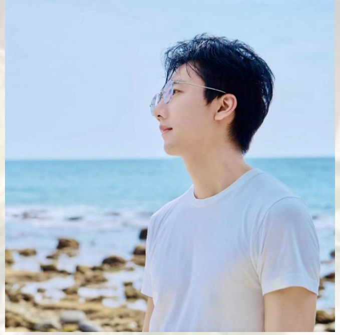
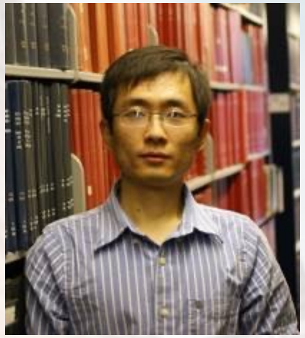
Large Language Models for Graphs: Progresses and Directions








Outline

香 港 大 學
THE UNIVERSITY OF HONG KONG

Time	Section	Presenter
13:30 - 13:40	Opening & Introduction	Chao Huang
13:40 - 14:25	Section 1: GNNs as Prefix	Jiabin Tang
14:25 - 15:10	Section 2: LLMs as Prefix	Lianghao Xia
15:10 - 15:40	Coffee Break	-
15:40 - 16:15	Section 3: LLMs-Graphs Intergration	Xubin Ren
16:15 - 17:00	Section 4: LLMs-Only	Xubin Ren & Jiabin Tang

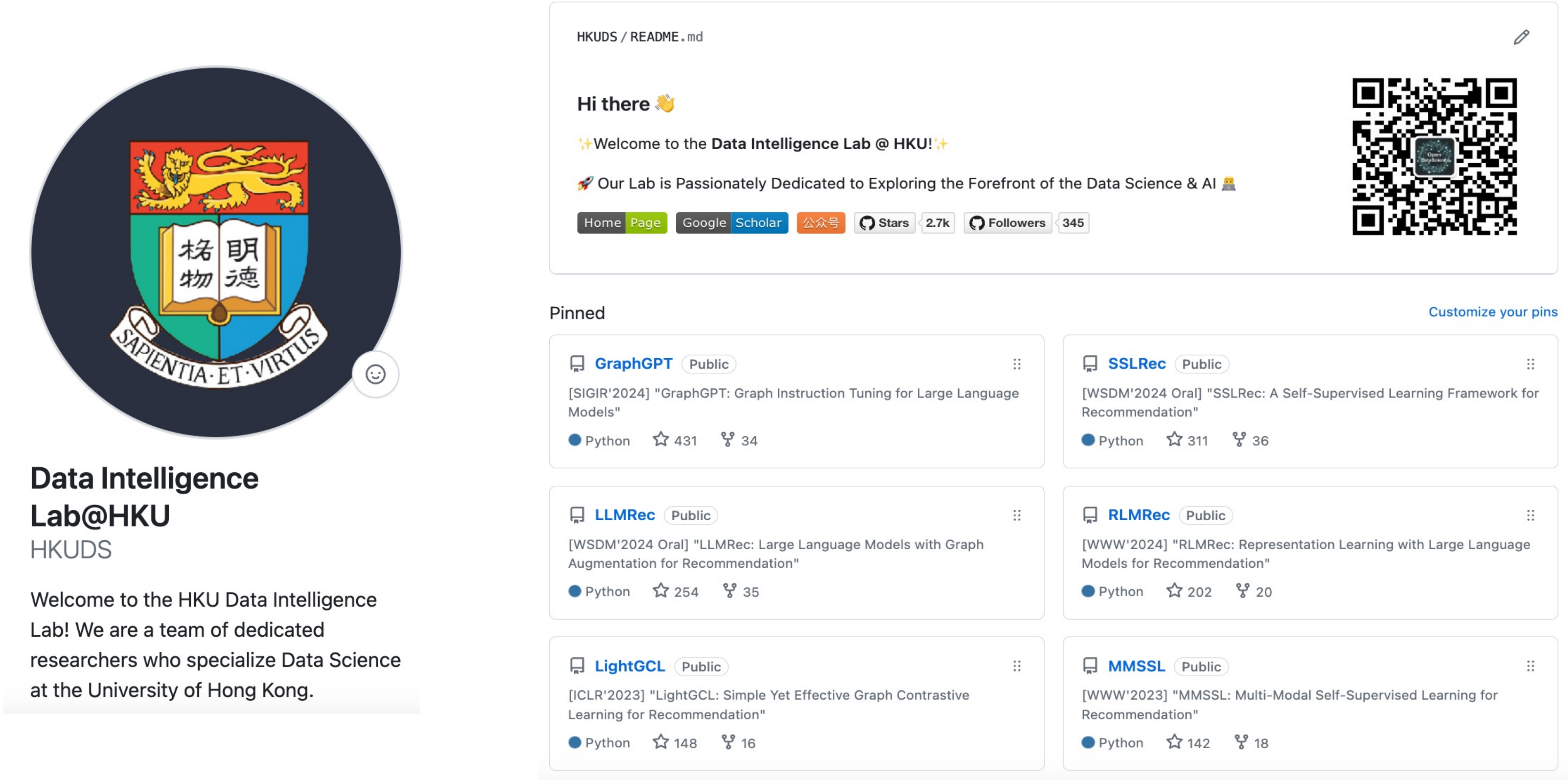
Q&A after each session

Tutorial Site

For more information about this tutorial!

Open-Sourced Research

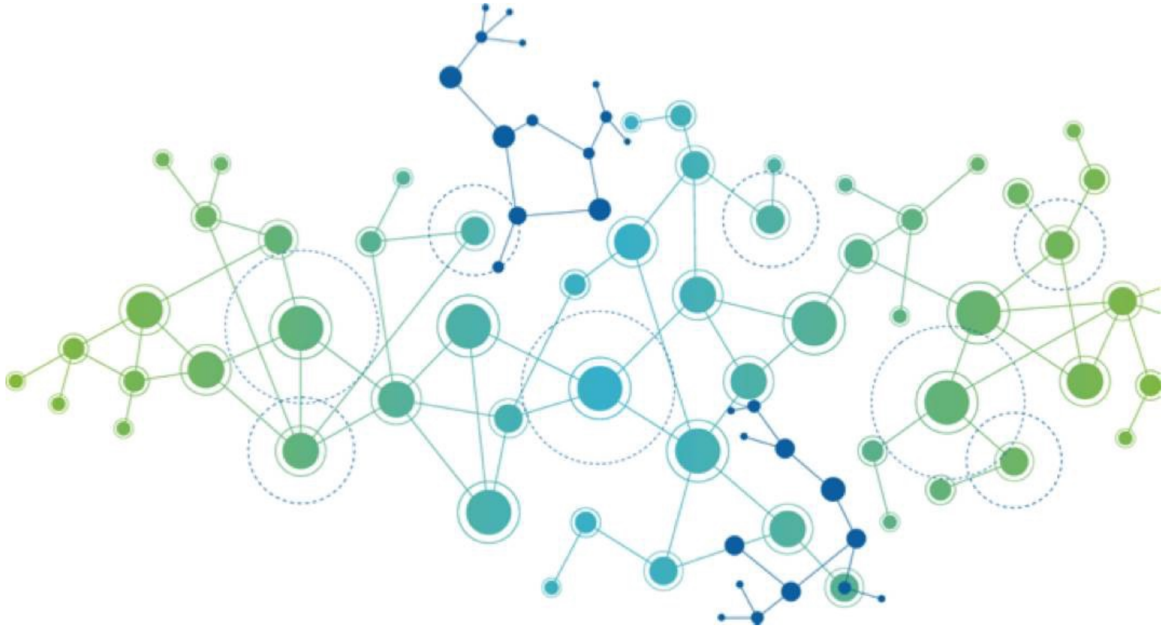
香 港 大 學
THE UNIVERSITY OF HONG KONG



HKUDS https://github.com/HKUDS

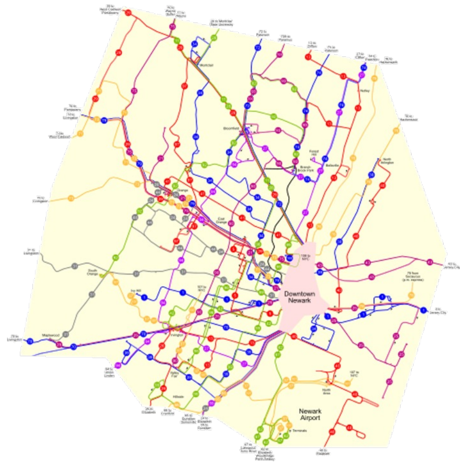
Graphs

Graphs are general language for describing and analyzing entities with relations/interactions

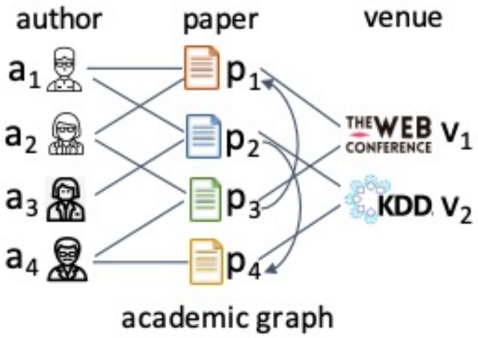


Graphs

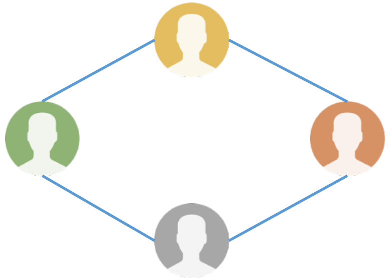
Social Graph

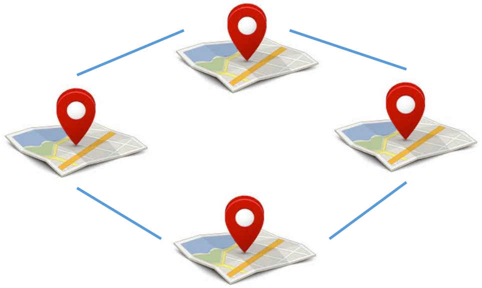


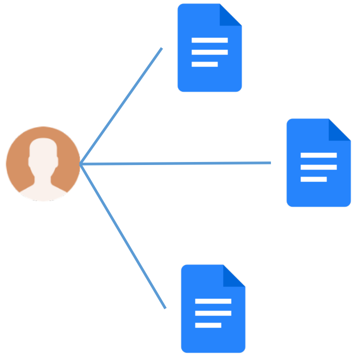
Transportation Graph



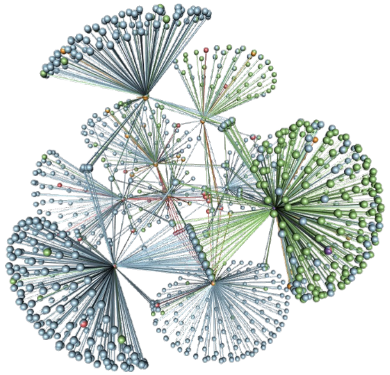
Academic Graph

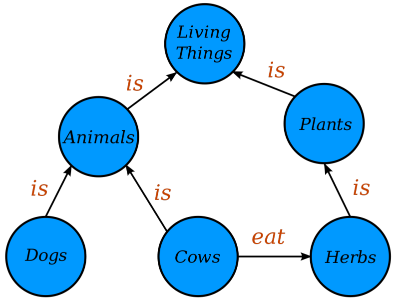






Graphs

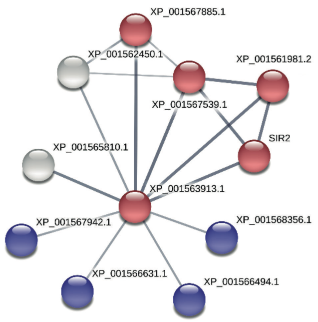


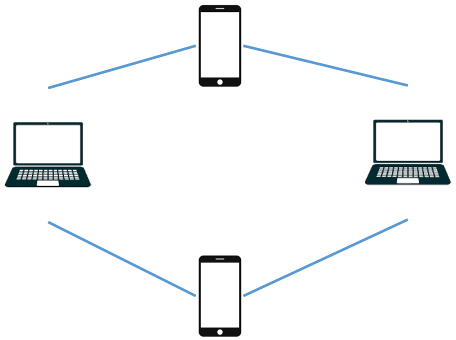


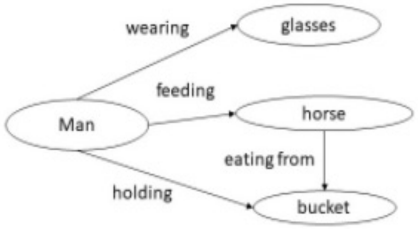
Protein Interaction Graph

Communication Graph

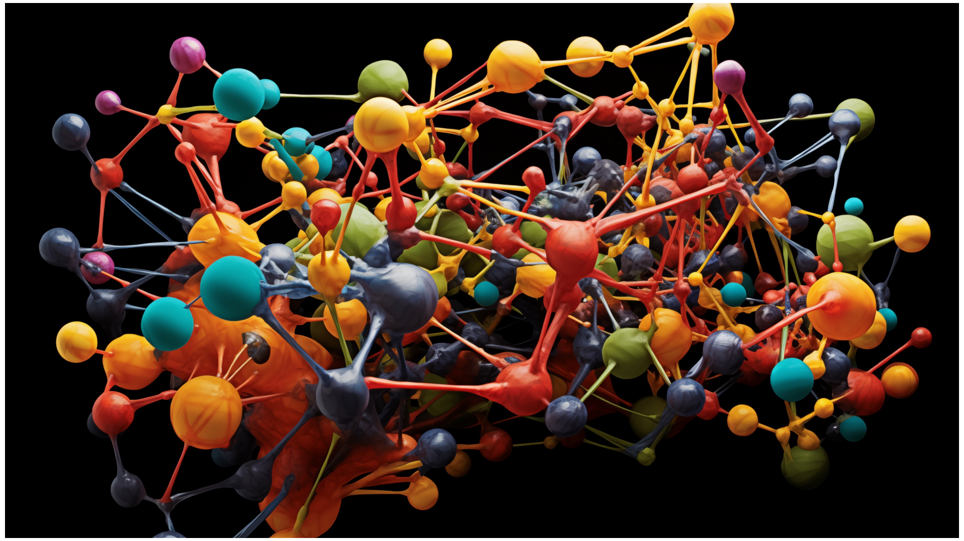
Knowledge Graph







Graph v.s Language



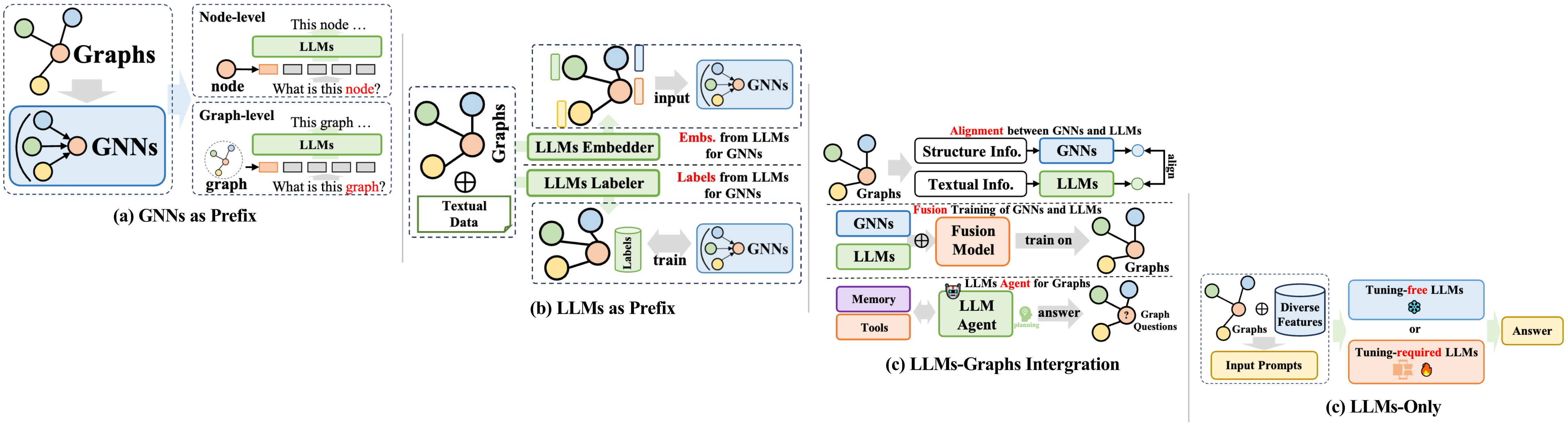
Massive amounts of information

- Social networks
- Recommendation
- Spatio-temporal prediction

Complex relational semantics

- Proteins
- Molecules
- Heterogeneous graphs

Graph + LLMs


Graphs
GNNs
Node-level
This node ...
LLMs
node
What is this node?
Graph-level
This graph ...
LLMs
graph
What is this graph?
(a) GNNs as Prefix
Graphs
Textual Data
input
GNNs
LLMs Embedder
Embs. from LLMs for GNNs
LLMs Labeler
Labels from LLMs for GNNs
Labels
train
GNNs
(b) LLMs as Prefix
Alignment between GNNs and LLMs
Graphs
Structure Info.
GNNs
Textual Info.
LLMs
align
Fusion Training of GNNs and LLMs
GNNs
LLMs
Fusion Model
train on
Graphs
LLMs Agent for Graphs
Memory
Tools
LLM Agent
answer
Graph Questions
(c) LLMs-Graphs Intergration
Graphs
Diverse Features
Input Prompts
Tuning-free LLMs
or
Tuning-required LLMs
Answer
(c) LLMs-Only

GNN as Prefix

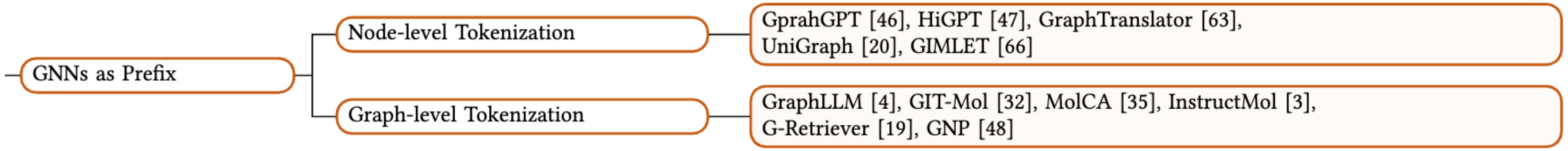
GNN→LLM

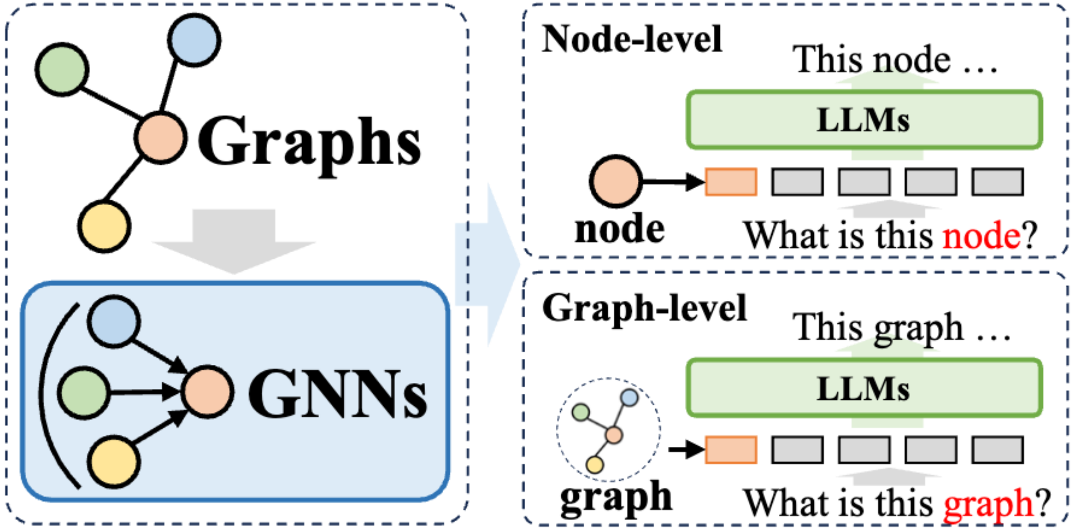
About Me (Jiabin Tang)

- First-year Ph.D. student majoring in Data Science at The University of Hong Kong, supervised by Dr. Chao Huang.
- **Research Interests**:
  - ➢ Large Language Models and other AIGC techniques
  - ➢ Graph Learning, Trustworthy Machine Learning
  - ➢ Deep Learning Applications, e.g., Spatio-Temporal Mining and Recommendation

GNN as Prefix

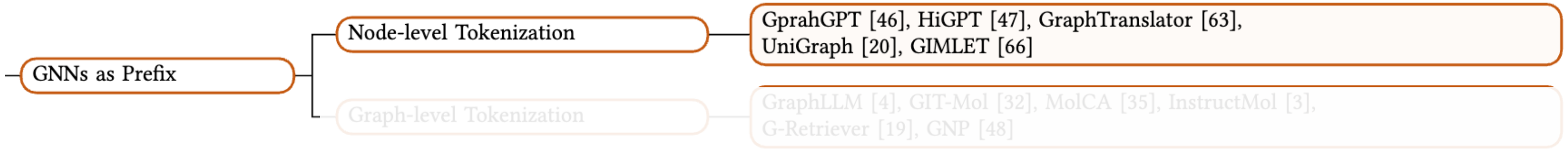
- **Node-level Tokenization**: retain the unique structural representation of each node as much as possible
- **Graph-level Tokenization**: abstracts node representations into unified graph representations through various "pooling" operations

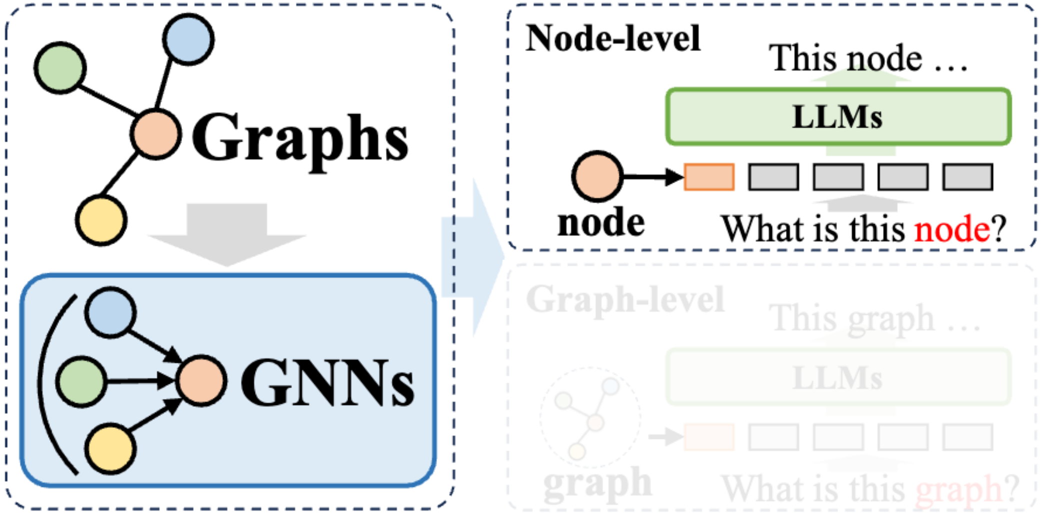




Node-level Tokenization

- **Motivation**:
  - make the LLM understand fine-grained node-level structural information and distinguish relationships
  - each node of the graph structure is input into the LLM





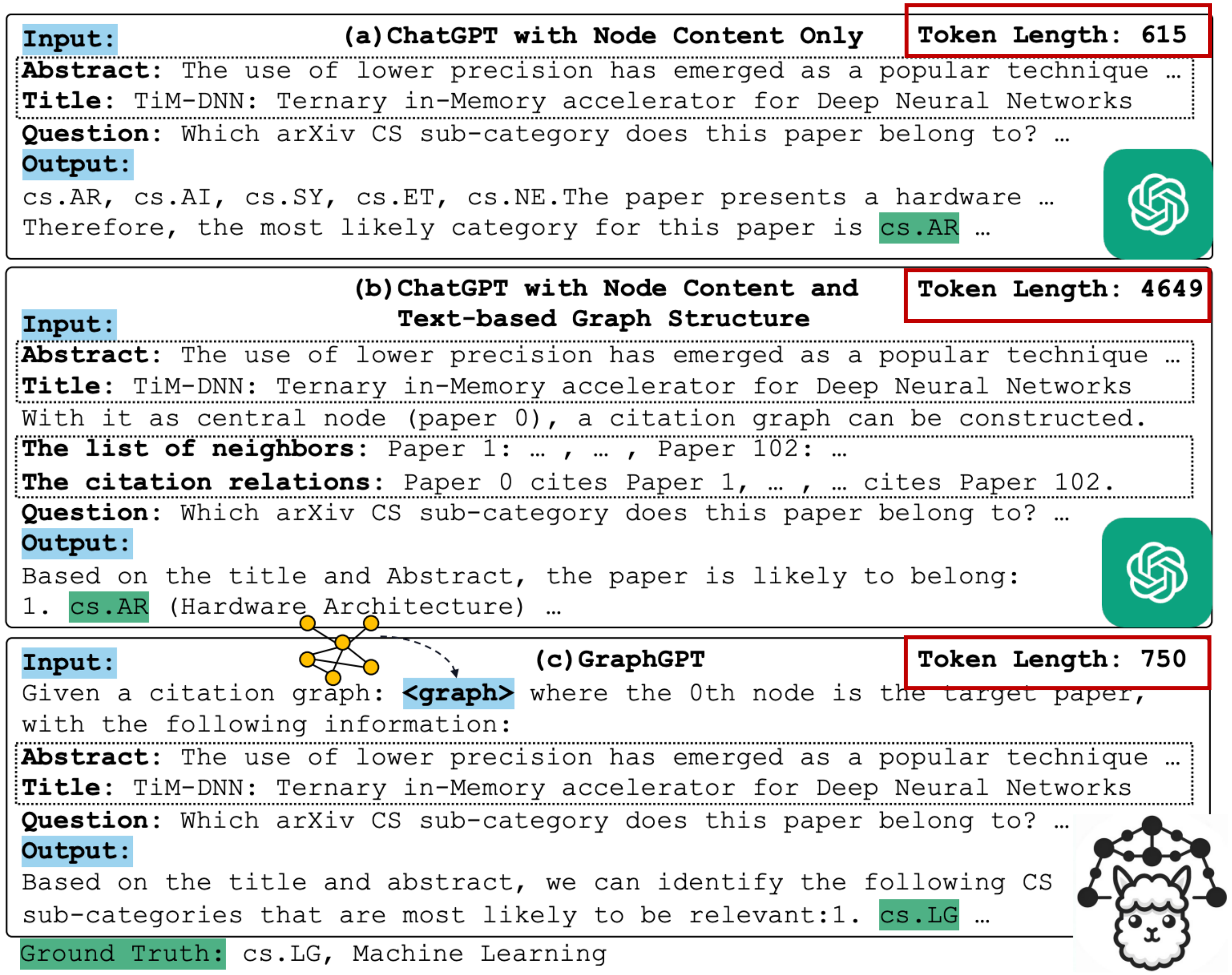
GNN as Prefix: GraphGPT

- **GraphGPT:** Graph Instruction Tuning for Large Language Models (SIGIR'24)
- **Backgrounds:**
  - ➢ **RQ1**: How can we feed graph structures into LLMs?
  - ➢ **RQ2**: How can we empower LLMs to understand graph structures?
  - ➢ **RQ3**: How to endow LLMs with the ability to reason step-by-step for <u>zero-shot complex</u> graph learning tasks.

香 港 大 學
THE UNIVERSITY OF HONG KONG

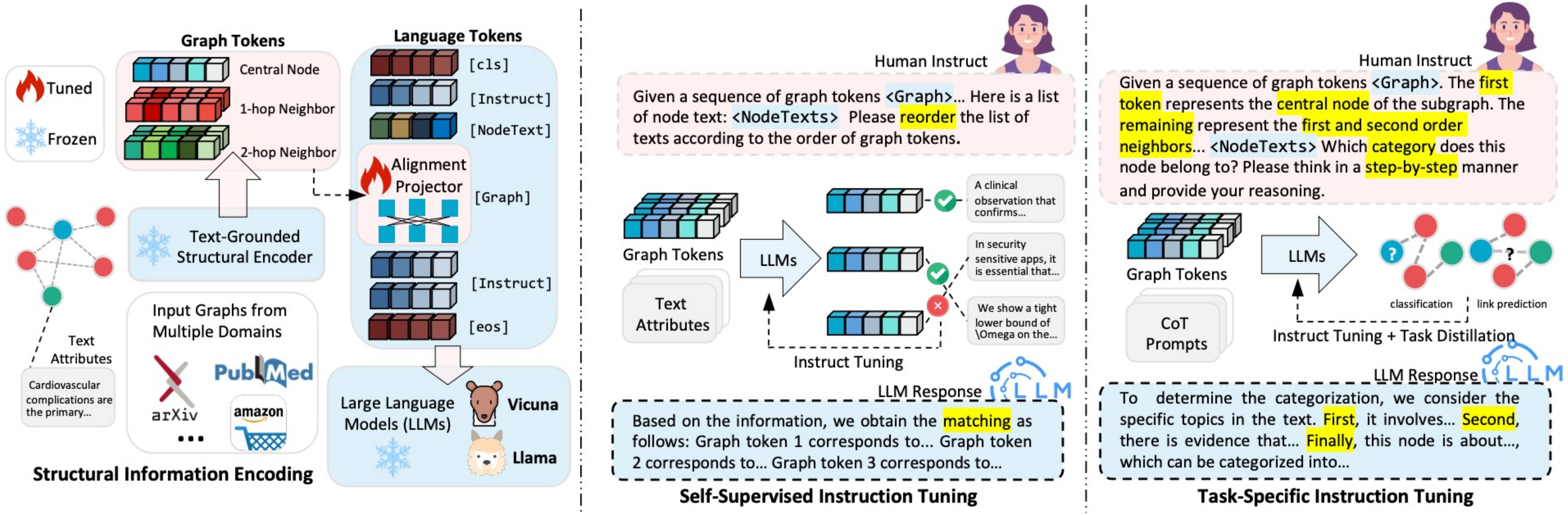
GraphGPT: RQ1

- **RQ1:** How can we feed graph structures into LLMs?
  - ➢ Without Graph Structure
    - ✓ fail when interdisciplinary field
  - ➢ Text-based Graph
    - ✓ fail when interdisciplinary field
    - ✓ unacceptably long token length
  - ➢ GraphGPT
    - ✓ effectively learning from graph
    - ✓ controllable token length



Overall Architecture

- **RQ1:** How can we feed graph structures into LLMs?
  - ➢ Graph is a sequence of “graph tokens”

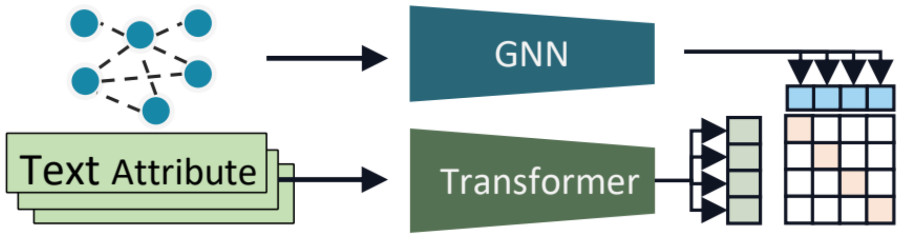


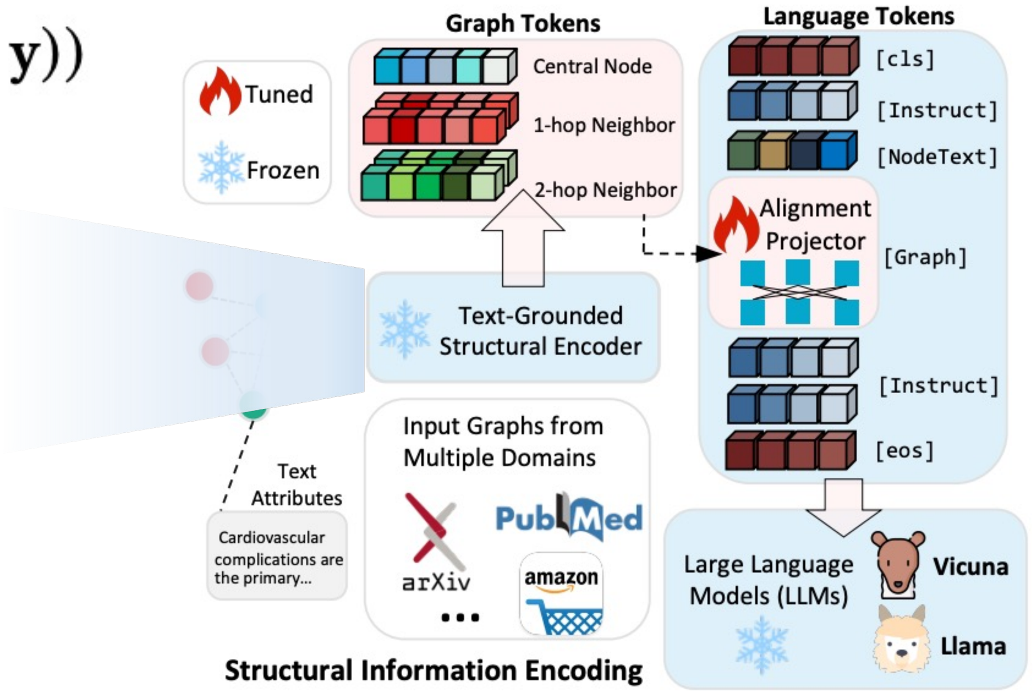
Text-Graph Grounding

- Initializing graph encoder with natural language alignment

$$\Gamma_i = (g_i^{(1)}(\hat{\mathrm{H}})g_i^{(2)}(\hat{\mathrm{T}})^\top) \cdot \exp(\tau)$$

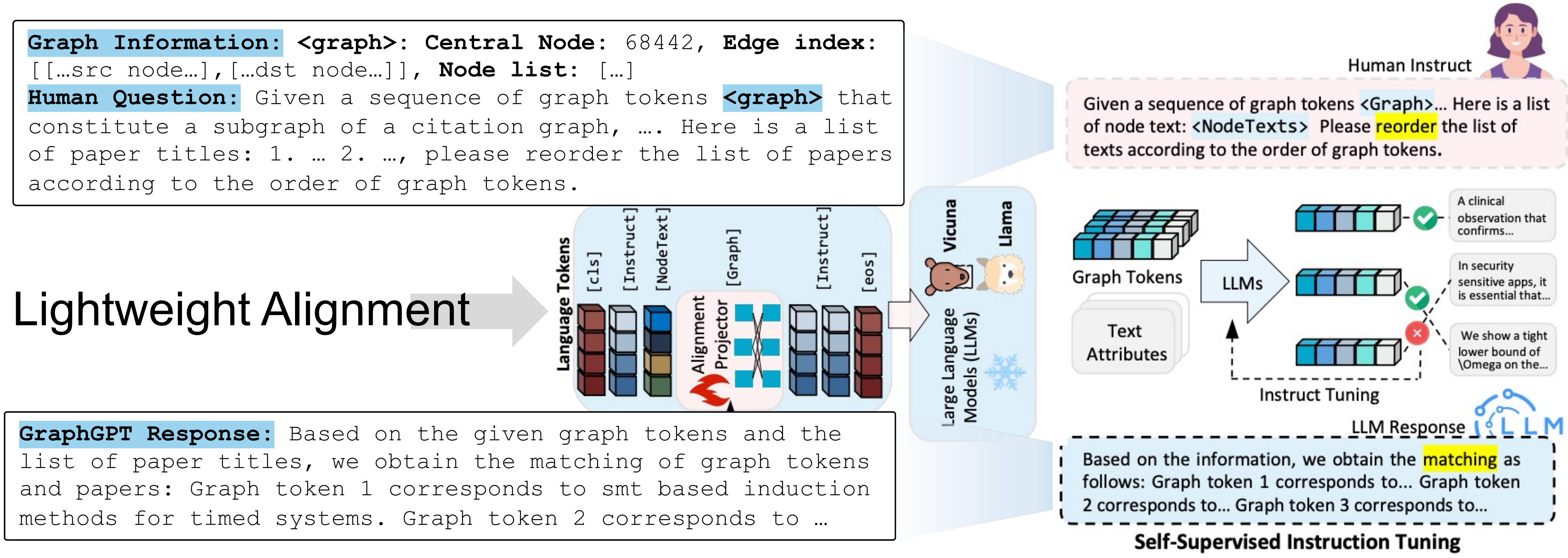
$$\mathcal{L} = \sum_i \frac{1}{2}\lambda_i(\mathrm{CE}(\Gamma_i, \mathbf{y}) + \mathrm{CE}(\Gamma_i^\top, \mathbf{y}))$$





Self-Supervised Instruction Tuning (RQ 2)

- let LLM match the graph tokens with the corresponding natural language content in the prompt.



