

Graph Prompt

提示学习在图神经网络中的探索

报告人：刘瑜

报告时间：2023.11.03



目录

1. 概述
 - 背景和挑战
 - 研究路线
 - GNN+Prompt、LLM+Prompt、GNN+LLM+Prompt
2. GNN+Prompt
 - GraphPrompt -- WWW2023
3. LLM+Prompt
 - DeepGPT: Deep Graph Prompt Tuning -- arXiv 2023.09
4. GNN+LLM+Prompt
 - GNP: Graph Neural Prompt -- arXiv 2023.09

目录

1. 概述
 - 背景和挑战
 - 研究路线
 - GNN+Prompt、LLM+Prompt、GNN+LLM+Prompt
2. GNN+Prompt
 - GraphPrompt -- WWW2023
3. LLM+Prompt
 - DeepGPT: Deep Graph Prompt Tuning -- arXiv 2023.09
4. GNN+LLM+Prompt
 - GNP: Graph Neural Prompt -- arXiv 2023.09

1. 概述

1. 背景和挑战

● 背景

近年来，“预训练、提示、微调”已经成为NLP领域的标准范式。LLMs基本统一了NLP领域的不同任务。然而，由于图的独特性质，图“预训练、提示、微调”面临很多挑战，图¹的统一大模型有待探索。

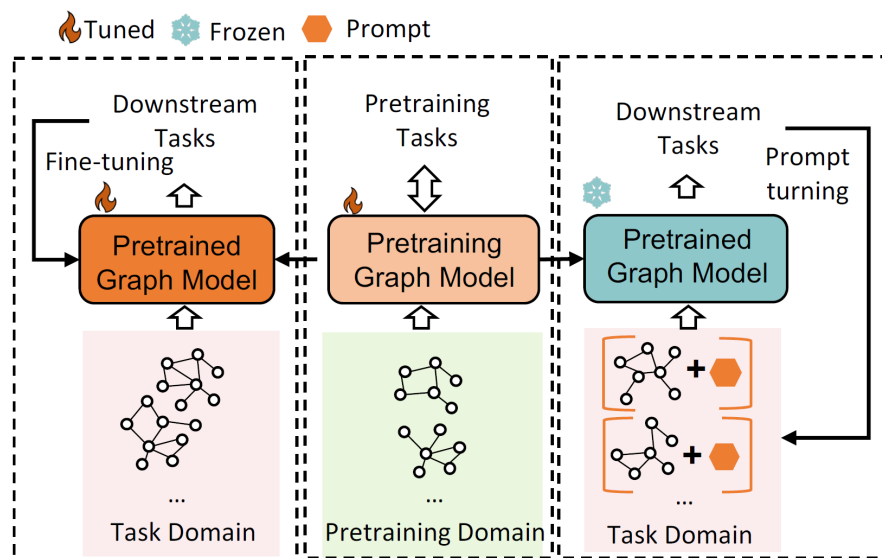


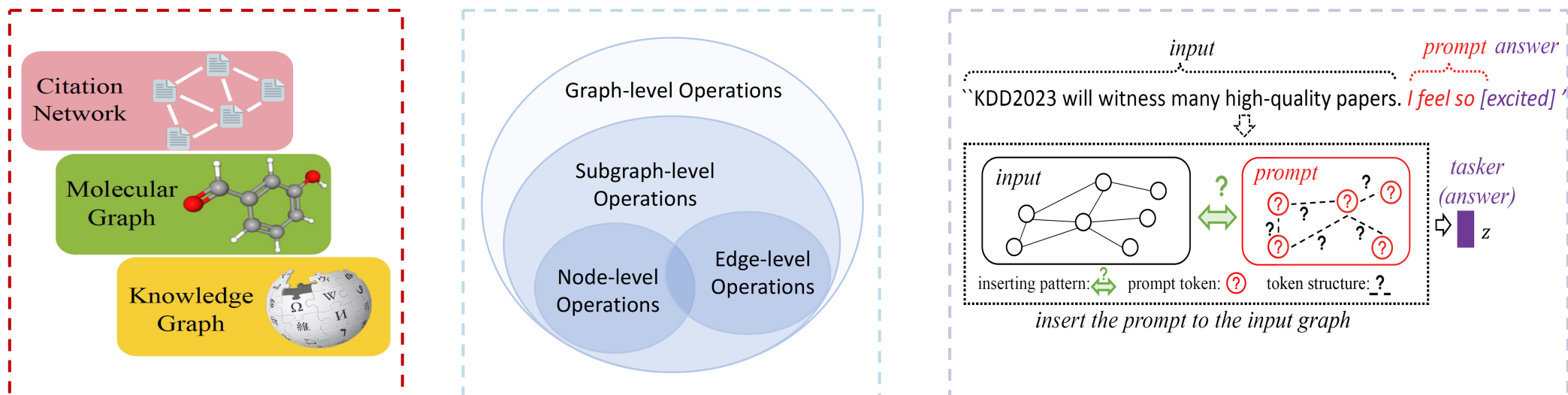
Figure 1: Fine-tuning, Pre-training, and Prompting.

1. 概述

1. 背景和挑战

● 挑战

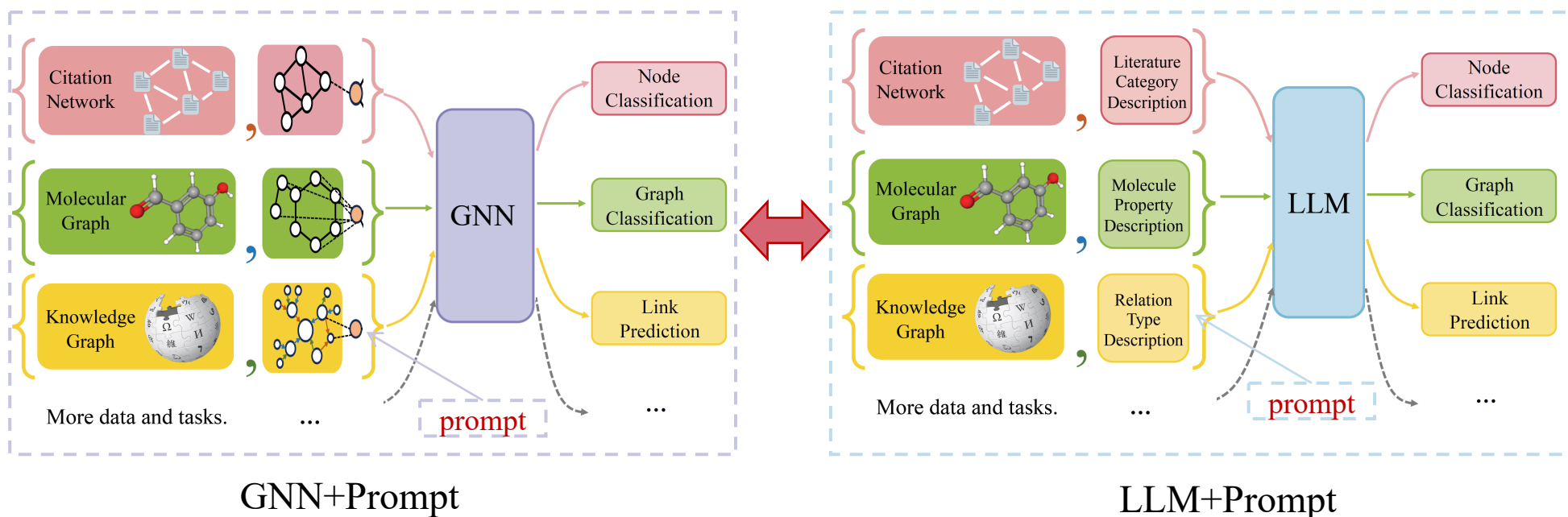
- 来自不同领域的图数据具有不同的结构和属性，如生物、化学和KG等。
- 图上的任务呈现多样化，包括节点级、边级和图级的分类和回归任务。
- 基于上下文的图提示学习模式尚不明确，处在初步探索阶段。



1. 概述

2. 研究路线

- ✓ GNN+Prompt: 基座模型采用GNN, 难点在于统一多种任务&设计Graph Prompt
- ✓ LLM+Prompt: 基座模型采用LLM, 重点在于设计Text Prompt
- ✓ GNN+LLM+Prompt: 两者结合, 互相帮助, GNN4LLM, LLM4GNN



One for All: Towards Training One Graph Model for All Classification Tasks

目录

1. 概述
 - 背景和挑战
 - 研究路线
 - GNN+Prompt、LLM+Prompt、GNN+LLM+Prompt
2. GNN+Prompt
 - GraphPrompt -- WWW2023
3. LLM+Prompt
 - DeepGPT: Deep Graph Prompt Tuning -- arXiv 2023.09
4. GNN+LLM+Prompt
 - GNP: Graph Neural Prompt -- arXiv 2023.09

2. GNN+Prompt

GRAPH PROMPT: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks

Zemin Liu*

National University of Singapore
Singapore
zeminliu@nus.edu.sg

Yuan Fang[†]

Singapore Management University
Singapore
yfang@smu.edu.sg

Xingtong Yu*

University of Science and Technology of China
China
yxt95@mail.ustc.edu.cn

Xinming Zhang[†]

University of Science and Technology of China
China
xinming@ustc.edu.cn

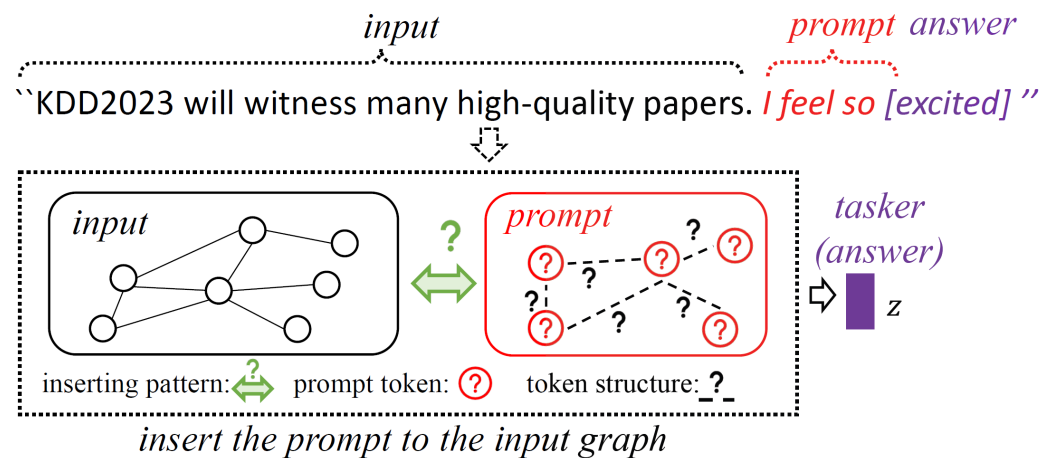
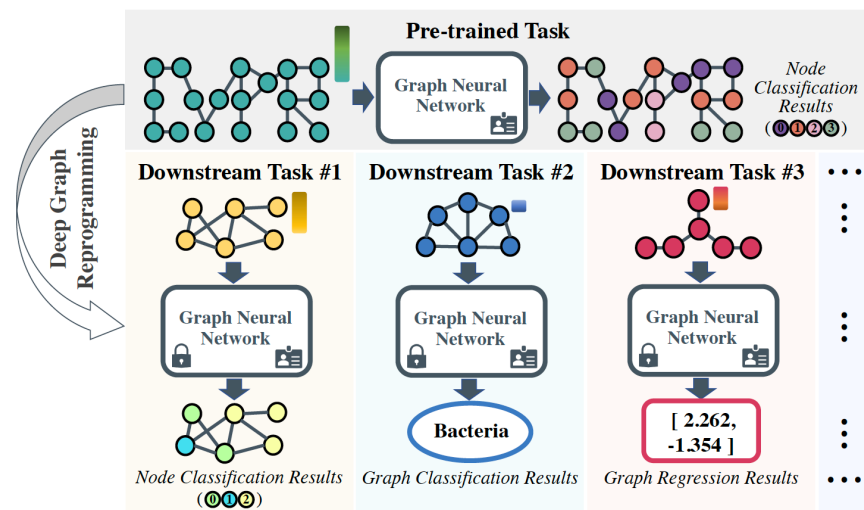
2. GraphPrompt

