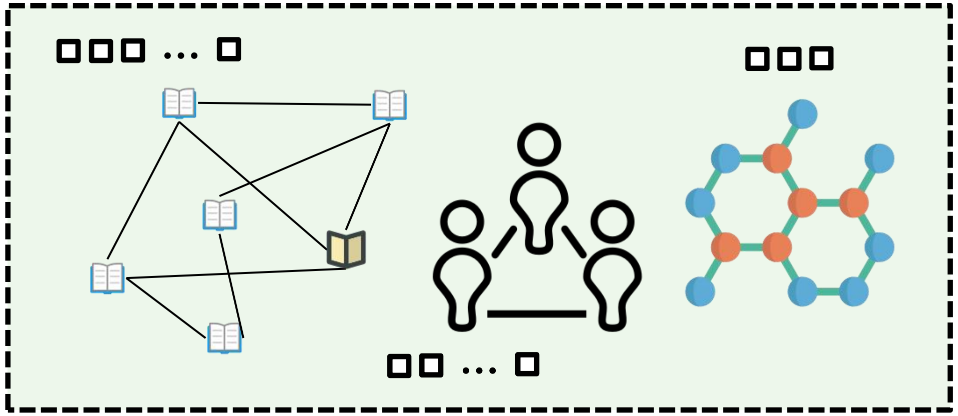
Graph samples

Challenge 2

- **Aligning semantic spaces (features) across graph datasets is more complex inherently in graph domains.**
  - Unlike the pure textual descriptions in NLP, in graph domains, many graphs are **not text-attributed or with specific feature semantics**. They have only latent feature vectors and we actually do not know how exactly each dimension means. Additionally, the **dimensions** are **far diversified**.