Task-Specific Instruction Tuning (RQ 2)

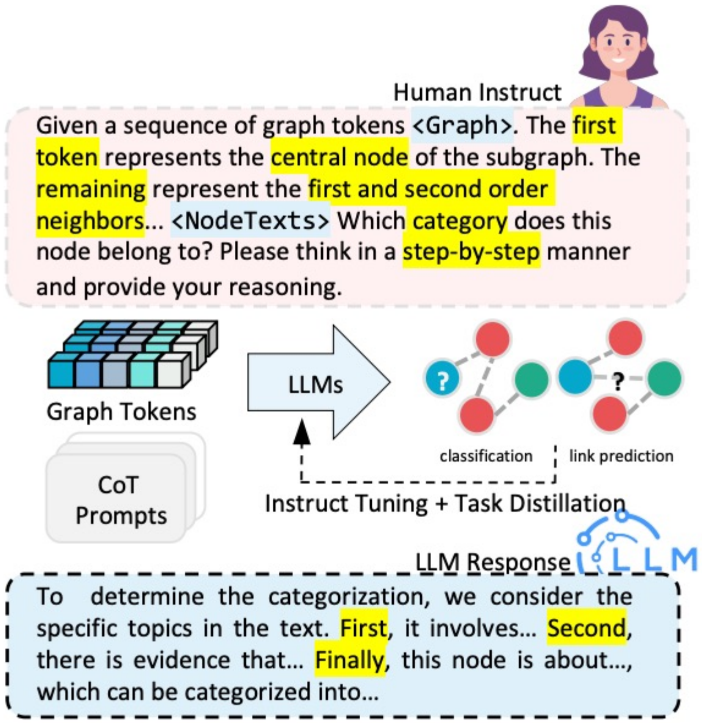
香 港 大 學
THE UNIVERSITY OF HONG KONG

- Instruction tuning for downstream tasks.

Graph Information: **<graph>: Central Node**: 2, **Edge index**: [[…src node…],[…dst node…]], **Node list**: […]
Human Question: Given a citation graph: **<graph>** where the 0th node is the target paper, with the following information: Abstract: … Title: … Question: Which arXiv CS sub-category does this paper belong to? …

Graph Information: **<graph>: Central Node 1**: 8471, **Edge index 1**: [[…src node…],[…dst node…]], **Node list 1**: […] **<graph>: Central Node 2**: 19368, **Edge index 2**: [[…src node…],[…dst node…]], **Node list 2**: […]
Human Question: Given a sequence of graph tokens: **<graph>** that constitute a subgraph of a citation graph, …. Abstract: … Titile: … and the other sequence of graph tokens: **<graph>**, … Abstract: … Title: …, are these two central nodes connected? Give me an answer of "yes" or "no".

GraphGPT Response: cs.IT, cs.LG, cs.SP, cs.CV, cs.NA. The paper discusses the Restricted Isometry …. So, it is likely to belong to cs.IT…

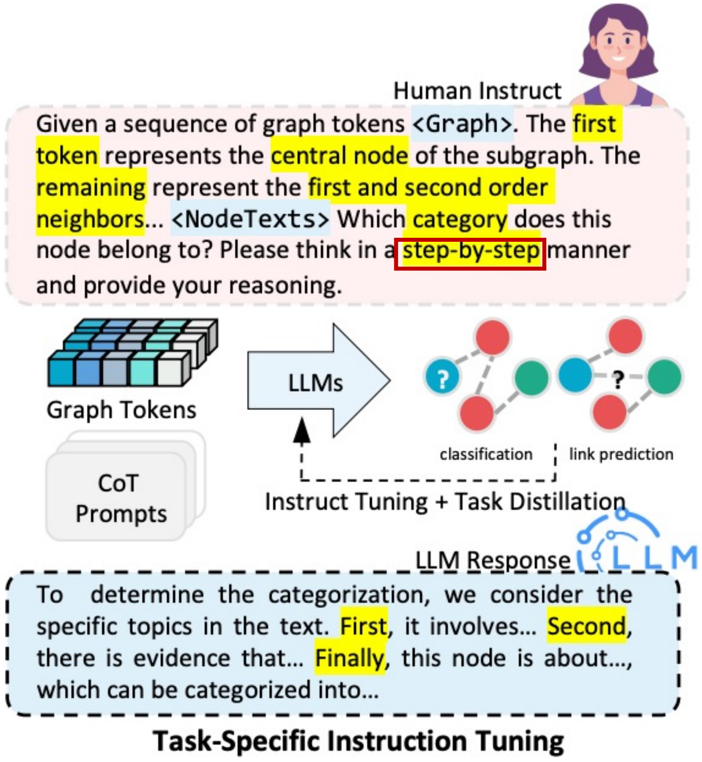


Task-Specific Instruction Tuning

Chain-of-Thought (CoT) Distillation (RQ 3)

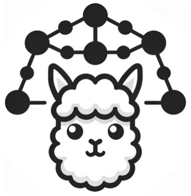
香 港 大 學
THE UNIVERSITY OF HONG KONG

- Distilling reasoning capabilities from a powerful model (ChatGPT) through the Chain-of-Thought (CoT).
  - What is COT? → … **Please think step by step.**



Powerful yet Closed-source and Cost

COT Distillation



-7B, Lightweight yet not "smart"

Experimental Results

香 港 大 學
THE UNIVERSITY OF HONG KONG

- Outperform SOTA not only **Supervised** but **Zero-shot** settings.

First !!

Dataset	Arxiv-Arxiv		Arxiv-PubMed		Arxiv-Cora		(Arxiv+PubMed)-Cora		(Arxiv+PubMed)-Arxiv	
Model	Accuracy	Macro-F1	acc	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
MLP	0.5179	0.2536	0.3940	0.1885	0.0258	0.0037	0.0220	0.0006	0.2127	0.0145
GraphSAGE	0.5480	0.3290	0.3950	0.1939	0.0328	0.0132	0.0132	0.0029	0.1281	0.0129
GCN	0.5267	0.3202	0.3940	0.1884	0.0214	0.0088	0.0187	0.0032	0.0122	0.0008
GAT	0.5332	0.3118	0.3940	0.1884	0.0167	0.0110	0.0161	0.0057	0.1707	0.0285
RevGNN	0.5474	0.3240	0.4440	0.3046	0.0272	0.0101	0.0217	0.0016	0.1309	0.0126
DGI	0.5059	0.2787	0.3991	0.1905	0.0205	0.0011	0.0205	0.0011	0.5059	0.2787
GKD	0.5570	0.1595	0.3645	0.2561	0.0470	0.0093	0.0406	0.0037	0.2089	0.0179
GLNN	0.6088	0.3757	0.4298	0.3182	0.0267	0.0115	0.0182	0.0092	0.3373	0.1115
NodeFormer	0.5922	0.3328	0.2064	0.1678	0.0152	0.0065	0.0144	0.0053	0.2713	0.0855
DIFFormer	0.5986	0.3355	0.2959	0.2503	0.0161	0.0094	0.0100	0.0007	0.1637	0.0234
baichuan-7B	0.0946	0.0363	0.4642	0.3876	0.0405	0.0469	0.0405	0.0469	0.0946	0.0363
vicuna-7B-v1.1	0.2657	0.1375	0.5251	0.4831	0.1090	0.0970	0.1090	0.0970	0.2657	0.1375
vicuna-7B-v1.5	0.4962	0.1853	0.6351	0.5231	0.1489	0.1213	0.1489	0.1213	0.4962	0.1853
GraphGPT-7B-v1.1-cot	0.4913	0.1728	0.6103	0.5982	0.1145	0.1016	0.1250	0.0962	0.4853	0.2102
GraphGPT-7B-v1.5-stage2	**0.7511**	**0.5600**	0.6484	0.5634	0.0813	0.0713	0.0934	0.0978	0.6278	0.2538
GraphGPT-7B-v1.5-std	0.6258	0.2622	**0.7011**	**0.6491**	0.1256	0.0819	0.1501	0.0936	0.6390	0.2652
GraphGPT-7B-v1.5-cot	0.5759	0.2276	0.5213	0.4816	**0.1813**	**0.1272**	**0.1647**	**0.1326**	**0.6476**	**0.2854**
p-val	$2.26e^{-9}$	$1.56e^{-10}$	$2.22e^{-7}$	$1.55e^{-9}$	$1.04e^{-9}$	$9.96e^{-6}$	$7.62e^{-8}$	$1.97e^{-7}$	$1.5e^{-13}$	$4.63e^{-6}$

Experimental Results

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Generalization Ability Investigation.**
  - ➢ More Data Boost Model Transfer Ability
  - ➢ More Data Yet No Forgetting
  - ➢ Generalization for Multitasking Graph Learner

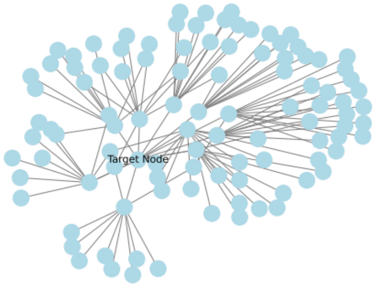
Dataset	Supervision. on Arxiv		Zero Shot on Cora	
Model	Acc	Macro-F1	Acc	Macro-F1
MLP	0.5179	0.2536	0.0220	0.0006
GraphSAGE	0.5480	0.3290	0.0132	0.0029
GCN	0.5267	0.3202	0.0187	0.0032
GAT	0.5332	0.3118	0.0161	0.0057
RvGNN	0.5474	0.3240	0.0217	0.0016
DGI	0.5059	0.2787	0.0205	0.0011
GKD	0.5570	0.1595	0.0406	0.0037
GLNN	0.6088	0.3757	0.0182	0.0092
NodeFormer	0.5922	0.3328	0.0144	0.0053
DIFFormer	0.5986	0.3355	0.0100	0.0007
baichuan-7b	0.0946	0.0363	0.0405	0.0469
vicuna-7B-v1.1	0.2657	0.1375	0.1090	0.0970
vicuna-7B-v1.5	0.4962	0.1853	0.1489	0.1213
Arxiv-std + PubMed-std	0.6390	0.2652	0.1501	0.0936
Arxiv-cot + PubMed-cot	0.6476	0.2854	0.1647	0.1326
Arxiv-mix + PubMed-mix	0.6139	0.2772	0.1544	0.1048
Arxiv-std + PubMed-std + Link	0.5931	0.2238	**0.1847**	**0.1579**
Arxiv-mix + Pubmed-mix + Link	**0.6874**	**0.3761**	0.1836	0.1494

Dataset	PubMed	
Model	AUC	AP
MLP	0.5583	0.5833
GAT	0.5606	0.6373
GraphSAGE	0.5041	0.5813
RevGNN	0.4538	0.5083
Node2Vec	0.6535	0.6885
w/o Link	0.5010	0.5005
only Link	0.6704	0.6087
Arxiv-std + PubMed-std + Link	**0.8246**	**0.8026**
Arxiv-mix + PubMed-mix + Link	0.6451	0.5886

Experimental Results

香 港 大 學
THE UNIVERSITY OF HONG KONG

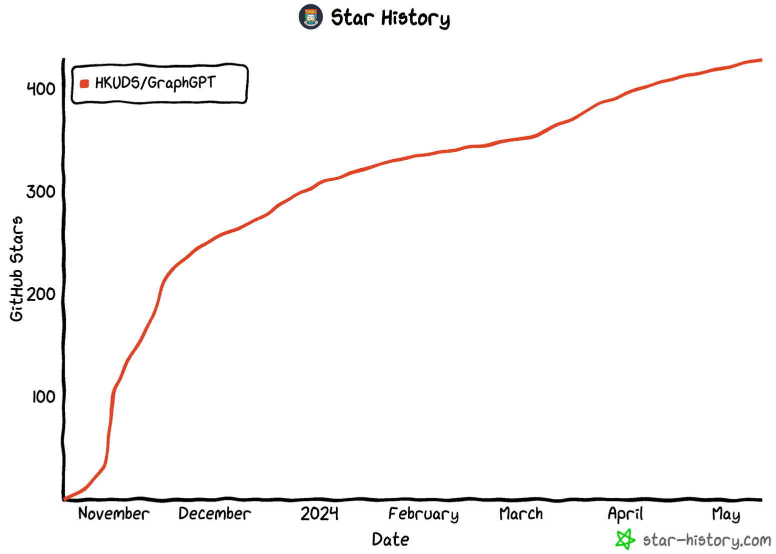
- Model Case Study.

Title: TiM-DNN: Ternary in-Memory accelerator for Deep Neural Networks. (with **102** first-order and second-order neighbors)	
Ground-Truth Category: cs.LG, Machine Learning	
ChatGPT (Node Content Only): cs.AR, cs.AI, cs.SY, cs.ET, cs.NE. The paper presents a hardware accelerator for deep neural networks (DNNs) that is specifically designed to execute ternary DNNs, which represent weights and activations by signed ternary values. Therefore, the most likely category for this paper is cs.AR (Hardware Architecture), which focuses on the design and analysis of hardware and systems for AI and machine learning applications. The paper also involves the optimization of compute and storage requirements of DNNs, which falls under the category of cs.AI (Artificial Intelligence). Additionally, the paper proposes a programmable, in-memory accelerator, which is a type of embedded system, and therefore, it could be categorized under cs.SY (Systems and Control). Furthermore, the paper discusses the use of ternary values for DNNs, which is an emerging technique to optimize the compute and storage requirements of DNNs, and therefore, it could be categorized under cs.ET (Emerging Technologies). Lastly, the paper proposes a new bit-cell called Ternary Processing Cells (TPCs), which could be categorized under cs.NE (Neural and Evolutionary Computing).	
ChatGPT (Node Content and Text-based Graph Structure): Based on the title "TiM-DNN: Ternary In-Memory Accelerator for Deep Neural Networks", the paper is likely to belong to the following arXiv CS sub-categories: 1. cs.AR (Hardware Architecture): The paper proposes a hardware accelerator, TiM-DNN, specifically designed for executing ternary deep neural networks. This falls under the domain of hardware architecture research. 2. cs.NE (Neural and Evolutionary Computing): The paper focuses on accelerating deep neural networks, which is a topic within the field of neural and evolutionary computing. 3. cs.CV (Computer Vision and Pattern Recognition): Deep neural networks are widely used in computer vision tasks, and the TiM-DNN accelerator could potentially be applied to accelerate computer vision algorithms. 4. cs.SY (Systems and Control): The TiM-DNN accelerator is a system-level solution for accelerating deep neural networks, so it could be categorized under systems and control research. 5. cs.DC (Distributed, Parallel, and Cluster Computing): While the paper does not explicitly mention distributed or parallel computing, the TiM-DNN accelerator is designed to optimize the compute and storage requirements of deep neural networks, which are often executed on distributed or parallel computing systems.	
GraphGPT: Based on the title and abstract of the paper, we can identify the following arXiv CS sub-categories that are most likely to be relevant:  1. cs.LG (Machine Learning): This sub-category is relevant because the paper discusses the use of ternary DNNs for language modeling and image classification, which are both tasks that involve learning. 2. cs.AI (Artificial Intelligence): This sub-category is relevant because the paper proposes a new accelerator for DNNs, which is designed to optimize the compute and storage requirements of complex DNNs. 3. cs.NE (Numerical Analysis): This sub-category is relevant because the paper discusses the use of ternary DNNs, which involve signed ternary values, and the use of TiM tiles, which are specialized memory arrays that perform massively parallel signed vector-matrix multiplications on ternary values. 4. cs.SY (Systems and Control Theory): This sub-category is relevant because the paper discusses the implementation of TiM-DNN in 32nm technology using an architectural simulator calibrated with SPICE simulations and RTL synthesis. 5. cs.AR (Hardware Architecture): This sub-category is relevant because the paper proposes a new accelerator for DNNs, which is designed to be programmable and in-memory.	

More Details

香 港 大 學
THE UNIVERSITY OF HONG KONG

- More details could be found as below:
  - Project page: https://graphgpt.github.io/ (QR code: )
  - Paper: https://arxiv.org/abs/2310.13023
  - Code: https://github.com/HKUDS/GraphGPT
  - Huggingface: Jiabin99/GraphGPT-7B-mix-all

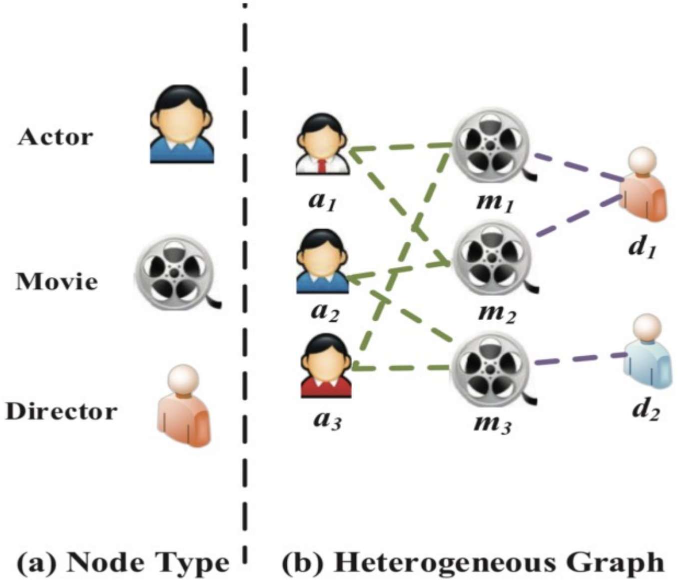


Don't be stingy with your stars!

GNN as Prefix: HiGPT

- **HiGPT:** Heterogeneous Graph Language Model
- **Backgrounds:**
  - **Heterogeneous Graphs**: $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathcal{T}, \mathcal{R}, \mathbf{X})$, where $\mathcal{T}$ and $\mathcal{R}$ signify the types of nodes and edges. $\mathbf{X} = \{X_{T_i} \in \mathbb{R}^{|\mathcal{V}_{T_i}| \times d_{T_i}}\}$ contains attributes associated with each node.
  - **Meta Relation:** a representation of the relationship between different types of nodes connected by an edge.

→ Meta relation $e = (u, v)$ $< \tau(u), \rho(e), \tau(v) >$



GNN as Prefix: HiGPT

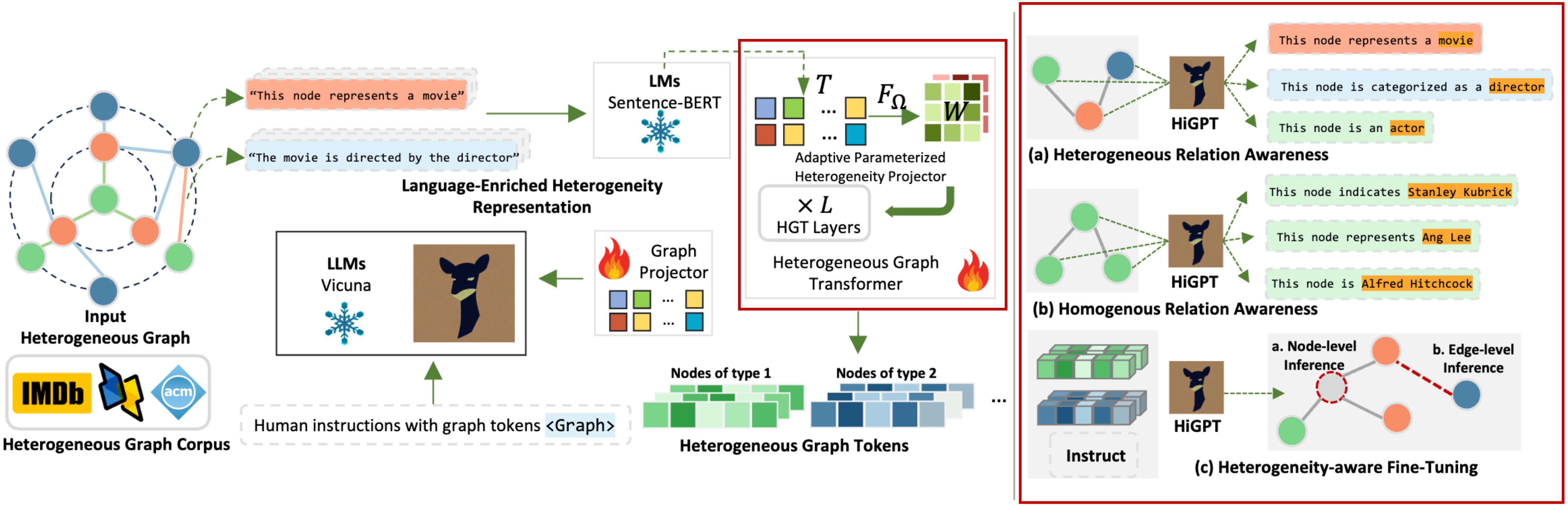
香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Challenges:**
  - ➢ **RQ1**: How can we deal with relation **type heterogeneity shift** across different heterogeneous graphs (One HG Model for All)?
  - ➢ **RQ2**: How can LLMs understand **complex heterogeneous graph structures** (node- and edge-types, structures)?
  - ➢ **RQ3**: How to address **data scarcity for model fine-tuning** for heterogeneous graph learning (few-shot, zero-shot).

TL;DR: **One Model for Any Heterogeneous Graph with Few Supervised Signals**

GNN as Prefix: HiGPT

- **Overview of HiGPT:**



GNN as Prefix: HiGPT

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **In-Context Heterogeneous Graph Tokenizer (RQ1):**
  - ➢ Graph Tokenization with Meta Projector
    - ✓ Graph Tokenization: with a Heterogeneous graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathcal{T}, \mathcal{R}, \mathbf{X})$, $\mathbf{H} = \text{HG-Tokenizer}(\mathbf{X}, \mathbf{A})$, where $\mathbf{X} = \{X_{T_i} \in \mathbb{R}^{|\mathcal{V}_{T_i}| \times d_{T_i}}\}$
    - ✓ HG-Tokenizer be implemented using various backbone HGNN, e.g. HGT.
    - ✓ Message Propagation and Aggregation of HGT:

$$\widetilde{h}_v^{(l)} = \bigoplus_{\forall u \in \mathcal{N}(v)} \left(\textbf{Attention}(u,e,v) \cdot \textbf{Message}(u,e,v)\right)$$

$$h_v^{(l)} = \mathcal{F}_{\Theta_1}^{\tau(v)}\left(\sigma\left(\widetilde{h}_v^{(l)}\right)\right) + h_v^{(l-1)}$$

$$= \mathbf{W}_1^{\tau(v)} \cdot \left(\sigma\left(\widetilde{h}_v^{(l)}\right)\right) + \mathbf{b}_1^{\tau(v)} + h_v^{(l-1)}$$

$$\textbf{Attention}(u,e,v) = \underset{\forall u \in \mathcal{N}(v)}{\text{Softmax}}\left(\big\Vert_{i \in [1,h]} \mathcal{F}_{\Theta_2}^{\tau(u)}\left(h_u^{(l-1)}\right) \mathbf{W}_1^{\rho(e)} \mathcal{F}_{\Theta_3}^{\tau(v)}\left(h_v^{(l-1)}\right)\right)$$

$$\textbf{Message}(u,e,v) = \big\Vert_{i \in [1,h]} \mathcal{F}_{\Theta_4}^{\tau(u)}\left(h_u^{(l-1)}\right) \mathbf{W}_2^{\rho(e)}$$

GNN as Prefix: HiGPT

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **In-Context Heterogeneous Graph Tokenizer (RQ1):**
  - Graph Tokenization with Meta Projector
    - ✓ Adaptive Parameterized Heterogeneity

$$\boxed{\Theta_i = \{\mathbf{W}_i^{\tau(v)}; \mathbf{b}_i^{\tau(v)}\} = \mathcal{F}_\Omega\left(\mathbf{T}^{\tau(v)}\right)}; \quad \boxed{\mathbf{W}_i^{\rho(e)} = \mathcal{F}_\Omega\left(\mathbf{T}^{\rho(e)}\right)}$$

Node-aware parameters

Edge-aware parameters

    - ✓ Language-Enriched Heterogeneity Representation

$$\mathbf{T}^{\tau(v)} = \text{Mean-Pooling}\left(\text{Sentence-BERT}\left(\mathbf{S}^{\tau(v)}\right)\right)$$

$$\mathbf{T}^{\rho(e)} = \text{Mean-Pooling}\left(\text{Sentence-BERT}\left(\mathbf{S}^{\rho(e)}\right)\right)$$

$$\mathbf{S}^{(\texttt{"movie"},\texttt{"to"},\texttt{"director"})} = \{$$
$$\texttt{"The movie is directed by the director"},$$
$$\texttt{"The film features direction by the director"}, \cdots\}$$

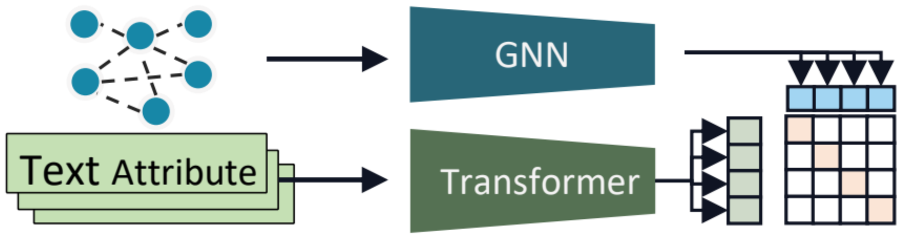
GNN as Prefix: HiGPT

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **In-Context Heterogeneous Graph Tokenizer (RQ1):**
  - Lightweight Text-Graph Contrastive Alignment

$$\hat{\mathbf{H}} = \text{norm}\left(\text{HG-Tokenizer}\left(\mathbf{X}\right)\right), \hat{\mathbf{T}} = \text{norm}\left(\text{LM-Tokenizer}\left(\mathbf{C}\right)\right)$$

$$\mathcal{L} = \frac{1}{2}\left(\text{CE}(\Lambda, \mathbf{y}) + \text{CE}(\Lambda^\top, \mathbf{y})\right), \Lambda = (\hat{\mathbf{H}}\hat{\mathbf{T}}^\top) \cdot \exp(\tau)$$



GNN as Prefix: HiGPT

- **Heterogeneous Graph Instruction Tuning (RQ2):**
  - ➢ Instruction Tuning with Heterogeneous Graph Corpus:
    - ✓ Heterogeneous Relation Awareness: inter-type token matching.
    - ✓ Homogeneous Relation Awareness: intra-type matching.

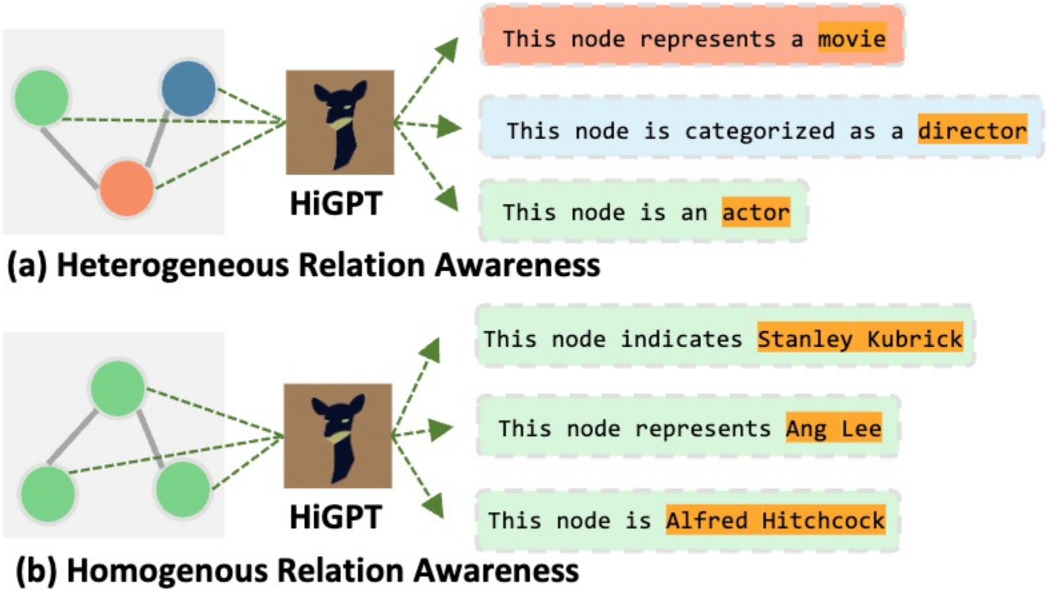
(a) Heterogeneous Instruction Pre-training

Given a heterogeneous graph about movies, there are 3 types of nodes: <DESC>. By performing random sampling, a **heterogeneous subgraph** is obtained. Separately nodes of different types are: 1. <graph>, 2. <graph>... Please sequentially **provide the types** for the node sequences.

(b) Homogenous Instruction Pre-training

Given a heterogeneous graph about papers, there are 4 types of nodes: <DESC>. ..., a **heterogeneous subgraph** is obtained. The nodes for "paper" are: <graph>. Also, a list of textual descriptions for the papers are: <DESC>. Please **reorder the text list based on the order of graph tokens**.

Prompt Templates



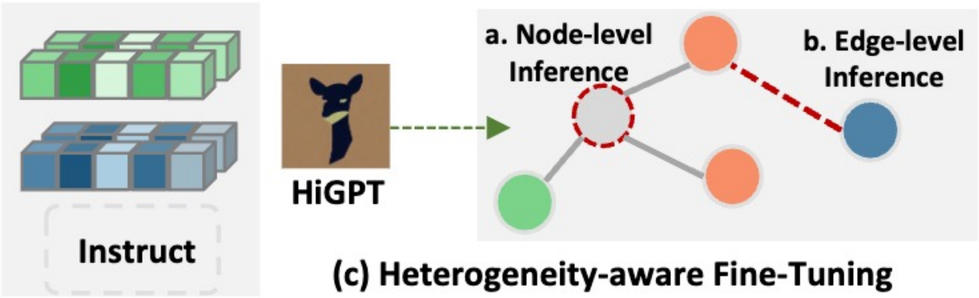
GNN as Prefix: HiGPT

- **Heterogeneous Graph Instruction Tuning (RQ2):**
  - ➢ Heterogeneity-aware Fine-Tuning:
    - ✓ customize the reasoning abilities of LLMs for specific downstream tasks.

(c) Heterogeneous Supervised Fine-Tuning

Given a heterogeneous graph about movies, there are 3 types of nodes: <DESC>. ..., a **heterogeneous subgraph** is obtained. There are nodes of different types: "movie" nodes: <graph>, <DESC> where the 0-th node is the central node. "actor" nodes: <graph>; "director" nodes: <graph>. **Which of the following classes does this movie belong to**: action, comedy, drama?