1. 动机

Pre-train & Fine-tune范式的预训练和下游任务目标不一致，该问题在图预训练上依旧存在。例如图的预训练任务主要是节点分类/边预测（二分类），而下游任务有节点/图分类（多分类）。

Prompt提示学习可以有效缩小两者的gap。GraphPrompt目前面临的两个问题：

1. 如何统一预训练和多种下游任务？
2. 如何在图上设计prompt，以便指导不同的下游任务有效地利用预训练的模型呢？



2. GraphPrompt

2. 创新

本文提出了**GraphPrompt**，一个在图上的 pretraining - prompting 框架。

问题1. 如何**统一**预训练和多种下游任务？

方法1. 将边级，节点级和图级任务统一到通用模板中：**图表示的相似性计算**。

问题2. 如何在图上**设计prompt**？

方法2. 引入一个**特定任务的learnable prompt**，引导下游任务利用预先训练的GNN模型中的相关知识。

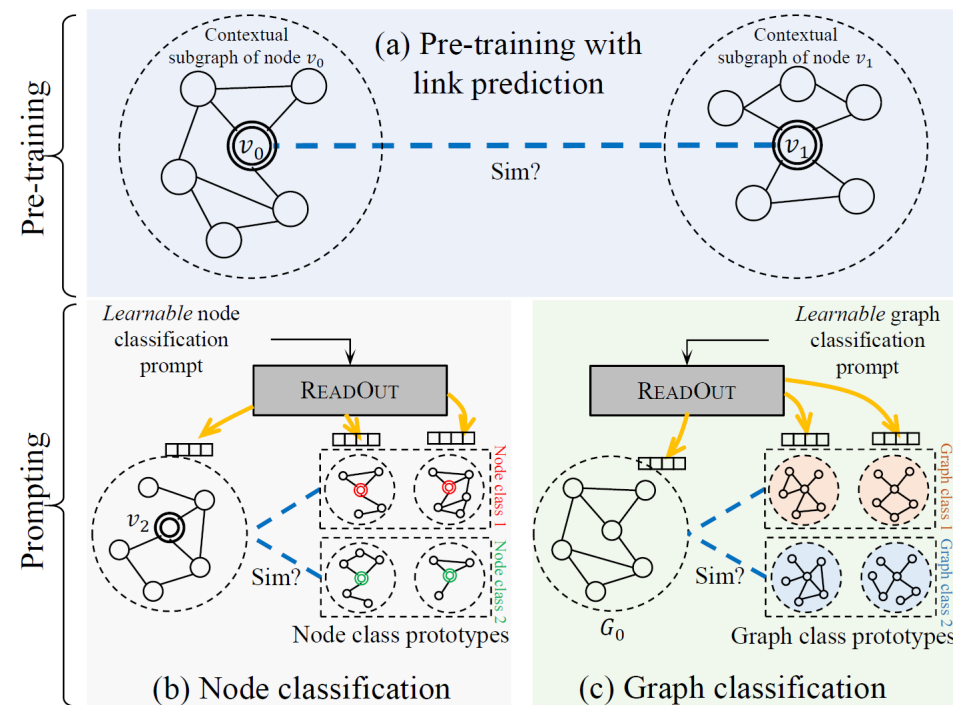
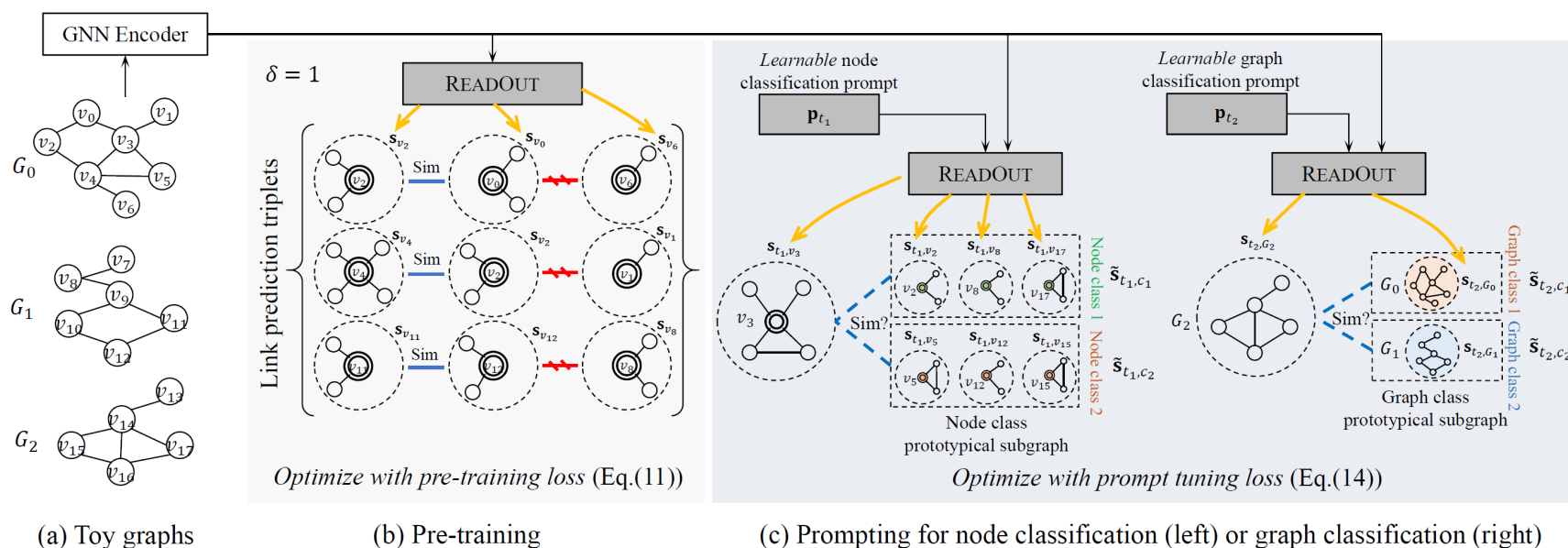


Figure 1: Illustration of the motivation. (a) Pre-training on graphs. (b/c) Downstream node/graph classification.

2. GraphPrompt

3. GraphPrompt模型

- ✓ 通用的任务模板：设计一个**通用模板**，来统一预训练和微调阶段的不同图任务
- ✓ Pre-Training阶段：采用边级的链接预测任务，来**预训练图模型**
- ✓ Prompt Tuning阶段：为区分不同的下游任务，在图上引入一个task-specific **learnable prompt**



2. GraphPrompt

模块一：设计通用模板

(1) 对于每个节点 v ，在原图上抽取一个上下文子图 S_v ，即常规局部子图抽取。

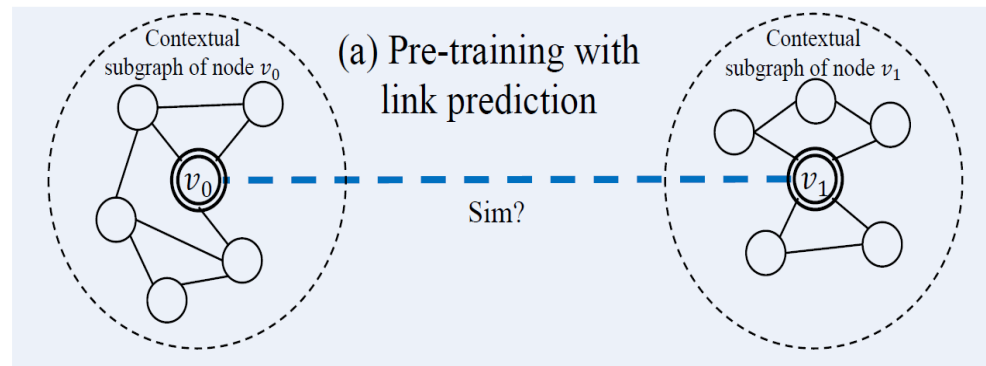
(2) 设计了一个通用模板，将边预测、节点分类和图分类任务重新定义为子图相似性学习，即计算两个子图的相似度。三个任务具体转化形式如下：

● 链接预测

给定一个节点对 (v, u) ，预测节点间是否有连边。
对于节点 v 和 u ，抽取两个子图 S_v 和 S_u ，得到子图表示 \mathbf{s}_v 和 \mathbf{s}_u ，计算两个图的相似度 $\text{sim}(\mathbf{s}_v, \mathbf{s}_u)$ 。

$$\mathbf{s}_x = \text{READOUT}(\{\mathbf{h}_v : v \in V(S_x)\}).$$

ReadOut 操作本质上是一个聚合函数，用于将子图中的节点表示融合为单个子图表示，例如sum，max，min，attention等。



2. GraphPrompt

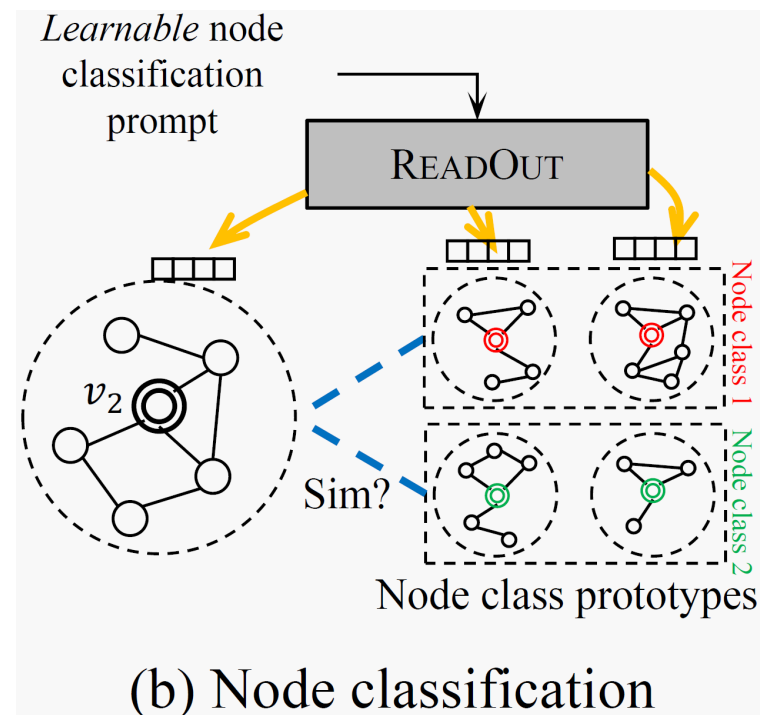
模块一：设计通用模板

● 节点分类

- 给定一个节点 v ，计算该节点的类别标签。
- 对于节点 v 抽取一个子图 s_v ，对于每个类别 $c \in C$ ，引入一个**节点类别原型子图** \tilde{s}_c (node class prototypical subgraph)，计算这两个图的相似度 $\text{sim}(s_v, \tilde{s}_c)$ 。