Unified feature spaces



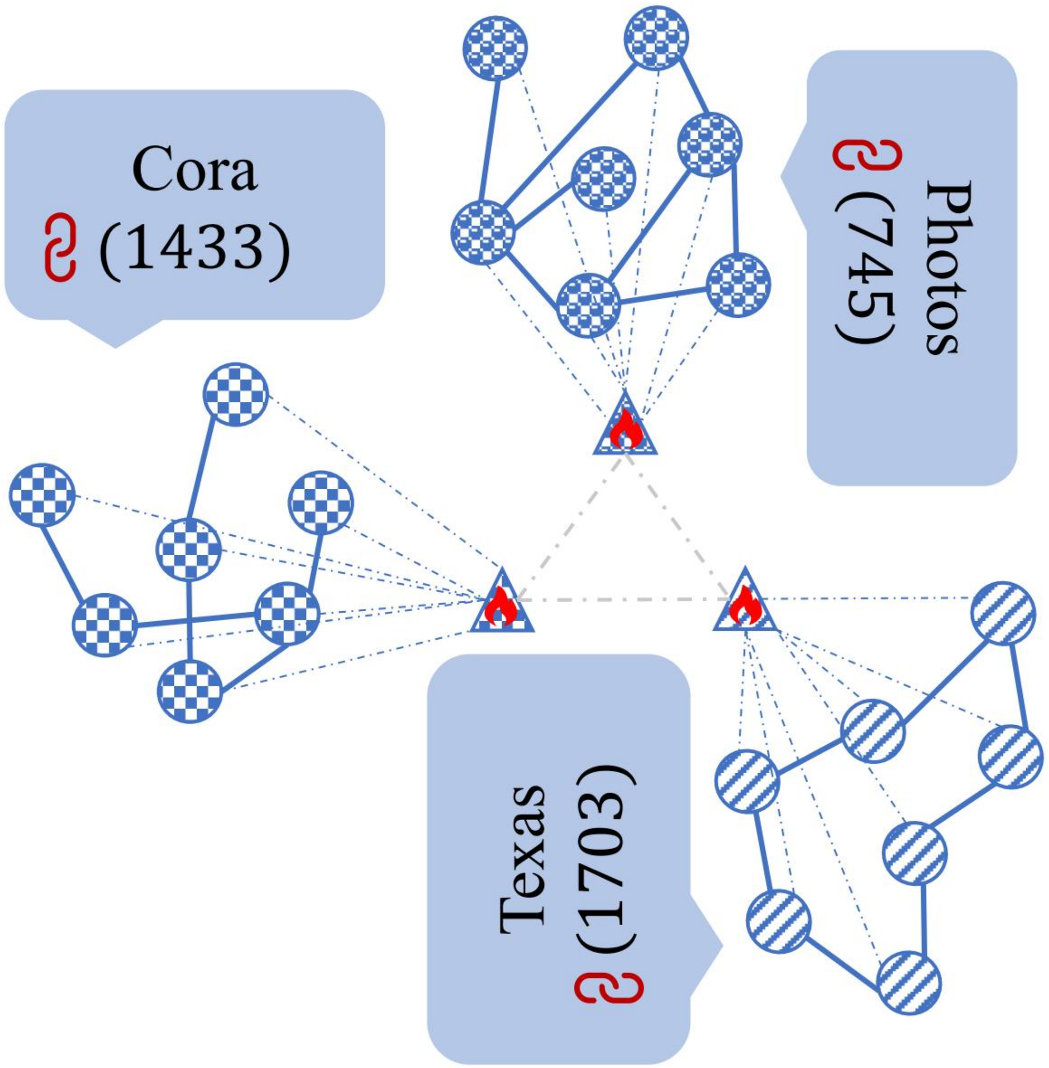
Diverse feature spaces

Our Solutions

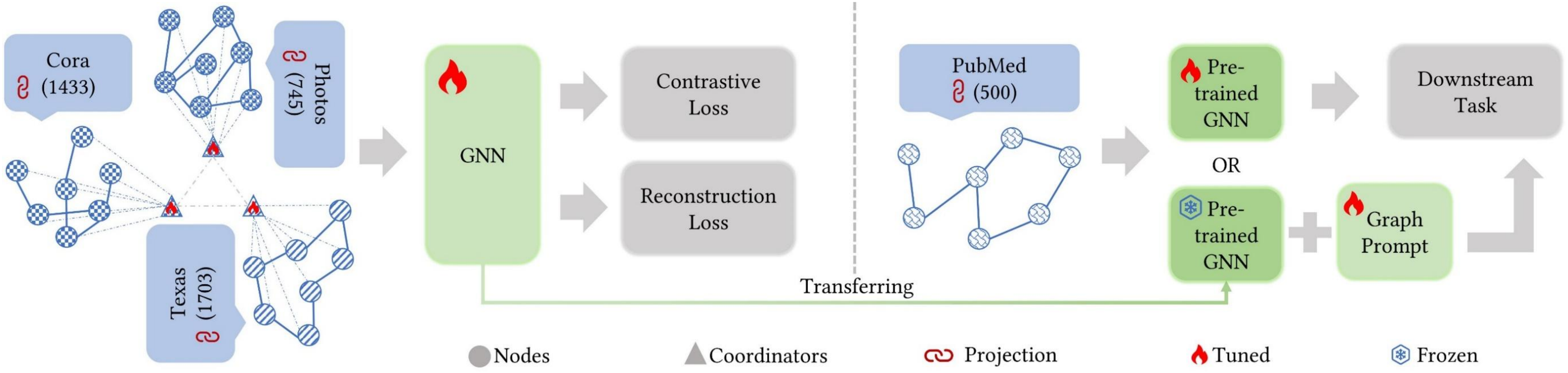
- We introduce the concept of "**coordinators**", which are some virtual nodes that function as dynamic bridges between disparate graph datasets, **prompting the integration across domains**.
- We design **a complete cross-domain pre-training framework** and provide two transferring components, which can ensure that the knowledge transferred is **not just relevant but also contextually enriched**.
- We carefully analyze **why our method works** and confirm the effectiveness of our method via extensive experiments.

Coordinators

- **Feature Projection**
  - Various features are aligned by a projecting module, such as 1433 -> 100, 745 -> 100, and 1703 -> 100.
- **Graph Coordinators**
  - Cross Connection between Coordinators and Datasets
  - Inner Connection within Coordinators



Unified Cross-domain Graph Pre-training



Based on carefully designed graph coordinators, we propose a complete cross-domain graph pre-training approach called **G**raph **CO**ordinators for **P**r**E**training (**GCOPE**), that harnesses the underlying commonalities across diverse graph datasets to enhance few-shot learning. Our novel methodology involves a unification framework that **amalgamates disparate graph datasets during the pretraining phase** to distill and transfer meaningful knowledge to target tasks.

Experiments

➤ Cross-domain Performance with Few-shot Learning Settings

An example:

Pretrain on:
- Cora
- Citeseer
- Pubmed
- Computers
- Photo
- Texas
- Cornell
- Chameleon
- Squirrel

Transfer to:
- Wisconsin

IMP (%) = Improvement Percentage

Training schemes	Methods	Cora Acc	Cora AUC	Cora F1	Citeseer Acc	Citeseer AUC	Citeseer F1	Pubmed Acc	Pubmed AUC	Pubmed F1	Computers Acc	Computers AUC	Computers F1	Photos Acc	Photos AUC	Photos F1
supervised	GCN	0.3012±.06	0.6444±.04	0.2591±.04	0.4358±.09	0.7234±.07	0.3583±.10	0.4210±.01	0.6040±.06	0.3026±.04	0.2602±.07	0.6773±.02	0.2428±.04	0.4603±.04	0.8458±.01	0.4592±.04
	GAT	0.3646±.04	0.6769±.03	0.3108±.04	0.3695±.05	0.7232±.06	0.3305±.04	0.4209±.04	0.5710±.06	0.3227±.07	0.3482±.07	0.6878±.05	0.2397±.05	0.4742±.08	0.8213±.02	0.4498±.07
	BWGNN	0.2543±.05	0.5563±.03	0.1971±.02	0.3599±.07	0.6954±.05	0.3112±.06	0.3976±.03	0.4934±.03	0.2686±.04	0.2768±.05	0.6273±.03	0.1864±.03	0.4113±.04	0.7769±.00	0.3883±.01
	FAGCN	0.3819±.03	0.6818±.04	0.3009±.09	0.5219±.08	0.8042±.03	0.4667±.08	0.4522±.02	0.5622±.04	0.4275±.07	0.4651±.04	0.7762±.02	0.3009±.07	0.5937±.05	0.8847±.00	0.5346±.03
IP + finetuning	GCL+GCN	0.2507±.06	0.6350±.03	0.2240±.03	0.3140±.02	0.6661±.04	0.2397±.02	0.4217±.02	0.5257±.05	0.2896±.07	0.2856±.04	0.6467±.03	0.1653±.06	0.5533±.01	0.8661±.01	0.5217±.01
	GCL+FAGCN	0.3892±.05	0.7228±.03	0.3619±.05	0.4461±.02	0.7781±.01	0.4126±.02	0.4532±.02	0.5708±.03	0.4168±.04	0.4371±.06	0.7616±.01	0.3450±.02	0.6273±.01	0.8710±.01	0.5406±.03
	Sim+GCN	0.2492±.02	0.5765±.03	0.1567±.04	0.2950±.06	0.6203±.06	0.1812±.06	0.3980±.01	0.5067±.02	0.2805±.01	0.2666±.10	0.6286±.01	0.1603±.03	0.4290±.04	0.7645±.02	0.3955±.02
	Sim+FAGCN	0.3957±.03	0.7284±.02	0.3585±.01	0.5101±.03	0.7969±.01	0.4615±.04	0.4398±.01	0.5535±.01	0.4225±.02	0.4393±.01	0.7718±.02	0.3100±.02	0.5704±.02	0.8543±.02	0.4984±.01
GCOPE + finetuning	GCL+GCN	0.3368±.02	0.6971±.04	0.2967±.03	0.3701±.03	0.7066±.02	0.3265±.05	0.4443±.04	0.5888±.04	0.4242±.04	0.3439±.03	0.7023±.01	0.2976±.03	0.5635±.02	0.8733±.00	0.5480±.02
	GCL+FAGCN	0.4618±.03	0.7597±.05	0.4388±.05	0.5631±.03	0.8258±.02	0.4953±.04	0.4591±.01	0.5512±.01	0.4203±.03	0.4465±.01	0.7747±.00	0.3432±.03	0.6329±.02	0.8850±.00	0.5935±.03
	Sim+GCN	0.2525±.05	0.5744±.03	0.1722±.06	0.3475±.05	0.6527±.05	0.2704±.05	0.4116±.00	0.5166±.04	0.2994±.03	0.3230±.01	0.6994±.00	0.2515±.00	0.4772±.03	0.7851±.01	0.4277±.02
	Sim+FAGCN	0.3875±.04	0.7163±.03	0.3355±.08	0.5704±.04	0.8425±.01	0.5178±.04	0.4727±.03	0.5587±.03	0.5672±.03	0.4677±.04	0.7875±.01	0.3823±.02	0.5985±.02	0.8757±.02	0.5556±.05
IMP (%)		11.23%	5.23%	14.63%	13.81%	4.26%	16.59%	5.02%	0.99%	25.32%	13.79%	6.28%	30.70%	10.31%	2.30%	12.18%