Prompt Templates

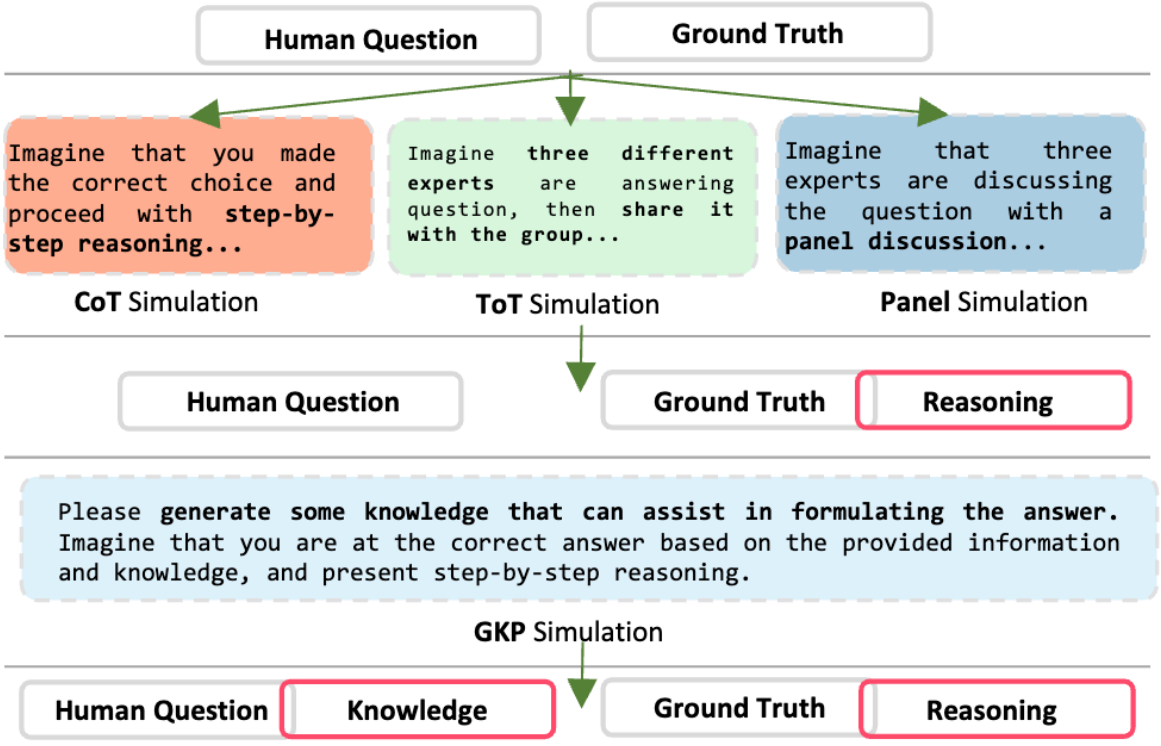


(c) Heterogeneity-aware Fine-Tuning

GNN as Prefix: HiGPT

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Mixture-of-Thought (MoT) for Graph Instruction Augmentation (RQ3):**
  - ➢ Mixture-of-Thought (MoT) Prompting:
    - ✓ Chain-of-Thought (CoT)
    - ✓ Tree-of-Thought (ToT)
    - ✓ PanelGPT
    - ✓ Generated Knowledge Prompting (GKP)
  - ➢ Instruction Augmentation with Priori Knowledge



GNN as Prefix: HiGPT

- **Mixture-of-Thought (MoT) for Graph Instruction Augmentation (RQ3):**
  - Mixture-of-Thought (MoT) Prompting:
  - Instruction Augmentation with Priori Knowledge

Prompting: I have a question as below: **{Human Question}** ; and the answer is **{Ground Truth}** ,**{MoT Simulations}**.
CoT Simulations: Imagine that you made the correct choice and proceed with step-by-step reasoning. Using the following format: Answer: …Reason: …
ToT Simulations: Imagine three different experts are answering question. All experts will write down 1 step of their thinking, then share it with the group. Then experts will go on to the next step. If any expert realizes they're wrong then they leave. Finally they make the correct choice.
Panel Simulations: Imagine that 3 experts are discussing the question with a panel discussion, trying to solve it step by step to make sure the result is correct and avoid penalty. And finally they make the correct choice.
ChatGPT Response: **{Answer&Reasoning}** **Augmented Instruction:** **{Human Question}** → **{Answer&Reasoning}**
GKP Simulations: Please generate some knowledge that can assist in formulating an answer, including, …. Imagine that you have arrived at the correct answer based on the provided information and knowledge, and present a step-by-step reasoning.
ChatGPT Response: **{Knowledge}+{Answer&Reasoning}** **Augmented Instruction:** **{Human Question}+{Knowledge}** → **{Answer&Reasoning}**

GNN as Prefix: HiGPT

- **Supervised and Zero-shot Performance Comparison:**

Datasets	Metric	train-on	test-on	SAGE	GAT	HAN	HGT	HetGNN	DMGI	HGMAE	HeCo	HiGPT-std	HiGPT-cot
Supervised	Mi-F1	IMDB-1	IMDB-1000	0.4663±0.0025	0.4567±0.0122	0.4890±0.0271	0.4977±0.0186	0.4790±0.0134	0.4570±0.0126	0.3609±0.0145	0.3874±0.0159	0.5090±0.0073	**0.5360±0.0065**
		IMDB-5	IMDB-1000	0.5010±0.0051	0.5170±0.0029	0.4840±0.0094	0.5003±0.0093	0.5020±0.0045	0.4413±0.0173	0.3652±0.0062	0.3385±0.0169	0.6180±0.0027	**0.6320±0.0085**
		IMDB-20	IMDB-1000	0.5930±0.0093	0.6117±0.0012	0.5763±0.0046	0.5750±0.0065	0.5957±0.0054	0.5497±0.0256	0.4107±0.0106	0.3781±0.0148	0.6090±0.0255	**0.6440±0.0075**
		IMDB-40	IMDB-1000	0.6170±0.0112	0.6261±0.0015	0.6198±0.0025	0.5923±0.0040	0.6177±0.0046	0.5813±0.0033	0.3946±0.0067	0.3927±0.0134	0.6260±0.0057	**0.6280±0.0071**
	Ma-F1	IMDB-1	IMDB-1000	0.4425±0.0068	0.3974±0.0183	0.4229±0.0104	0.4020±0.0112	0.4456±0.0036	0.4083±0.0288	0.3573±0.0117	0.4023±0.0137	0.4986±0.0141	**0.5247±0.0061**
		IMDB-5	IMDB-1000	0.4613±0.0086	0.4767±0.0098	0.4695±0.0037	0.4676±0.0153	0.4677±0.0145	0.4254±0.0124	0.3500±0.0080	0.3468±0.0213	0.6111±0.0091	**0.6243±0.0060**
		IMDB-20	IMDB-1000	0.5953±0.0095	0.6121±0.0024	0.5756±0.0051	0.5723±0.0056	0.5969±0.0055	0.5495±0.0270	0.4065±0.0089	0.3904±0.0172	0.6068±0.0146	**0.6398±0.0083**
		IMDB-40	IMDB-1000	0.6182±0.0107	0.6254±0.0009	0.6224±0.0057	0.5909±0.0068	0.6234±0.0038	0.5786±0.0064	0.3866±0.0072	0.3988±0.0147	**0.6265±0.0090**	0.6237±0.0059
	AUC	IMDB-1	IMDB-1000	0.6079±0.0061	0.6151±0.0065	0.6234±0.0252	0.6249±0.0170	0.6107±0.0075	0.5780±0.0130	0.5274±0.0058	0.5712±0.0099	0.6565±0.0146	**0.6685±0.0037**
		IMDB-5	IMDB-1000	0.6309±0.0049	0.6372±0.0012	0.6102±0.0059	0.6197±0.0152	0.6290±0.0022	0.5832±0.0132	0.5262±0.0041	0.5067±0.0228	0.7308±0.0125	**0.7310±0.0086**
		IMDB-20	IMDB-1000	0.6976±0.0064	0.7122±0.0020	0.6815±0.0052	0.6801±0.0048	0.7005±0.0030	0.6657±0.0179	0.5766±0.0064	0.5541±0.0145	0.7227±0.0034	**0.7424±0.0113**
		IMDB-40	IMDB-1000	0.7171±0.0069	0.7210±0.0014	0.7204±0.0015	0.6970±0.0060	0.7145±0.0035	0.6860±0.0027	0.5488±0.0049	0.5653±0.0105	0.7323±0.0036	**0.7331±0.0074**
Zero-shot	Mi-F1	IMDB-1	DBLP-1000	0.2353±0.0372	0.1893±0.0373	0.2653±0.0203	0.2573±0.0519	0.2900±0.0638	-	-	-	0.3180±0.0072	**0.3500±0.0073**
		IMDB-5	DBLP-1000	0.2607±0.0082	0.2737±0.0176	0.2577±0.0094	0.2453±0.0458	0.2427±0.0452	-	-	-	0.3180±0.0044	**0.3620±0.0047**
		IMDB-20	DBLP-1000	0.2810±0.0289	0.2780±0.0033	0.2710±0.0000	0.2803±0.0208	0.2333±0.0353	-	-	-	0.3840±0.0088	**0.4180±0.0083**
		IMDB-40	DBLP-1000	0.2400±0.0324	0.2847±0.0053	0.2710±0.0000	0.2937±0.0005	0.2027±0.0345	-	-	-	0.3320±0.0087	**0.3630±0.0045**
	Ma-F1	IMDB-1	DBLP-1000	0.0963±0.0132	0.1169±0.0089	0.1047±0.0063	0.1016±0.0169	0.1778±0.0629	-	-	-	0.2048±0.0068	**0.2472±0.0070**
		IMDB-5	DBLP-1000	0.1042±0.0028	0.1291±0.0145	0.1024±0.0030	0.1138±0.0296	0.0971±0.0148	-	-	-	0.1917±0.0046	**0.2773±0.0085**
		IMDB-20	DBLP-1000	0.1448±0.0573	0.1274±0.0060	0.1066±0.0000	0.1143±0.0116	0.1008±0.0191	-	-	-	0.3142±0.0074	**0.3733±0.0051**
		IMDB-40	DBLP-1000	0.1068±0.0060	0.1588±0.0078	0.1066±0.0000	0.1268±0.0105	0.0984±0.0161	-	-	-	0.2331±0.0069	**0.2912±0.0056**
	AUC	IMDB-1	DBLP-1000	0.4999±0.0001	0.4513±0.0295	0.5000±0.0000	0.5000±0.0000	0.5206±0.0306	-	-	-	0.5222±0.0069	**0.5406±0.0040**
		IMDB-5	DBLP-1000	0.4978±0.0030	0.4908±0.0078	0.5000±0.0000	0.5031±0.0043	0.4998±0.0003	-	-	-	0.5184±0.0081	**0.5493±0.0091**
		IMDB-20	DBLP-1000	0.5154±0.0213	0.4918±0.0020	0.5000±0.0000	0.5011±0.0016	0.4957±0.0060	-	-	-	0.5669±0.0041	**0.5907±0.0089**
		IMDB-40	DBLP-1000	0.5027±0.0031	0.4976±0.0021	0.5000±0.0000	0.5008±0.0006	0.4884±0.0164	-	-	-	0.5296±0.0070	**0.5508±0.0086**
	Mi-F1	IMDB-1	ACM-1000	0.3293±0.0418	0.3567±0.0053	0.3407±0.0111	0.3240±0.0014	0.3743±0.0434	-	-	-	0.4160±0.0106	**0.4540±0.0089**
		IMDB-5	ACM-1000	0.3820±0.0113	0.3787±0.0057	0.3630±0.0086	0.3160±0.0169	0.3583±0.0198	-	-	-	0.4580±0.0173	**0.4880±0.0131**
		IMDB-20	ACM-1000	0.2807±0.0074	0.3013±0.0188	0.3133±0.0031	0.3530±0.0000	0.2840±0.0226	-	-	-	**0.5080±0.0129**	0.5030±0.0064
		IMDB-40	ACM-1000	0.3173±0.0005	0.2393±0.0144	0.2697±0.0194	0.3560±0.0099	0.3180±0.0016	-	-	-	0.4750±0.0149	**0.5050±0.0077**
	Ma-F1	IMDB-1	ACM-1000	0.2647±0.0269	0.2908±0.0131	0.2250±0.0416	0.1631±0.0005	0.3139±0.0468	-	-	-	0.3949±0.0078	**0.4177±0.0124**
		IMDB-5	ACM-1000	0.3208±0.0130	0.3009±0.0137	0.2782±0.0026	0.1969±0.0301	0.3087±0.0225	-	-	-	0.4336±0.0085	**0.4510±0.0114**
		IMDB-20	ACM-1000	0.2694±0.0091	0.2422±0.0098	0.2412±0.0050	0.2094±0.0501	0.2715±0.0181	-	-	-	**0.4964±0.0075**	0.4877±0.0070
		IMDB-40	ACM-1000	0.3117±0.0017	0.2141±0.0071	0.2313±0.0132	0.2749±0.0122	0.3144±0.0017	-	-	-	0.4176±0.0116	**0.4585±0.0089**
	AUC	IMDB-1	ACM-1000	0.4934±0.0247	0.5248±0.0038	0.5128±0.0086	0.5000±0.0000	0.5318±0.0295	-	-	-	0.5672±0.0040	**0.5969±0.0082**
		IMDB-5	ACM-1000	0.5433±0.0082	0.5415±0.0047	0.5282±0.0073	0.4950±0.0134	0.5256±0.0145	-	-	-	0.5991±0.0103	**0.6224±0.0054**
		IMDB-20	ACM-1000	0.4601±0.0048	0.4772±0.0137	0.4877±0.0029	0.5038±0.0053	0.4625±0.0163q	-	-	-	**0.6352±0.0094**	0.6318±0.0068
		IMDB-40	ACM-1000	0.4867±0.0013	0.4320±0.0108	0.4545±0.0146	0.5148±0.0043	0.4872±0.0006	-	-	-	0.6138±0.0047	**0.6360±0.0051**

GNN as Prefix: HiGPT

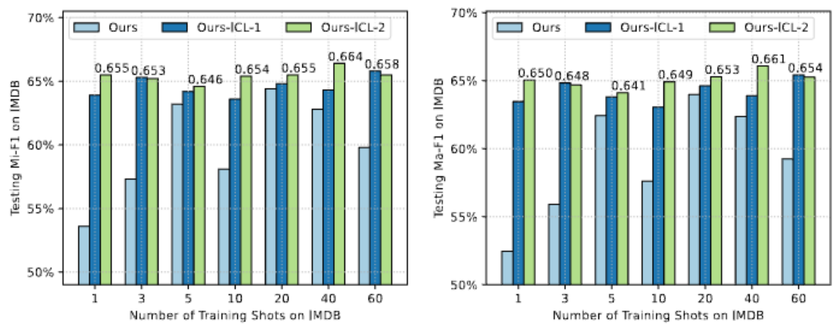
香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Graph In-Context Learning:**
  - ➢ 1-shot Beat 60-shot with Graph ICL
  - ➢ Enhanced Transferability with our Graph ICL
  - ➢ Benefit of Irrelevant Graph Examples

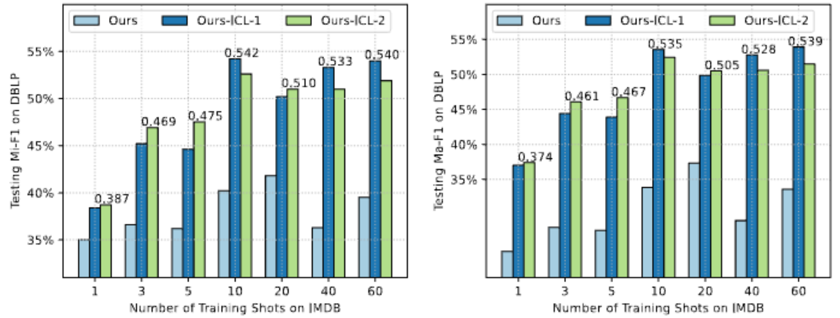
IMDB-ICL-1:
Q: Given a heterogeneous graph about internet movie… **{Human Question}**
A: **{Ground Truth Answer&Reasoning}**
Q: Given a heterogeneous graph about internet movie… **{Human Question}**
IMDB-ICL-2:
Q: Given a heterogeneous graph about internet movie… **{Human Question}**
A: **{Ground Truth Answer&Reasoning}**
Q: Given a heterogeneous graph about internet movie… **{Human Question}**
A: **{Ground Truth Answer&Reasoning}**
Q: Given a heterogeneous graph about internet movie… **{Human Question}**
ACM-ICL-DBLP:
Q: Given a heterogeneous academic network graph about computer science from DBLP website … **{Human Question}**
A: **{Ground Truth Answer&Reasoning}**
Q: Given a heterogeneous academic network graph about computer science collected from ACM website… **{Human Question}**

With Graph ICL

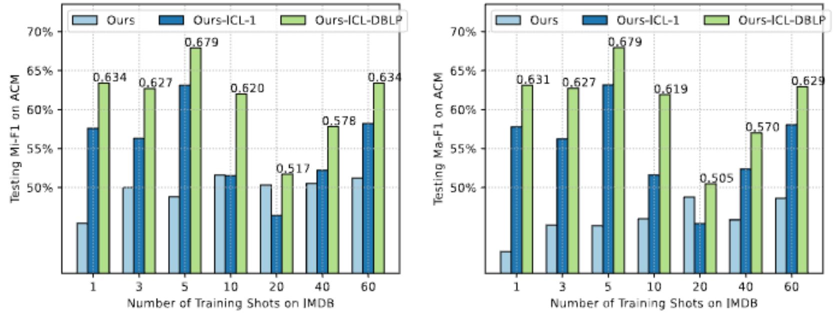
Without optimization



(a) IMDB-IMDB@Mi-F1, Ma-F1



(b) IMDB-DBLP@Mi-F1, Ma-F1



(c) IMDB-ACM@Mi-F1, Ma-F1

GNN as Prefix: HiGPT

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Case Study:**

Question: Given a heterogeneous graph about internet movie, there are three types of nodes, namely: movie, actor, director. The relationships (meta paths) between different nodes include: [movie is directed by director], [movie has actor]. By performing random sampling of 2-hop 10 neighbors centered on the target movie node, a heterogeneous subgraph is obtained. In the subgraph, "movie" nodes: <graph>, where the 0-th node is the central node that represents a movie with the following information: Name: Live Free or Die Hard; Director's name: Len Wiseman; Actors' names: Bruce Willis|Jonathan Sadowski|Cyril Raffaelli; Plot keywords: fbi|hacker|independence day|police|terrorist; "actor" nodes: <graph>; "director" nodes: <graph>. Question: Which of the following classes does this movie belong to: action, comedy, drama?

Ground-Truth Category: Action

+ Give likely categories directly. **HiGPT (Direct Response):** Action.

+ Please think about the categorization in a step by step manner and avoid making false associations. Then provide your reasoning.
HiGPT (CoT Response without Format Constraint): Answer: Action Reasoning: The movie "Live Free or Die Hard" has several plot keywords that suggest an action genre, such as "fbi", "hacker", "independence day", "police", and "terrorist". These keywords imply that the movie is likely to have a lot of action scenes, chases, and possibly some thrilling moments. Additionally, the director Len Wiseman has directed other action movies in the past, such as "Underworld" and "Total Recall". This further supports the idea that "Live Free or Die Hard" belongs to the action genre. Finally, the actors in the movie also contribute to the action genre. Bruce Willis, who plays the lead role, is known for his roles in action movies such as "Die Hard", "Pulp Fiction", and "The Fifth Element". Jonathan Sadowski and Cyril Raffaelli, who also appear in the movie, have acted in other action movies and TV shows. Overall, based on the plot keywords, the director's past work, and the actors' roles, it is reasonable to conclude that "Live Free or Die Hard" belongs to the action genre.

+ Please think about the categorization in a step by step manner and avoid making false associations. Then provide your reasoning. Using the following format: Answer: [The answer] Reason: ...
HiGPT (CoT Response with Format Constraint): Answer: Action Reason: Based on the plot keywords provided, the movie seems to involve elements of action, such as the FBI, hackers, police, and terrorists. These keywords suggest that the movie may have a high level of action and suspense, which is a common characteristic of action movies. Additionally, the presence of Bruce Willis, a well-known action star, further supports the classification of this movie as an action film.

+ Imagine three different experts are answering this question. All experts will write down 1 step of their thinking, then share it with the group. Then all experts will go on to the next step, etc. If any expert realizes they're wrong at any point then they leave.
HiGPT (ToT Response): Expert 1: First, I will analyze the plot keywords of the movie. The keywords "fbi", "hacker", "independence day", "police", and "terrorist" suggest that the movie might belong to the action genre. However, the presence of the word "independence day" could also indicate that the movie might be a combination of action and drama. Expert 2: Next, I will look at the director's name. Len Wiseman is known for directing the "Underworld" film series, which are primarily action movies. Therefore, it is likely that "Live Free or Die Hard" belongs to the action genre. Expert 3:Finally, I will examine the actors' names. Bruce Willis is an action star, and Jonathan Sadowski and Cyril Raffaelli are relatively unknown actors. Therefore, it is more likely that the movie belongs to the action genre. Based on the analysis of the plot keywords, the director's name, and the actors' names, it is likely that "Live Free or Die Hard" belongs to the action genre.

+ 3 experts are discussing the question with a panel discussion, trying to solve it step by step, and make sure the result is correct and avoid penalty
HiGPT (Panel Discussion Response): Expert 1: Let's start by analyzing the plot keywords. The keywords "fbi", "hacker", "independence day", "police", and "terrorist" suggest that this movie might belong to the action genre. Expert 2: That's a good point. However, we should also consider the director's previous works. Len Wiseman is known for directing the "Underworld" series, which are primarily action movies. Therefore, it's likely that this movie also belongs to the action genre. Expert 3: I agree with both of you. However, we should also consider the actors' names. Bruce Willis is known for his roles in action movies, and Jonathan Sadowski and Cyril Raffaelli are relatively unknown actors. Therefore, it's possible that this movie might also belong to the drama genre. Expert 1: That's a good point. We should consider all possible genres. Let's perform a random sampling of 2-hop 10 neighbors centered on the target movie node to obtain a heterogeneous subgraph. Expert 2: Great idea. Let's see what we can find. Expert 3: Agreed. After performing the random sampling, the experts found that the movie "Live Free or Die Hard" belongs to the action genre.

More Details

- More details could be found as below:
  - ➢ Project page: https://higpt-hku.github.io/
  - ➢ Paper: To be released on arXiv this week
  - ➢ Code: https://github.com/HKUDS/HiGPT
  - ➢ Huggingface: https://huggingface.co/Jiabin99/HiGPT

README Apache-2.0 license

HiGPT: Heterogeneous Graph Language Model

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin and Chao Huang*. (*Correspondence)

Data Intelligence Lab@University of Hong Kong, Baidu Inc.

Project Page | Demo Page | Paper PDF | Youtube • 中文博客

This repository hosts the code, data and model weight of **HiGPT**.

News

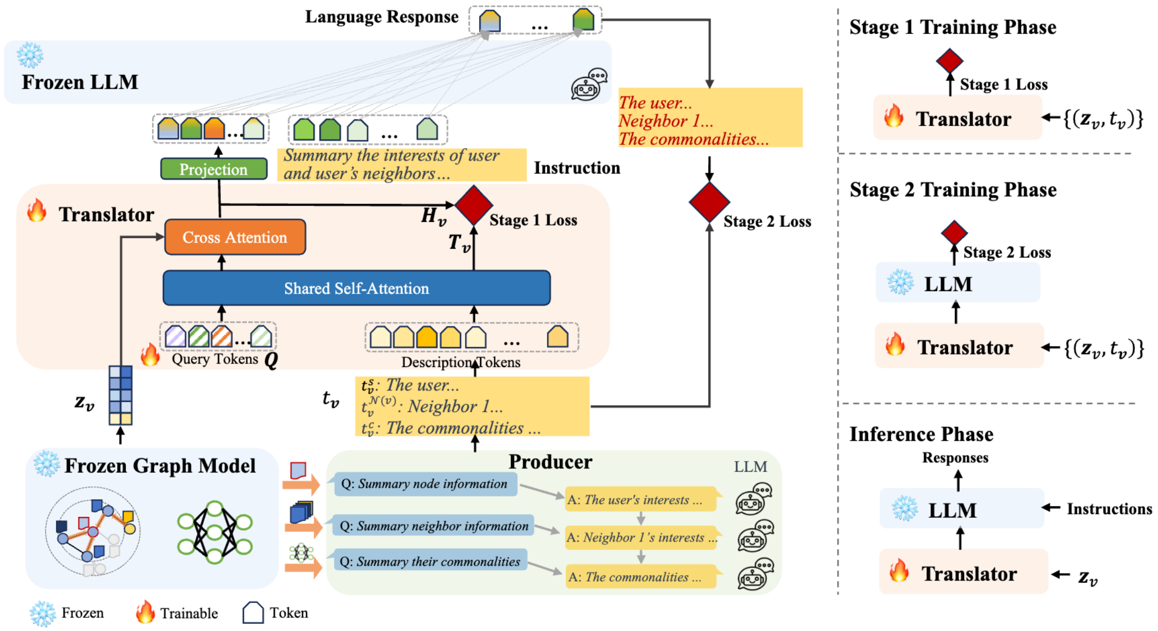
We have made significant updates to the **models** used in our HiGPT on **Huggingface**. We highly recommend referring to the table below for further details:

Huggingface Address	Description
https://huggingface.co/Jiabin99/In-Context-HGT	The trained in-context heterogeneous graph tokenizer using our lightweight text-graph contrastive alignment.
https://huggingface.co/Jiabin99/HiGPT	It's the checkpoint of our HiGPT based on Vicuna-7B-v1.5 tuned on 60 shots IMDB graph instruction data.

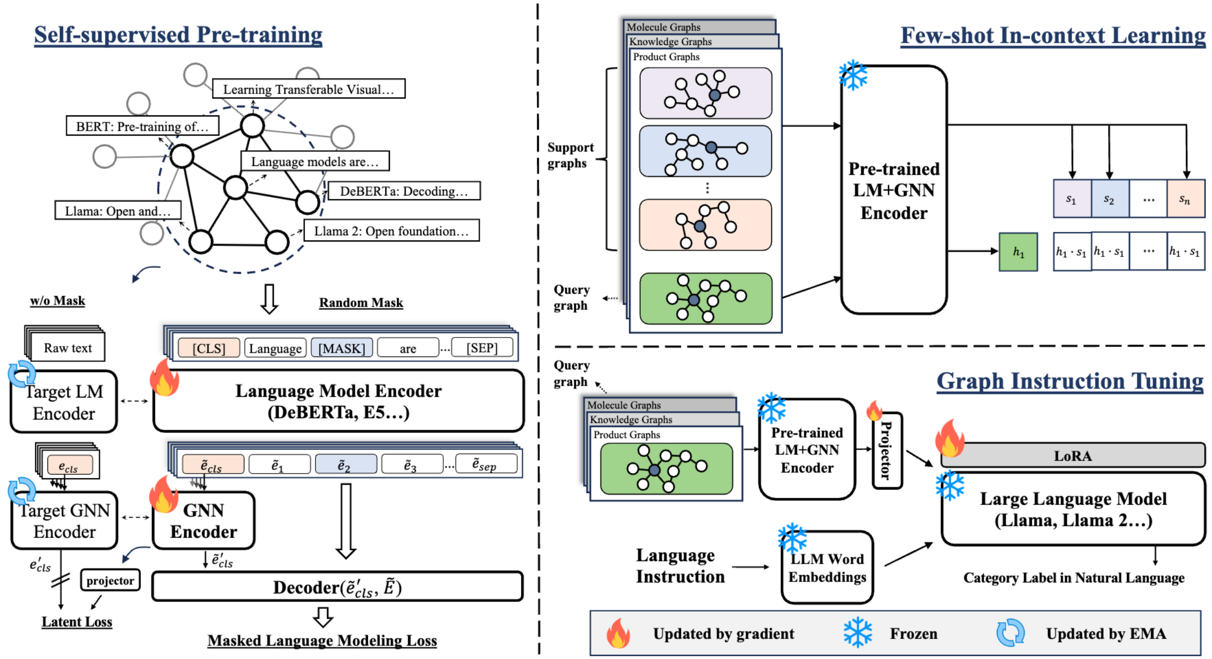
Don't be stingy with your stars!

Node-level Tokenization

香 港 大 學
THE UNIVERSITY OF HONG KONG



GraphTranslator

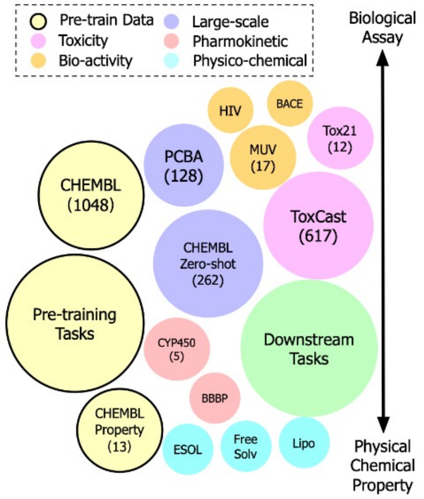


UniGraph

[GraphTranslator] Aligning Graph Model to Large Language Model for Open-ended Tasks

[UniGraph] Learning a Cross-Domain Graph Foundation Model From Natural Language

Node-level Tokenization

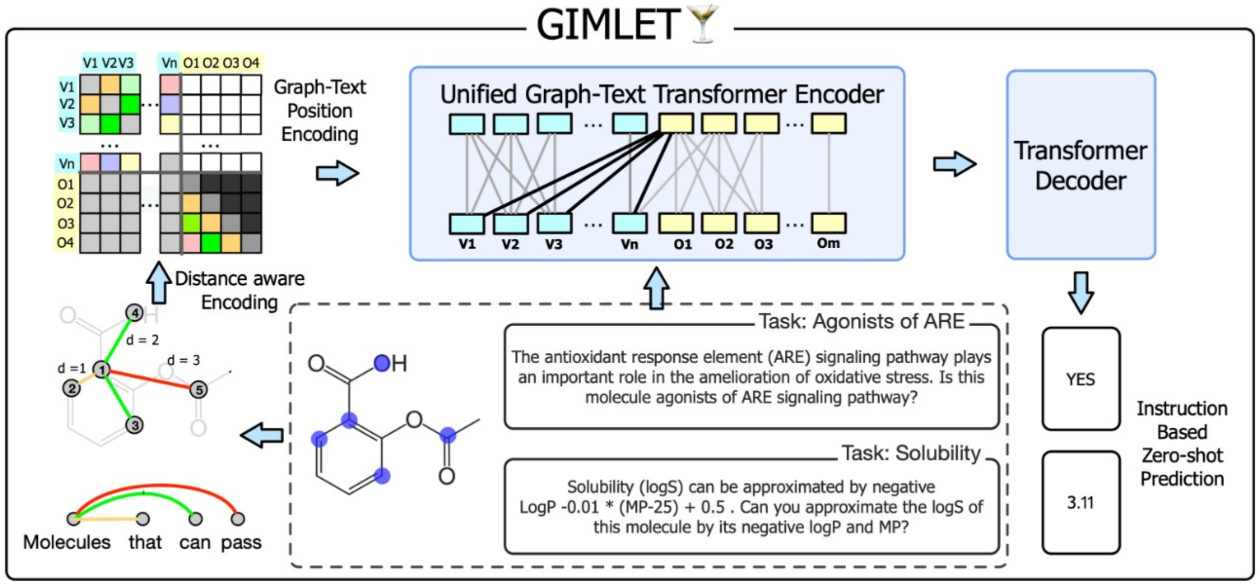


Heavy atoms counting (From Chembl property)
"Heavy atom refers to any atom that is not hydrogen. How many heavy atoms do the molecule have?"

Inhibitors of Schistosoma Mansoni Peroxiredoxins (From Chembl)
"The functional assay is named qHTS Assay for the Inhibitors of Schistosoma Mansoni Peroxiredoxins. It is related to two other pubchem assays, namely Confirmation Concentration-Response Assay for Inhibitors of the Schistosoma mansoni Redox Cascade and Schistosoma Mansoni Peroxiredoxins (Prx2) and thioredoxin glutathione reductase (TGR) coupled assay. The assay category is also confirmatory and it pertains to the Schistosoma mansoni organism. Is this molecule effective to the assay?"

Inhibition of receptor SF-1 (From MUV)
"The nuclear receptor SF-1 (steroidogenic factor-1) is expressed in the pituitary, testes, ovaries, and adrenal gland and regulates steroid hormone production at many levels, including direct regulation of expression of major P450 enzymes involved in steroid hormone synthesis. Is this molecule inhibitor of SF-1?"

Toxicity to ARE signaling pathway (From Tox21)
"Oxidative stress has been implicated in the pathogenesis of a variety of diseases ranging from cancer to neurodegeneration. The antioxidant response element (ARE) signaling pathway is important in the amelioration of oxidative stress. Is this molecule agonists of antioxidant response element (ARE) signaling pathway?"