节点类别原型子图表示 \tilde{s}_c ，通过抽取 k 个<标签类别是 c >的节点，并将它们的子图的图表示聚合得到。

$$\tilde{s}_c = \frac{1}{k} \sum_{(v_i, \ell_i) \in D, \ell_i = c} s_{v_i}. \quad \ell_j = \arg \max_{c \in C} \text{sim}(s_{v_j}, \tilde{s}_c).$$



2.1 GraphPrompt

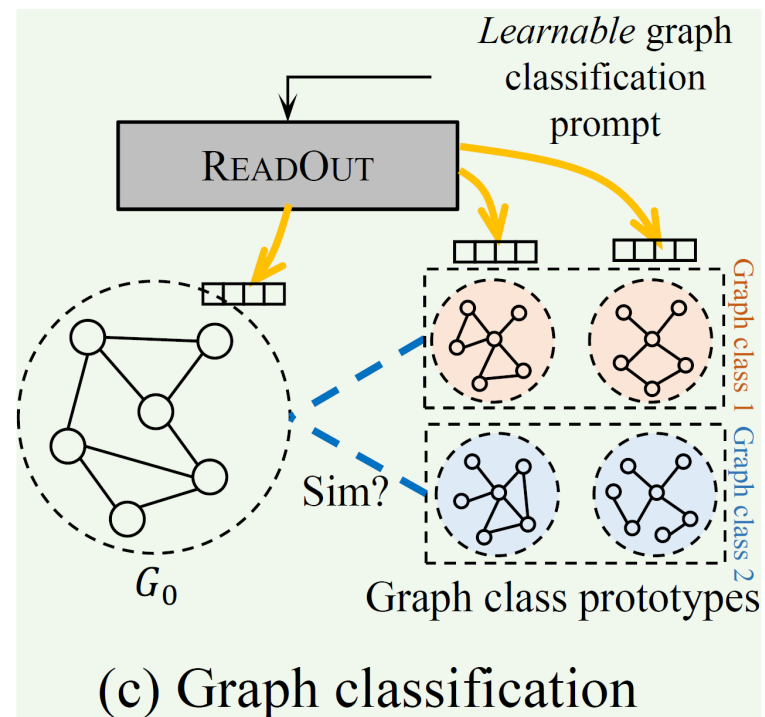
模块一：设计通用模板

- 图分类

给定一个图 G ，计算这个图的类别标签。
对于图 G ，抽取的子图就是图本身 $s_G = G$ 。对每个类别 $c \in \mathcal{C}$ ，引入一个图类别原型子图（**graph class prototypical subgraph**） \tilde{s}_c ，计算这两个图表示的相似度 $\text{sim}(s_G, \tilde{s}_c)$ 。

图类别原型子图表示 \tilde{s}_c ，通过抽取类别是 c 的 k 个图，并将它们的子图表示聚合得到。

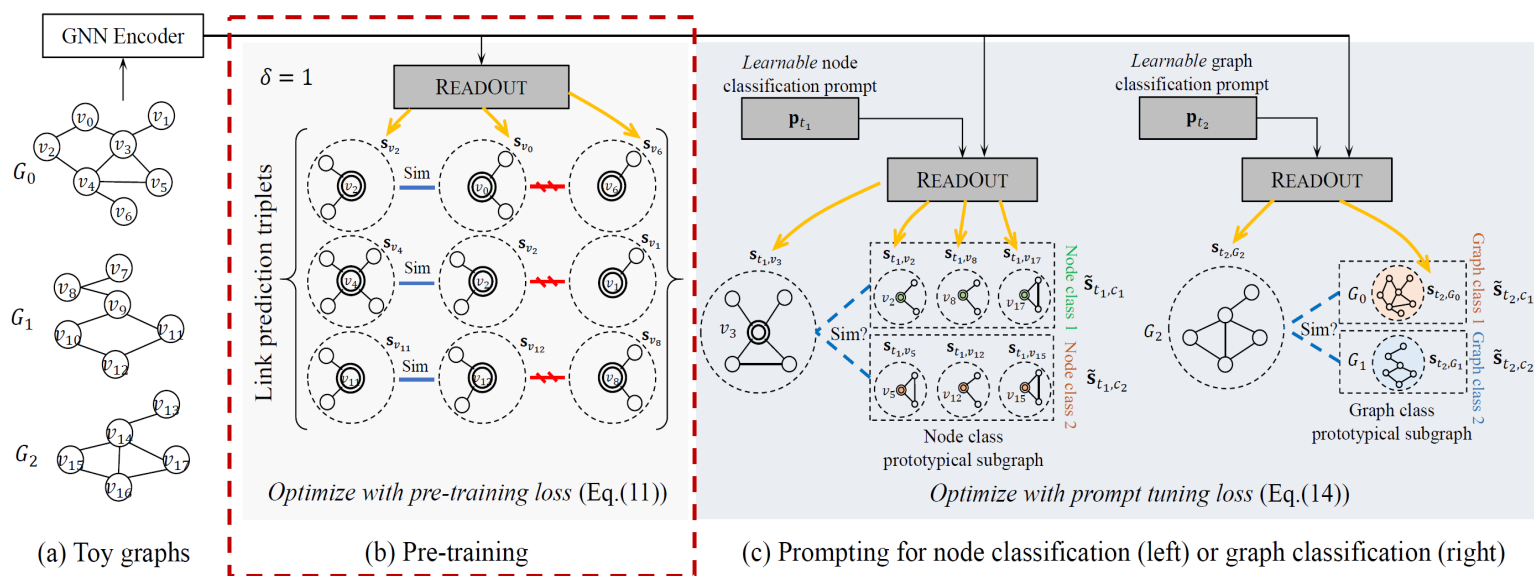
$$\tilde{s}_c = \frac{1}{k} \sum_{(G_i, L_i) \in \mathcal{D}, L_i=c} s_{G_i}. \quad L_j = \arg \max_{c \in \mathcal{C}} \text{sim}(s_{G_j}, \tilde{s}_c).$$



2. GraphPrompt

模块二：Pre-Training阶段

当统一了不同的图任务后，在预训练阶段，利用无标签的图数据，通过构造链接预测任务，来预训练图网络GNNs。给定图 G 的一个节点 v ，从节点 v 的邻居节点中采样节点 a 作为正样本，随机采样 b 作为负样本。



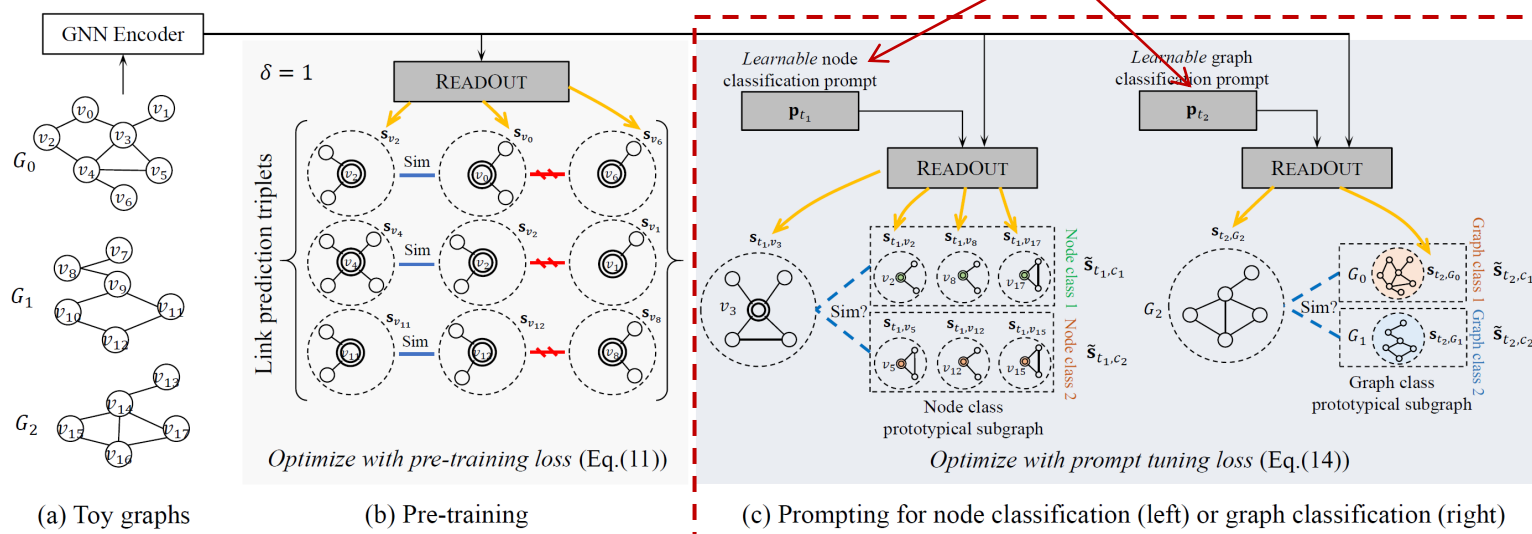
2. GraphPrompt

模块三：Prompt Tuning阶段

为了区分不同的下游任务，引入一个 **learnable prompt**，添加到图表示的计算中。具体地，对于一个下游任务 t ，将可学习的prompt记为 \mathbf{p}_t ，指导每个下游任务的ReadOut操作。

$$\mathbf{s}_x = \text{READOUT}(\{\mathbf{h}_v : v \in V(S_x)\}).$$

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



2. GraphPrompt

两个阶段的损失函数

预训练的损失函数如下：

$$\mathcal{L}_{\text{pre}}(\Theta) = - \sum_{(v,a,b) \in \mathcal{T}_{\text{pre}}} \ln \frac{\exp(\text{sim}(\mathbf{s}_v, \mathbf{s}_a)/\tau)}{\sum_{u \in \{a,b\}} \exp(\text{sim}(\mathbf{s}_v, \mathbf{s}_u)/\tau)},$$

下游任务的损失函数如下：

$$\mathcal{L}_{\text{prompt}}(\mathbf{p}_t) = - \sum_{(x_i, y_i) \in \mathcal{T}_t} \ln \frac{\exp(\text{sim}(\mathbf{s}_{t,x_i}, \tilde{\mathbf{s}}_{t,y_i})/\tau)}{\sum_{c \in Y} \exp(\text{sim}(\mathbf{s}_{t,x_i}, \tilde{\mathbf{s}}_{t,c})/\tau)},$$

2. GraphPrompt

4. 实验结果

● 数据集

- Flickr : 图片分析网络
- PROTEINS : 蛋白质网络
- COX2 : 分子结构图
- ENZYMES : 化学 (酶) 分子图
- BZR : 化学分子图

Table 1: Summary of datasets.