Training schemes	Methods	Wisconsin Acc	Wisconsin AUC	Wisconsin F1	Texas Acc	Texas AUC	Texas F1	Cornell Acc	Cornell AUC	Cornell F1	Chameleon Acc	Chameleon AUC	Chameleon F1	Squirrel Acc	Squirrel AUC	Squirrel F1
supervised	GCN	0.6290±.05	0.8320±.04	0.4871±.14	0.5812±.08	0.6731±.04	0.4557±.10	0.3263±.04	0.5666±.01	0.3151±.03	0.2393±.03	0.5310±.04	0.1923±.03	0.2093±.00	0.5263±.01	0.1889±.01
	GAT	0.6009±.02	0.8346±.01	0.5217±.05	0.6300±.08	0.5854±.08	0.4282±.13	0.3275±.14	0.5306±.03	0.1497±.04	0.2342±.02	0.5205±.04	0.1379±.03	0.2118±.00	0.5195±.02	0.1160±.01
	BWGNN	0.5620±.05	0.8463±.02	0.5189±.05	0.7438±.10	0.6642±.07	0.6274±.22	0.3150±.09	0.5938±.06	0.2190±.05	0.2206±.02	0.5039±.03	0.1540±.03	0.2155±.00	0.5149±.00	0.1664±.02
	FAGCN	0.5222±.05	0.7905±.0310	0.4725±.06	0.6900±.06	0.7185±.01	0.5334±.12	0.2938±.06	0.6573±.04	0.2872±.05	0.2575±.02	0.5515±.02	0.1941±.01	0.2181±.00	0.5202±.00	0.1875±.02
IP + finetuning	GCL+GCN	0.5249±.03	0.7876±.03	0.4415±.05	0.7350±.01	0.7210±.02	0.5636±.09	0.4175±.04	0.6350±.02	0.3500±.04	0.2249±.02	0.5213±.00	0.1432±.03	0.2118±.01	0.5059±.01	0.1110±.03
	GCL+FAGCN	0.6063±.04	0.8356±.01	0.5555±.07	0.7425±.03	0.7034±.03	0.6141±.09	0.2588±.04	0.6262±.04	0.2442±.04	0.2443±.00	0.5530±.01	0.1875±.01	0.2223±.00	0.5307±.00	0.1740±.02
	Sim+GCN	0.5258±.04	0.7927±.05	0.4604±.06	0.6338±.05	0.6024±.07	0.4269±.14	0.3438±.13	0.5954±.09	0.2168±.09	0.2271±.01	0.5183±.02	0.1578±.03	0.2133±.00	0.5133±.01	0.1550±.02
	Sim+FAGCN	0.6335±.02	0.8557±.00	0.5830±.04	0.6725±.14	0.6922±.04	0.5906±.10	0.2725±.05	0.6433±.04	0.2617±.04	0.2748±.01	0.5652±.00	0.2011±.00	0.2170±.00	0.5213±.00	0.1716±.01
GCOPE + finetuning	GCL+GCN	0.6606±.04	0.8487±.01	0.5952±.04	0.7738±.06	0.7387±.01	0.6763±.08	0.3975±.10	0.6694±.04	0.3120±.04	0.2411±.01	0.5564±.00	0.2210±.00	0.2245±.00	0.5207±.01	0.1741±.00
	GCL+FAGCN	0.6579±.03	0.8531±.01	0.5649±.00	0.7125±.02	0.6693±.02	0.6300±.03	0.4013±.05	0.6897±.01	0.3160±.02	0.2886±.00	0.5898±.00	0.2320±.00	0.2257±.00	0.5257±.00	0.1885±.01
	Sim+GCN	0.5412±.03	0.8059±.02	0.4509±.06	0.6137±.18	0.6900±.03	0.4674±.10	0.3675±.09	0.6045±.04	0.2339±.04	0.2573±.02	0.5467±.01	0.1852±.01	0.2180±.00	0.5147±.00	0.1783±.00
	Sim+FAGCN	0.7321±.00	0.9305±.00	0.6873±.01	0.7950±.03	0.7451±.01	0.7042±.03	0.5925±.01	0.8069±.03	0.4626±.03	0.2894±.01	0.5662±.02	0.2192±.02	0.2193±.00	0.5370±.00	0.1984±.01
IMP (%)		12.57%	4.58%	13.76%	6.65%	6.08%	16.87%	37.66%	14.29%	29.62%	11.97%	6.01%	25.36%	3.25%	1.06%	16.39%

Research Survey for Further Study

- **Graph Prompting Research**
  - Xiangguo Sun, Jiawen Zhang, Xixi Wu, Hong Cheng, Yun Xiong, Jia Li. Graph Prompt Learning: A Comprehensive Survey and Beyond. https://arxiv.org/abs/2311.16534
- **Graph Meets Large Language Model**
  - A Survey of Graph Meets Large Language Model: Progress and Future Directions. Survey paper at IJCAI2024.