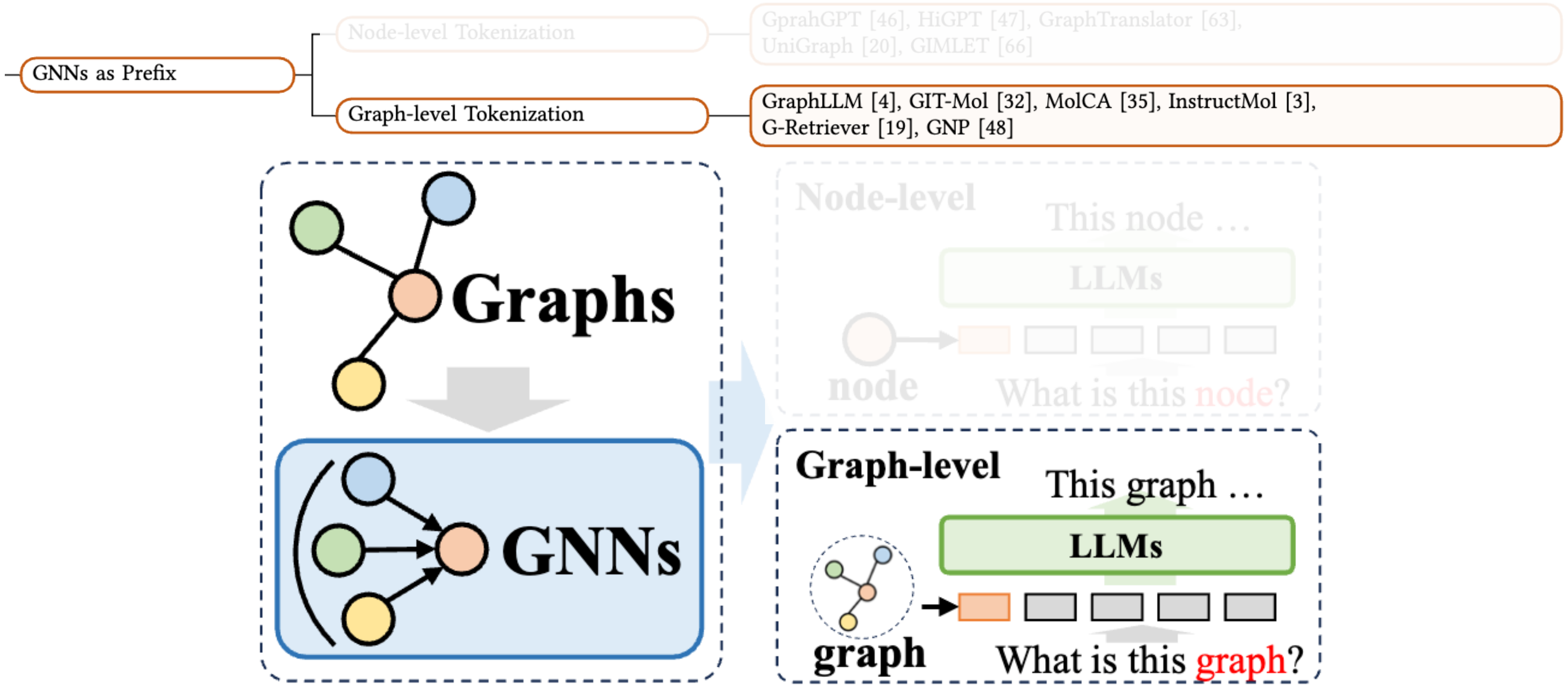


GIMLET

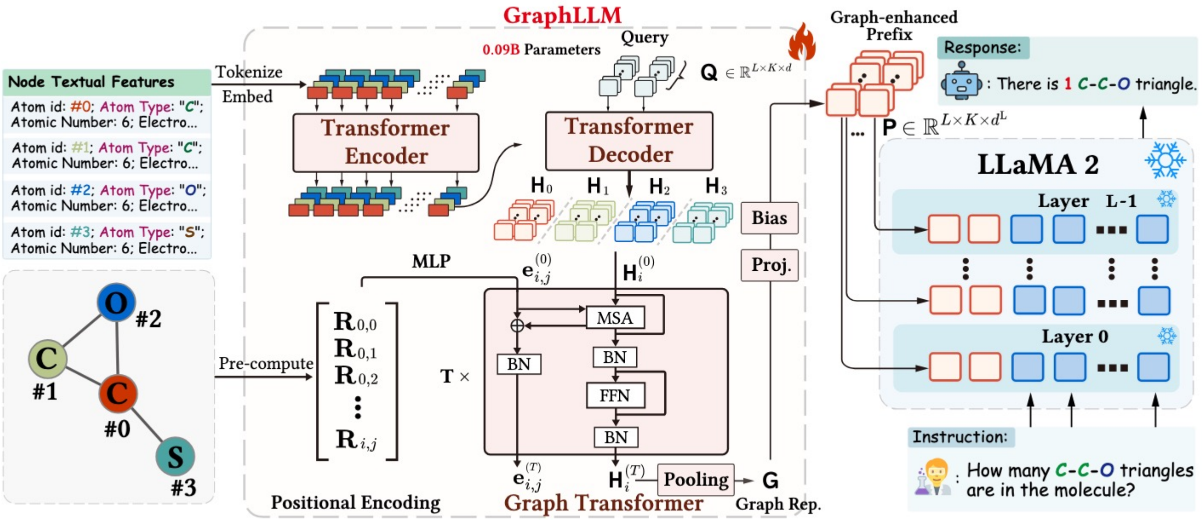
[GIMLET] A Unified Graph-Text Model for Instruction-Based Molecule Zero-Shot Learning

Graph-level Tokenization

- **Motivation**:
  - capture high-level global semantic information of the graph structure.
  - the graph is compressed into a fixed-length token sequence using a specific pooling method



Graph-level Tokenization



GraphLLM

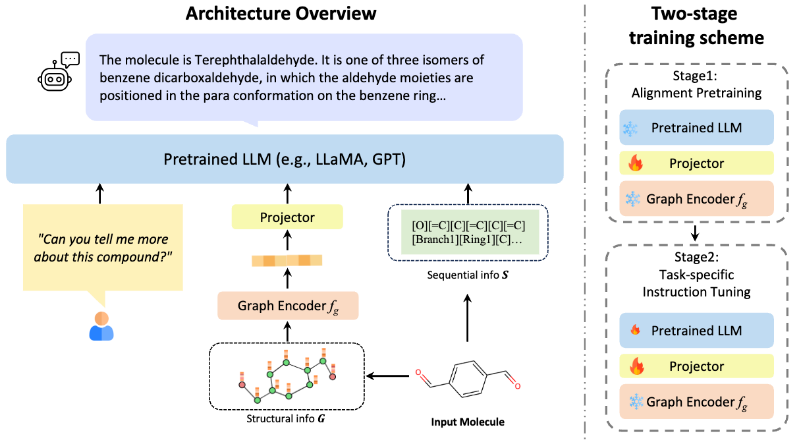


Figure 2. Overview of **InstructMol** model architecture design and two-stage training paradigm. The example molecule in the figure is *Terephthalaldehyde* [62] (CID 12173).

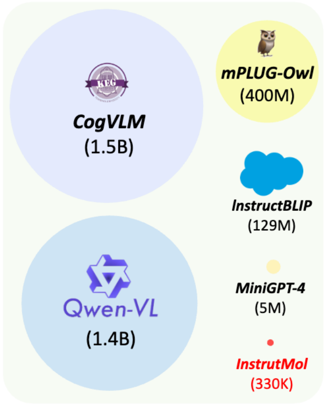


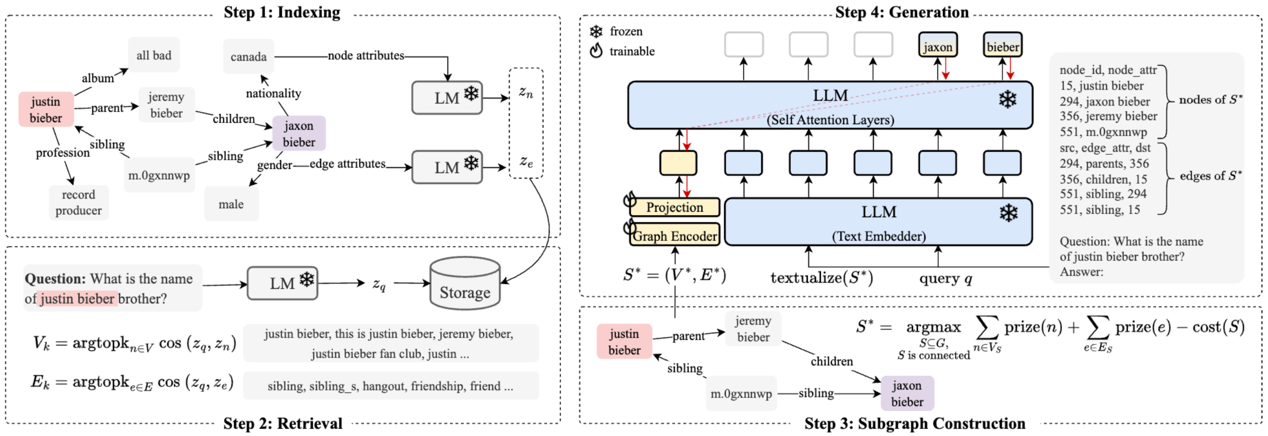
Figure 3. Comparison of biomolecule-domain molecule-text dataset scale with existing general domain vision-language datasets [4, 13, 71, 81, 87].

InstructMol

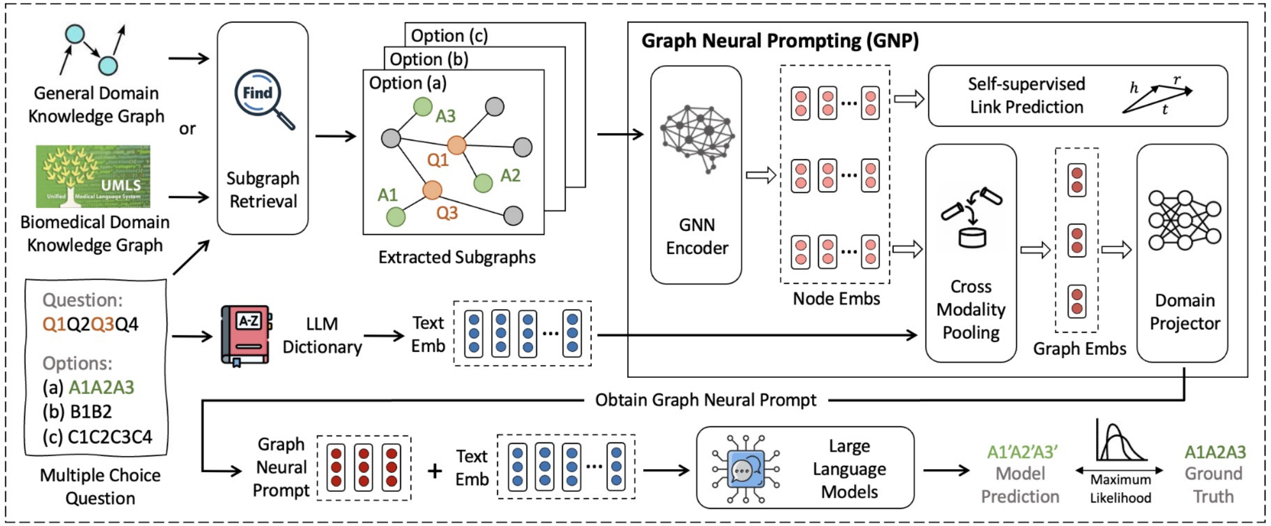
[GraphLLM] Boosting Graph Reasoning Ability of Large Language Model

[InstructMol] Multi-Modal Integration for Building a Versatile and Reliable Molecular Assistant in Drug Discovery

Graph-level Tokenization



G-Retriever



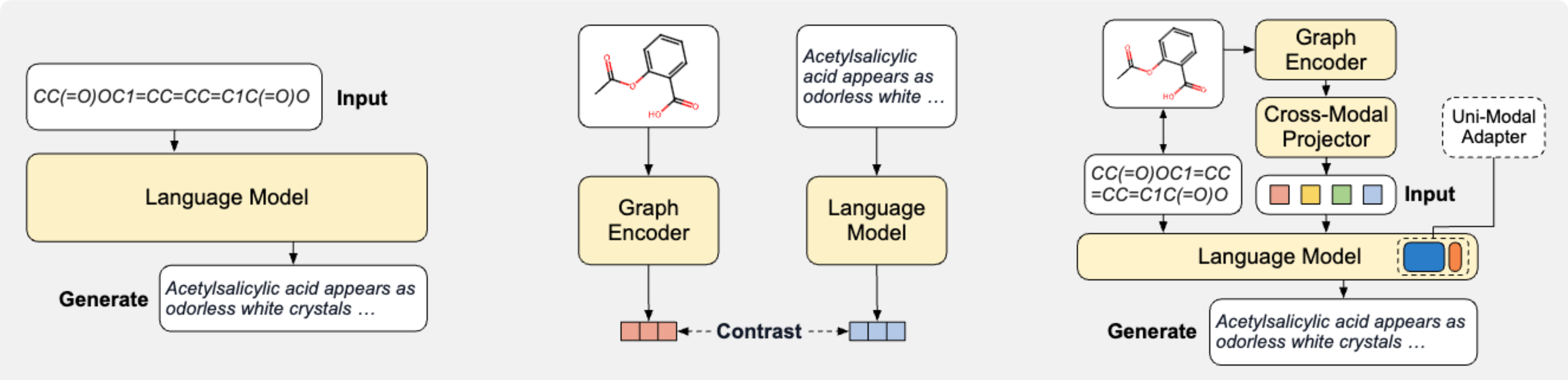
GNP

[G-Retriever] Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering

[GNP] Graph Neural Prompting with Large Language Models

GNN as Prefix: MolCA

- **MolCA:** Molecular Graph-Language Modeling with Cross-Modal Projector and Uni-Modal Adapter (EMNLP’23)
- Existing molecular language modeling methods



(a) 1D language modeling. (b) Cross-modal contrastive learning. (c) MolCA.

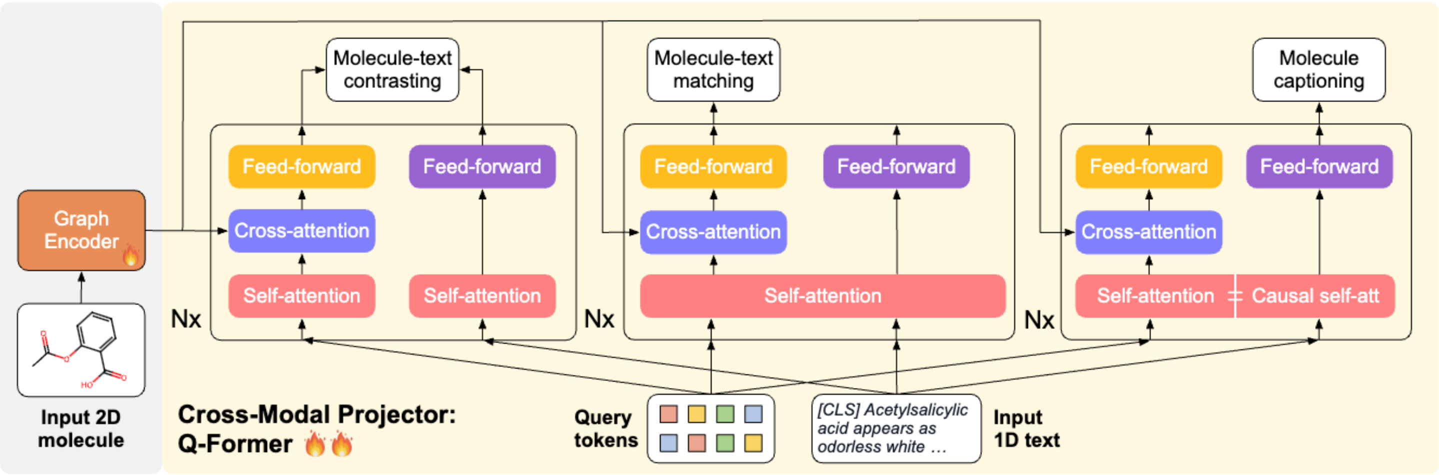
- Three-stage training pipeline:

Stage	Model Architecture	Downstream Task
Pretrain stage 1	Graph Encoder → Cross-Modal Projector	Cross-modal retrieval
Pretrain stage 2	Graph Encoder → Cross-Modal Projector → Language Model	-
Fine-tune stage	Graph Encoder → Cross-Modal Projector → Language Model	Generation tasks: Molecule caption, IUPAC name prediction

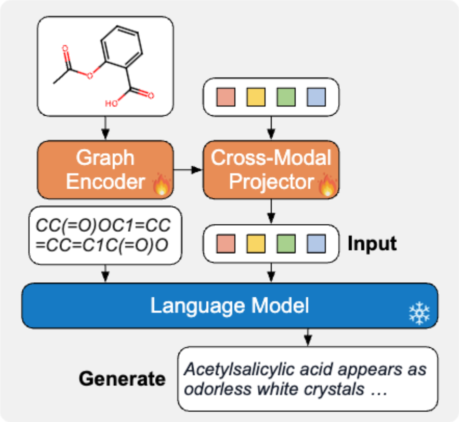
GNN as Prefix: MolCA

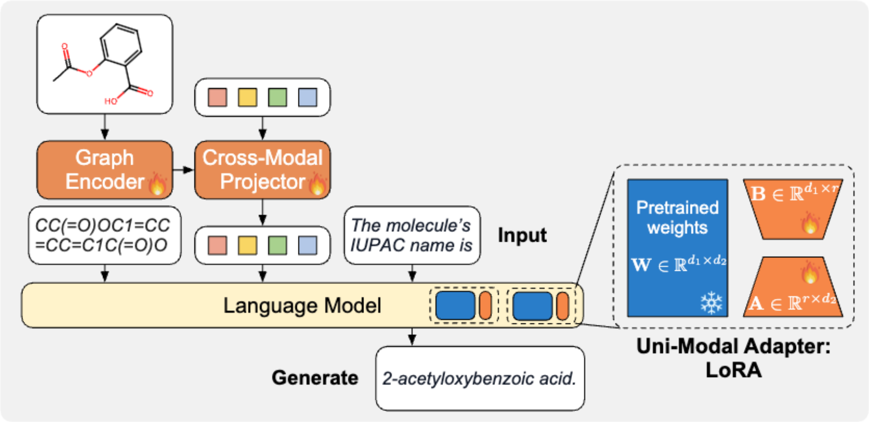
香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Pretrain stage 1**:



- **Pretrain stage 2** by molecule captioning & **Fine-tune stage** for molecule-to-text generation





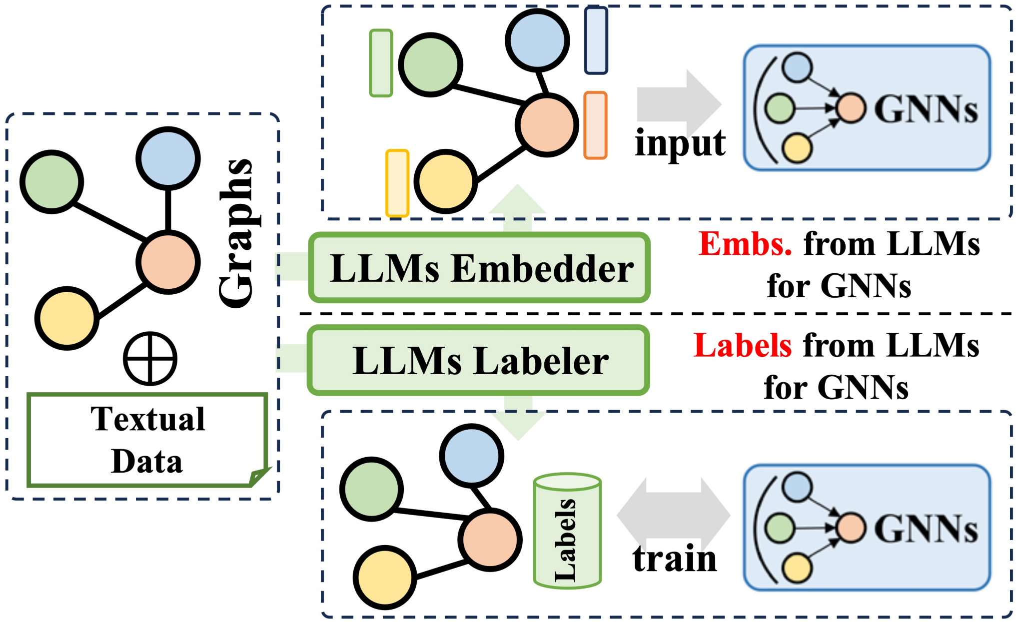
Subset	Size	Avg mol len	Min text len	Avg text len
Pretrain	298083	35	1	16
Train	12000	32	20	60
Valid	1000	32	20	61
Test	2000	31	20	60

LLM as Prefix

LLM→GNN

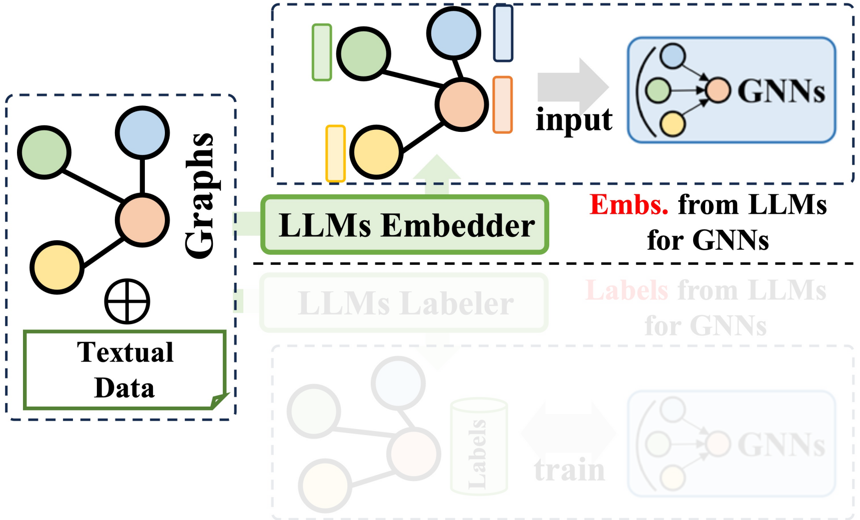
LLM as Prefix

- LLMs provide graph embeddings for GNNs
- LLMs provide graph labels for GNNs



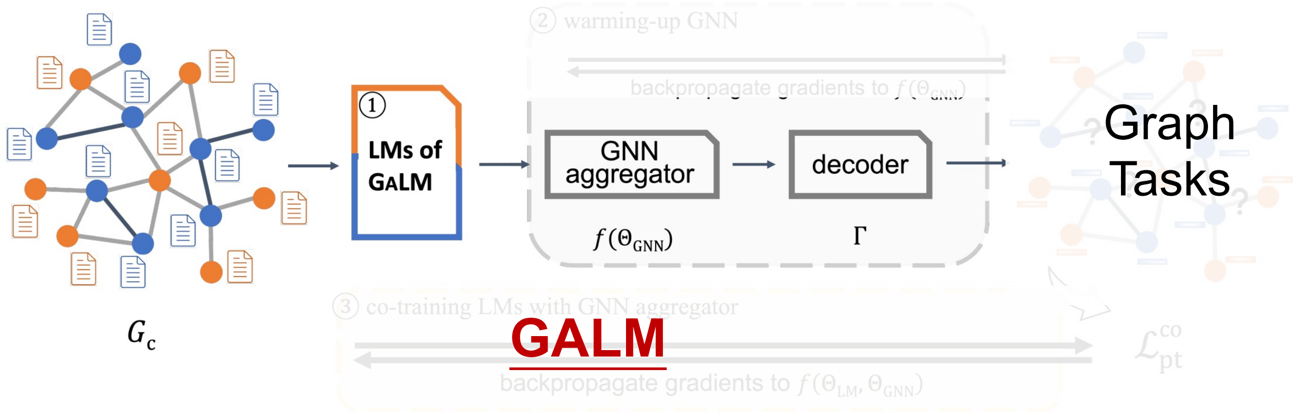
LLM as Embedder

- **Motivation**: diverse node embs/feats causing
  - Low generalization ability for GNNs
  - Low representation quality in the initial phase
- Leverage LLMs' powerful
  - Language summarization/processing abilities
  - Text representation abilities
- Text-Attributed Graphs (TAGs)

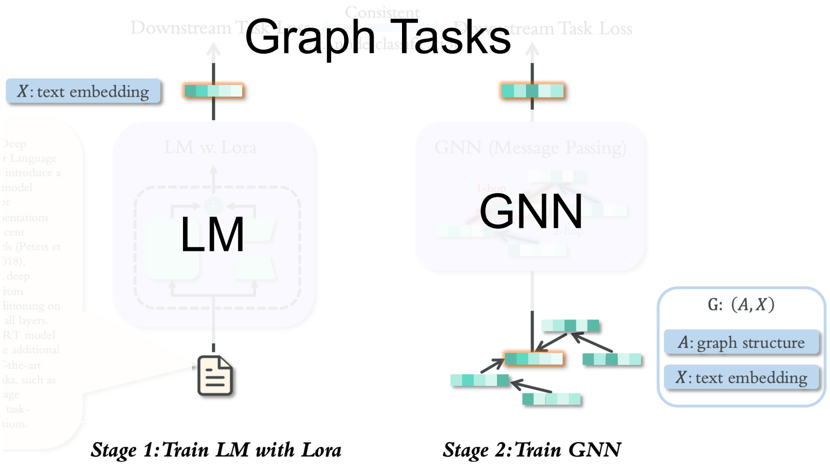


LLM as Embedder

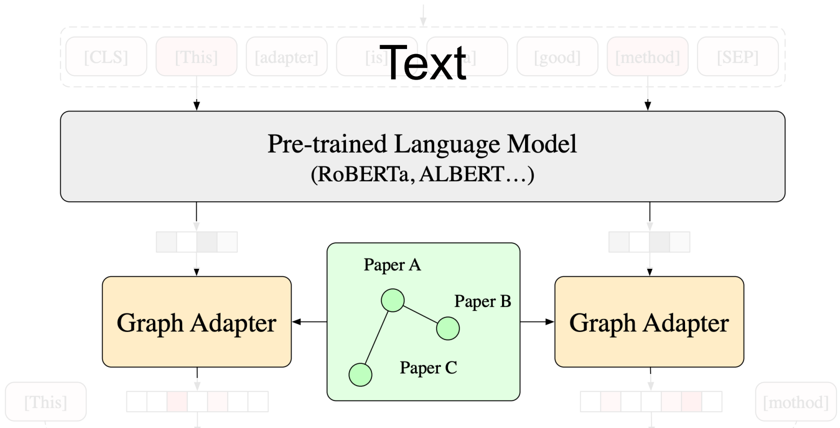
- LLM embs. as initial embs. of GNNs



GALM



SimTeG



G-Prompt

[G-Prompt] Prompt-based Node Feature Extractor for Few-shot Learning on Text-Attributed Graphs

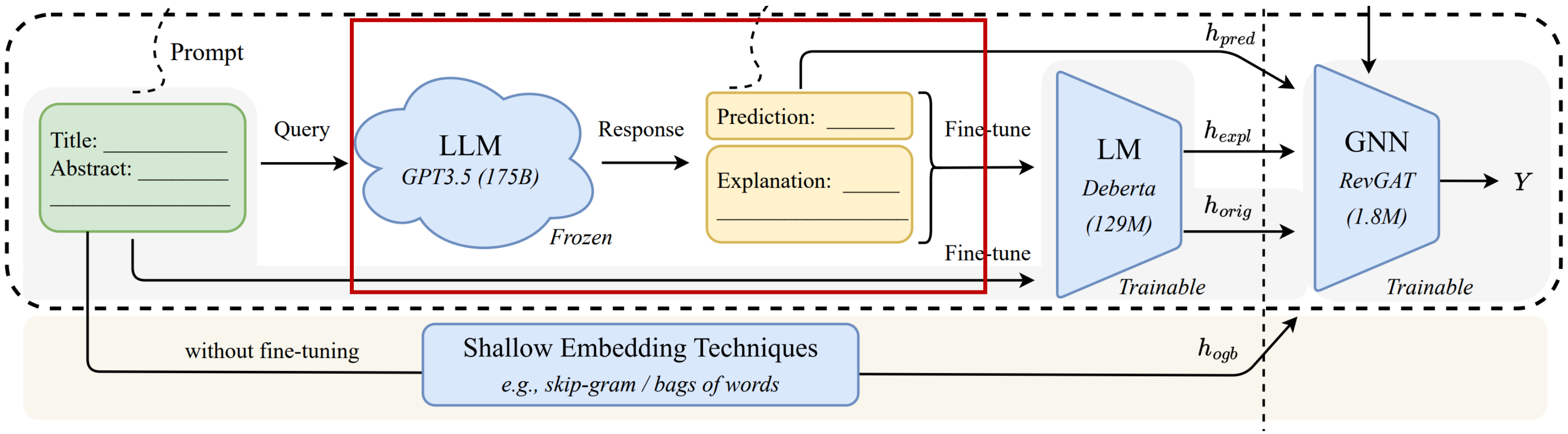
[SimTeG] Simteg: A frustratingly simple approach improves textual graph learning

[GALM] Graph-Aware Language Model Pre-Training on a Large Graph Corpus Can Help Multiple Graph Applications

LLM as Embedder

香港大學 THE UNIVERSITY OF HONG KONG

- Title, Abstract, Prediction & Explanation (TAPE)
- Leverage (L)LMs for
  - Text feature enhancement
  - Text-based embedding

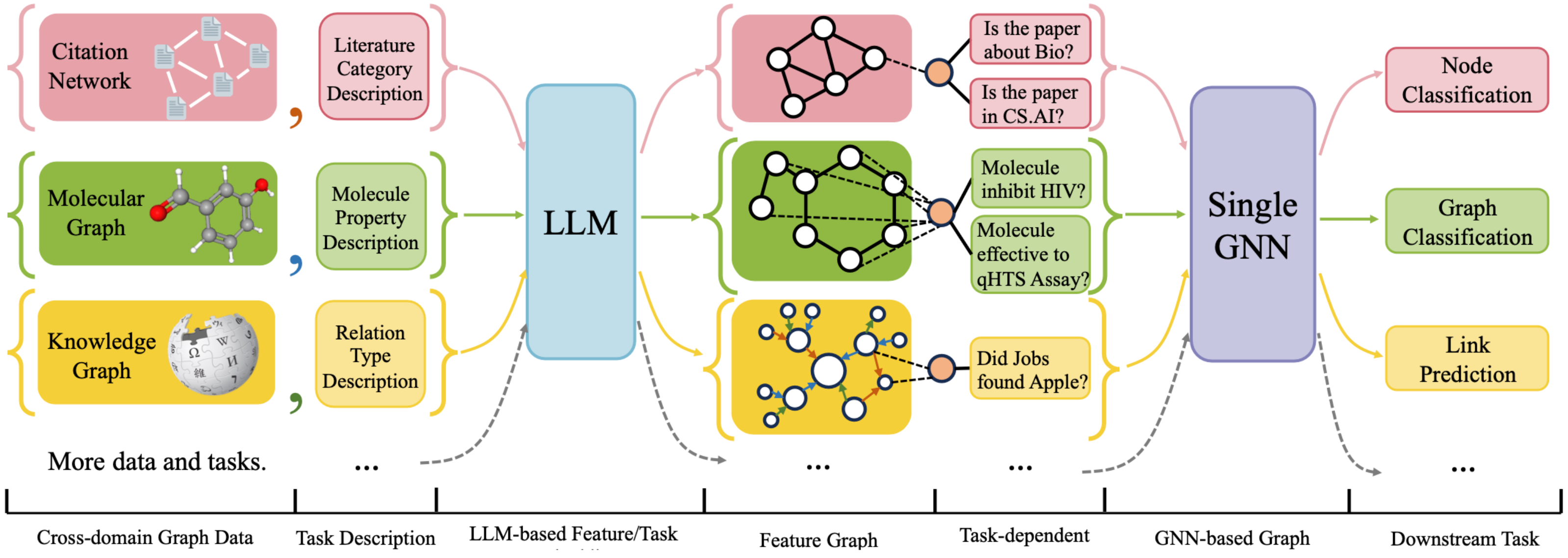


[TAPE] Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning

LLM as Embedder

香 港 大 學
THE UNIVERSITY OF HONG KONG

- One for All (OFA)
  - ➢LLM generates embeddings for features & tasks
  - ➢Unified formulation for different graph tasks