	Graphs	Graph classes	Avg. nodes	Avg. edges	Node features	Node classes	Task (N/G)
Flickr	1	-	89,250	899,756	500	7	N
PROTEINS	1,113	2	39.06	72.82	1	3	N, G
COX2	467	2	41.22	43.45	3	-	G
ENZYMES	600	6	32.63	62.14	18	3	N, G
BZR	405	2	35.75	38.36	3	-	G

● 基准模型

- 4个GNN方法 : GCN , GraphSAGE , GAT和GIN
- 3个图预训练模型 : DGI , InfoGraph , 和GraphCL
- 1个图提示模型 : GPPT

2. GraphPrompt

4. 实验结果

Table 2: Accuracy evaluation on **node classification**.

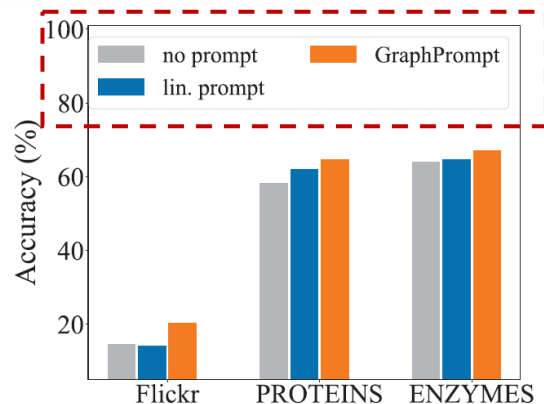
All tabular results are in percent, with best **bolded** and runner-up underlined.

Methods	Flickr 50-shot	PROTEINS 1-shot	ENZYMES 1-shot
GCN	9.22 ± 9.49	59.60 ± 12.44	61.49 ± 12.87
GRAPHSAGE	13.52 ± 11.28	59.12 ± 12.14	61.81 ± 13.19
GAT	16.02 ± 12.72	58.14 ± 12.05	60.77 ± 13.21
GIN	10.18 ± 5.41	<u>60.53</u> ± 12.19	<u>63.81</u> ± 11.28
DGI	17.71 ± 1.09	54.92 ± 18.46	63.33 ± 18.13
GRAPHCL	18.37 ± 1.72	52.00 ± 15.83	58.73 ± 16.47
GPPT	<u>18.95</u> ± 1.92	50.83 ± 16.56	53.79 ± 17.46
GRAPHPROMPT	20.21 ± 11.52	63.03 ± 12.14	67.04 ± 11.48

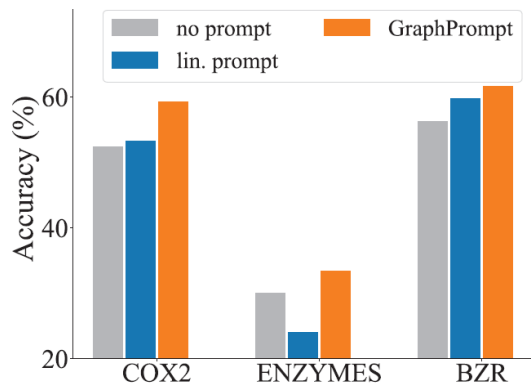
Table 3: Accuracy evaluation on **graph classification**.

Methods	PROTEINS 5-shot	COX2 5-shot	ENZYMES 5-shot	BZR 5-shot
GCN	54.87 ± 11.20	51.37 ± 11.06	20.37 ± 5.24	56.16 ± 11.07
GRAPHSAGE	52.99 ± 10.57	52.87 ± 11.46	18.31 ± 6.22	57.23 ± 10.95
GAT	48.78 ± 18.46	51.20 ± 27.93	15.90 ± 4.13	53.19 ± 20.61
GIN	<u>58.17</u> ± 8.58	51.89 ± 8.71	20.34 ± 5.01	57.45 ± 10.54
INFOGRAPH	54.12 ± 8.20	54.04 ± 9.45	20.90 ± 3.32	57.57 ± 9.93
GRAPHCL	56.38 ± 7.24	<u>55.40</u> ± 12.04	<u>28.11</u> ± 4.00	<u>59.22</u> ± 7.42
GRAPHPROMPT	64.42 ± 4.37	59.21 ± 6.82	31.45 ± 4.32	61.63 ± 7.68

2. GraphPrompt



(a) Node classification



(b) Graph classification

Figure 5: Ablation Study

Table 4: Study of parameter efficiency on node classification.

Methods	Flickr		PROTEINS		ENZYMES	
	Params	FLOPs	Params	FLOPs	Params	FLOPs
GIN	22,183	240,100	5,730	12,380	6,280	11,030
GPPT	4,096	4,582	1,536	1,659	1,536	1,659
GRAPHPROMPT	96	96	96	96	96	96
GRAPHPROMPT-ft	21,600	235,200	6,176	13,440	6,176	10,944

目录

1. 概述
 - 背景和挑战
 - 研究路线
 - GNN+Prompt、LLM+Prompt、GNN+LLM+Prompt
2. GNN+Prompt
 - GraphPrompt -- WWW2023
3. LLM+Prompt
 - DeepGPT: Deep Graph Prompt Tuning -- arXiv 2023.09
4. GNN+LLM+Prompt
 - GNP: Graph Neural Prompt -- arXiv 2023.09

3. LLM+Prompt

Deep Prompt Tuning for Graph Transformers

Reza Shirkavand,¹ Heng Huang,¹

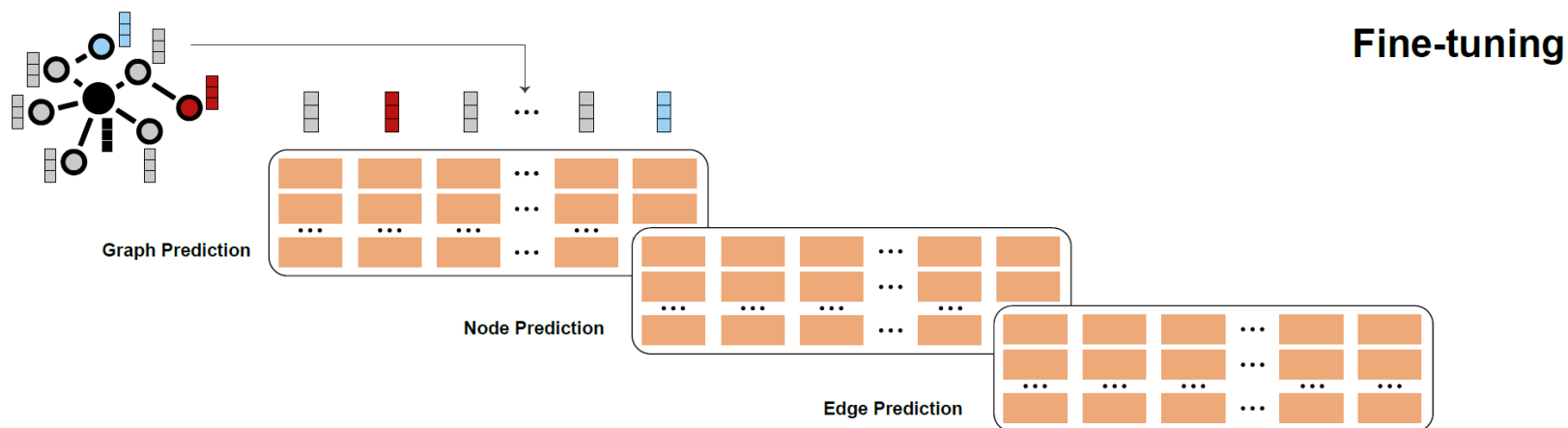
¹ University of Maryland
rezashkv@umd.edu, heng@umd.edu,

3. DeepGPT

1. 动机

随着Transformer在NLP领域取得的成功，Graph Transformers通过解决传统GNNs面临的挑战，如表达受限、过平滑等问题，在图任务上越来越受欢迎。

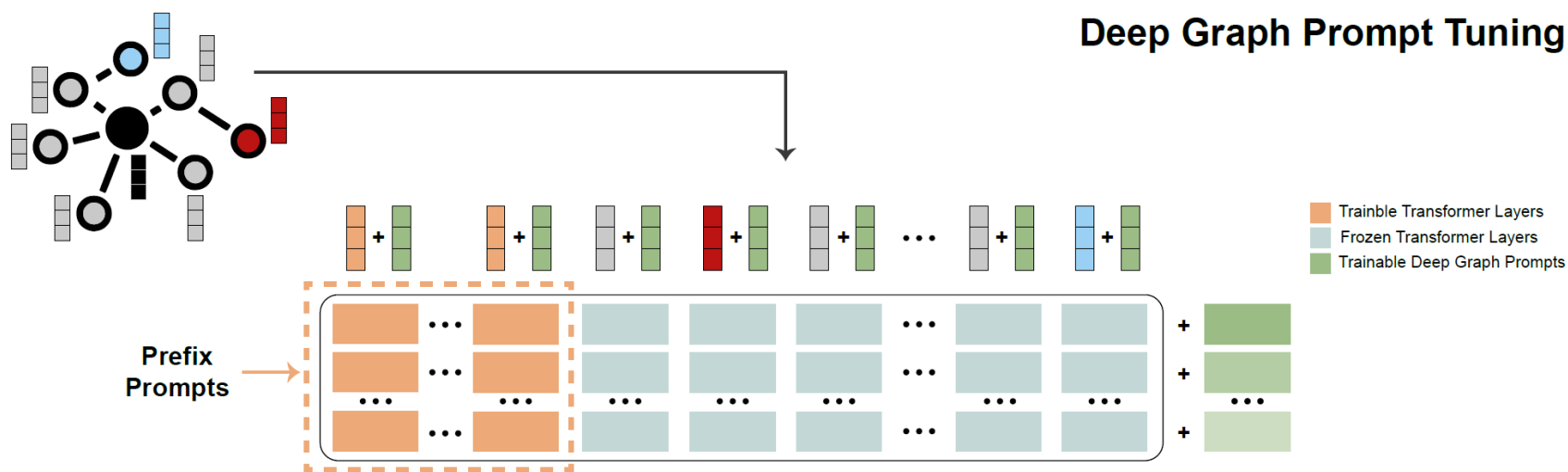
1. Transformer的多层叠加和self-attention的平方复杂度给图上的预测任务带来了挑战。
2. Fine-tuning既耗费数据资源又占用内存空间。相反，Prompt-tuning可以很好解决这些问题，已经称为NLP新的范式，但是这个技术在Graph transformer中尚待实验探索。