More applications of graph prompt

➤ **Graph prompt for Protein Multimer Structure**

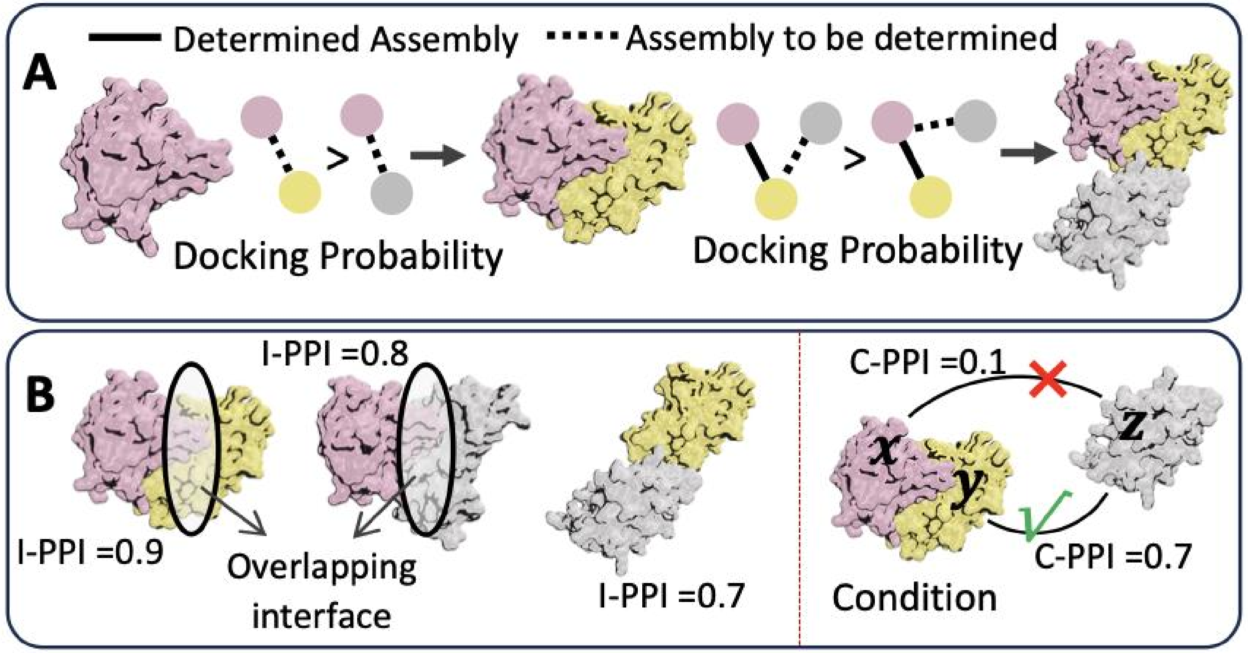


Figure 1: (**A**). Step-wise assembly for MSP. (**B**). Motivation for extending I-PPI to C-PPI.

Ziqi Gao, Xiangguo Sun, Zijing Liu, Yu Li, Hong Cheng, Jia Li. Protein Multimer Structure Prediction via Prompt Learning. ICLR 2024

More applications of graph prompt

➢ Graph prompt for Protein Multimer Structure

In Figure 10, we demonstrate that PromptMSP can successfully assemble unknown multimers, where no chain has a similarity higher than 40% with any chain in the training set.

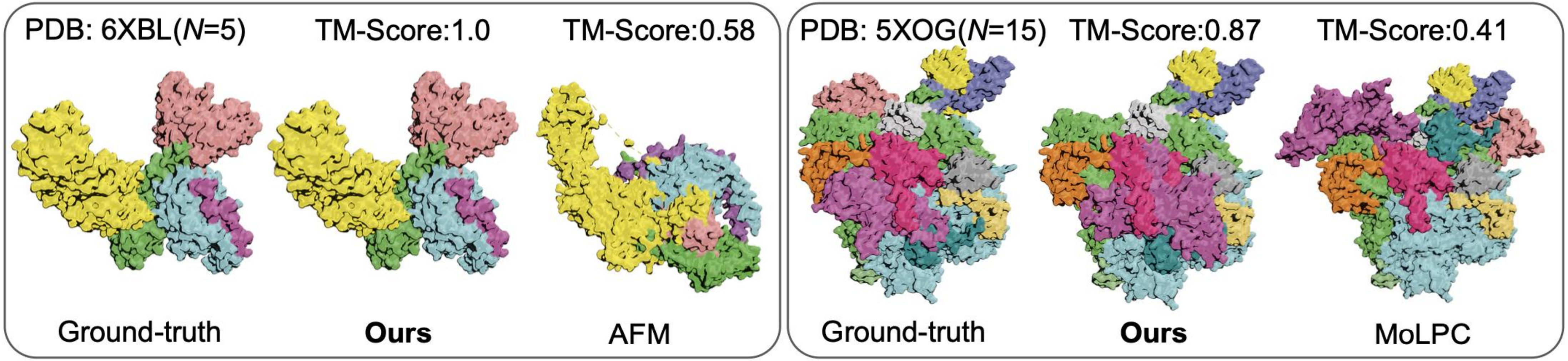
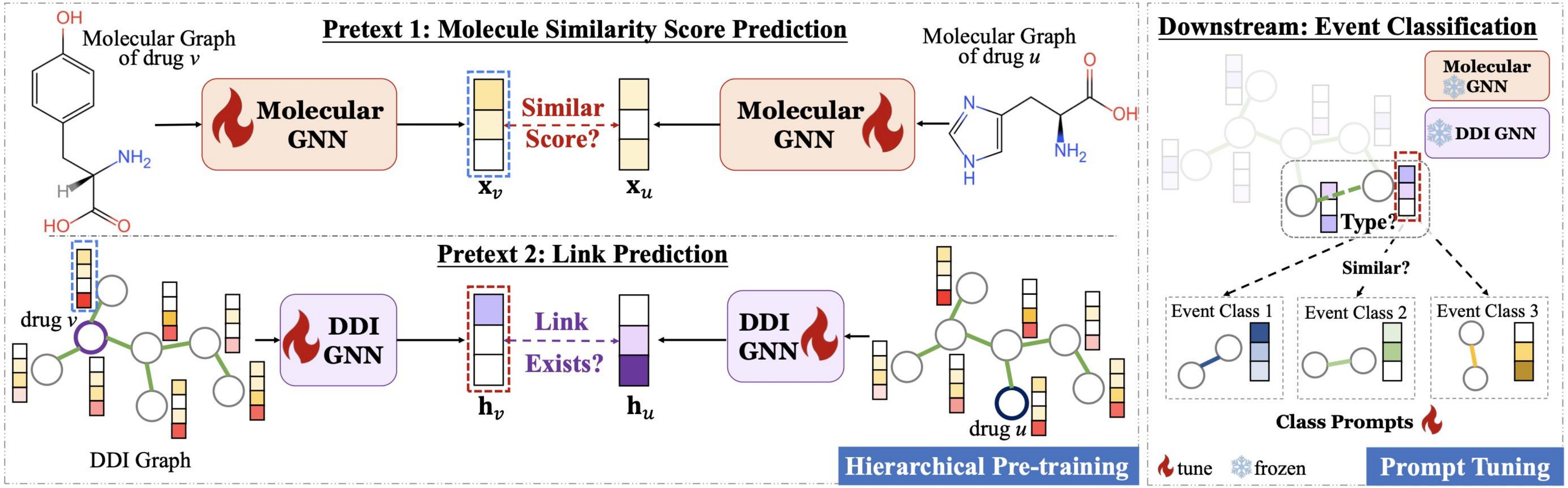


Figure 10: Visualization of multimers with chain numbers of 5 and 15. They are both successfully predicted by PROMPTMSP. For 5XOG, our model correctly predicted 12 out of 14 assembly actions.