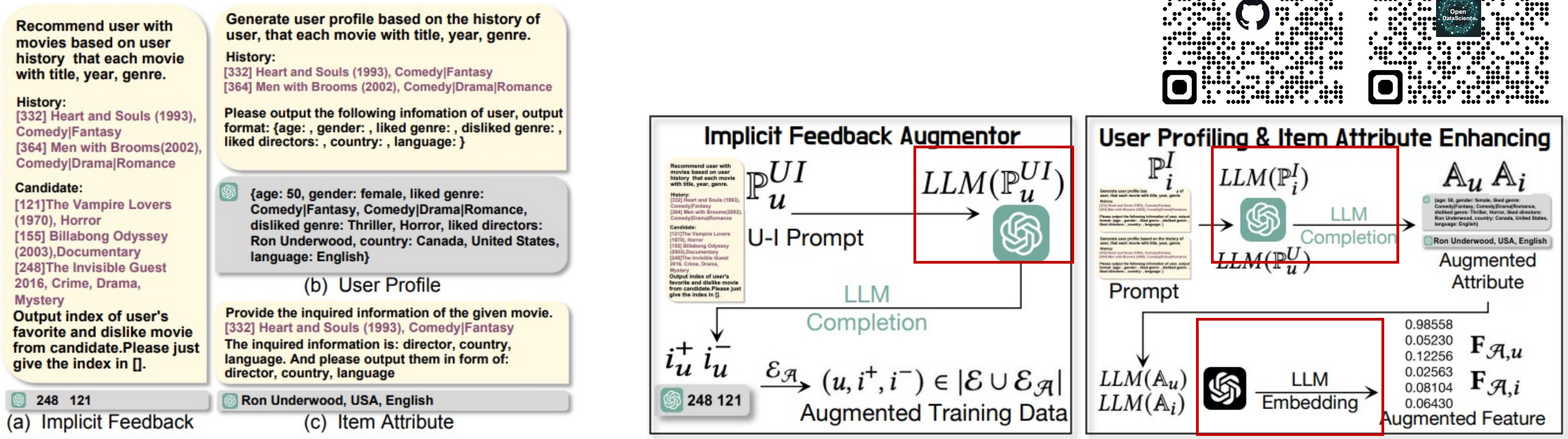


[OFA] One for All: Towards Training One Graph Model for All Classification Tasks

LLM as Embedder

香 港 大 學
THE UNIVERSITY OF HONG KONG

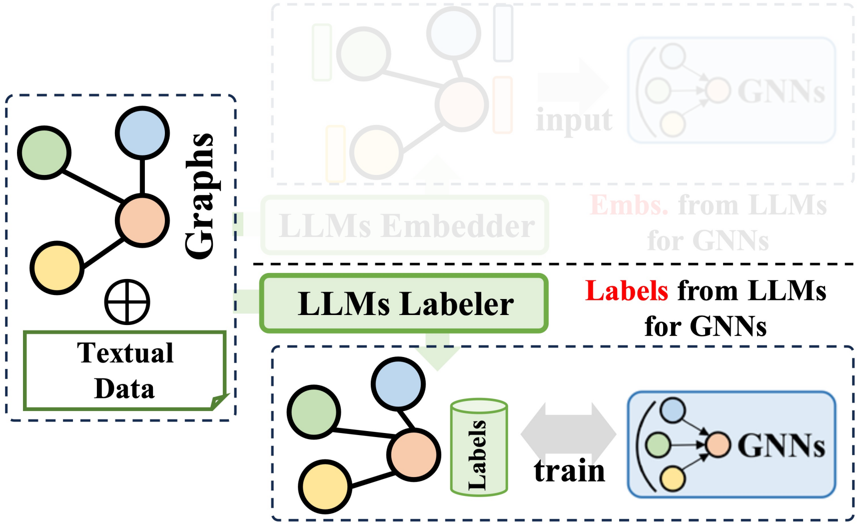
- LLMRec: use LLM to augment rec graphs in:
  - ➢user-item interaction edges, user/item node attributes
  - ➢Structure, text augmentation and embedding

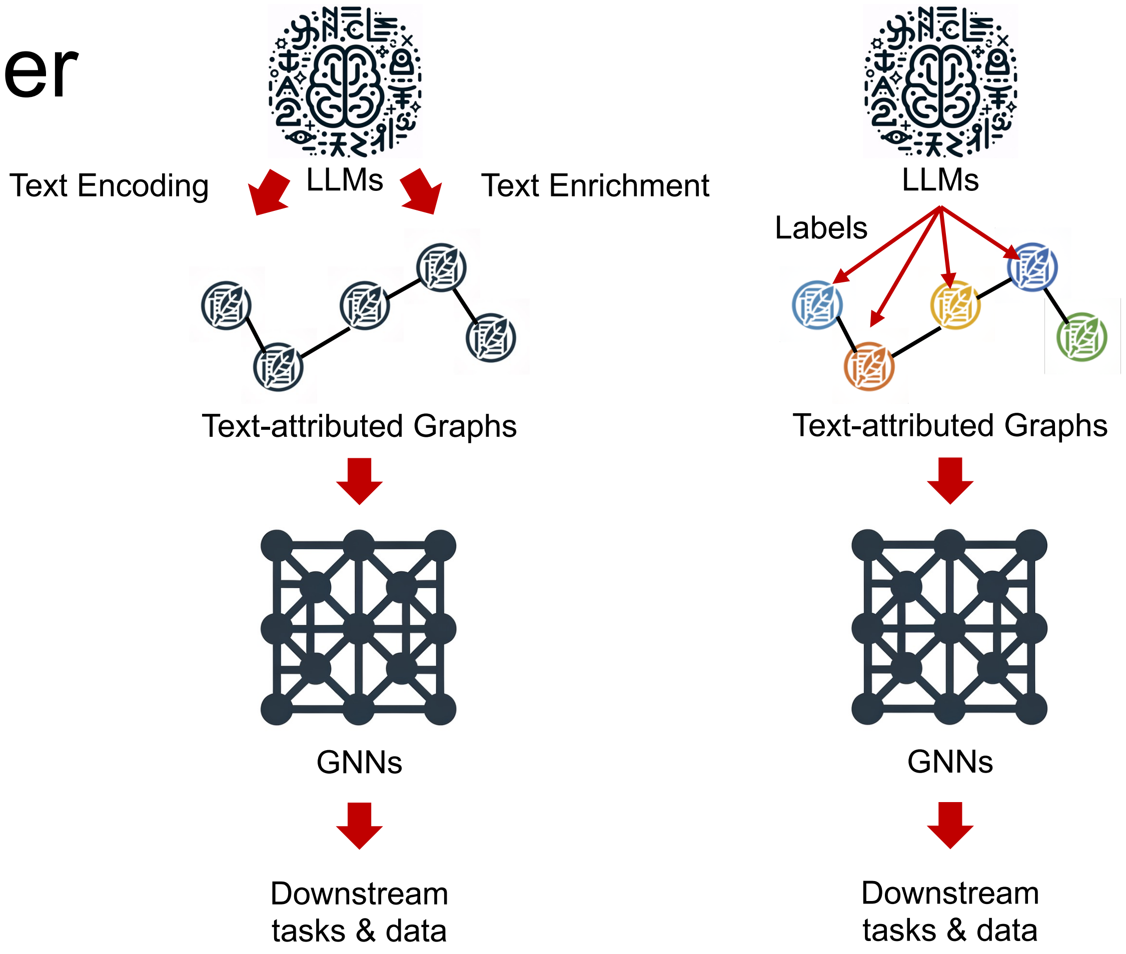


LLMRec: Large Language Models with Graph Augmentation for Recommendation

LLM as Labeler

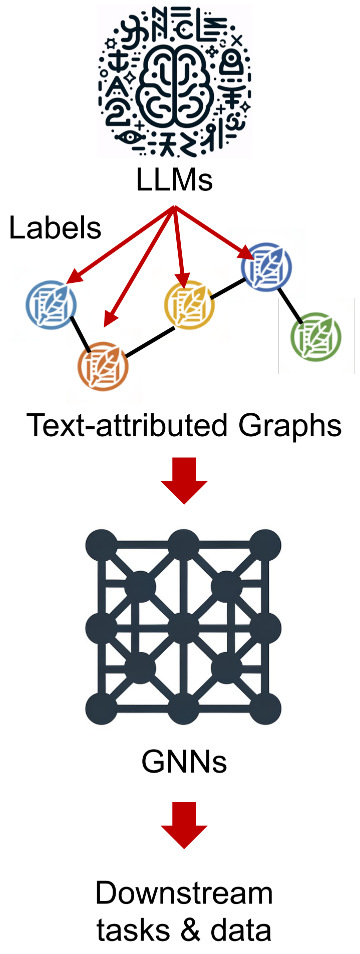
- Comparing to LLM as Embedder





LLM as Labeler

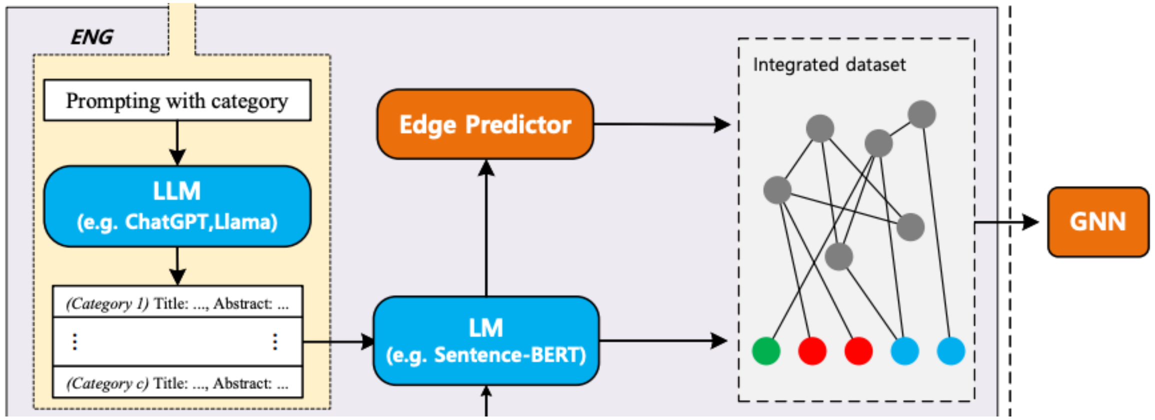
- **Motivation**: Graph labels are insufficient
  - Node class labels
  - Link labels
  - Representations are low-quality
- Label generation based on
  - Language understanding & reasoning
  - Quality semantic representation learning
  - Knowledge about the world

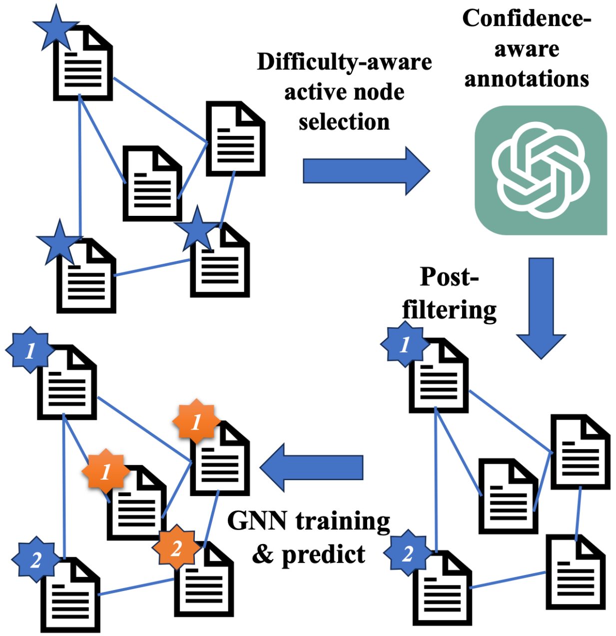


LLM as Labeler

香 港 大 學
THE UNIVERSITY OF HONG KONG

- LLM-GNN, ENG
  - Generate node labels using LLM
  - Generalize to graphs with different label sets



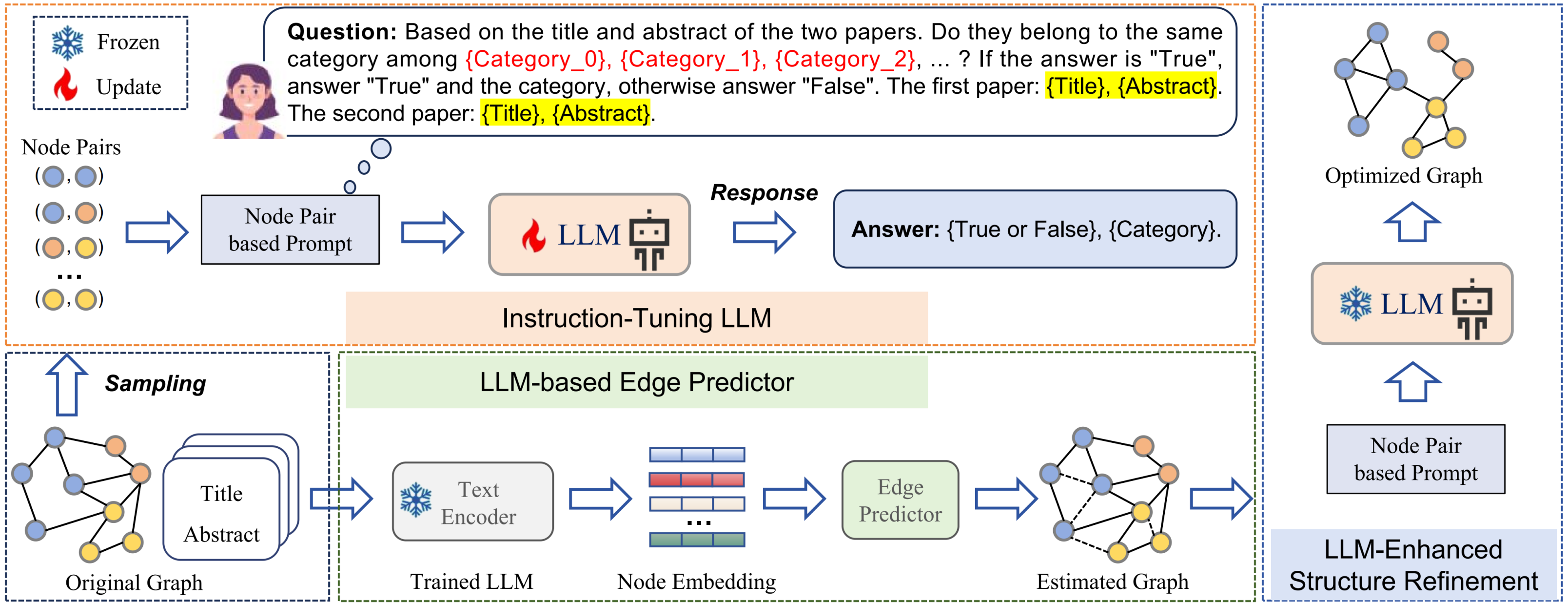


[LLM-GNN] Label-free Node Classification on Graphs with Large Language Models (LLMs)
[ENG] Empower Text-Attributed Graphs Learning with Large Language Models (LLMs)

LLM as Labeler

香 港 大 學
THE UNIVERSITY OF HONG KONG

- GraphEdit
  - ➢ Generate link labels by instruction-tuning LLM
  - ➢ Target: graph structure learning for downstream tasks



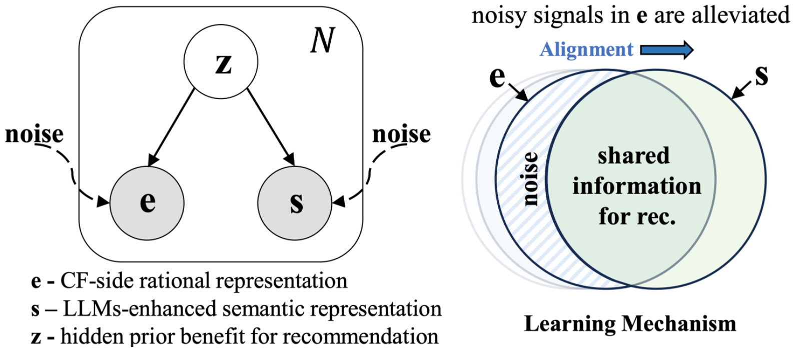
GraphEdit: Large Language Models for Graph Structure Learning

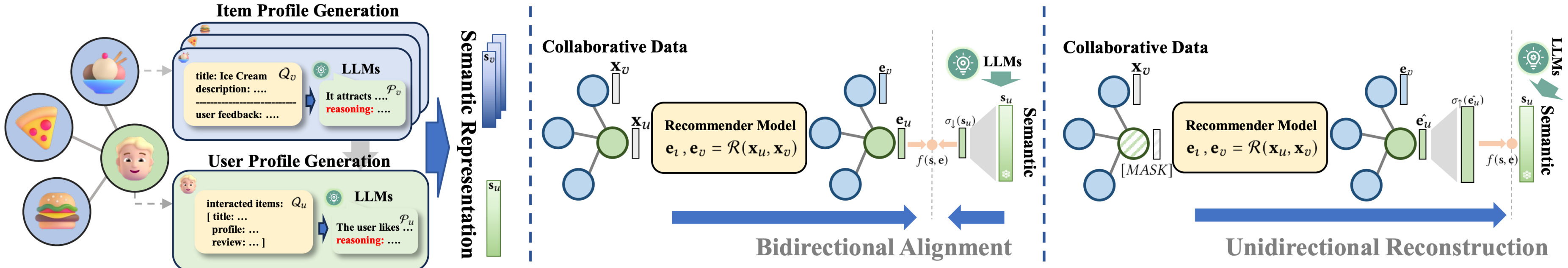
arXiv

LLM as Labeler

香 港 大 學
THE UNIVERSITY OF HONG KONG

- RLMRec
  - ➢ Generate representation labels using LLM
  - ➢ Minimize noise by infomax
  - ➢ Use LLM for
    - ✓ text generation & representation





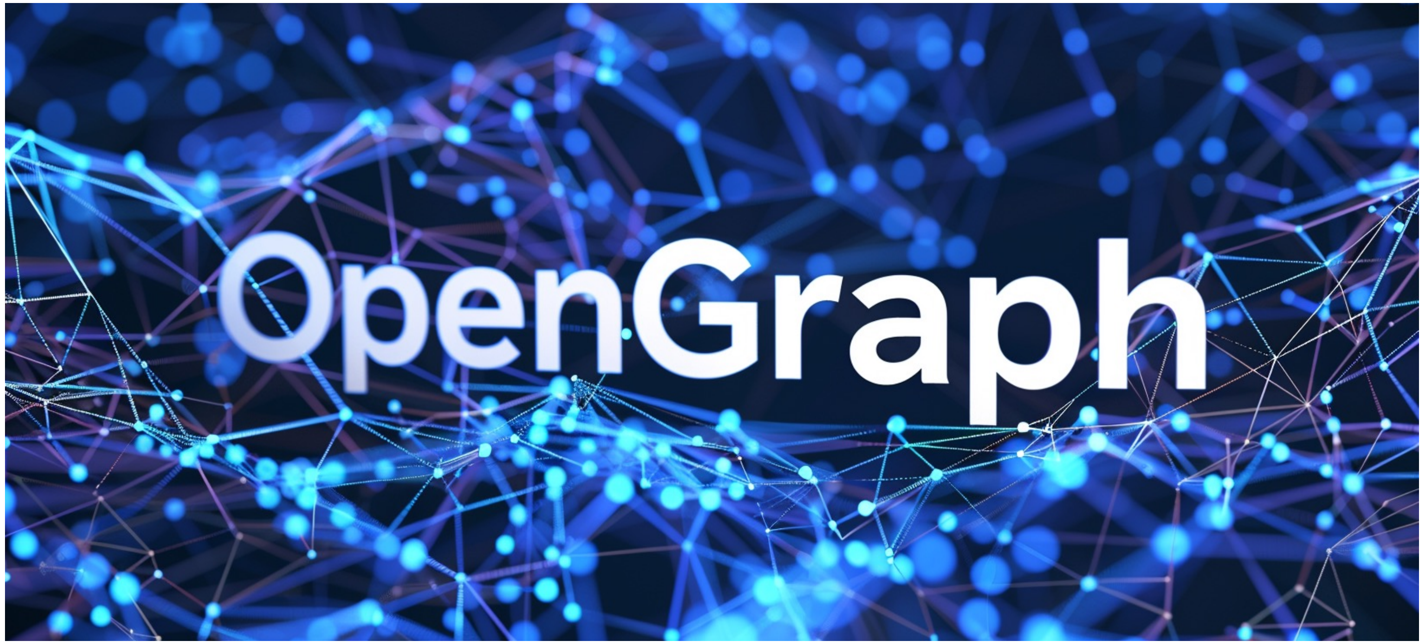
(a) Profile Generation via Reasoning (b) Contrastive Alignment (c) Generative Alignment

[RLMRec] Representation Learning with Large Language Models for Recommendation

LLM as Labeler

香 港 大 學
THE UNIVERSITY OF HONG KONG

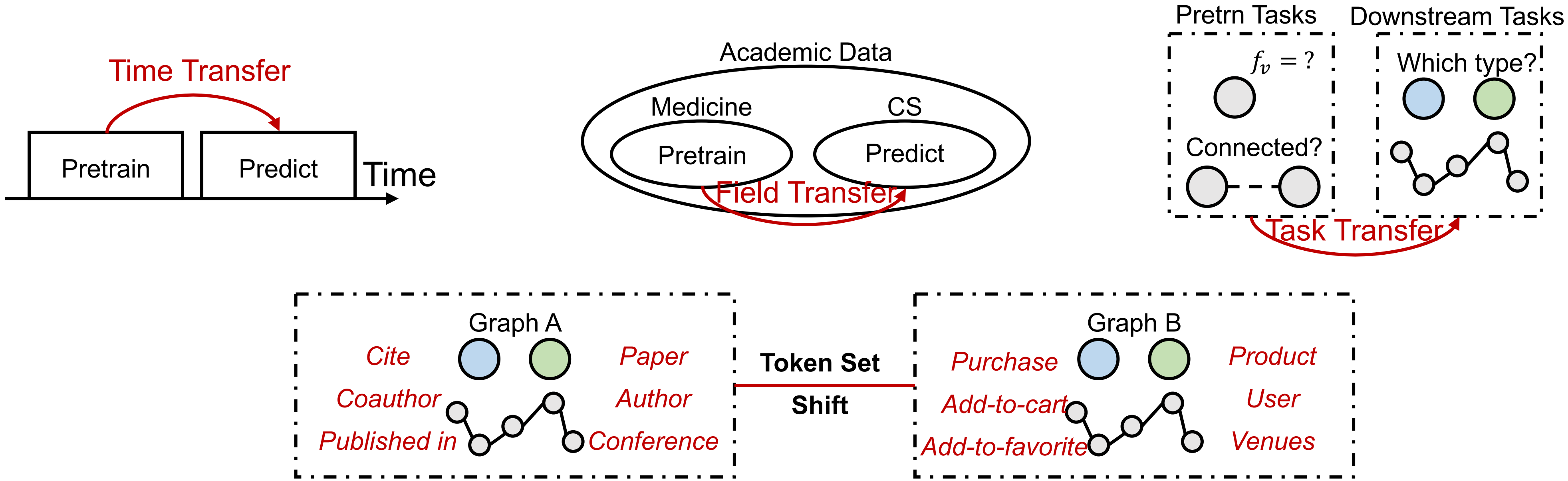
- OpenGraph: Towards Graph Foundation Models
  - ➢Generate domain-specific graph data using LLM



arXiv

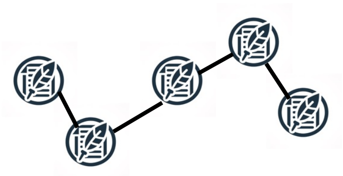
Generalizability of Graph Models

- Self-supervised pretraining benefits generalization ability
  - *e.g.* GraphCL, DGI, GraphPrompt
  - *Pretrain* and *fine/prompt tuning* for task/field/time transfer
  - Cannot handle **token set shift**



Generalizability of Graph Models

- Text-based generalization ability with LLMs
  - e.g. OFA, GraphGPT, ZeroG
  - Address **token set shift** with LLMs for text-based data
  - Cannot handle structure shifts and text-less scenarios



Text-attributed Graphs

There are 4 job applicants numbered from 0 to 3, and 5 jobs numbered from 0 to 4. Each applicant is interested in some of the jobs. Each job can only accept one applicant and a job applicant can be appointed for only one

Graphs in Text

Large-scale Graphs without Text

?

- Generalization ability for structures.
- What if text features are insufficient?
- Prompt for different tasks?

[OFA] One for All: Towards Training One Graph Model for All Classification Tasks

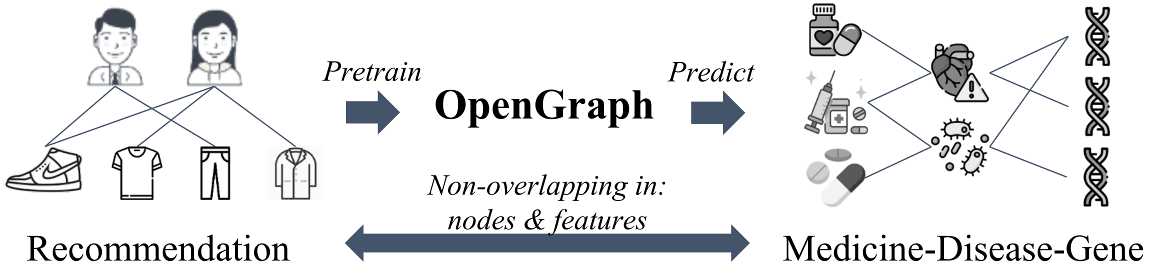
[GraphGPT] GraphGPT: Graph instruction tuning for large language models

[ZeroG] ZeroG: Investigating Cross-dataset Zero-shot Transferability in Graphs

OpenGraph: Challenges

香 港 大 學
THE UNIVERSITY OF HONG KONG

- Zero-shot Graph Generalization across Graphs



- Challenges:
  - Token set shift across graphs: **Unified Graph Tokenizer**
  - Efficient node-wise relation modeling
  - Domain-specific data scarcity

OpenGraph: Challenges

香 港 大 學
THE UNIVERSITY OF HONG KONG

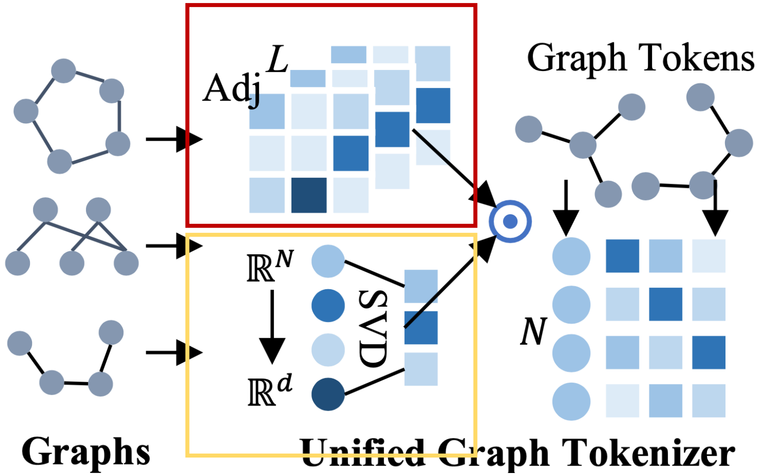
- Efficient node-wise relation modeling
  - Pairwise relation learning by Transformer
  - Large number of tokens in graphs
  - **Scalable Graph Transformer**

- Domain-specific data scarcity
  - How to collect training data covering different downstream domains?
  - **Knowledge Distillation from LLMs**

Unified Graph Tokenizer

- $\mathcal{G} \rightarrow \{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_{|\mathcal{V}|}\}$ for any graph

Smoothed high-order adjacency



Topology-aware projection

Feature augmentation

- High-order connectivity
- Sparse → dense

$\mathbb{R}^N \rightarrow \mathbb{R}^d$

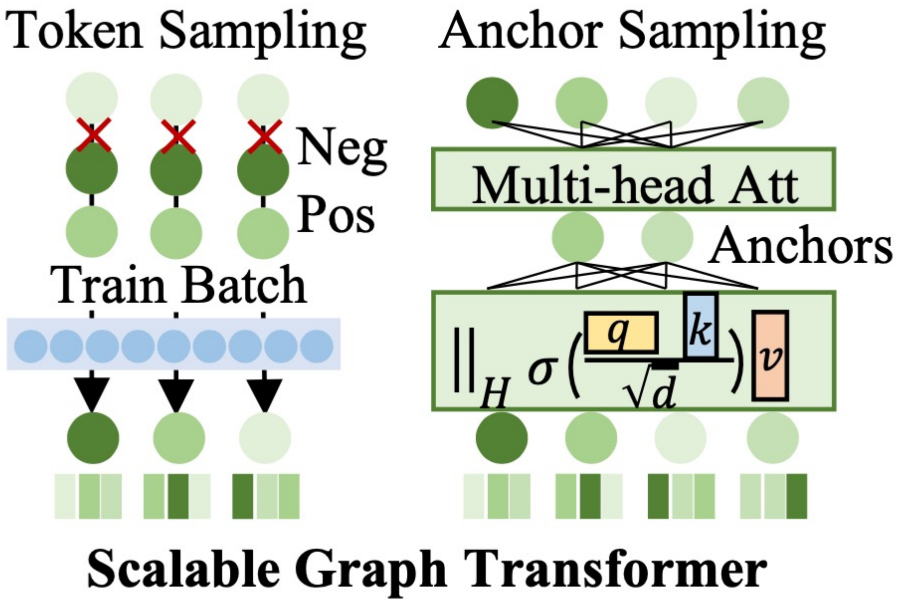
- Representations given by FastSVD
- Cross-graph unification

Scalable Graph Transformer

- In-efficiency of transformers caused by
  - ➢Long token sequence
  - ➢Pairwise relation learning

Full token sequence

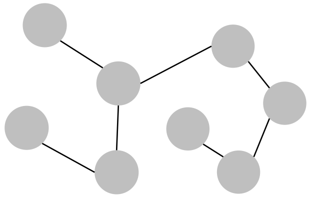
Token seq with only train batch

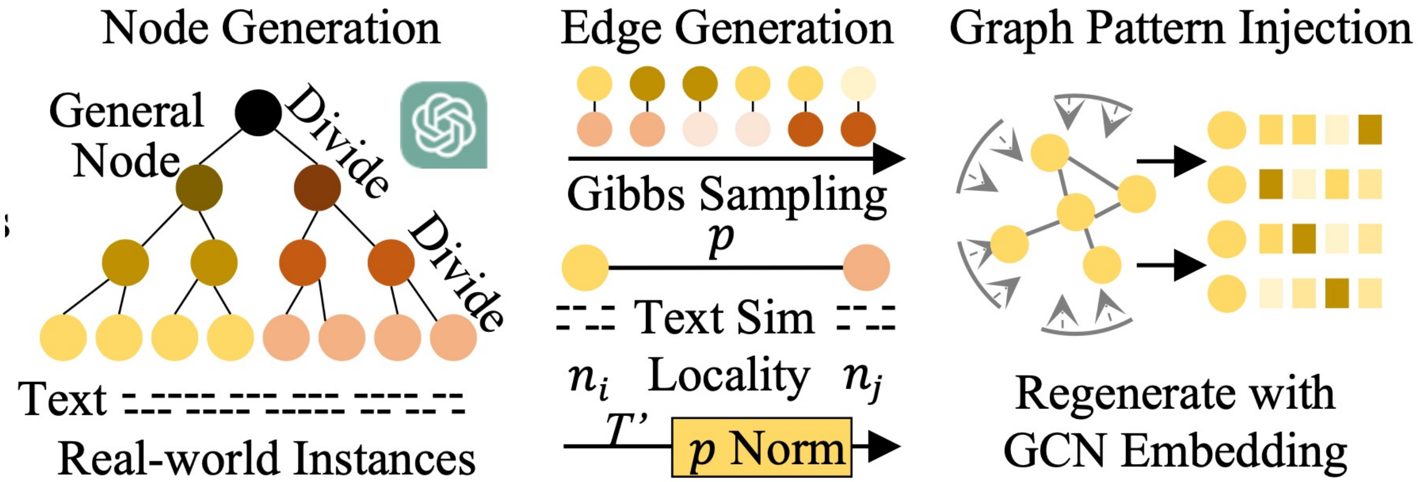


Relations between input tokens & sampled anchors

Distillation from LLMs

Motivation LLM $\xrightarrow[\textit{knowledge}]{\textit{Downstream}}$ OpenGraph

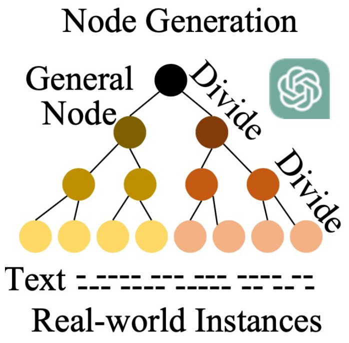




Knowledge Distillation from Large Language Models

Distillation from LLMs

- Node Generation
  - ➢ Tree-of-prompt Algorithm



Products

↓ ***Prompt***

Clothing, Electronics, Shoes, Healthcare, Baby, …

↓

Women's Clothing, Men's Clothing, Computers, Printers, Vitamins, …

↓

Hoodies, Dresses, Suits, Ties, Laptops, MacBook, Vitamin C, Calcium, …

↓

Instances: White hooded sweater; Men's clothing, Shorts, Golf Shorts, …

Prompt Template

List all distinct sub-categories of **{entity_name}** within the **{prefix}** category in the context of **{scenario_desc}**, ensuring a finer level of granularity. The sub-categories should not overlap with each other. And a sub-category should be a smaller subset of **{entity_name}**. Directly present the list EXACTLY following the form: "sub-category a, sub-category b, sub-category c, ..." without other words, format symbols, new lines, serial numbers.