3. DeepGPT

2. 创新

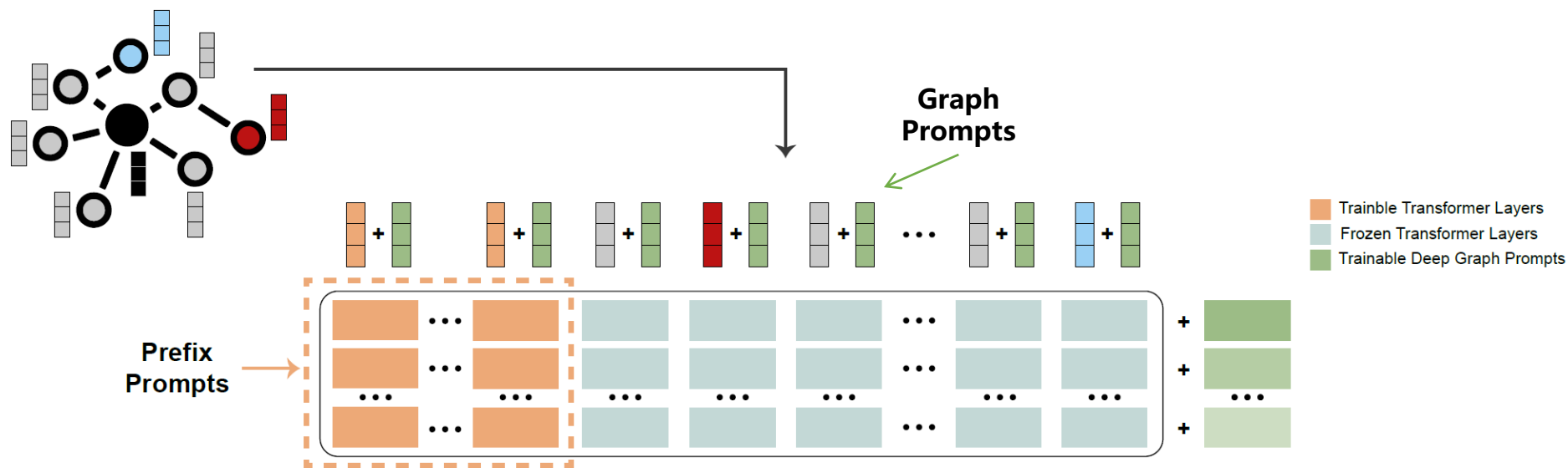
- ✓ 本文提出了DeepGPT，一种专门为Graph transformer设计的prompt tuning方法。
- ✓ 通过引入了可学习的graph prompts和prefix prompts（即soft prompt），在不改变其参数的情况下指导graph transformer。



3. DeepGPT

3. DeepGPT模型

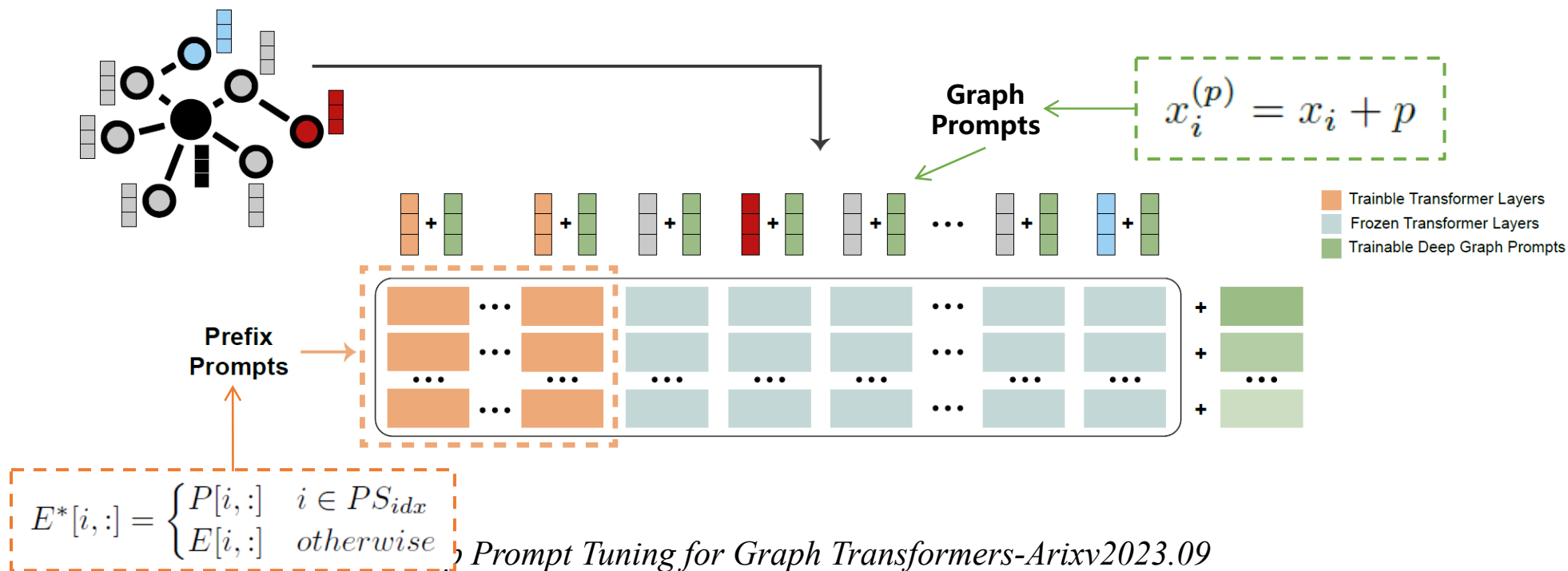
- ✓ Graph Prompts : 一个可学习的graph token , 添加到序列的每个节点上 (绿色)
- ✓ Prefix Prompts : 多个可学习的task-specific tokens , 添加到序列的前面 (黄色)



3. DeepGPT

3. DeepGPT模型

- ✓ Graph Prompts : 一个可学习的graph token , 添加到序列的每个节点上 (绿色)
- ✓ Prefix Prompts : 多个可学习的task-specific tokens , 添加到序列的前面 (黄色)



3. DeepGPT

4. 实验结果

● 数据集

- Pre-training : 在OGB-LSC数据集上训练模型, 量子化学回归数据集。
- Prompt Tuning : 在OGB多领域 (生物图、分子图、学术图、电商图和KG图) 和Moleculenet分子数据上。这两个数据集都包含了多个任务, 如回归、二分类或多分类任务。

● 基准模型

- 3个Graph Transformer方法 : Graphormer , GraphGPS和Line Graph Transformer (LiGhT)。
- 3个GNN方法 : GatedGCN , GINE和PNA

Table 5: Overview of graph-level prediction datasets used in this study.

Dataset	#Graphs	Prediction Task	Metric
BACE(Wu et al. 2018)	1513	binary classification	AUROC
BBBP(Wu et al. 2018)	2039	binary classification	AUROC
ClinTox(Wu et al. 2018)	1478	2-task classification	AUROC
Estrogen(Gaulton et al. 2012)	3122	2-task classification	AUROC
MetStab(Podlowska and Kafel 2018)	2267	2-task classification	AUROC
SIDER(Wu et al. 2018)	1427	27-task classification	AUROC
Tox21(Wu et al. 2018)	7831	12-task classification	AUROC
ToxCast(Wu et al. 2018)	8575	617-task classification	AUROC
ESOL(Wu et al. 2018)	1128	regression	RMSE
FreeSolv(Wu et al. 2018)	642	regression	RMSE
Lipophilicity(Wu et al. 2018)	4200	regression	RMSE
ogbg-molhiv(Hu et al. 2020a)	41127	binary classification	AUROC
ogbg-molpcba(Hu et al. 2020a)	437929	128-task classification	Average Precision
PCQM4Mv2(Hu et al. 2021)	3746620	regression	Mean Abs. Error

3. DeepGPT

4. 实验结果

Table 1: Comparison of the performance of DeepGPT, Fine Tuning (FT) and MPGNN baselines on classification benchmarks. Entries marked with an asterisk (*) indicate the use of Performer self-attention. Results represent the mean and standard deviation from 5-fold cross-validation.

Model	#Param.	Classification Dataset (\uparrow)							
		BACE	BBBP*	ClinTox*	Estrogen*	MetStab*	SIDER	ToxCast*	Tox21
Graphormer (Ying et al. 2021) FT	48M	0.704 \pm 0.018	0.778 \pm 0.043	0.839 \pm 0.114	0.673 \pm 0.034	0.607 \pm 0.015	0.550 \pm 0.007	0.665 \pm 0.032	0.582 \pm 0.051
Graphormer DeepGPT	100K	0.883 \pm 0.022	0.914 \pm 0.016	0.884 \pm 0.030	0.941 \pm 0.006	0.871 \pm 0.004	0.646 \pm 0.003	0.725 \pm 0.011	0.813 \pm 0.010
GraphGPS (Rampásek et al. 2022) FT	14M (*: 103M)	0.898 \pm 0.020	0.888 \pm 0.015	0.910 \pm 0.046	0.933 \pm 0.014	0.897 \pm 0.025	0.610 \pm 0.005	0.722 \pm 0.011	0.832 \pm 0.009
GraphGPS DeepGPT	50K (*: 500K)	0.892 \pm 0.022	0.901 \pm 0.015	0.907 \pm 0.044	0.944 \pm 0.010	0.899 \pm 0.019	0.609 \pm 0.020	0.735 \pm 0.005	0.832 \pm 0.012
LiGhT (Li, Zhao, and Zeng 2022) FT	90M	0.880 \pm 0.012	0.902 \pm 0.021	0.857 \pm 0.035	0.942 \pm 0.012	0.902 \pm 0.012	0.670 \pm 0.005	0.745 \pm 0.005	0.844 \pm 0.004
LiGhT DeepGPT	370K	0.873 \pm 0.020	0.917 \pm 0.012	0.862 \pm 0.056	0.950 \pm 0.010	0.912 \pm 0.011	0.671 \pm 0.011	0.757 \pm 0.011	0.843 \pm 0.004
GatedGCN (Bresson and Laurent 2017)	2.8M	0.833 \pm 0.013	0.887 \pm 0.025	0.893 \pm 0.041	0.919 \pm 0.008	0.848 \pm 0.018	0.599 \pm 0.015	0.683 \pm 0.005	0.807 \pm 0.011
GINE (Hu et al. 2020b)	1.2M	0.599 \pm 0.045	0.613 \pm 0.024	0.559 \pm 0.044	0.492 \pm 0.024	0.540 \pm 0.016	0.584 \pm 0.025	0.629 \pm 0.025	0.714 \pm 0.024
PNA (Corso et al. 2020)	1.8M	0.845 \pm 0.021	0.903 \pm 0.018	0.867 \pm 0.011	0.927 \pm 0.009	0.738 \pm 0.024	0.583 \pm 0.012	0.673 \pm 0.008	0.793 \pm 0.015

3. DeepGPT

4. 实验结果

Table 2: Comparison of the performance of DeepGPT, Fine Tuning (FT) and MPGNN baselines on Moleculenet regression benchmarks.