Ziqi Gao, Xiangguo Sun, Zijing Liu, Yu Li, Hong Cheng, Jia Li. Protein Multimer Structure Prediction via Prompt Learning. ICLR 2024

More applications of graph prompt

➢ Graph prompt for Drug-Drug Interaction



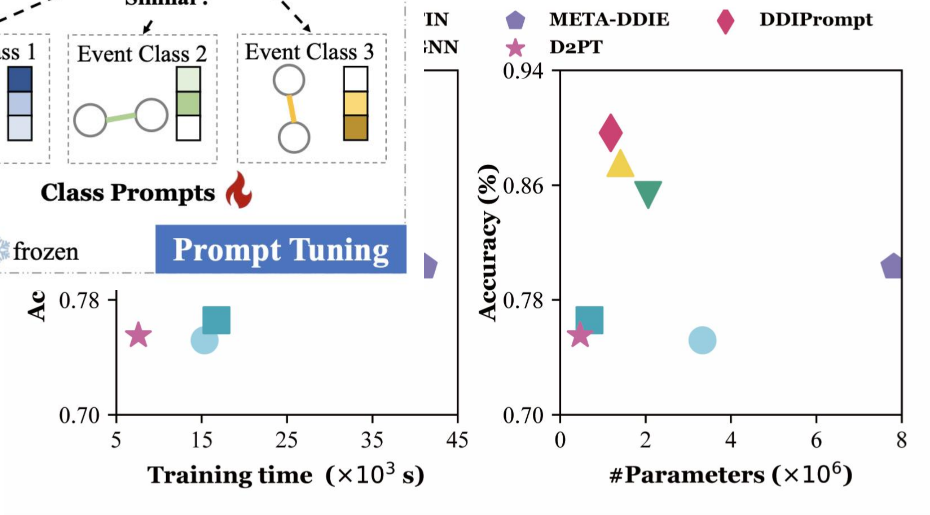
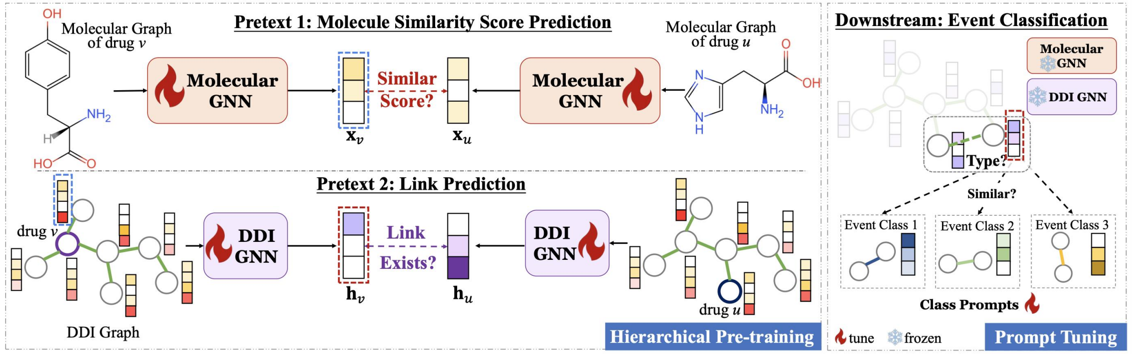


Figure 3: **Efficiency analysis on Ryu's dataset.**

Yingying Wang, Yun Xiong, Xixi Wu, Xiangguo Sun, Jiawei Zhang. DDIPrompt: Drug-Drug Interaction Event Prediction based on Graph Prompt Learning. CIKM 2024