Prompt Example

entity_name = "women's clothing"
scenario_desc = "e-commerce platform like Amazon"
prefix = "products, clothing"

Examples of Generated Nodes

products, Clothing, Women's clothing, Sweaters, Crewneck sweaters
products, Clothing, Men's clothing, Costumes, Scary costumes
products, Clothing, Outerwear, Vests, Sweater Vests
products, Shoes, Flats, T-strap flats, Open toe T-strap flats
products, Shoes, Ballet flats, Ankle strap ballet flats, Nude ankle strap ballet flats
products, Jewelry, Jewelry Sets, Choker, Gothic Choker
products, Electronics, office electronics, Calculators, Scientific Calculators
products, Books, Non-fiction, Self-Help, Codependency

Prompt Example

entity_name = "Restaurant"
scenario_desc = "venue rating platform like Yelp"
prefix = "business venues"

Examples of Generated Nodes

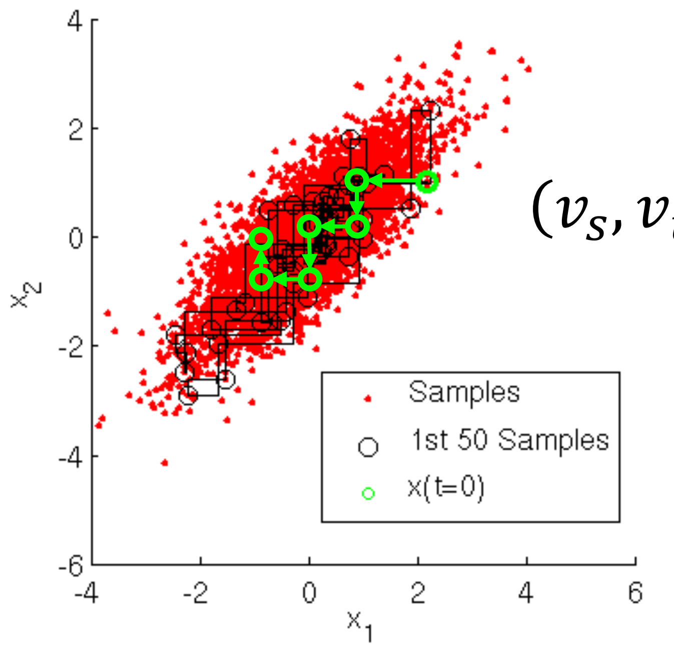
business venues, Restaurant, American, Barbecue, BBQ fusion
business venues, Restaurant, Buffet, Chinese buffet, Seafood
business venues, Cafe, Tea house, Tea room, British tea house
business venues, Cafe, Brunch spot, Buffet brunch, Vegan buffet
business venues, Bar, Karaoke Bar, Karaoke DJ nights, live band karaoke
business venues, Nightclub, Live Music Venue, Jazz Club, Latin Jazz Club
business venues, Fast Food Restaurant, Smoothie Bar, Specialty smoothie bar, Fresh fruit smoothie bar
business venues, Drive-Thru Restaurant, Fast food, Pizza place, Coal-fired pizza

Distillation from LLMs

- Edge Generation
  - ➢Gibbs sampling
  - ➢LLM-based estimation for

$$p(\boxed{\mathbf{a}^t \oplus v_{t'}} | \boxed{\mathbf{a}^t}) = \sum_{v_i} a_i^t (\mathbf{h}_i / \|\mathbf{a}^t\|_0)^\top \cdot \mathbf{h}_{t'}$$

new sample old sample LLM embeddings

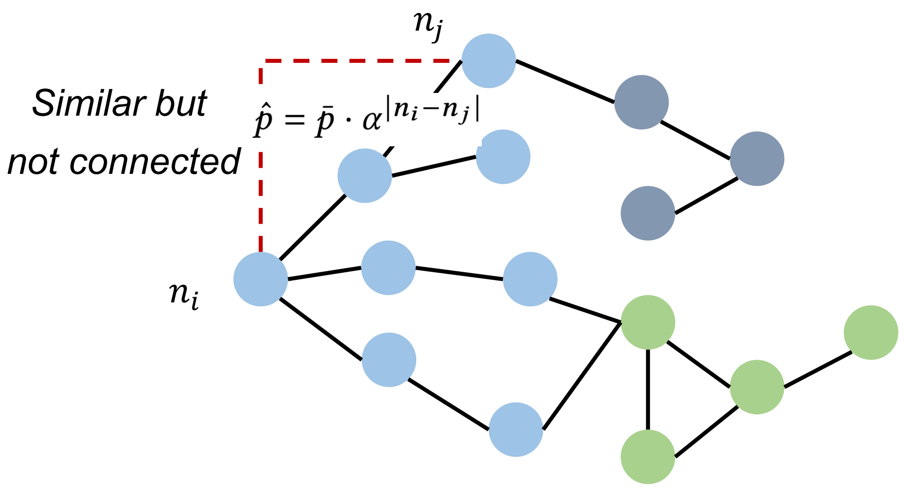


(v_s, v_t)

Distillation from LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- Techniques for Edge Generation $p(\mathbf{a}^t \oplus v_{t'} | \mathbf{a}^t)$
  - Dynamic Probability Normalization
    - ✓ Maintain last *T* estimations
    - ✓ Normalize to the range close to [0, 1] $\bar{p} = (p - \mu)/(4\sigma)$
  - Node Locality Incorporation



Distillation from LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- Graph Topological Pattern Injection

Node Gen. → Edge Gen. → Graph → GCN → GCN-smoothed Embeddings

Experiments

香 港 大 學
THE UNIVERSITY OF HONG KONG

- Train with synthetic data, test on real data

Table 4: Statistics of experimental datasets.

Dataset		# Nodes	# Edges	# Features	# Classes
Link	OGBL-ddi	4,267	1,334,889	0	N/A
	OGBL-collab	235,868	1,285,465	128	
	ML-1M	9,746	720,152	0	
	ML-10M	80,555	7,200,040	0	
	Amazon-book	144,242	2,380,730	0	
Node	Cora	2,708	10,556	1433	7
	Citeseer	3,327	9,104	3,703	6
	Pubmed	19,717	88,648	500	3
Generated	Gen0	46,861	454,276	0	N/A
	Gen1	51,061	268,007	0	
	Gen2	32,739	240,500	0	

Overall Comparison

1) Superior 0-shot prediction capability. 2) Existing pretraining methods may fail for cross-data generalization

Table 1: Performance comparison between our OpenGraph (zero-shot) and baseline methods (one-shot, five-shot) on the link prediction task (measured by *Recall@N* for $N = 20, 40$) and node classification task (measured by *Accuracy* and *Macro F1 Score*).

Dataset		ogbl-ddi		ogbl-collab		ML-1M		ML-10M		Amazon-book		Cora		Citeseer		Pubmed	
Metric		R@20	R@40	R@20	R@40	R@20	R@40	R@20	R@40	R@20	R@40	Acc	MacF1	Acc	MacF1	Acc	MacF1
MF	1-shot	0.0087	0.0161	0.0261	0.0349	0.0331	0.0604	0.1396	0.1956	0.0034	0.0043	0.1710	0.1563	0.1740	0.1727	0.3470	0.3346
	5-shot	0.0536	0.0884	0.0412	0.0609	0.0987	0.1584	0.2060	0.2989	0.0196	0.0284	0.1500	0.1422	0.1520	0.1484	0.3540	0.3435
MLP	1-shot	0.0195	0.0336	0.0112	0.0185	0.0548	0.1019	0.1492	0.2048	0.0017	0.0028	0.2300	0.1100	0.2590	0.1993	0.4430	0.3114
	5-shot	0.0621	0.1038	0.0115	0.0185	0.0851	0.1470	0.2362	0.2563	0.0092	0.0152	0.3930	0.3367	0.3690	0.3032	0.5240	0.4767
GCN	1-shot	0.0279	0.0459	0.0206	0.0321	0.0432	0.0849	0.1760	0.2086	0.0096	0.0160	0.3180	0.1643	0.3200	0.2096	0.4270	0.3296
	5-shot	0.0705	0.1312	0.0366	0.0513	0.1054	0.1656	0.2127	0.2324	0.0251	0.0408	0.5470	0.5008	0.4910	0.4190	0.509	0.4455
GAT	1-shot	0.0580	0.1061	0.0258	0.0372	0.0245	0.0520	0.1615	0.2476	0.0047	0.0079	0.2420	0.1687	0.2810	0.2025	0.4720	0.3657
	5-shot	0.0711	0.1309	0.0340	0.0505	0.1506	0.2267	0.2002	0.2883	0.0228	0.0392	0.585	0.5438	0.4940	0.4441	0.5780	0.5582
GIN	1-shot	0.0530	0.1004	0.0163	0.0247	0.0466	0.0884	0.1541	0.2388	0.0069	0.0114	0.3190	0.1753	0.2820	0.1705	0.4410	0.3064
	5-shot	0.0735	0.1441	0.0311	0.0458	0.1458	0.2344	0.1926	0.2829	0.0252	0.0418	0.5400	0.4941	0.521	0.4696	0.5070	0.4547
DGI	1-shot	0.0315	0.0617	0.0255	0.0385	0.0486	0.0863	0.1868	0.2716	0.0081	0.0142	0.3150	0.1782	0.2840	0.1791	0.4290	0.3163
	5-shot	0.0821	0.1426	0.0345	0.0502	0.1687	0.2573	0.2303	0.3063	0.0300	0.0492	0.4880	0.4606	0.4450	0.4062	0.4890	0.4509
GPF	1-shot	0.0503	0.0856	0.0027	0.0048	0.1099	0.1702	0.1599	0.2326	0.0072	0.0128	0.3080	0.1952	0.3110	0.1984	0.4220	0.2670
	5-shot	0.0839	0.1460	0.0027	0.0047	0.0817	0.1392	0.2014	0.2994	0.0179	0.0310	0.5550	0.5233	0.4690	0.4223	0.5150	0.4934
GPrompt	1-shot	0.0541	0.1102	0.0138	0.0207	0.0797	0.1310	0.1362	0.2073	0.0074	0.0120	0.3540	0.1596	0.2800	0.1519	0.4710	0.3705
	5-shot	0.0769	0.1396	0.0157	0.0231	0.1340	0.2166	0.2157	0.3147	0.0287	0.0464	0.5510	0.5098	0.5570	0.5211	0.5130	0.4520
GraphCL	1-shot	0.0603	0.1112	0.0265	0.0398	0.0390	0.0799	0.1655	0.2529	0.0047	0.0077	0.2430	0.1548	0.2980	0.1630	0.4070	0.4130
	5-shot	0.0740	0.1368	0.0311	0.0456	0.1416	0.2138	0.2019	0.3075	0.0270	0.0440	0.5610	0.5330	0.4300	0.3683	0.5230	0.5024
OpenGraph	0-shot	**0.0921**	**0.1746**	**0.0421**	**0.0639**	**0.1911**	**0.2978**	**0.2370**	**0.3265**	**0.0485**	**0.0748**	**0.7504**	**0.7426**	**0.6097**	**0.5821**	**0.6869**	**0.6537**

Graph Tokenizer Study

香 港 大 學
THE UNIVERSITY OF HONG KONG

Effectiveness of 1) Smoothing, 2) Topology-aware projection

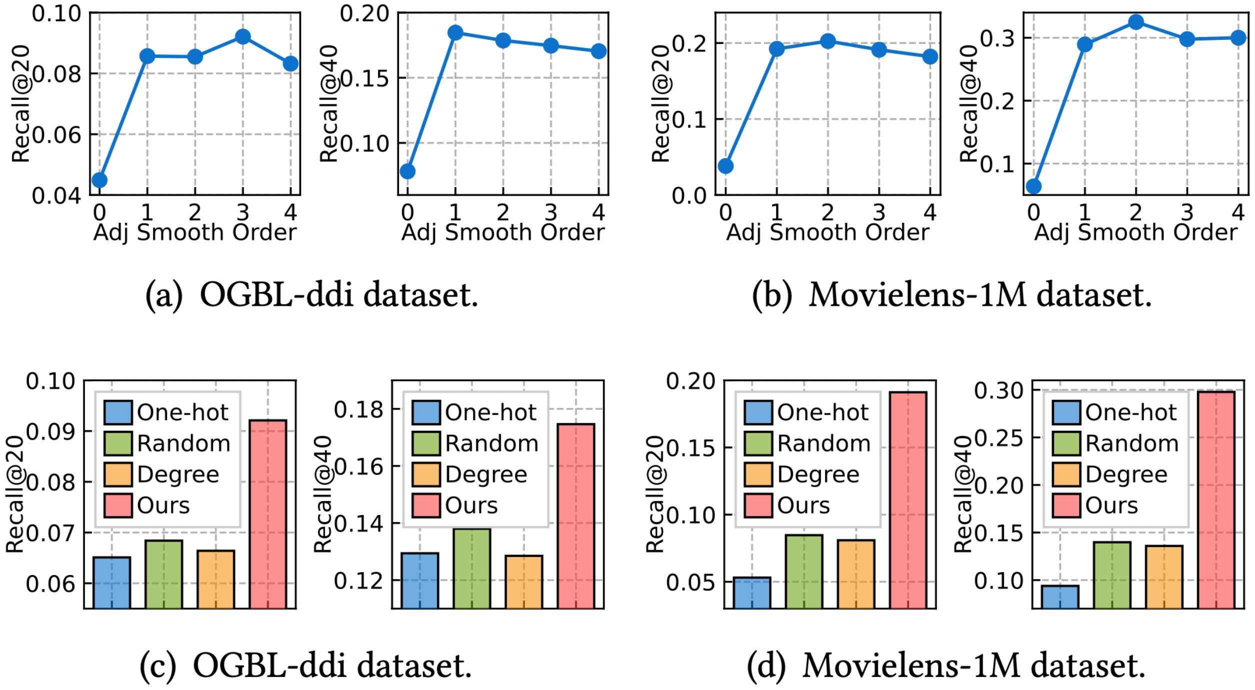


Figure 2: Influence of graph tokenizer configurations.

Pre-training Data Study

1) Ablation datasets, 2) Unrelated real data,
3) Related real data, 4) Ours synthetic data

Table 2: Influence of utilizing different pre-training datasets.

Test Dataset	Pre-training Dataset -Norm	-Loc	-Topo	Yelp2018	Gowalla	ML-10M	Gen
ogbl-ddi	0.0737	<u>0.0893</u>	0.0656	0.0588	0.0770	0.0692	**0.0921**
ML-1M	0.0572	0.1680	0.0850	0.0599	0.0485	**0.2030**	<u>0.1911</u>
ML-10M	0.0982	0.1636	0.1017	0.1629	0.0910	**0.2698**	<u>0.2370</u>
Cora	0.4985	0.4864	0.4342	0.3715	<u>0.5943</u>	0.2780	**0.7504**
Citeseer	0.3944	0.3691	<u>0.5743</u>	0.2651	0.4300	0.2003	**0.6097**
Pubmed	0.4501	0.5015	0.4876	0.3317	<u>0.5148</u>	0.3652	**0.6869**

Sampling Strategy Study

- Sequence & anchor sampling improves memory & computational efficiency
- Sequence sampling has positive effect on performance

Table 3: Impact of sampling strategies on the efficiency and model performance in the scalable graph transformer.

OGBL-ddi	Train Mem	Test Mem	Train Time	Test Time	Recall@20
-Seq-Anc	5420MiB	1456MiB	22.72s	13.88s	0.0966
-Anc	3360MiB	1456MiB	18.19s	13.73s	0.1107
-Seq	2456MiB	1202MiB	16.45s	12.09s	0.0930
OpenGraph	2358MiB	1202MiB	15.45s	12.09s	0.1006

ML-10M	Train Mem	Test Mem	Train Time	Test Time	Recall@20
-Seq-Anc	OOM	OOM	–	–	–
-Anc	4996MiB	OOM	73.15s	–	–
-Seq	23140MiB	4550MiB	158.60s	84.78s	0.2772
OpenGraph	4470MiB	4550MiB	68.79s	54.17s	0.2816

Implementation of OpenGraph

- Environment Setup
  - python==3.10.13
  - torch==1.13.0
  - numpy==1.23.4
  - scipy==1.9.3

- Download pre-trained models using the link in Models/readme

- Code Structure

```
./
│   └── README.md
│   ├── History/ ## Training history of pre-trained models ##
│   ├── Models/ ## Pre-trained models ##
│   ├── datasets/
│   ├── graph_generation/ ## Code and examples for graph generation ##
│   ├── imgs/ ## Images used in readme ##
│   ├── link_prediction/ ## code for link prediction and pre-training ##
│   │   ├── data_handler.py
│   │   ├── main.py
│   │   ├── model.py
│   │   └── params.py
│   │   ├── Utils/
│   │   │   └── TimeLogger.py
│   ├── node_classification/ ## code for testing on node classification ##
│   │   ├── data_handler.py
│   │   ├── main.py
│   │   ├── model.py
│   │   └── params.py
│   │   ├── Utils/
│   │   │   └── TimeLogger.py
```

Run Model

香 港 大 學
THE UNIVERSITY OF HONG KONG

Usage

To reproduce the test performance reported in the paper, run the following command lines:

```
cd link_prediction/
python main.py --load pretrn_gen1 --epoch 0 # test on OGBL-Collab, ML-1M, ML-10M
python main.py --load pretrn_gen0 --tstdata amazon-book --epoch 0 # test on Amazon-Book
python main.py --load pretrn_gen2 --tstdata ddi --epoch 0 # test on OGBL-ddi
cd ../node_classification/
python main.py --load pretrn_gen1 --tstdata cora # test on Cora
python main.py --load pretrn_gen1 --tstdata citeseer # test on Citeseer
python main.py --load pretrn_gen1 --tstdata pubmed # test on Pubmed
```

To re-pretrain OpenGraph by yourself, run the following command lines:

```
cd ../link_prediction/
python main.py --save pretrn_gen1
python main.py --trndata gen0 --tstdata amazon-book --save pretrn_gen0
python main.py --trndata gen2 --tstdata ddi --save pretrn_gen2
```

To explore pretraining with multiple different pre-training and testing datasets, modify `trn_datasets` and `tst_datasets` in line 241 of `link_prediction/main.py`.

Graph Generation

Graph Data Generation

The graph generation code is in `graph_generation/`. A toy dataset of small size is given. You need to fill in your OpenAI key in `Utils.py` and `itemCollecting_dfsIterator.py` first. To generate your dataset, modify the `descs` and `hyperparams` dicts, and follow the following procedure:

```
cd graph_generation/
python itemCollecting_dfsIterator.py
python instance_number_estimation_hierarchical.py
python embedding_generation.py
python human_item_generation_gibbsSampling_embedEstimation.py
python make_adjs.py
```

Prompt Template
List all distinct sub-categories of **{entity_name}** within the **{prefix}** category in the context of **{scenario_desc}**, ensuring a finer level of granularity. The sub-categories should not overlap with each other. And a sub-category should be a smaller subset of **{entity_name}**. Directly present the list EXACTLY following the form: "sub-category a, sub-category b, sub-category c, ..." without other words, format symbols, new lines, serial numbers.

Prompt Example
entity_name = "women's clothing"
scenario_desc = "e-commerce platform like Amazon"
prefix = "products, clothing"

Examples of Generated Nodes
products, Clothing, Women's clothing, Sweaters, Crewneck sweaters
products, Clothing, Men's clothing, Costumes, Scary costumes
products, Clothing, Outerwear, Vests, Sweater Vests
products, Shoes, Flats, T-strap flats, Open toe T-strap flats
products, Shoes, Ballet flats, Ankle strap ballet flats, Nude ankle strap ballet flats
products, Jewelry, Jewelry Sets, Choker, Gothic Choker
products, Electronics, office electronics, Calculators, Scientific Calculators
products, Books, Non-fiction, Self-Help, Codependency

Code of OpenGraph

`link_prediction/ ## c`	
`├── data_handler.py`	Data processing and accessing.
`├── main.py`	Run training and testing.
`├── model.py`	Model/Algorithm implementation.
`└── params.py`	Hyperparameter definition.
`├── Utils/`	
`│   └── TimeLogger.py`	Logger.

Code of OpenGraph

- main.py
  - ➢ if name == '__main__'
  - ➢ class Exp

```python
if __name__ == '__main__':
    os.environ['CUDA_VISIBLE_DEVICES'] = args.gpu
    if len(args.gpu.split(',')) > 1:
        args.devices = ['cuda:0', 'cuda:1']
    else:
        args.devices = ['cuda:0', 'cuda:0']
    args.devices = list(map(lambda x: t.device(x), args.devices))
    logger.saveDefault = True
    setproctitle.setproctitle('OpenGraph')

    log('Start')
    trn_datasets = ['gen1']
    tst_datasets = ['ml1m', 'ml10m', 'collab']

    # trn_datasets = ['gen2']
    # tst_datasets = ['ddi']

    # trn_datasets = ['gen0']
    # tst_datasets = ['amazon-book']

    if len(args.tstdata) != 0:
        tst_datasets = [args.tstdata]
    if len(args.trndata) != 0:
        trn_datasets = [args.trndata]
    trn_datasets = list(set(trn_datasets))
    tst_datasets = list(set(tst_datasets))
    multi_handler = MultiDataHandler(trn_datasets, tst_datasets)
    log('Load Data')

    exp = Exp(multi_handler)
    exp.run()
```

Code of OpenGraph

- main.py
  - ➢if name == '__main__'
  - ➢class Exp

```
class Exp:
    def __init__(self, multi_handler):
                    self.metrics['Test' + handler.data_name + met] = list()

    def make_print(self, name, ep, reses, save, data_name=None):
        return ret

    def run(self):
        self.save_history()

    def add_res_to_summary(self, summary, res):
            summary[key] += res[key]

    def print_model_size(self):
        print(f'Non-trainable params: {non_trainable_params/1e6}')

    def prepare_model(self):
        self.print_model_size()

    def train_epoch(self):
        return ret

    def test_epoch(self, tst_loader, tst_handler):
        return ret

    def calc_recall_ndcg(self, topLocs, tstLocs, batIds):
        return allRecall, allNdcg

    def save_history(self):
        log('Model Saved: %s' % args.save_path)

    def load_model(self):
        log('Model Loaded')
```

Code of OpenGraph

- model.py

```python
class InitialProjector(nn.Module): ...
        return self.proj_embeds
```

Topology-aware projection

```python
class TopoEncoder(nn.Module): ...
        return embeds
```

High-order smoothing

```python
class GraphTransformer(nn.Module): ...
        return embeds
```

Graph Transformer

```python
class GTLayer(nn.Module): ...
        return embeds
```

Graph Transformer layer

```python
class FeedForwardLayer(nn.Module): ...
        return self.act(self.linear(embeds))
```

Feed-forward layer

```python
class Masker(nn.Module): ...
        return t.sparse.FloatTensor(adj._indices(), newVals, adj.shape)
```

Efficient graph masker for edge MAE training

```python
class OpenGraph(nn.Module): ...
        return all_preds
```

OpenGraph

Code of OpenGraph

- class OpenGraph

```python
class OpenGraph(nn.Module):
    def __init__(self):
        super(OpenGraph, self).__init__()
        self.topoEncoder = TopoEncoder().to(args.devices[0])
        self.graphTransformer = GraphTransformer().to(args.devices[1])
        self.masker = Masker().to(args.devices[0])

    def forward(self, adj, initial_projector, user_num):
        topo_embeds = self.topoEncoder(adj, initial_projector(), user_num).to(args.devices[1])
        final_embeds = self.graphTransformer(topo_embeds)
        return final_embeds

    def pred_norm(self, pos_preds, neg_preds): ...
        return pos_preds, neg_preds

    def cal_loss(self, batch_data, adj, initial_projector): ...
        return pre_loss + reg_loss, loss_dict

    def pred_for_test(self, batch_data, adj, initial_projector, cand_size, rerun_embed=True): ...
        return all_preds
```

Code of OpenGraph

- data_handler.py

```python
class MultiDataHandler: ...
        trn_handler.initial_projector = InitialProjector(trn_handler.asym_adj)

class DataHandler: ...
            self.tst_loader = data.DataLoader(tst_data, batch_size=args.tst_batch, shuffle=False, num_workers=0)

class TstData(data.Dataset): ...
        return self.tst_nodes[idx]

class TrnData(data.Dataset): ...
        return ancs, poss, negs, adj_id
```

Code of OpenGraph

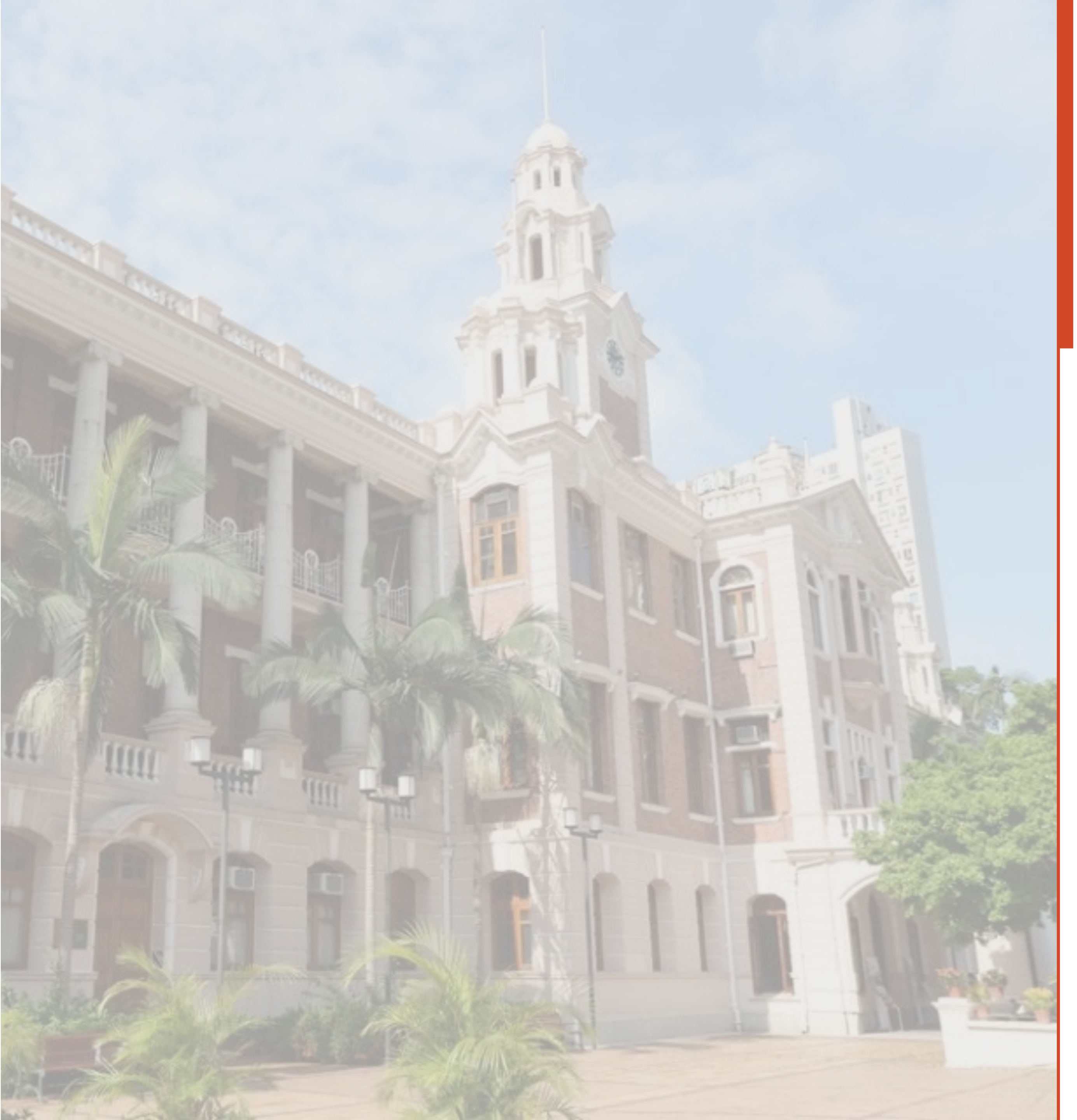
香 港 大 學
THE UNIVERSITY OF HONG KONG

```python
class MultiDataHandler:
    def __init__(self, trn_datasets, tst_datasets):
        all_datasets = list(set(trn_datasets + tst_datasets))
        self.trn_handlers = []
        self.tst_handlers = []
        for data_name in all_datasets:
            trn_flag = data_name in trn_datasets
            tst_flag = data_name in tst_datasets
            handler = DataHandler(data_name, trn_flag, tst_flag)
            if trn_flag:
                self.trn_handlers.append(handler)
            if tst_flag:
                self.tst_handlers.append(handler)
        self.make_joint_trn_loader()

    def make_joint_trn_loader(self):
        trn_data = TrnData(self.trn_handlers)
        self.trn_loader = data.DataLoader(trn_data, batch_size=1, shuffle=True, num_workers=0)

    def remake_initial_projections(self):
        for i in range(len(self.trn_handlers)):
            self.remake_one_initial_projection(i)

    def remake_one_initial_projection(self, idx):
        trn_handler = self.trn_handlers[idx]
        trn_handler.initial_projector = InitialProjector(trn_handler.asym_adj)
```



Q & A

Lianghao Xia

Personal Information

Group Information

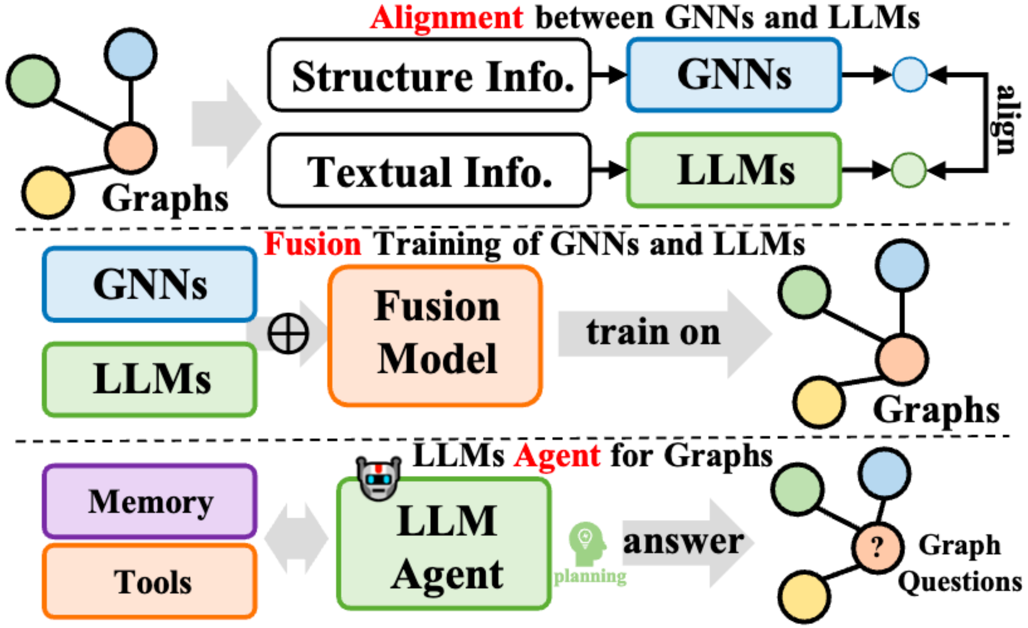
LLMs-Graph Intergration

LLMs↔Graphs

LLMs-Graphs Intergration

香 港 大 學
THE UNIVERSITY OF HONG KONG

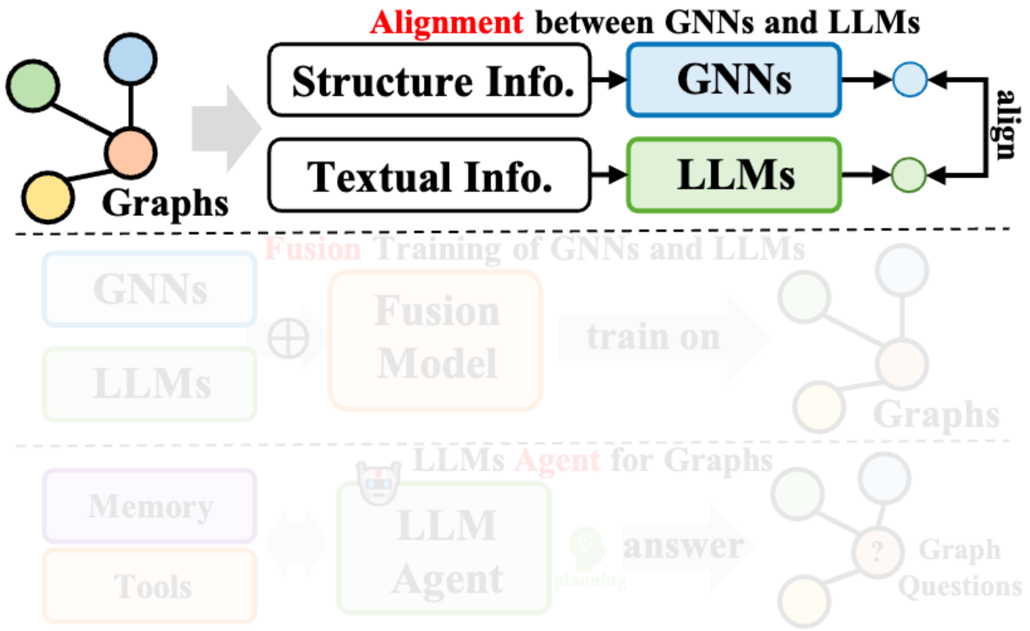
- Conduct **Alignment** between GNNs and LLMs
- Create a **Fused Model** with LLM and GNN
- Build **Agents** by LLMs for Graph Tasks



Alignment bet. GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

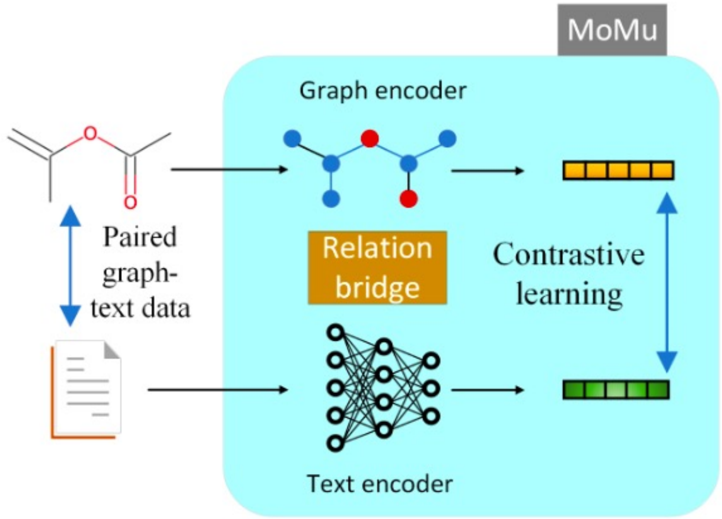
- **Motivation**: Align the feature spaces of GNNs and LLMs
  - GNNs and LLMs process different types of data
  - Distinct feature spaces in GNNs and LLMs
- **Methods**
  - Contrastive Learning
  - EM Iterative Training



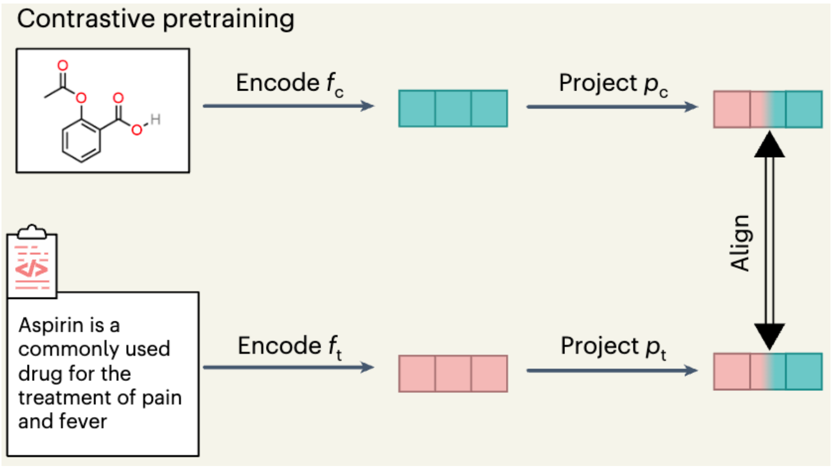
Alignment bet. GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Several Methods in BioScience Leverage Contrastive Alignment**
  - Both **MoMu** and **MoleculeSTM** uses GIN to handle molecular graphs and BERT to handle text data.
  - **MoMu** can directly imagine new molecules from textual descriptions with a pre-trained model **MoFlow**.
  - **MoleculeSTM** achieves molecule retrieval and editing with text descriptions.