Model	Moleculenet Regression Dataset (\downarrow)		
	FreeSolv	ESOL	Lipo
Graphormer FT	1.680 ± 0.013	0.925 ± 0.061	0.909 ± 0.237
Graphormer DeepGPT	1.668 ± 0.114	0.943 ± 0.024	0.834 ± 0.046
GraphGPS FT	1.380 ± 0.234	0.772 ± 0.102	0.673 ± 0.022
GraphGPS DeepGPT	1.415 ± 0.254	0.685 ± 0.130	0.528 ± 0.056
LiGhT FT	0.955 ± 0.051	0.565 ± 0.041	0.530 ± 0.019
LiGhT DeepGPT	0.983 ± 0.079	0.579 ± 0.038	0.548 ± 0.028
GatedGCN	2.685 ± 1.766	1.373 ± 0.219	0.568 ± 0.065
GINE	6.971 ± 3.892	3.075 ± 1.034	1.440 ± 0.027
PNA	2.479 ± 0.556	1.463 ± 0.341	0.684 ± 0.024

Table 3: Comparison of the performance of DeepGPT, Fine Tuning (FT) and MPGNN baselines on OGB classification benchmark.

Model	OGB Classification Dataset (\uparrow)	
	MOLHIV (AUROC)	MOLPCBA (AP)
Graphormer FT	0.805 ± 0.005	0.313 ± 0.003
Graphormer DeepGPT	0.804 ± 0.021	0.289 ± 0.009
GraphGPS FT	0.806 ± 0.007	0.301 ± 0.013
GraphGPS DeepGPT	0.801 ± 0.015	0.297 ± 0.020
LiGhT FT	0.787 ± 0.008	0.295 ± 0.006
LiGhT DeepGPT	0.799 ± 0.010	0.270 ± 0.007
GatedGCN	0.809 ± 0.016	0.264 ± 0.021
GINE	0.679 ± 0.055	-
PNA	0.782 ± 0.013	0.257 ± 0.006

目录

1. 概述
 - 背景和挑战
 - 研究路线
 - GNN+Prompt、LLM+Prompt、GNN+LLM+Prompt
2. GNN+Prompt
 - GraphPrompt -- WWW2023
3. LLM+Prompt
 - DeepGPT: Deep Graph Prompt Tuning -- arXiv 2023.09
4. GNN+LLM+Prompt
 - GNP: Graph Neural Prompt -- arXiv 2023.09

4. GNN+LLM+Prompt

Graph Neural Prompting with Large Language Models

**Yijun Tian¹, Huan Song², Zichen Wang², Haozhu Wang²,
Ziqing Hu², Fang Wang², Nitesh V. Chawla¹, Panpan Xu²**

¹University of Notre Dame

²Amazon

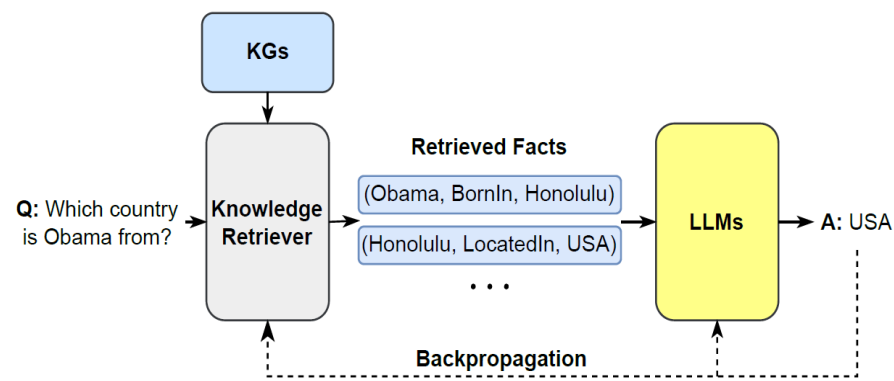
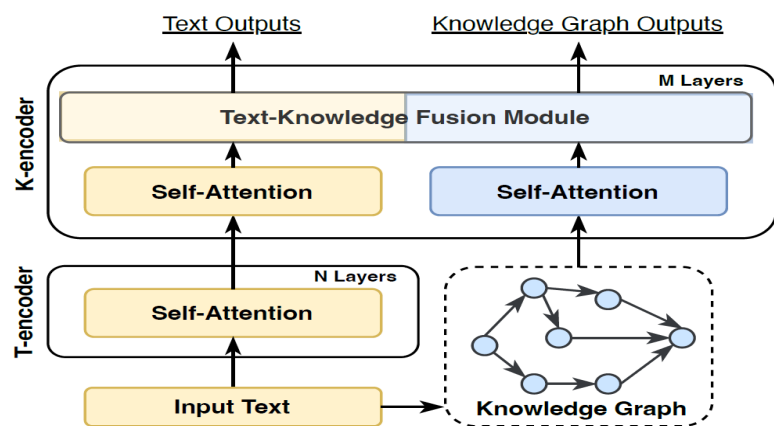
{yijun.tian, nchawla}@nd.edu, {huanso, zichewan, haozhuw, ziqinghu, fwfang, xupanpan}@amazon.com

4. GNP

1. 动机

LLMs在NLP领域展现了惊人的效果，但它在捕捉知识方面仍然表现出固有的局限性。

1. 先前工作通过联合训练和定制模型架构的方式，探索利用KG增强LLM的推理能力，但是庞大的模型参数会导致高昂的计算代价。
2. 如何利用预训练LLM+KG，并避免从头训练一个定制化模型仍然是一个悬而未决的问题。一种直接的方法是将KG中三元组作为LLM输入的一部分，但是这样容易引入噪声。



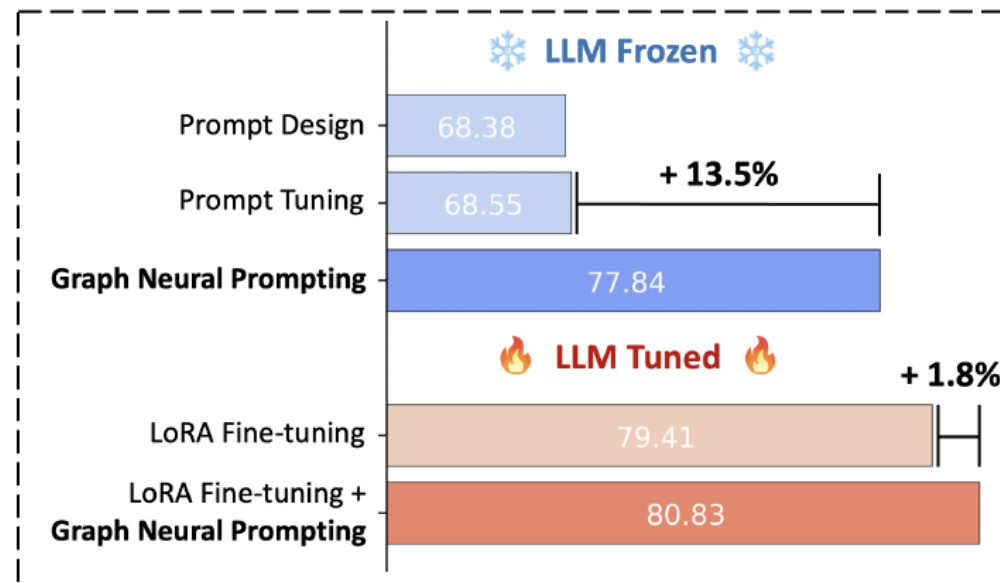
4. GNP

2. 创新

1. 本文提出了GNP, Graph Neural Prompting, 一个即插即用的方法, 利用GNN从KGs中学习有益的知识并融入LLMs。

2. GNP将文本信息和结构信息编码得到子图表示, 将其视为graph prompting, 然后拼接到LLM的输入前面, 以提供指导instructions。

3. 实验结果显示, 当LLM被冻结时, GNP将基线提高了+13.5%。此外, 通过LoRA方法微调LLM, GNP提升了+1.8%。



4. GNP

3. 问题定义

- Multiple Choice QA

给定一个问题 Q ，候选答案集合 $A = \{a_k\}_{k=1}^K$ 和一个可选的上下文 C ，该问答任务是设计一个模型 \mathcal{F}_θ 从候选答案中选择正确答案。

- Prompting LLMs for QA

一个通用prompt的QA框架是，将问题 Q ，上下文 C 和候选答案 A 拼接成文本序列 X ，设计一系列提示 P 并将其添加到 X 前面，然后利用LLM模型生成预测结果。

$$y' = f([P, X])$$

Question:


What is the best way to guess a babies eye color?

Options:

- (a) The surroundings they are born in
- (b) Their parents usual diet
- (c) Just take a random guess

(d) The genealogy records of their family

Model Prediction:

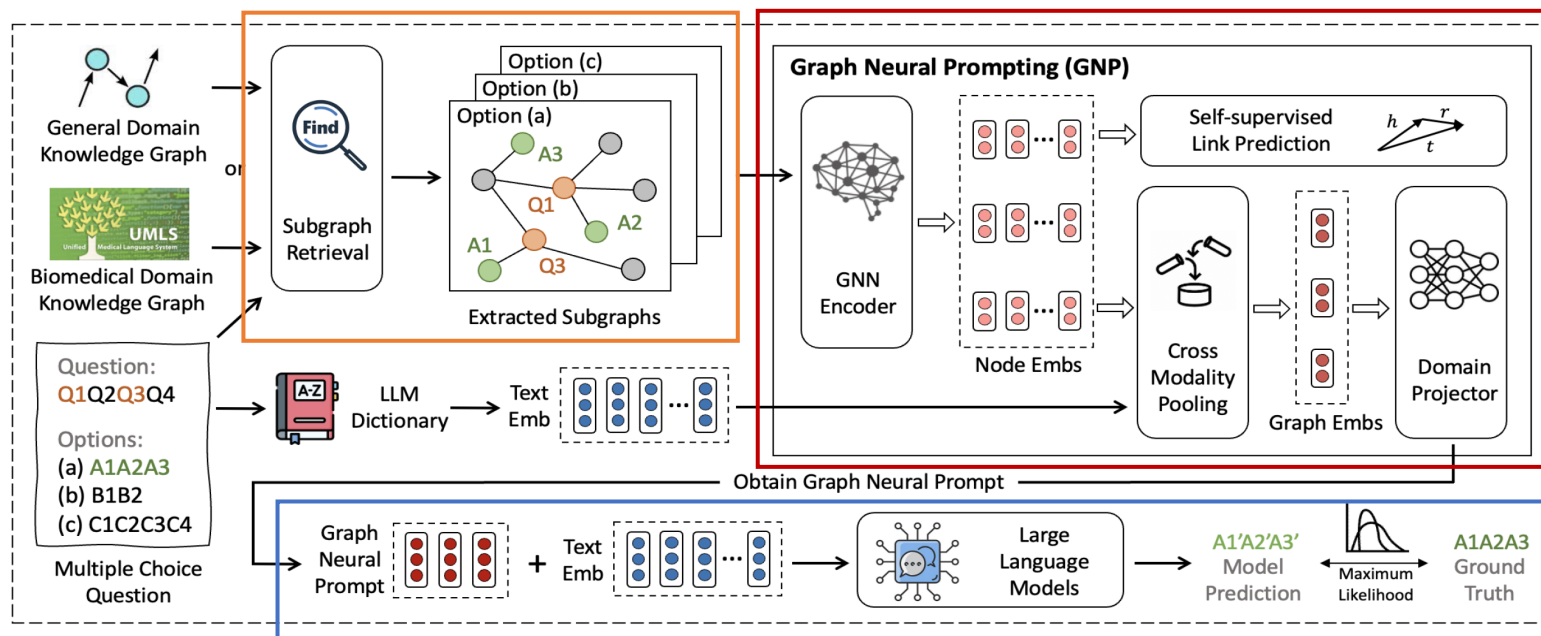
Prompt Tuning: (c) Just take ... 