More applications of graph prompt

➤ **Graph prompt for Drug-Drug Interaction**



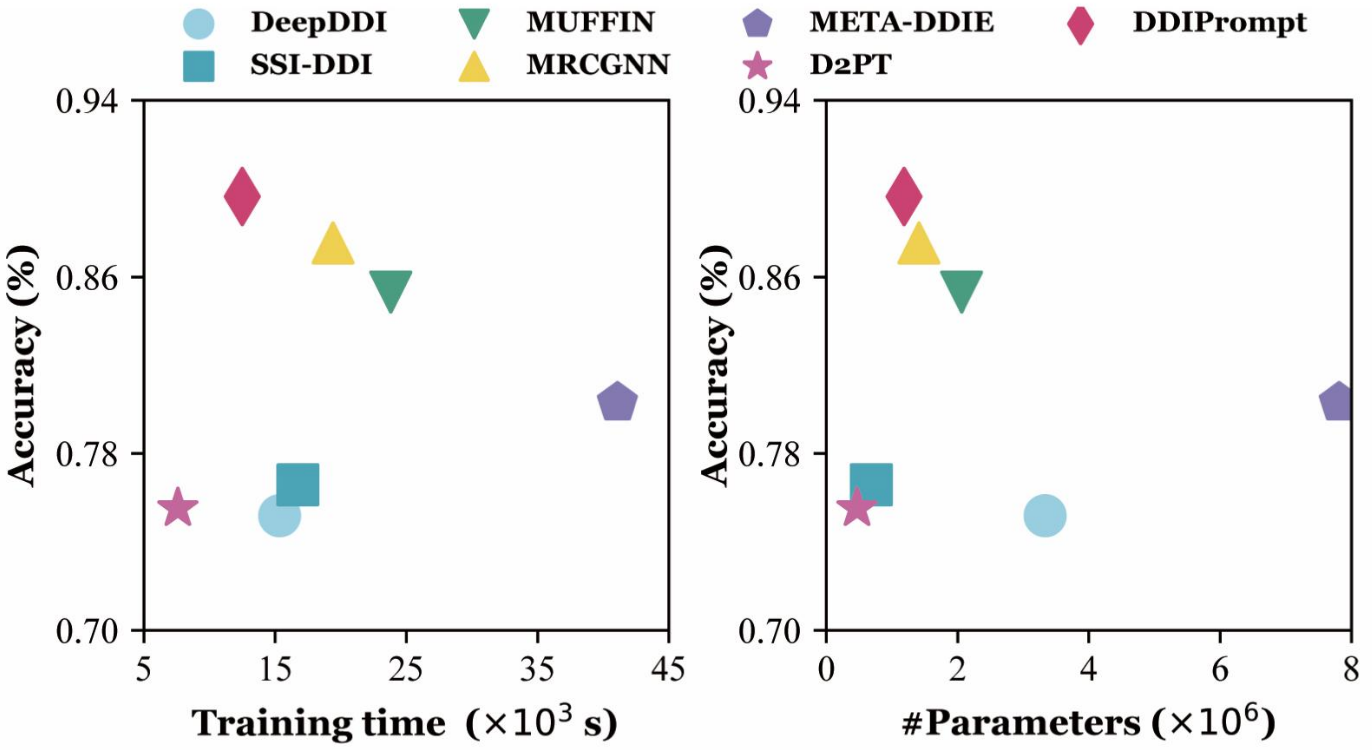


Figure 3: **Efficiency analysis on Ryu's dataset.**

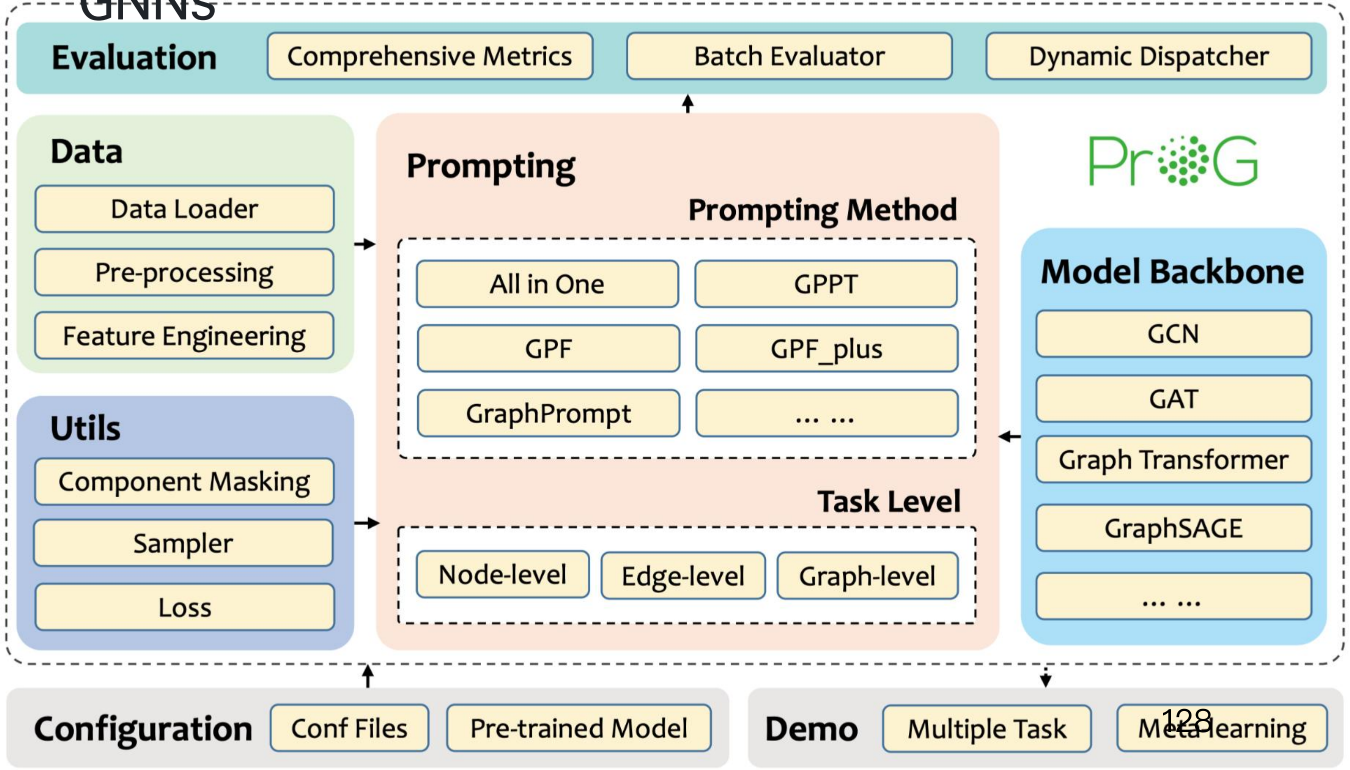
Yingying Wang, Yun Xiong, Xixi Wu, Xiangguo Sun, Jiawei Zhang. DDIPrompt: Drug-Drug Interaction Event Prediction based on Graph Prompt Learning. CIKM 2024

Graph Prompt Tool

- We develop a powerful tool to help researchers easily conduct various graph prompting approaches.

https://github.com/sheldonresearch/ProG

A library built upon PyTorch to easily conduct single or multi-task prompting for pre-trained GNNs



Graph Prompt Tool

```python
import prompt_graph as ProG
from ProG.pretrain import Edgepred_GPPT, Edgepred_Gprompt, GraphCL, SimGRACE, NodePrePrompt,
from ProG.utils import seed_everything
from ProG.utils import mkdir, get_args
from ProG.data import load4node,load4graph

args = get_args()
seed_everything(args.seed)


if args.task == 'SimGRACE':
    pt = SimGRACE(dataset_name = args.dataset_name, gnn_type = args.gnn_type, hid_dim = args
if args.task == 'GraphCL':
    pt = GraphCL(dataset_name = args.dataset_name, gnn_type = args.gnn_type, hid_dim = args.
if args.task == 'Edgepred_GPPT':
    pt = Edgepred_GPPT(dataset_name = args.dataset_name, gnn_type = args.gnn_type, hid_dim =
if args.task == 'Edgepred_Gprompt':
    pt = Edgepred_Gprompt(dataset_name = args.dataset_name, gnn_type = args.gnn_type, hid_di
if args.task == 'DGI':
    pt = DGI(dataset_name = args.dataset_name, gnn_type = args.gnn_type, hid_dim = args.hid_
if args.task == 'NodeMultiGprompt':
    nonlinearity = 'prelu'
    pt = NodePrePrompt(args.dataset_name, args.hid_dim, nonlinearity, 0.9, 0.9, 0.1, 0.001,
if args.task == 'GraphMultiGprompt':
    nonlinearity = 'prelu'
    pt = GraphPrePrompt(graph_list, input_dim, out_dim, args.dataset_name, args.hid_dim, nor
if args.task == 'GraphMAE':
    pt = GraphMAE(dataset_name = args.dataset_name, gnn_type = args.gnn_type, hid_dim = args
                  mask_rate=0.75, drop_edge_rate=0.0, replace_rate=0.1, loss_fn='sce', alpha
pt.pretrain()
```