[MoMu, 2022]



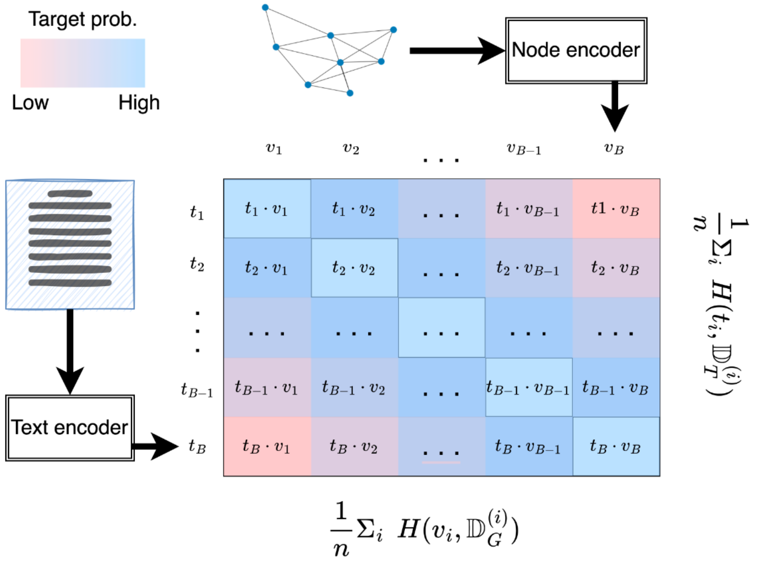
[MoleculeSTM, 2022]

[1] A Molecular Multimodal Foundation Model Associating Molecule Graphs with Natural Language, arXiv 2022

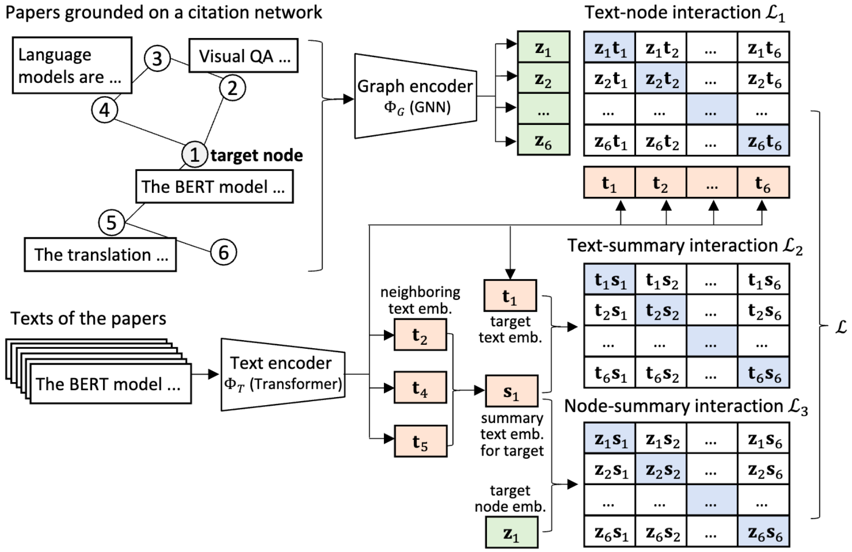
[2] Multi-modal Molecule Structure–text Model for Text-based Retrieval and Editing, Nature Machine Intelligence 2022

Alignment bet. GNNs and LLMs

- **Several Methods in Graph Learning Leverage Contrastive Alignment**
  - **ConGraT** designs node-level alignment of GNN-embeddings and LM-embeddings, and show improvement on node and text classification as well as link prediction tasks.
  - **G2P2** proposes text-node, text-summary, and node-summary alignment and improves the text classification performance in low-resource environments



[ConGraT, 2023]



[G2P2, 2023]

[1] ConGraT: Self-Supervised Contrastive Pretraining for Joint Graph and Text Embeddings, arXiv 2023
[2] Prompt Tuning on Graph-augmented Low-resource Text Classification, arXiv 2023

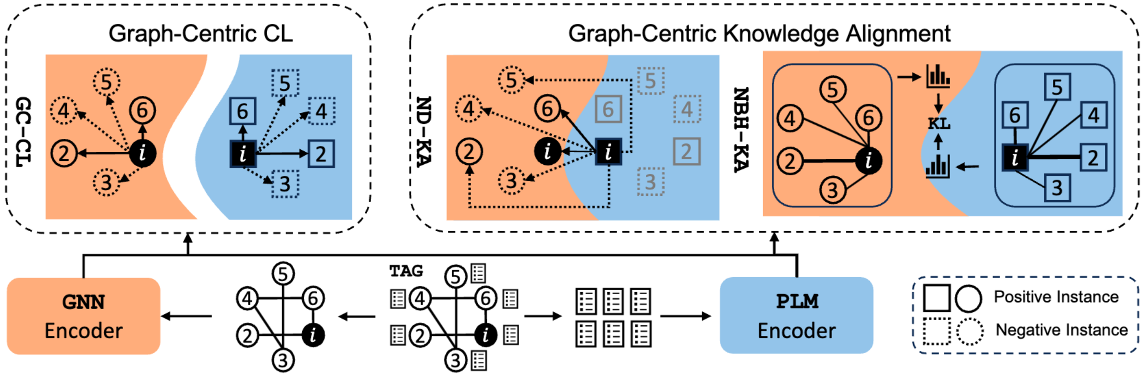
Alignment bet. GNNs and LLMs

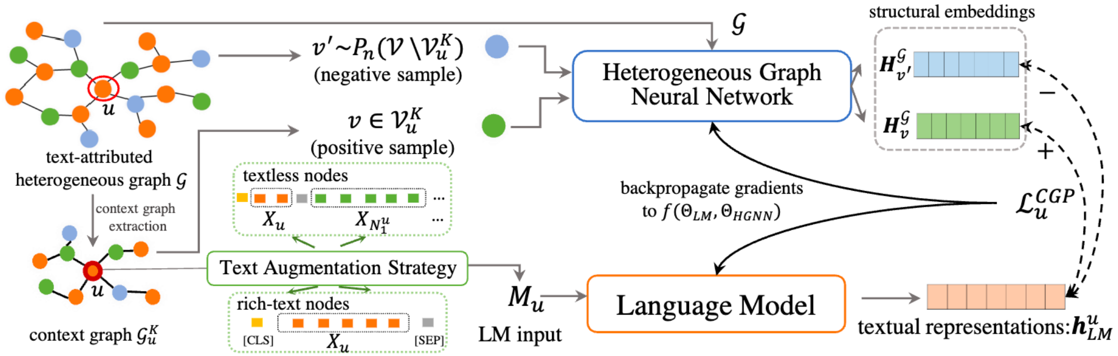
香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Several Methods in Graph Learning Leverage Other Alignment**

[GRENADE, 2023]

- **Graph-Centric Contrastive Learning**
- **Graph-Centric Knowledge Alignment**
  - Minimize KL Divergence of the Distribution from two Encoders (GNN and PLM).





[THLM, 2023]

- **Learning on Heterogenous Graphs**
- **THLM** uses a <u>positive-negative classification task</u> with negative sampling to improve the alignment of embeddings from two different modalities.

[1] Grenade: Graph-Centric Language Model for Self-Supervised Representation Learning on Text-Attributed Graphs, arXiv 2023

[2] Pretraining Language Models with Text-Attributed Heterogeneous Graphs, arXiv 2023

Alignment bet. GNNs and LLMs

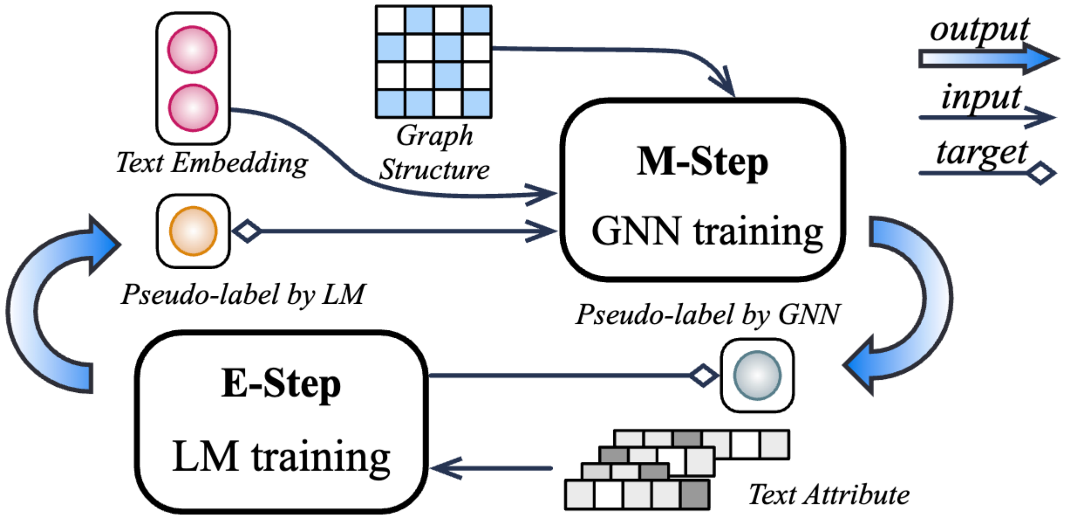
- **Graph and Language Learning by Expectation Maximization (GLEM)**

  - **Language Models (LMs)** for Node Classification

$$\mathbf{h}_n = \mathrm{SeqEnc}_{\theta_1}(\mathbf{s}_n)$$
$$p_\theta(\mathbf{y}_n|\mathbf{s}_n) = \mathrm{Cat}(\mathbf{y}_n \mid \mathrm{softmax}(\mathrm{MLP}_{\theta_2}(\mathbf{h}_n)))$$

  - **Graph Neural Networks (GNNs)** for Node Classification

$$\mathbf{h}_n^{(l)} = \sigma(\mathrm{AGG}_\phi(\mathrm{MSG}_\phi(\mathbf{h}_{\mathrm{NB}(n)}^{(l-1)}), A))$$
$$p_\phi(\mathbf{y}_n|A) = \mathrm{Cat}(\mathbf{y}_n \mid \mathrm{softmax}(\mathbf{h}_n^{(L)}))$$



[1] Learning on Large-scale Text-attributed Graphs via Variational Inference, ICLR 2023

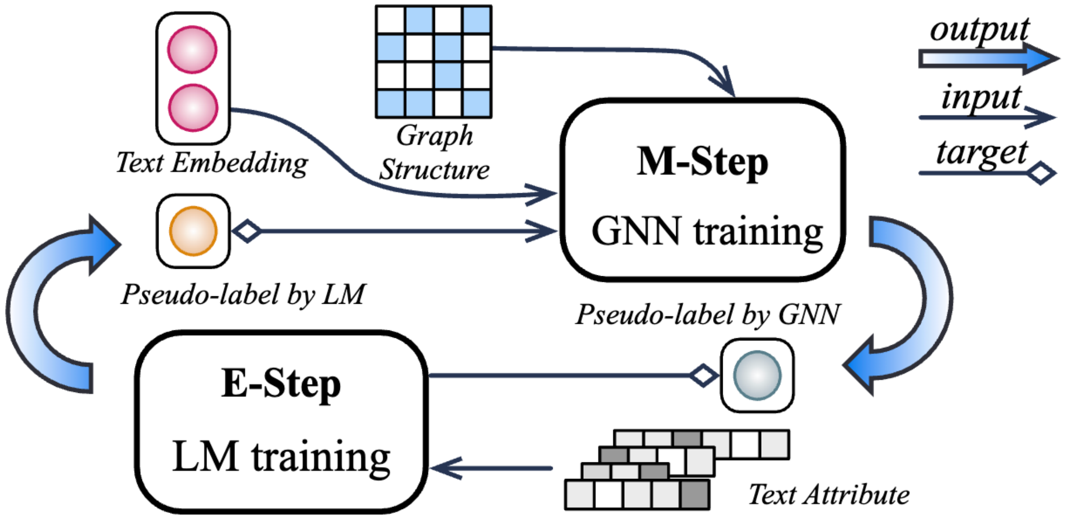
Alignment bet. GNNs and LLMs

- **Graph and Language Learning by Expectation Maximization (GLEM)**

  - **GLEM** leverages a variational **EM framework** for optimization

    - In the **E-step**, the GNN is fixed, and the LM mimics the labels inferred by the GNN.

    - In the **M-step**, the LM is fixed, and the GNN is optimized by using the node representations learned by the LM as features and the node labels inferred by the LM as target



[1] Learning on Large-scale Text-attributed Graphs via Variational Inference, ICLR 2023

Alignment bet. GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Graph and Language Learning by Expectation Maximization (GLEM)**
  - **GLEM** achieves better performance improvements with various GNN backbones

Datasets	Methods		GNN					LM		
			X_{OGB}	X_{GIANT}	X_{PLM}	GLEM-GNN	G↑	LM-Ft	GLEM-LM	L↑
Arxiv	GCN	*val*	73.00 ± 0.17	74.89 ± 0.17	47.56 ± 1.91	76.86 ± 0.19	3.86	75.27 ± 0.09	76.17 ± 0.47	0.90
		test	71.74 ± 0.29	73.29 ± 0.10	48.19 ± 1.47	<u>75.93 ± 0.19</u>	4.19	74.13 ± 0.04	75.71 ± 0.24	1.58
	SAGE	*val*	72.77 ± 0.16	75.95 ± 0.11	56.16 ± 0.46	76.45 ± 0.05	3.68	75.27 ± 0.09	75.32 ± 0.04	0.6
		test	71.49 ± 0.27	74.35 ± 0.14	56.39 ± 0.82	75.50 ± 0.24	4.01	74.13 ± 0.04	74.53 ± 0.12	1.44
	GAMLP	*val*	62.20 ± 0.11	75.01 ± 0.02	71.14 ± 0.19	76.95 ± 0.14	14.75	75.27 ± 0.09	75.64 ± 0.30	0.44
		test	56.53 ± 0.02	73.35 ± 0.14	70.15 ± 0.22	75.62 ± 0.23	19.09	74.13 ± 0.04	74.48 ± 0.41	2.04
	RevGAT	*val*	75.01 ± 0.10	77.01 ± 0.09	71.40 ± 0.23	77.49 ± 0.17	2.48	75.27 ± 0.09	75.75 ± 0.07	0.48
		test	74.02 ± 0.18	75.90 ± 0.19	70.21 ± 0.30	**76.97 ± 0.19**	2.95	74.13 ± 0.04	75.45 ± 0.12	1.32
Products	SAGE	*val*	91.99 ± 0.07	93.47 ± 0.14	86.74 ± 0.31	93.84 ± 0.12	1.85	91.82 ± 0.11	92.71 ± 0.15	0.71
		test	79.21 ± 0.15	82.33 ± 0.37	71.09 ± 0.65	83.16 ± 0.19	3.95	79.63 ± 0.12	81.25 ± 0.15	1.61
	GAMLP	*val*	93.12 ± 0.03	93.99 ± 0.04	91.65 ± 0.17	94.19 ± 0.01	1.07	91.82 ± 0.11	90.56 ± 0.04	-1.26
		test	83.54 ± 0.09	83.16 ± 0.07	80.49 ± 0.19	85.09 ± 0.21	1.55	79.63 ± 0.12	82.23 ± 0.27	2.60
	SAGN+	*val*	93.02 ± 0.04	93.64 ± 0.05	92.78 ± 0.04	94.00 ± 0.03	0.98	91.82 ± 0.11	92.01 ± 0.05	0.21
		test	84.35 ± 0.09	<u>86.67 ± 0.09</u>	84.20 ± 0.39	**87.36 ± 0.07**	3.01	79.63 ± 0.12	84.83 ± 0.04	5.17
Papers	GAMLP	*val*	71.17 ± 0.14	72.70 ± 0.07	69.78 ± 0.07	71.71 ± 0.09	0.54	68.05 ± 0.03	69.94 ± 0.16	1.89
		test	67.71 ± 0.20	69.33 ± 0.06	65.94 ± 0.10	68.25 ± 0.14	0.54	63.52 ± 0.06	64.80 ± 0.06	1.78
	GAMLP+	*val*	71.59 ± 0.05	73.05 ± 0.04	69.87 ± 0.06	73.54 ± 0.01	1.95	68.05 ± 0.03	71.16 ± 0.45	3.11
		test	68.25 ± 0.11	<u>69.67 ± 0.05</u>	66.36 ± 0.09	**70.36 ± 0.02**	2.11	63.52 ± 0.06	66.71 ± 0.25	3.19

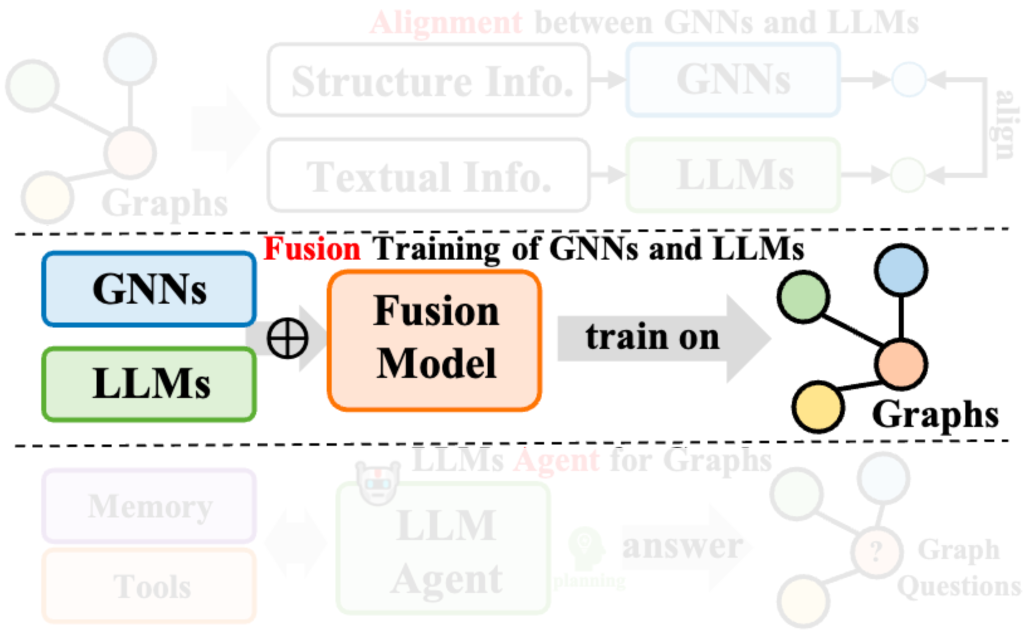
[1] Learning on Large-scale Text-attributed Graphs via Variational Inference, ICLR 2023

Fusion of GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Motivation**: Achieve a higher level of integration between LLMs and GNNs

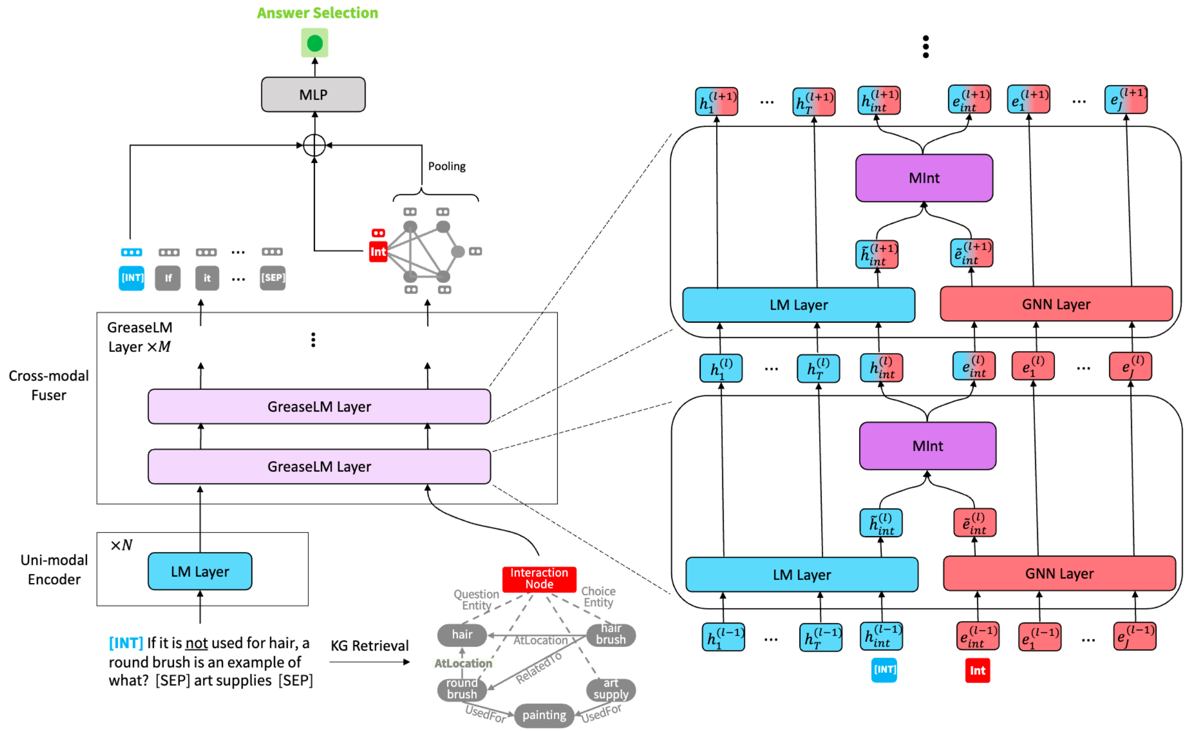
- **Methods**
  - Fuse Transformer Layer with Graph Neural Network Layer



Fusion of GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **GreaseLM** uses a cross-modal fusion component to inject information from the KG into language representation and information from language into KG representation.



- An **interaction node** will connect all nodes in the graph and conduct information fusion with the latent rep. of a specific token in the sequence.

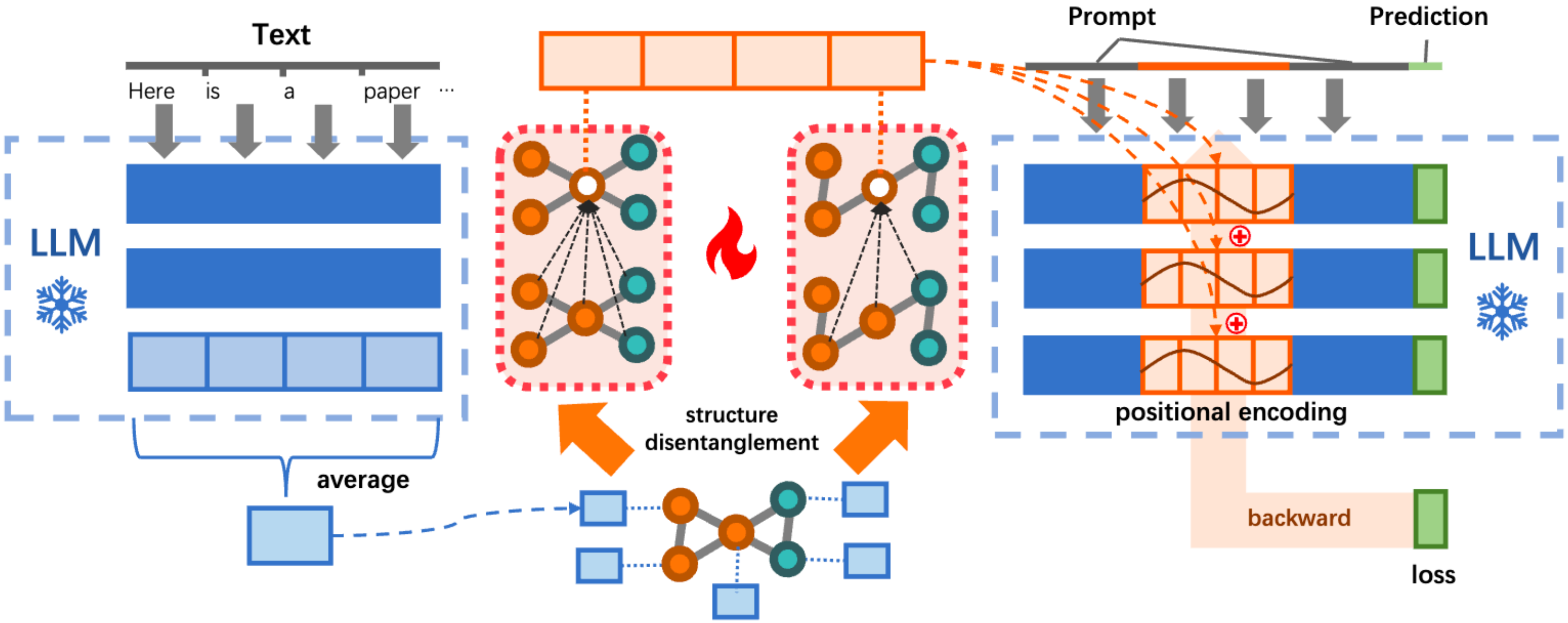
- GreaseLM achieves high performance on **Question-Answering tasks** with the structured knowledge from graphs.

[1] GreaseLM: Graph REASoning Enhanced Language Models for Question Answering, ICLR2022

Fusion of GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Disentangled Graph-Text Learner (DGTL)**
  - DGTL injects the disentangled graph neural network representation into each layer of the large language models.
  - DGTL achieves high performance on both <u>citation network</u> and <u>e-commerce graph</u> tasks.

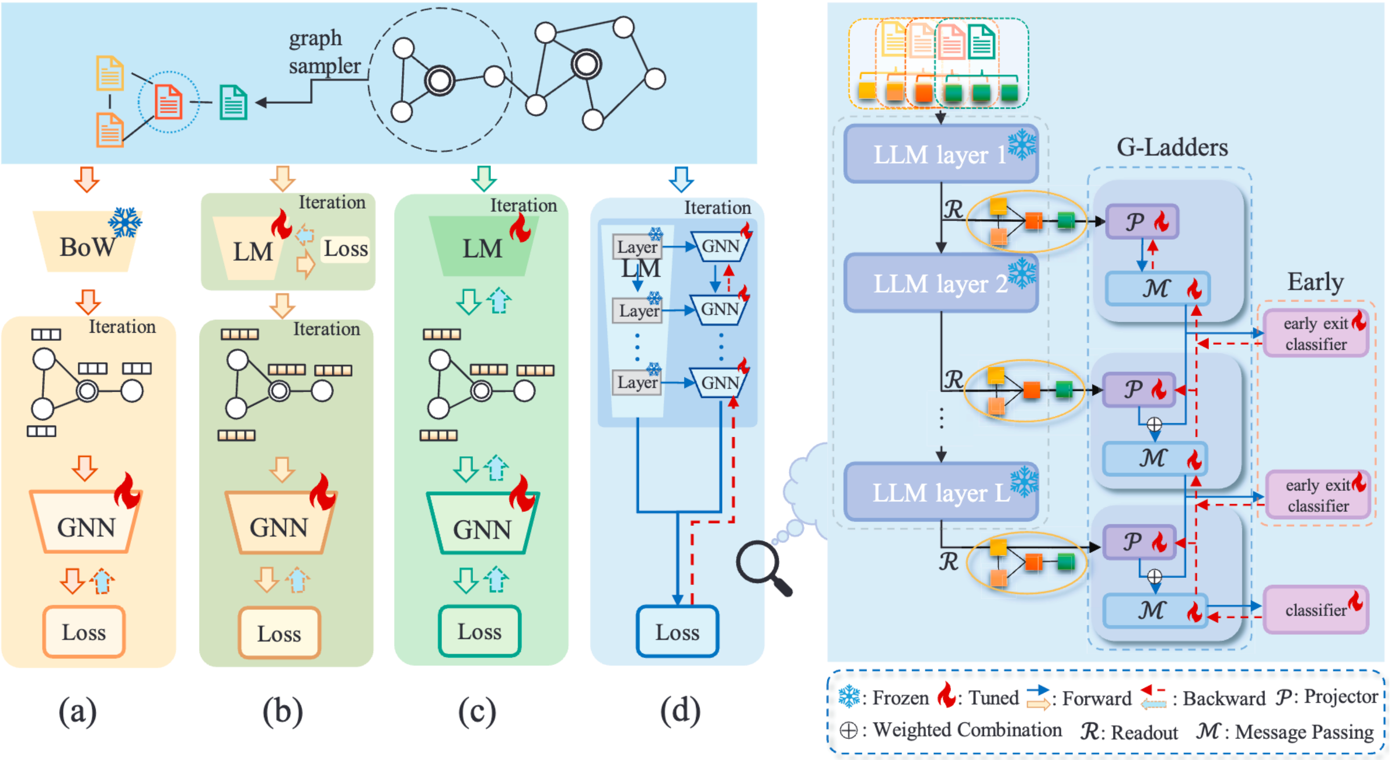


[1] Disentangled Representation Learning with Large Language Models for Text-Attributed Graphs, arXiv 2024

Fusion of GNNs and LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Efficient tuninG algorIthm for large laNguage models on tExtual graphs (EGINE)**



- ENGINE introduces a lightweight and adaptable **G-Ladder** module that is added to each layer of the LLMs.
- The **G-Ladder** employs a message-passing mechanism to incorporate structural information into the LLM.
- The **output of G-Ladder** is then used for classification from downstream tasks.

[1] Efficient Tuning and Inference for Large Language Models on Textual Graphs, arXiv 2024

Fusion of GNNs and LLMs

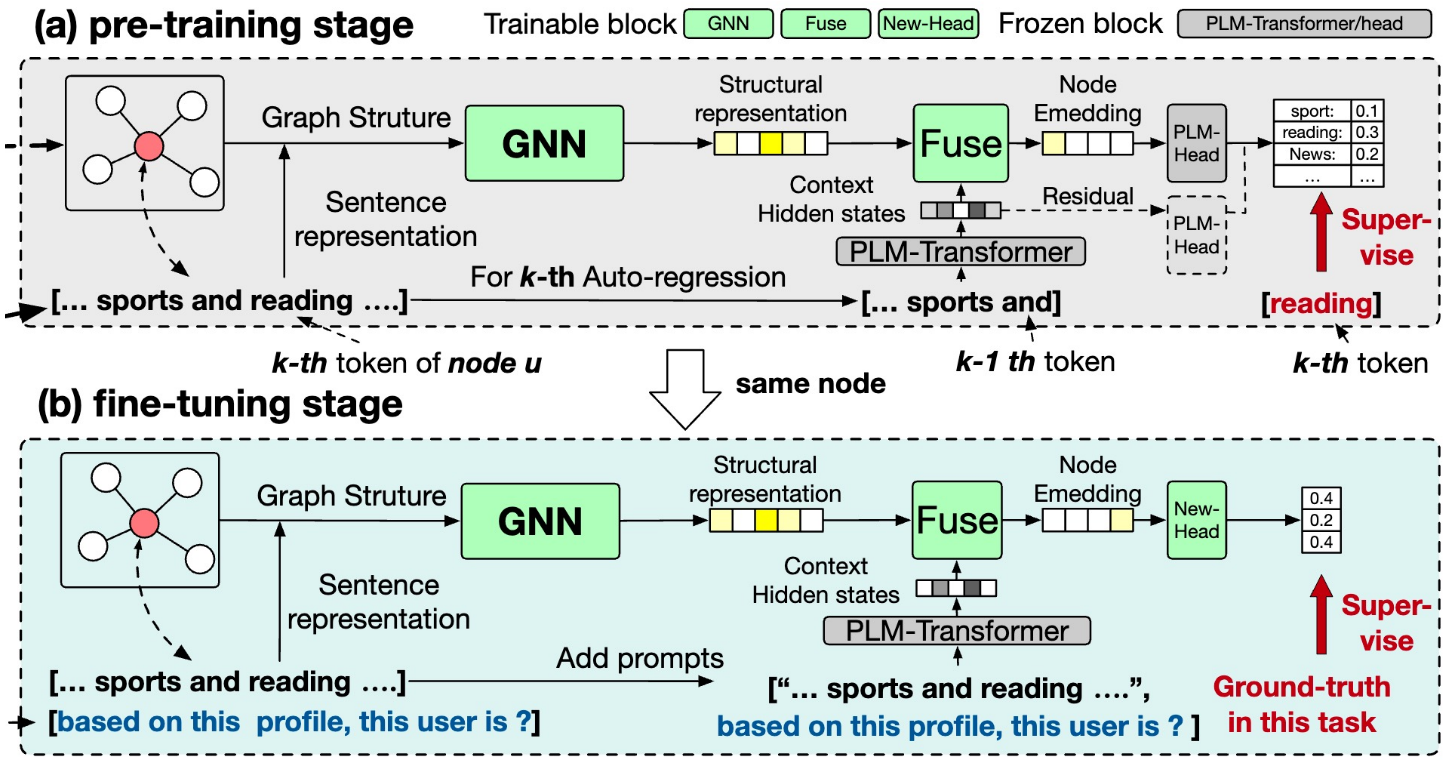
香 港 大 學
THE UNIVERSITY OF HONG KONG

- **GraphAdapter** uses a fusion module (e.g., MLPs) to combine the structural representations obtained from GNNs with the contextual hidden states of LLMs.