GNP: (d) The genealogy ... 

4. GNP

4. GNP模型

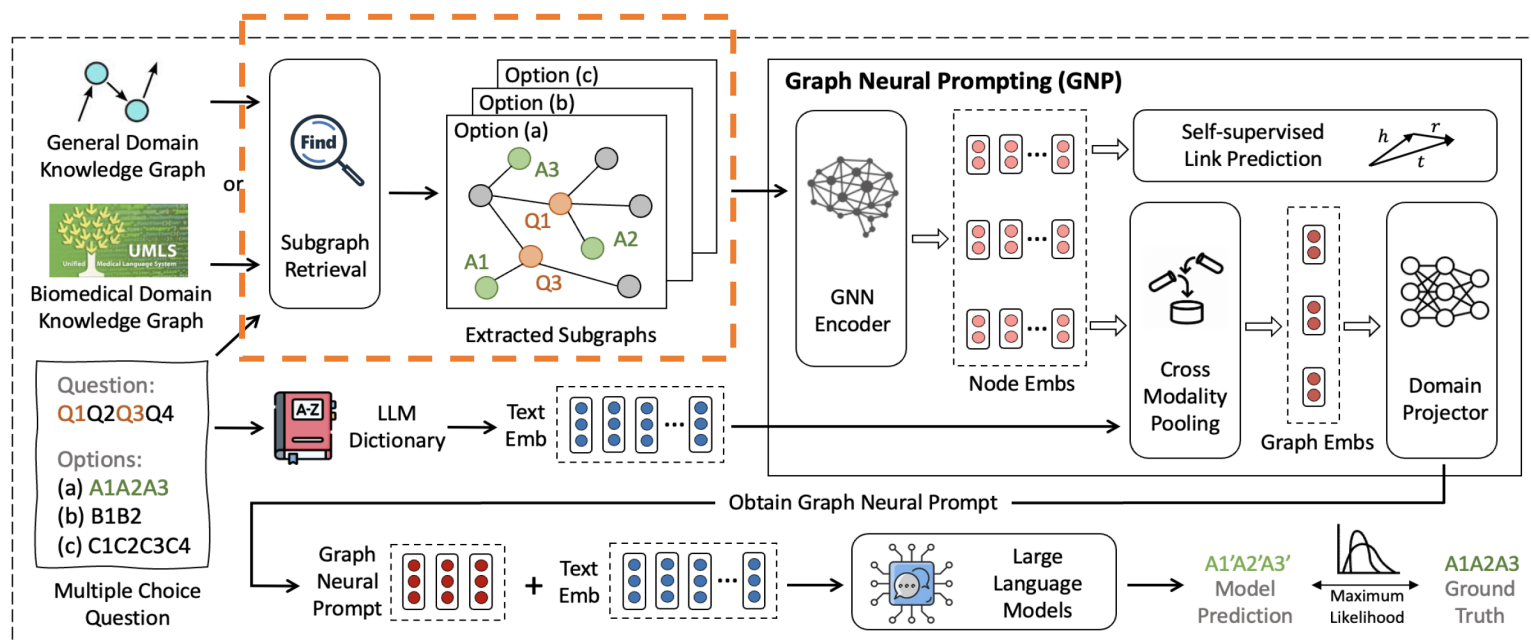
- ✓ Subgraph Retrieval : 根据问题和候选答案集中实体, 从KG中抽取k-hop局部子图 g'
- ✓ Graph Neural Prompting : 将子图 g' 编码得到图表示, 再经过映射得到graph prompting
- ✓ LLM Prediction : 将graph prompting和文本表示拼接输入LLM, 预测出答案。



4. GNP

模块一：Subgraph Retrieval

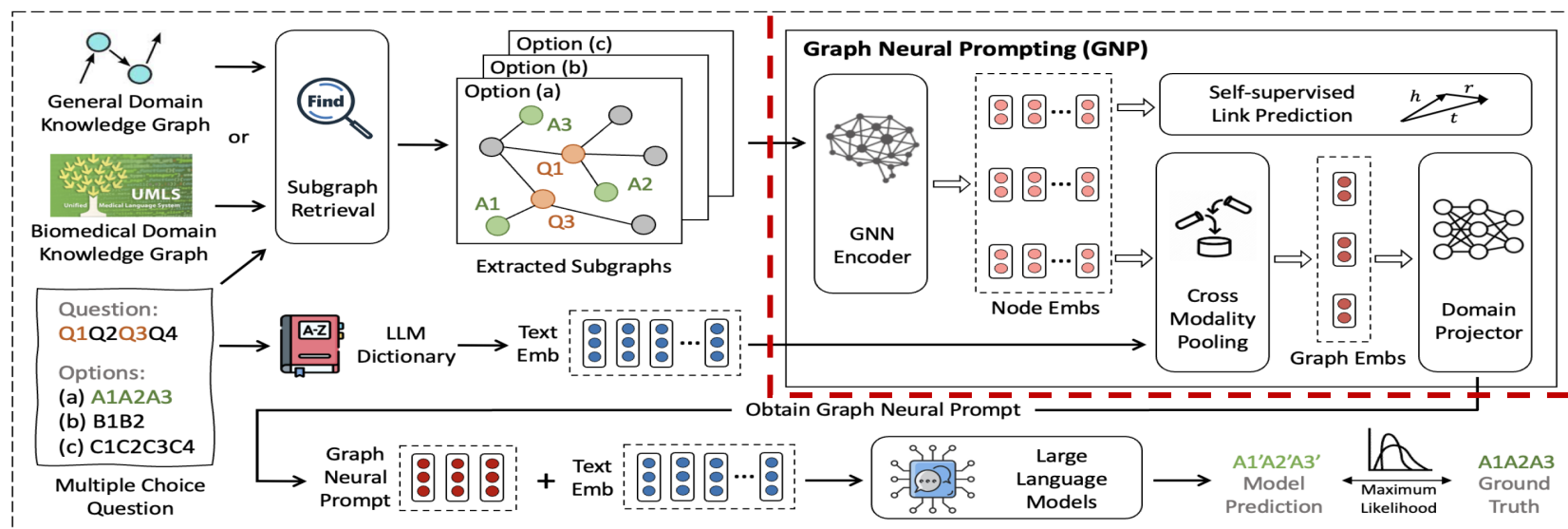
1. 将查询 Q ，候选答案 $A = \{a_k\}_{k=1}^K$ ，和上下文 C 和拼接，得到输入文本 X ；
2. 将输入文本 X 的tokens，链接到KG的entities；
3. 基于这些实体从KG上抽取2-hop邻居和关系，得到局部子图 g' 。



4. GNP

模块二：Graph Neural Prompting

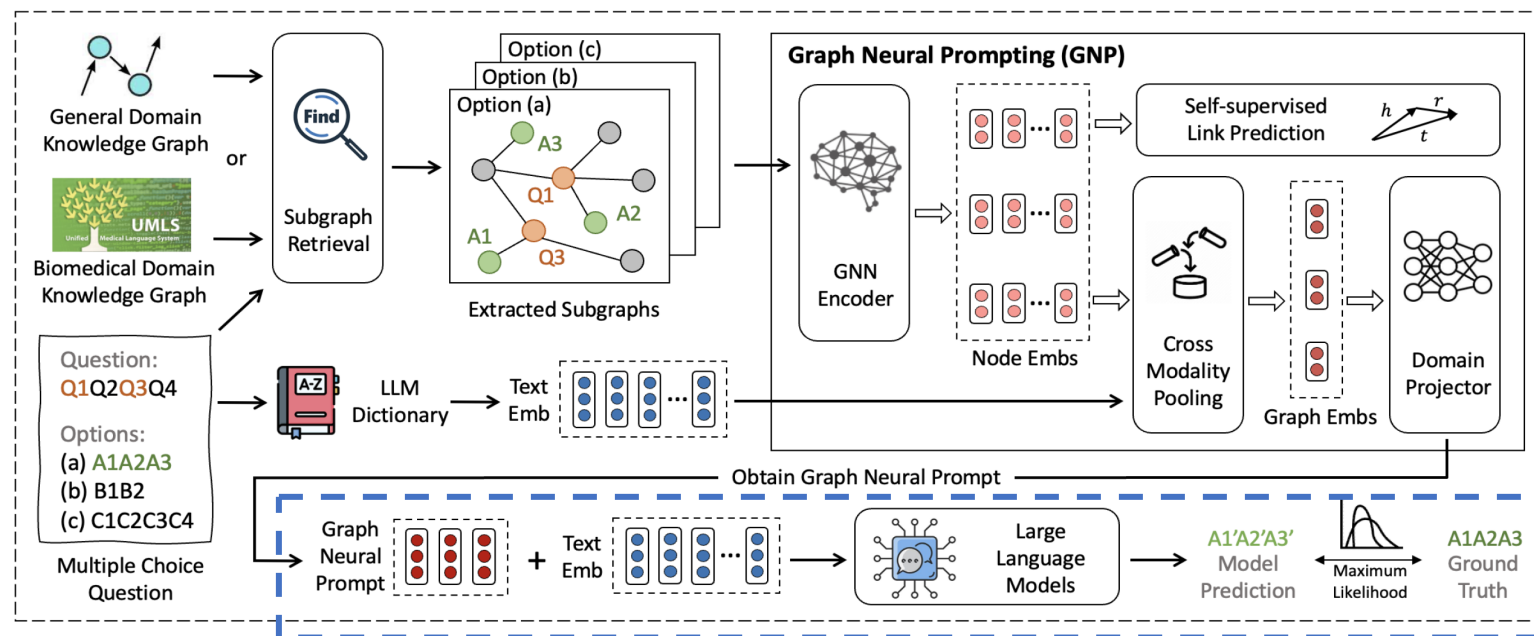
1. GNN encoder：利用GNN对抽取的子图编码，得到图中**节点表示**；
2. Cross-modality pooling：设计跨模态注意力融合文本表示和节点表示，得到**融入文本感知的节点表示**，并将其平均池化后得到**图表示**；
3. Domain projector：将图表示经过映射层得到**graph prompting**；
4. Self-supervised link prediction：设计了一个自监督的链接预测任务，多任务训练。



4. GNP

模块三：LLM Prediction

1. 将graph prompting拼接在文本表示的前面，得到完整的输入序列；
2. 利用LLM，预测出答案。



4. GNP

5. 实验结果

● 数据集

本文在通用领域和生物领域上进行实验，使用的KGs和数据集如下：

- KGs : ConceptNet常识知识图谱和Unified Medical Language System (UMLS) 生物图谱。
- 数据集：4个常识推理数据集 (OpenBookQA、ARC、PIQA和Riddle)和2个生物推理数据集 (PubMedQA和BioASQ)

● 实验设置

LLM有两个设置，一是固定模型参数，二是使用Lora微调模型

● 基准模型

- LLM-only : 不使用prompt
- KG Flattening REL/BFS : 按照相关度分数或宽度搜索将图中节点展开成序列
- KAPING OH/TH : 注入KG中一跳或两跳的邻居信息
- Prompt Tuning : 利用soft prompt

4. GNP

5. 实验结果

Table 1: Overall experimental results on commonsense reasoning and biomedical reasoning tasks.