Graph Prompt Tool

```
import prompt_graph as ProG
from ProG.tasker import NodeTask, LinkTask, GraphTask

if args.task == 'GraphTask':
    input_dim, output_dim, dataset = load4graph(args.dataset_name)

if args.task == 'NodeTask':
    tasker = NodeTask(pre_train_model_path = args.pre_train_model_path,
                    dataset_name = args.dataset_name, num_layer = args.num_layer,
                    gnn_type = args.gnn_type, hid_dim = args.hid_dim, prompt_type = args.prc
                    epochs = args.epochs, shot_num = args.shot_num, device=args.device, lr =
                    batch_size = args.batch_size, data = data, input_dim = input_dim, output


if args.task == 'GraphTask':
    tasker = GraphTask(pre_train_model_path = args.pre_train_model_path,
                    dataset_name = args.dataset_name, num_layer = args.num_layer, gnn_type =
                    shot_num = args.shot_num, device=args.device, lr = args.lr, wd = args.de
                    batch_size = args.batch_size, dataset = dataset, input_dim = input_dim,

_, test_acc, std_test_acc, f1, std_f1, roc, std_roc, _, _= tasker.run()
```

Graph Prompt Tool

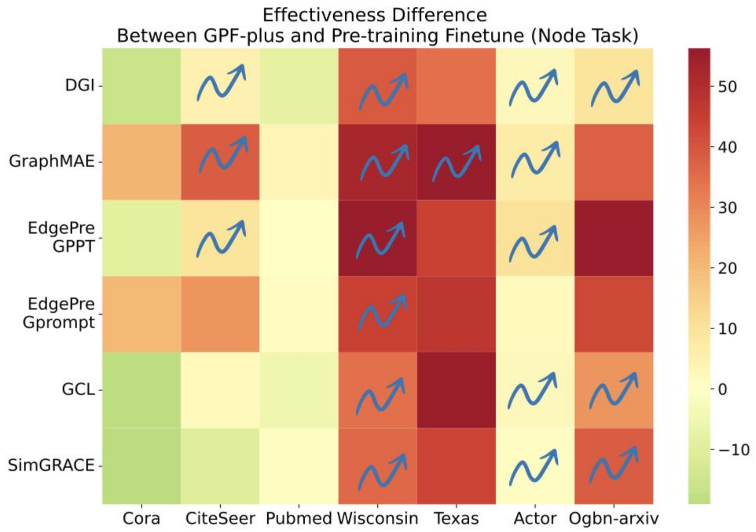
Supportive graph prompt approaches currently (keep updating):

- [All in One] X. Sun, H. Cheng, J. Li, B. Liu, and J. Guan, "All in One: Multi-Task Prompting for Graph Neural Networks," KDD, 2023
- [GPF Plus] T. Fang, Y. Zhang, Y. Yang, C. Wang, and L. Chen, "Universal Prompt Tuning for Graph Neural Networks," NeurIPS, 2023.
- [GraphPrompt] Liu Z, Yu X, Fang Y, et al. Graphprompt: Unifying pre-training and downstream tasks for graph neural networks. The Web Conference, 2023.
- [GPPT] M. Sun, K. Zhou, X. He, Y. Wang, and X. Wang, "GPPT: Graph Pre-Training and Prompt Tuning to Generalize Graph Neural Networks," KDD, 2022
- [GPF] T. Fang, Y. Zhang, Y. Yang, and C. Wang, "Prompt tuning for graph neural networks," arXiv preprint, 2022.

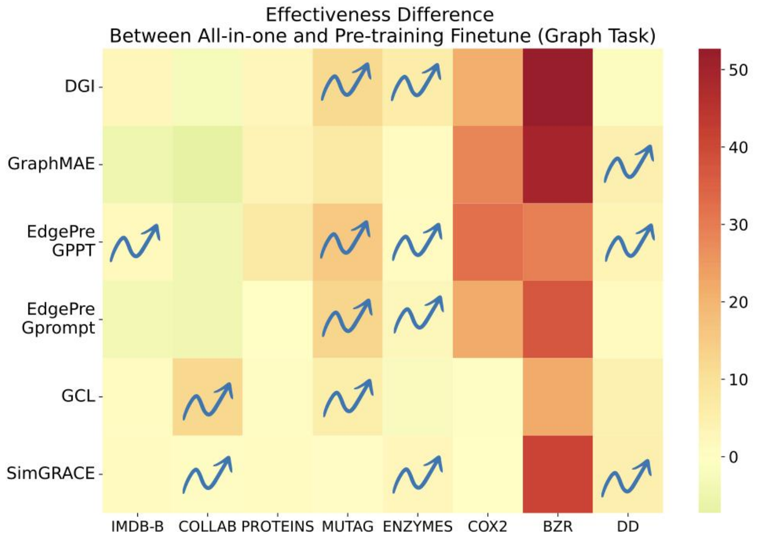
Graph Prompt Tool

Table 3: Performance on 1-shot graph classification. The best results for each dataset are highlighted in bold with a dark red background. The second-best are underlined with a light red background.

Methods\Datasets	IMDB-B	COLLAB	PROTEINS	MUTAG	ENZYMES	COX2	BZR	DD
Supervised	$57.30_{\pm 0.98}$	$47.23_{\pm 0.61}$	$56.36_{\pm 7.97}$	$65.20_{\pm 6.70}$	$20.58_{\pm 2.00}$	$27.08_{\pm 1.95}$	$25.80_{\pm 6.53}$	$55.33_{\pm 6.22}$
Pre-train & Fine-tune	$57.75_{\pm 1.22}$	$48.10_{\pm 0.23}$	$63.44_{\pm 3.64}$	$65.47_{\pm 5.89}$	$22.21_{\pm 2.79}$	$76.19_{\pm 5.41}$	$34.69_{\pm 8.50}$	$57.15_{\pm 4.32}$
GPPT	$50.15_{\pm 0.75}$	$47.18_{\pm 5.93}$	$60.92_{\pm 2.47}$	$60.40_{\pm 15.43}$	$21.29_{\pm 3.79}$	$78.23_{\pm 1.38}$	$59.32_{\pm 11.22}$	$57.69_{\pm 6.89}$
All-in-one	$60.07_{\pm 4.81}$	$51.66_{\pm 0.26}$	$66.49_{\pm 6.26}$	$79.87_{\pm 5.34}$	$23.96_{\pm 1.45}$	$76.14_{\pm 5.51}$	$79.20_{\pm 1.65}$	$59.72_{\pm 1.52}$
Gprompt	$54.75_{\pm 12.43}$	$48.25_{\pm 13.64}$	$59.17_{\pm 11.26}$	$73.60_{\pm 4.76}$	$22.29_{\pm 3.50}$	$54.64_{\pm 9.94}$	$55.43_{\pm 13.69}$	$57.81_{\pm 2.68}$
GPF	$59.65_{\pm 5.06}$	$47.42_{\pm 11.22}$	$63.91_{\pm 3.26}$	$68.40_{\pm 5.09}$	$22.00_{\pm 1.25}$	$65.79_{\pm 17.72}$	$71.67_{\pm 14.71}$	$59.36_{\pm 1.18}$
GPF-plus	$57.93_{\pm 1.62}$	$47.24_{\pm 0.29}$	$62.92_{\pm 2.78}$	$65.20_{\pm 6.04}$	$22.92_{\pm 1.64}$	$33.78_{\pm 1.52}$	$71.17_{\pm 14.92}$	$57.62_{\pm 2.42}$



(a) GPF-plus (1-shot node classification Task).



(b) All-in-one (1-shot graph classification Task).

Graph Prompt Tool

➢ **We released a repository for a comprehensive collection of research papers, datasets, other resources.**

README   MIT license

Awesome-Graph-Prompt

A collection of AWESOME things about performing prompt learning on Graphs.

awesome | Stars 298

Recently, the workflow of **"pre-train, fine-tune"** has been shown less effective and efficient when dealing with diverse downstream tasks on graph domain. Inspired by the prompt learning in natural language processing (NLP) domain, the **"pre-train, prompt"** workflow has emerged as a promising solution.

This repo aims to provide a curated list of research papers that explore the prompt learning on graphs. **It is based on our Survey Paper:** Graph Prompt Learning: A Comprehensive Survey and Beyond. We will try to make this list updated frequently. If you found any error or any missed paper, please don't hesitate to open issues or pull requests. 🌹

Table of Contents

Prompt with LLMs on graphs

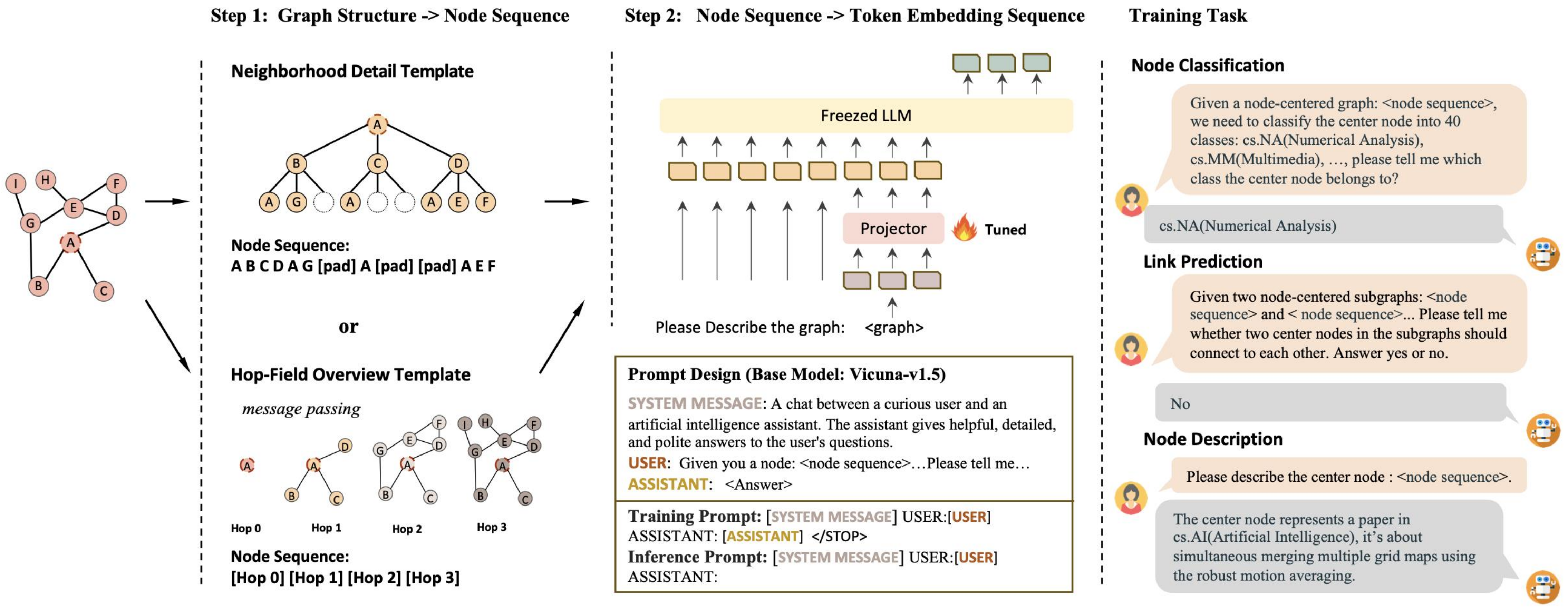


Figure 1. Illustration of LLaGA framework and its prompt design paradigm.

Chen R, Zhao T, Jaiswal A, et al. Llaga: Large language and graph assistant.ICML 2024.

Future Directions

- We are still waiting for "ChatGPT Moment" in graphs.
- How powerful is the graph prompt in manipulating data?
- How helpful is the graph prompt for more general graph model?

Q&A