LLM	Setting	Method	Commonsense Reasoning				Biomedical Reasoning		Total
			OBQA	ARC	PIQA	Riddle	PubMedQA	BioASQ	
FLAN-T5 xlarge (3B)	LLM Frozen	LLM-only	69.20	68.24	58.43	53.73	71.50	65.85	64.49
		Prompt Designs*	72.20	70.99	60.94	52.75	70.50	67.48	65.33
		KG Flattening REL	61.80	64.12	57.56	43.33	69.25	65.04	60.18
		KG Flattening BFS	62.80	63.86	56.69	44.12	69.25	65.04	60.29
		KAPING TH	58.80	63.52	52.34	40.78	70.00	65.04	58.41
		KAPING OH	60.00	63.09	51.69	41.37	70.00	65.04	58.53
		Prompt Tuning	72.20	70.64	60.83	53.33	72.00	66.67	65.95
		GNP	79.80	71.85	61.48	66.86	76.75	89.43	74.36
		Δ_{PT}	$\uparrow 10.53\%$	$\uparrow 1.71\%$	$\uparrow 1.07\%$	$\uparrow 25.37\%$	$\uparrow 6.60\%$	$\uparrow 34.14\%$	$\uparrow 12.76\%$
	LLM Tuned	Full Fine-tuning	82.80	73.30	63.55	74.12	76.25	91.06	76.85
		LoRA	80.40	71.33	63.76	72.94	76.25	92.68	76.23
		LoRA + GNP	83.40	72.45	64.31	75.49	76.25	92.68	77.43
		Δ_{LoRA}	$\uparrow 3.73\%$	$\uparrow 1.57\%$	$\uparrow 0.86\%$	$\uparrow 3.50\%$	N/A	N/A	$\uparrow 1.58\%$
		LLM-only	76.80	68.93	56.58	61.37	71.75	65.85	66.88
FLAN-T5 xxlarge (11B)	LLM Frozen	Prompt Designs*	79.60	74.16	58.00	60.59	71.25	66.67	68.38
		KG Flattening REL	72.80	66.78	56.80	53.53	69.50	66.67	64.35
		KG Flattening BFS	72.40	66.95	56.37	54.90	68.75	65.85	64.20
		KAPING TH	60.60	57.25	53.21	48.43	68.75	66.67	59.15
		KAPING OH	60.00	56.65	52.99	47.65	69.25	66.67	58.87
		Prompt Tuning	78.80	74.85	61.26	61.37	70.00	65.04	68.55
		GNP	87.20	78.20	63.66	70.98	76.75	90.24	77.84
		Δ_{PT}	$\uparrow 10.66\%$	$\uparrow 4.48\%$	$\uparrow 3.92\%$	$\uparrow 15.66\%$	$\uparrow 9.64\%$	$\uparrow 38.75\%$	$\uparrow 13.54\%$
		LLM Tuned	Full Fine-tuning	89.40	76.82	65.61	80.78	78.00	92.68
	LoRA		88.60	78.54	65.61	74.90	77.75	91.06	79.41
LoRA + GNP	89.60		78.71	65.94	76.67	79.75	94.31	80.83	
Δ_{LoRA}	$\uparrow 1.13\%$		$\uparrow 0.22\%$	$\uparrow 0.50\%$	$\uparrow 2.36\%$	$\uparrow 2.57\%$	$\uparrow 3.57\%$	$\uparrow 1.79\%$	

4. GNP

5. 实验结果

Table 2: Results of ablation study!

LLM	Variant	Commonsense		Biomedical	
		OBQA	ARC	PubMedQA	BioASQ
FLAN-T5 xlarge (3B)	w/o CMP	78.00	69.44	76.00	86.18
	w/o SLP	78.80	69.18	75.75	88.62
	w/o DP	73.00	70.30	76.25	83.74
	GNP	79.80	71.85	76.75	89.43
FLAN-T5 xxlarge (11B)	w/o CMP	85.20	76.91	75.75	87.80
	w/o SLP	83.60	76.74	73.25	89.43
	w/o DP	79.40	74.59	71.75	85.37
	GNP	87.20	78.20	76.25	90.24

Table 3: Results of integrating different model designs.

LLM	Design	Commonsense		Biomedical	
		OBQA	ARC	PubMedQA	BioASQ
FLAN-T5 xlarge (3B)	GNP	79.80	71.85	76.75	89.43
	+ DLP	79.80	70.30	75.50	89.43
	+ RGNN	79.00	71.49	75.50	89.43
FLAN-T5 xxlarge (11B)	GNP	87.20	78.20	76.25	90.24
	+ DLP	86.20	76.05	75.00	88.62
	+ RGNN	85.20	76.48	75.25	89.43

Question:

What is the best way to guess a babies eye color?

Options:

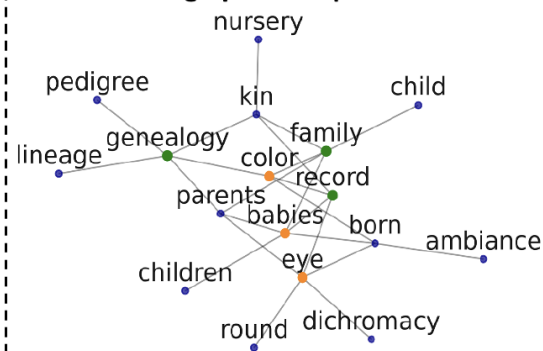
- (a) The surroundings they are born in
- (b) Their parents usual diet
- (c) Just take a random guess
- (d) The genealogy records of their family

Model Prediction:

Prompt Tuning: (c) Just take ... ❌

GNP: (d) The genealogy ... ✅

Retrieved subgraph from question entities:



Question:

were there fossil fuels in the ground when humans evolved?

Options:

- (a) This was only created by humans
- (b) Humans predate fossil fuel formation
- (c) Significant supplies accumulated prior
- (d) None of these

Model Prediction:

Prompt Tuning: (b) Humans pre ... ❌

GNP: (c) Significant supplies ... ✅

Retrieved subgraph from question entities:

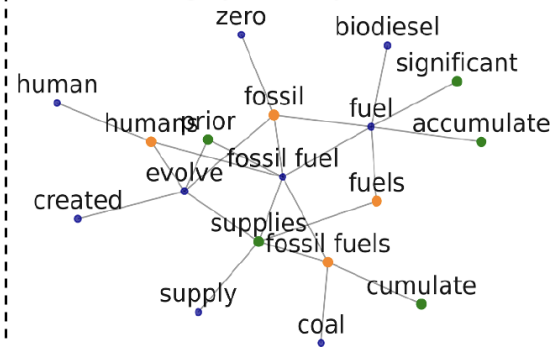


Figure 5: Case study on two QA examples from OBQA

总结

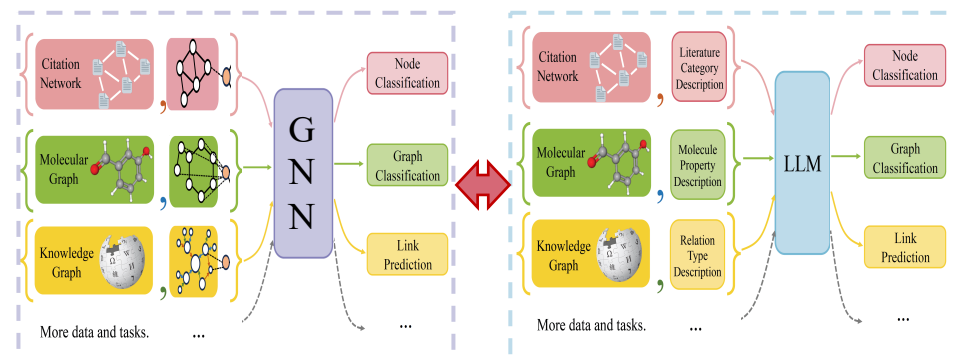
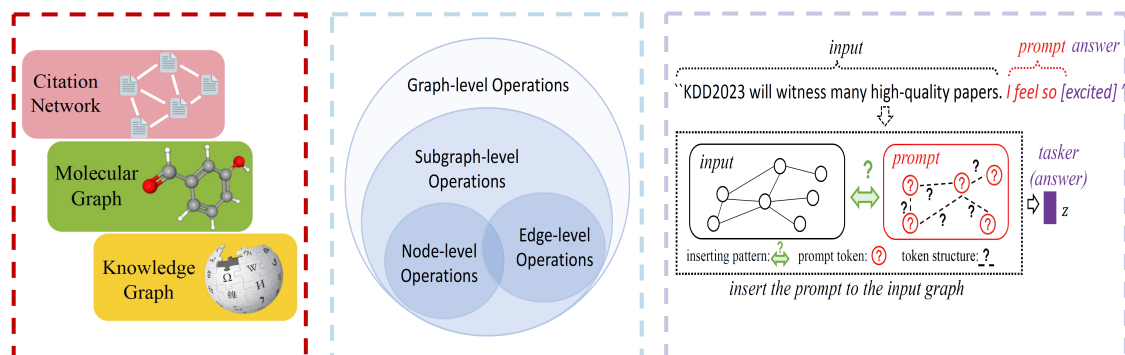
Graph Prompt

1. 挑战

来自不同领域的图数据；图上的任务呈现多样化；图提示学习模式不明确

2. 研究路线

GNN+Prompt、LLM+Prompt、GNN+LLM+Prompt



论文列表

- Graph Prompt
 - [KDD22] GPPT: Graph Pre-training and Prompt Tuning to Generalize Graph Neural Networks
 - [WWW23] GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks
 - [NeurIPS23] GPT-Universal Prompt Tuning for Graph Neural Networks
 - [KDD23] All in One: Multi-Task Prompting for Graph Neural Networks
 - [NeurIPS23] Prodigy: Enabling In-context Learning Over Graphs
 - [CVPR23] Deep Graph Reprogramming
 - [arXiv 2023.09] DeepGPT- Deep Prompt Tuning for Graph Transformers
 - [arXiv 2023.09] Graph Neural Prompting with Large Language Models
 - [arXiv 2023.09] One for All: Towards Training One Graph Model for All Classification Tasks