- This results in a **fused representation** that can be used for supervised training and prompting.

- **Pre-training:** Trains the GNN adapter through an autoregressive task, specifically predicting the next word.

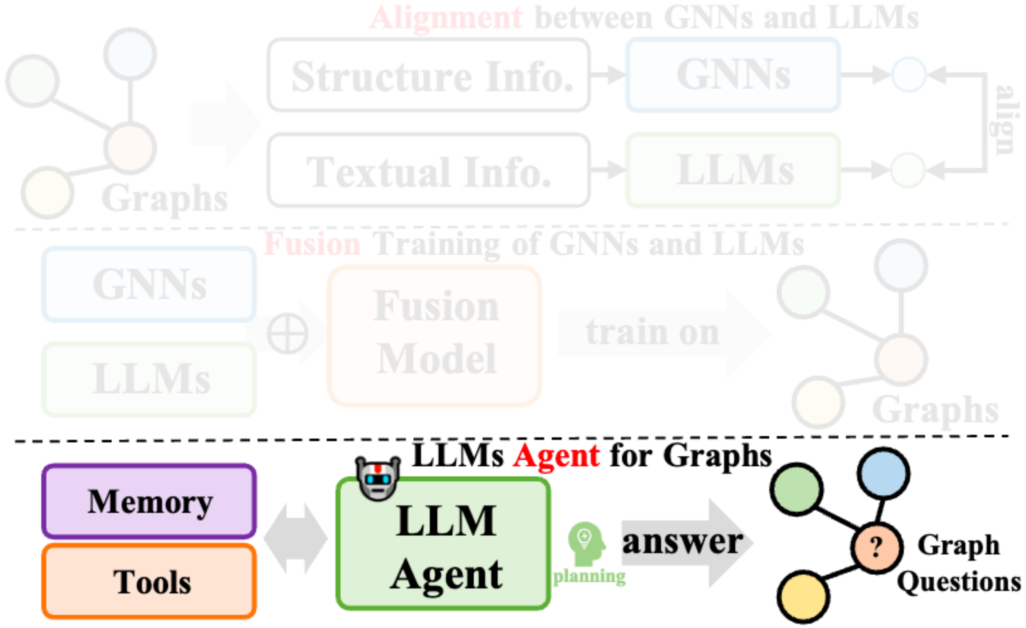
- **Fine-tuning:** insert task-specific prompts based on different downstream tasks to generate task-relevant node reps.



LLMs Agent for Graphs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Motivation**: Build LLM-based Agent for Graph Tasks
  - The model is capable of solving problems step-by-step by utilizing external tools and interacting with the graph.
- **Methods**
  - Perception Module
  - Memory Module
  - Action Module

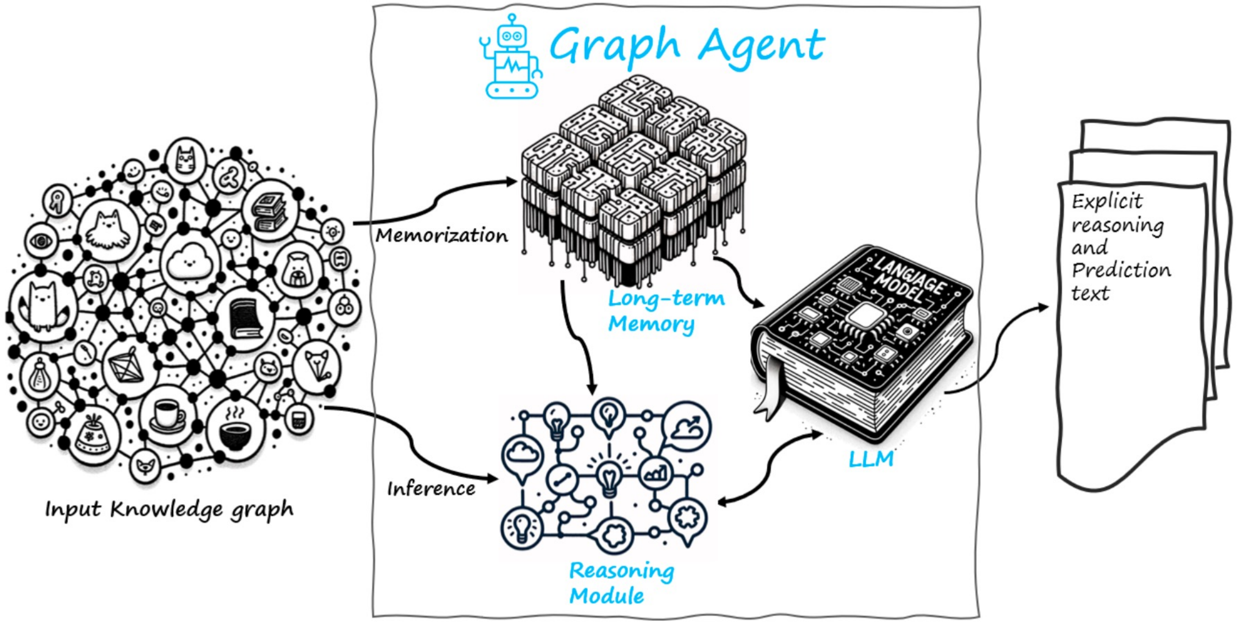


LLMs Agent for Graphs

- **Graph Agent (GA)** builds memory module and reasoning module for LLMs to build an agent for graph tasks.

  - **Memory Module:** Graph Agent converts graph data into textual descriptions and generates embedding vectors, which are stored in long-term memory.

  - **Reasoning Module:** During inference, GA retrieves similar samples from long-term memory and integrates them into a structured prompt, which is used by LLMs to explain the potential reasons for node classification or edge connection.

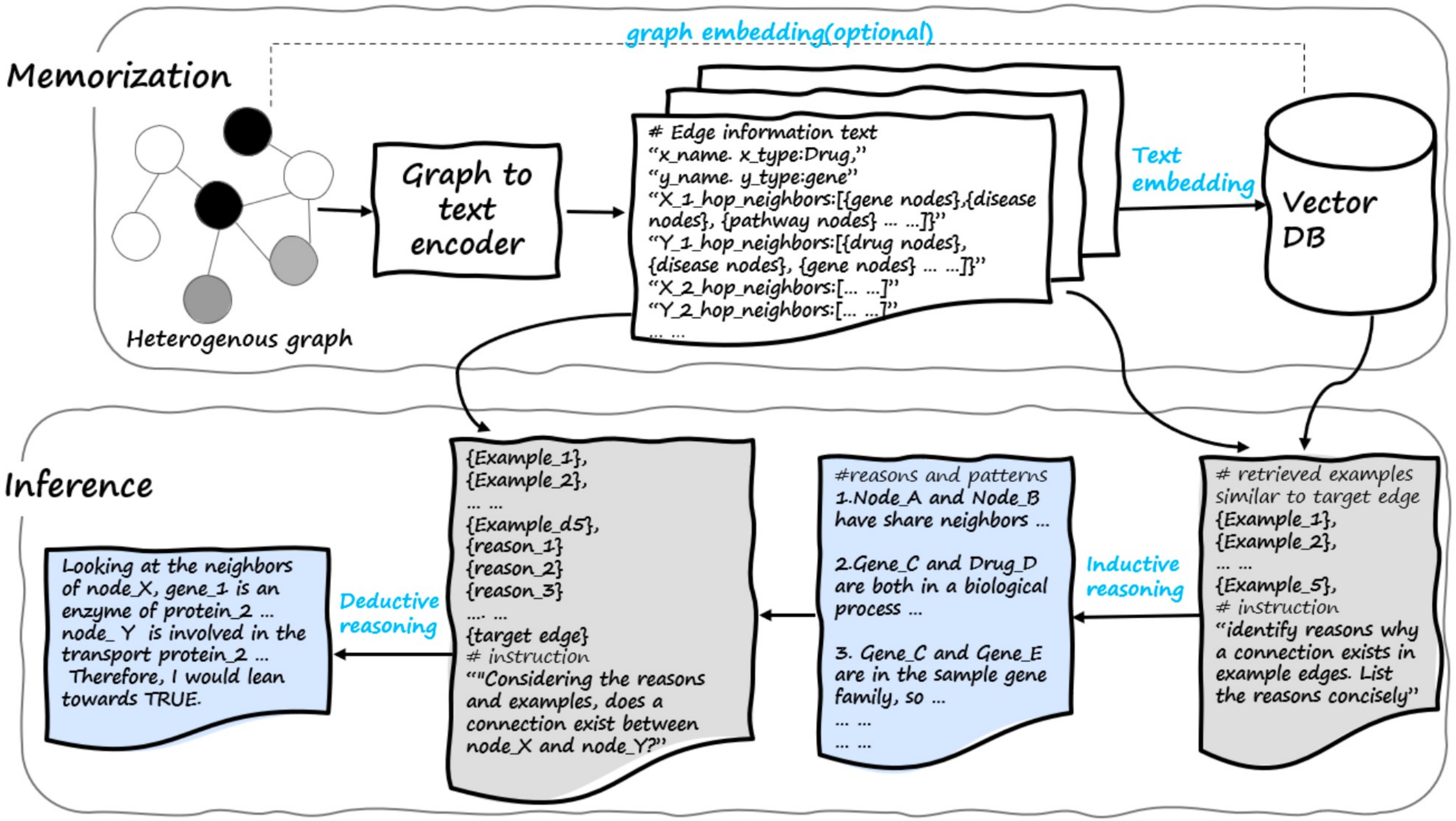


[1] Graph Agent: Explicit Reasoning Agent for Graphs, arXiv 2023

LLMs Agent for Graphs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Graph Agent (GA)** has an inductive-deductive reasoning paradigm.



[1] Graph Agent: Explicit Reasoning Agent for Graphs, arXiv 2023

LLMs Agent for Graphs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Readi** proposes a Reasoning-PathEditing framework to solve questions with Knowledge Graphs (KGs).

  - **Step 1: Reasoning Path Generation**

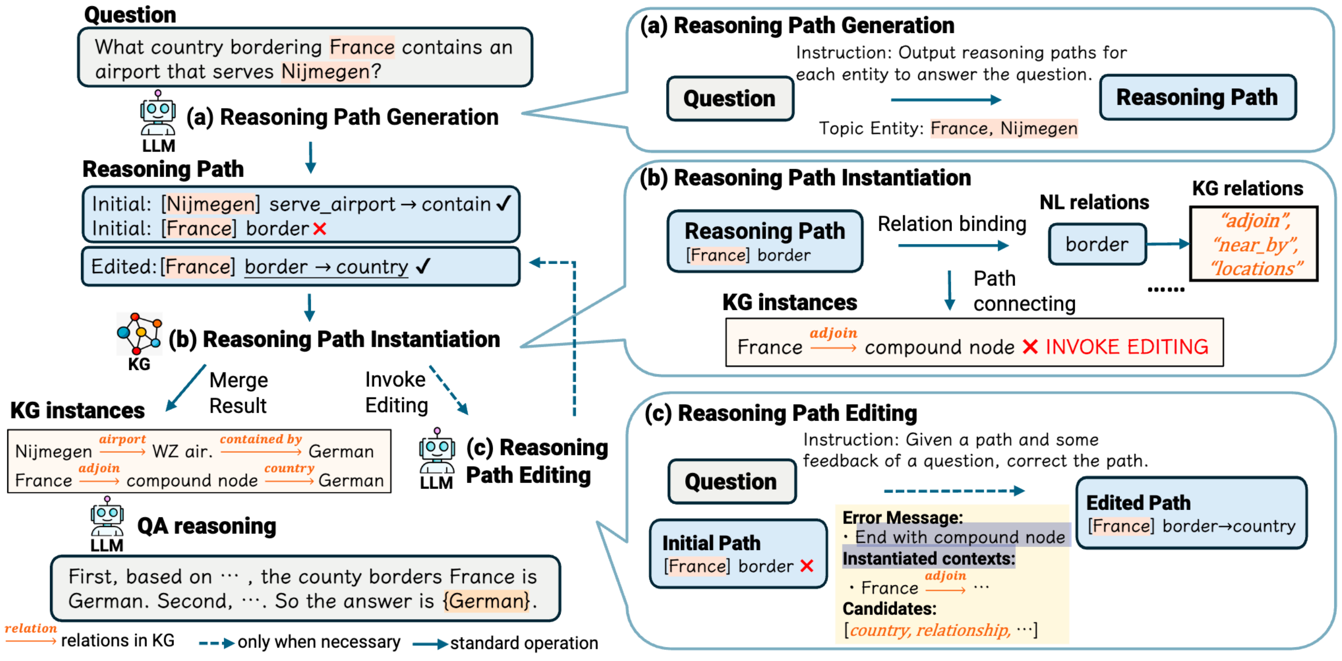
    Leverage LLMs to generate reasoning paths for the given questions.

  - **Step 2: Path Editing**

    If the path can not be initialized on the KG, then edit it with error message by LLMs.

  - **Step 3: Path Instantiation and Answering**

    Instantiating the paths and merge the paths into prompts for LLMs to given the final answer.



[1] Call Me When Necessary: LLMs can Efficiently and Faithfully Reason over Structured Environments, arXiv 2024

LLMs Agent for Graphs

- **Readi** achieves high performances on the QA tasks with knowledge graphs.

Methods	WebQSP	CWQ	MQA-1H	MQA-2H	MQA-3H
Training-based Method					
EmbedKGQA (Saxena et al., 2020)	66.6	-	**97.5**	98.8	94.8
NSM (He et al., 2021)	67.7	47.6	97.1	99.9	98.9
TransferNet (Shi et al., 2021)	71.4	48.6	97.5	**100***	**100***
SR+NSM+E2E (Zhang et al., 2022)	69.5	49.3	-	-	-
UniKGQA (Jiang et al., 2023c)	75.1	50.7	**97.5**	99.0	99.1
ReasoningLM (Jiang et al., 2023b)	78.5	**69.0***	96.5	98.3	92.7
RoG (Luo et al., 2024)	**85.7***	62.6	-	-	84.8
Inference-based Method					
Davinci-003 (Ouyang et al., 2022)	48.7	-	52.1	25.3	42.5
GPT3.5 (OpenAI, 2022)	65.7	44.7	61.9	31.0	43.2
GPT4 (OpenAI, 2023)	70.7	52.1	71.8	52.5	49.2
AgentBench (Liu et al., 2024b)	47.8	24.8	-	-	-
StructGPT (Jiang et al., 2023a)	69.6	-	97.1	97.3	87.0
Readi-GPT3.5	74.3	55.6	98.4	**99.9**	**99.4**
Readi-GPT4	**78.7**	**67.0**	**98.5***	**99.9**	99.2

[1] Call Me When Necessary: LLMs can Efficiently and Faithfully Reason over Structured Environments, arXiv 2024

LLMs Agent for Graphs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Reasoning on Graphs (RoG)** answers graph-retaled questions in 3 steps: planning, retrieval, and reasoning.

  - **Planning**

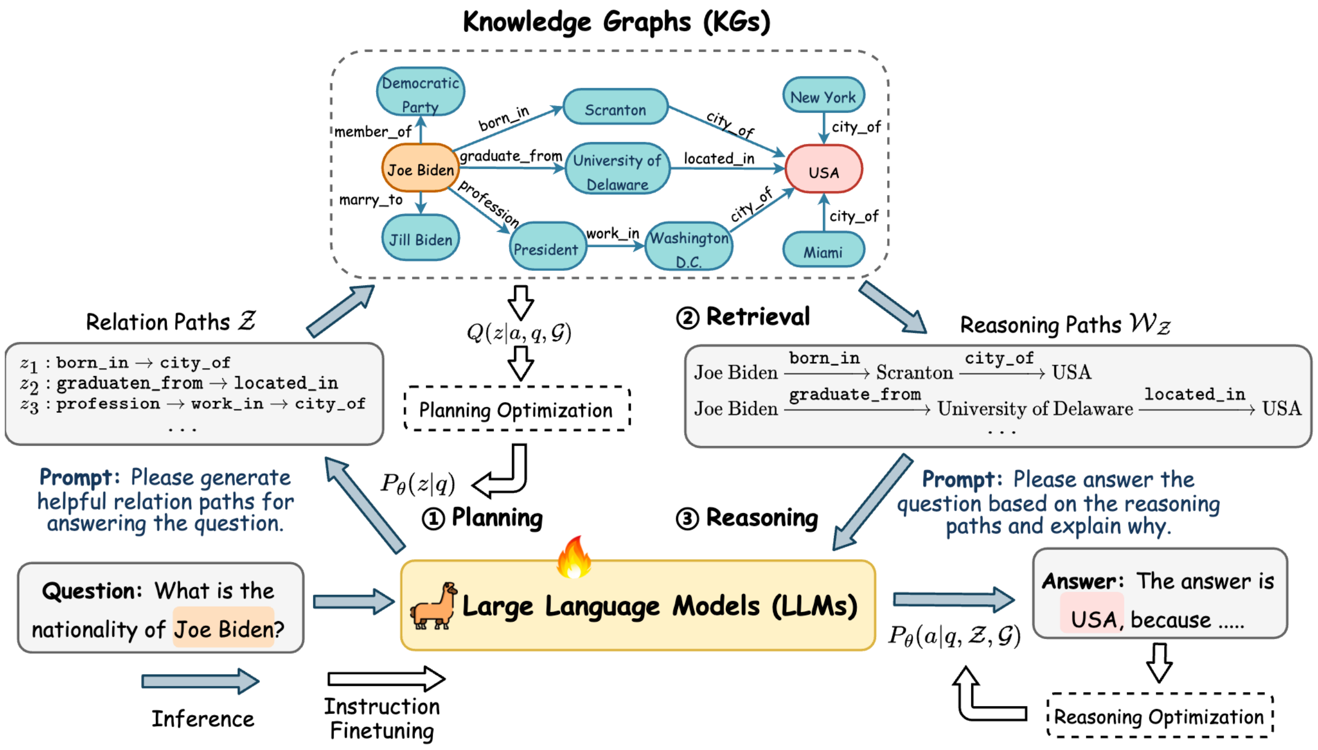
    LLMs generates a set of associated paths based on the structured information of the knowledge graph according to the problem

  - **Retrieval**

    it uses the associated paths generated in the planning stage to retrieve the corresponding reasoning paths from the KG.

  - **Reasoning** 🔥

    Finally, it uses the retrieved reasoning paths to generate the answer and explanation for the problem using LLMs.



LLMs Agent for Graphs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **RoG** also achieves high performances on the QA tasks with knowledge graphs.

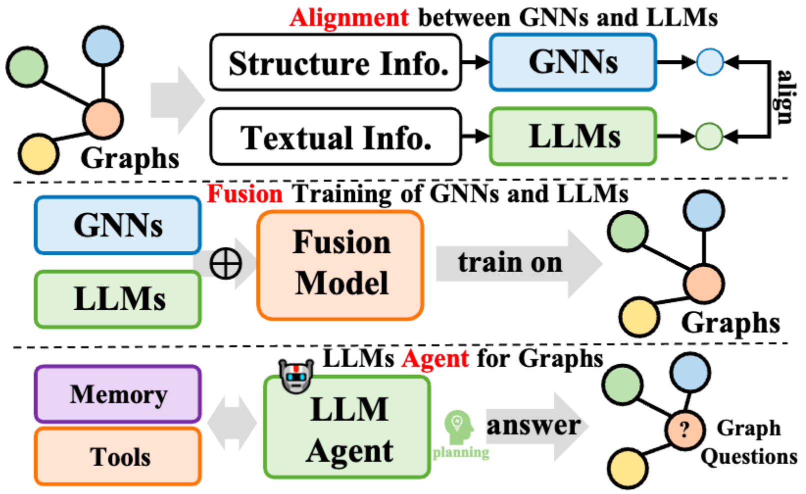
Table 1: Performance comparison with different baselines on the two KGQA datasets.

Type	Methods	WebQSP		CWQ	
		Hits@1	F1	Hits@1	F1
Embedding	KV-Mem (Miller et al., 2016)	46.7	34.5	18.4	15.7
	EmbedKGQA (Saxena et al., 2020)	66.6	-	45.9	-
	NSM (He et al., 2021)	68.7	62.8	47.6	42.4
	TransferNet (Shi et al., 2021)	71.4	-	48.6	-
	KGT5 Saxena et al. (2022)	56.1	-	36.5	-
Retrieval	GraftNet (Sun et al., 2018)	66.4	60.4	36.8	32.7
	PullNet (Sun et al., 2019)	68.1	-	45.9	-
	SR+NSM (Zhang et al., 2022)	68.9	64.1	50.2	47.1
	SR+NSM+E2E (Zhang et al., 2022)	69.5	64.1	49.3	46.3
Semantic Parsing	SPARQL (Sun et al., 2020)	-	-	31.6	-
	QGG (Lan & Jiang, 2020)	73.0	73.8	36.9	37.4
	ArcaneQA (Gu & Su, 2022)	-	75.3	-	-
	RnG-KBQA (Ye et al., 2022)	-	76.2	-	-
LLMs	Flan-T5-xl (Chung et al., 2022)	31.0	-	14.7	-
	Alpaca-7B (Taori et al., 2023)	51.8	-	27.4	-
	LLaMA2-Chat-7B (Touvron et al., 2023)	64.4	-	34.6	-
	ChatGPT	66.8	-	39.9	-
	ChatGPT+CoT	75.6	-	48.9	-
LLMs+KGs	KD-CoT (Wang et al., 2023b)	68.6	52.5	55.7	-
	UniKGQA (Jiang et al., 2022)	77.2	72.2	51.2	49.1
	DECAF (DPR+FiD-3B) (Yu et al., 2022a)	82.1	**78.8**	-	-
	RoG	**85.7**	70.8	**62.6**	**56.2**

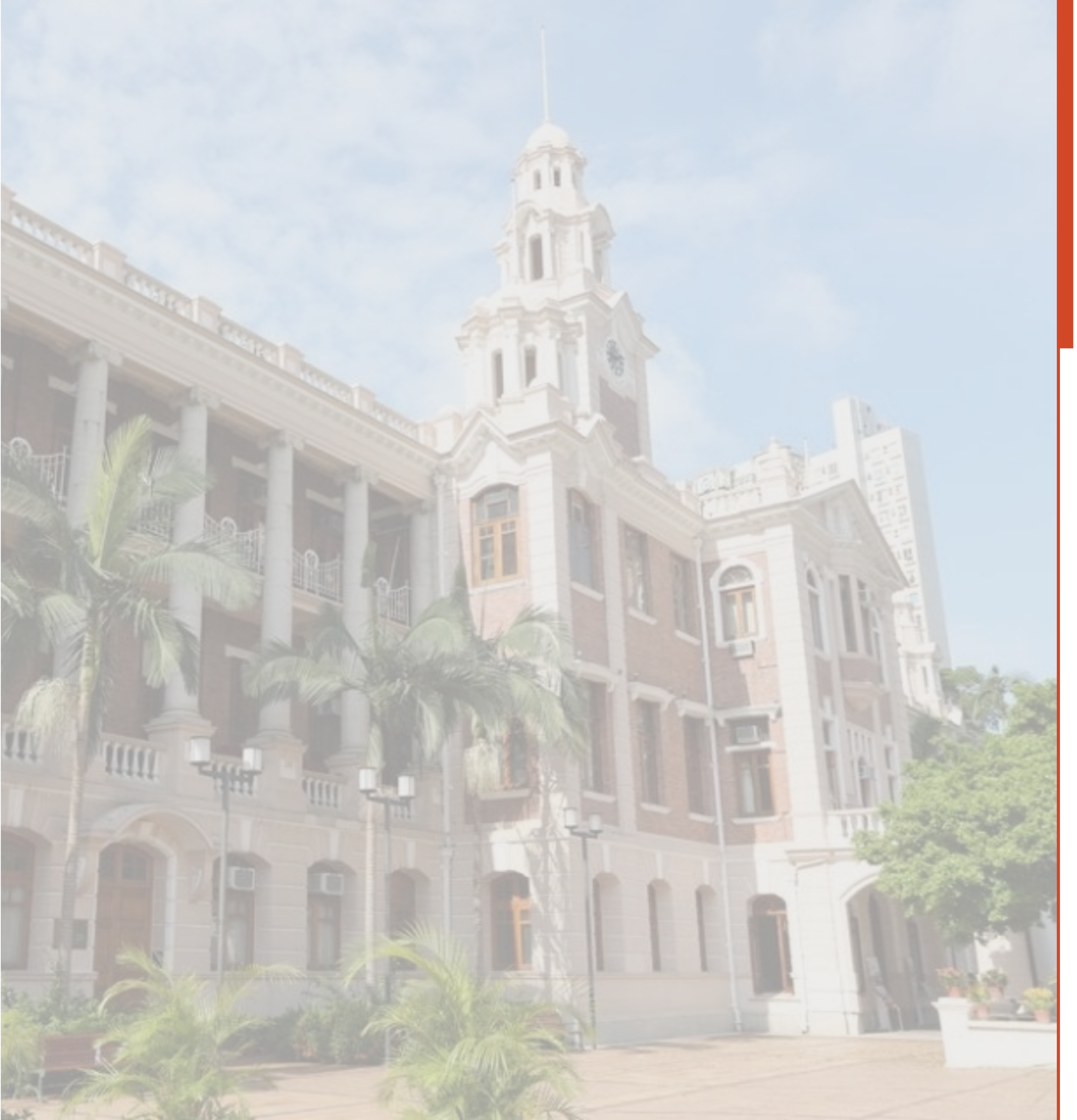
[1] Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning, ICLR 2024

LLMs-Graphs Intergration

香 港 大 學
THE UNIVERSITY OF HONG KONG



- LLMs-Graphs Intergration
  - Alignment between GNNs and LLMs
    - MoMu [43], ConGraT [2], G2P2 [55], GRENADE [27], MoleculeSTM [33], THLM [70], GLEM [67]
  - Fusion Training of GNNs and LLMs
    - GreaseLM [64], DGTL [40], ENGINE [69], GraphAdapter [24]
  - LLMs Agent for Graphs
    - Graph Agent [53], Readi [9], RoG [36]



Q & A

Xubin Ren

Personal Information

@xubinrencs

Github Page

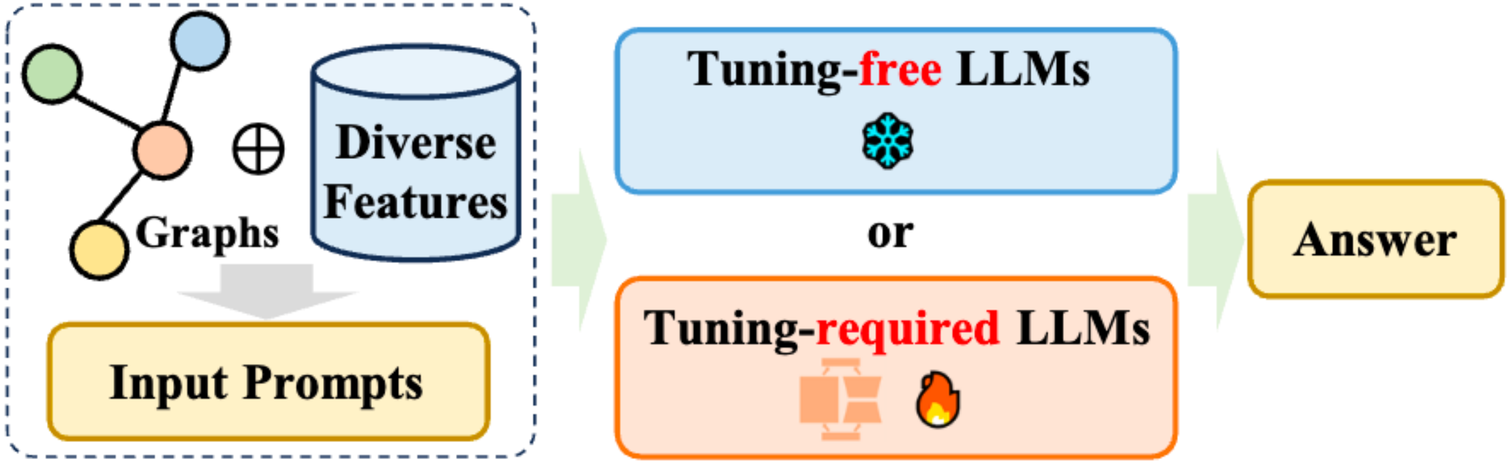
Group Information

LLMs-Only

LLMs

LLMs-Only

- Constructing **proper prompts** for LLMs to answer graph-related answers!
- **Tuning-free:** Design prompts from graph structured that LLMs can directly understand.
- **Tuning-required:** Instruction tuning LLMs to align the knowledge of graphs.



Tuning-free LLMs

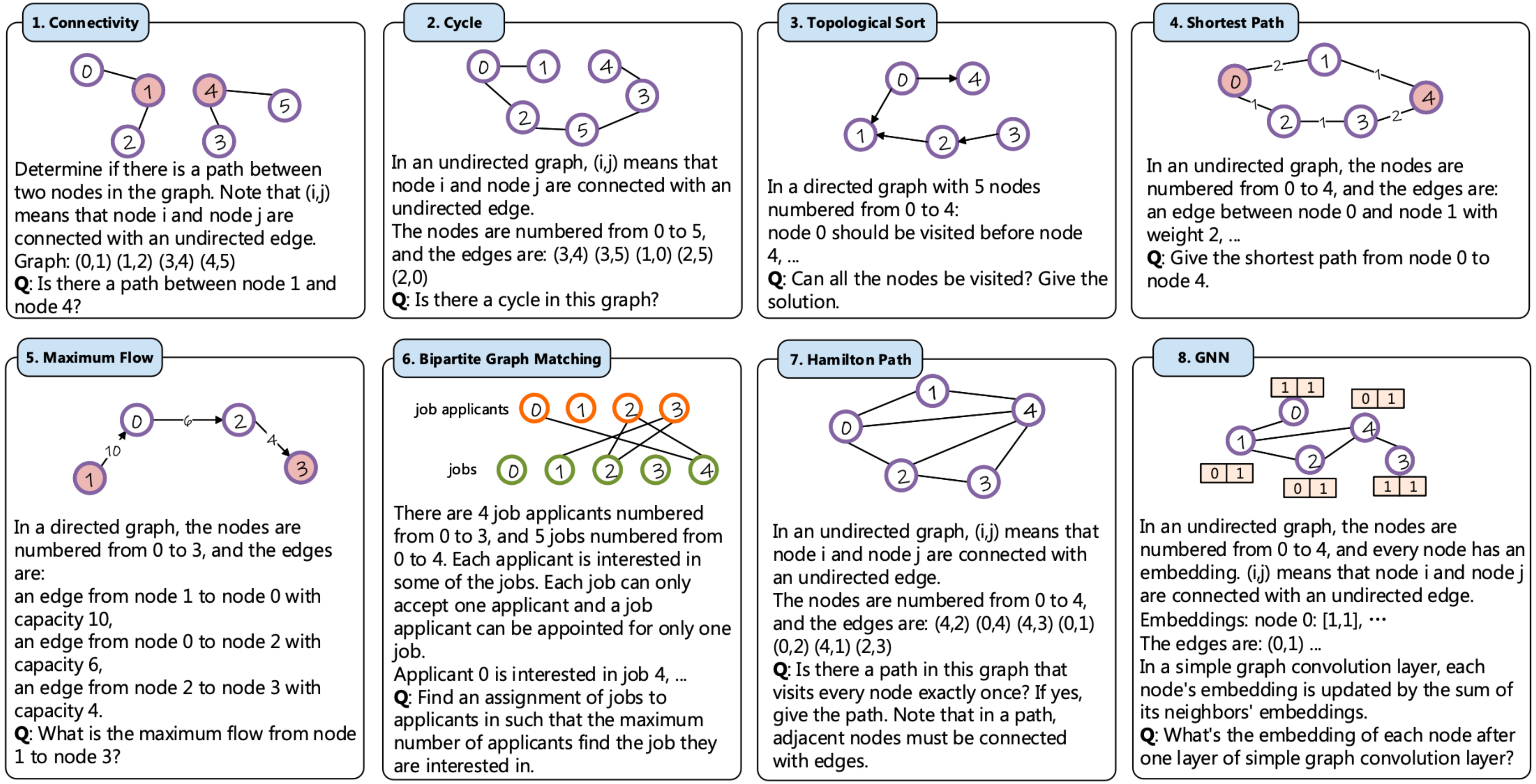
- **Motivation**: Translate the structure data of graphs into natural languages that LLMs can directly reasoning.

- **Challenges**
  - How to sequence the graph information (e.g., edges, nodes) ?
  - How to benchmark LLMs?

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **NLGraph** proposes a benchmark for graph-based problems and introduces instruction-based approach.
  - **NLGraph Benchmark** has 8 graph-related tasks (e.g., link prediction, shortest path, hamilton path).



[1] Can Language Models Solve Graph Problems in Natural Language?, NeurIPS 2023

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **NLGraph** proposes a benchmark for graph-based problems and introduces instruction-based approach.
  - **Promprting Techniques** can greatly improves the baseline performance (~37%-57%).
  - **CoT:** Chain-of-Thought
  - **SC :** Self-Consistency

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**

[1] Can Language Models Solve Graph Problems in Natural Language?, NeurIPS 2023

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **GPT4Graph** designs structure understanding and semantic understanding tasks.
  - **Structure Understanding Tasks:** Degree Calculation, Link Prediction …
  - **Semantic Understanding Tasks:** Question-Answering, Node Classification, Graph Classification …

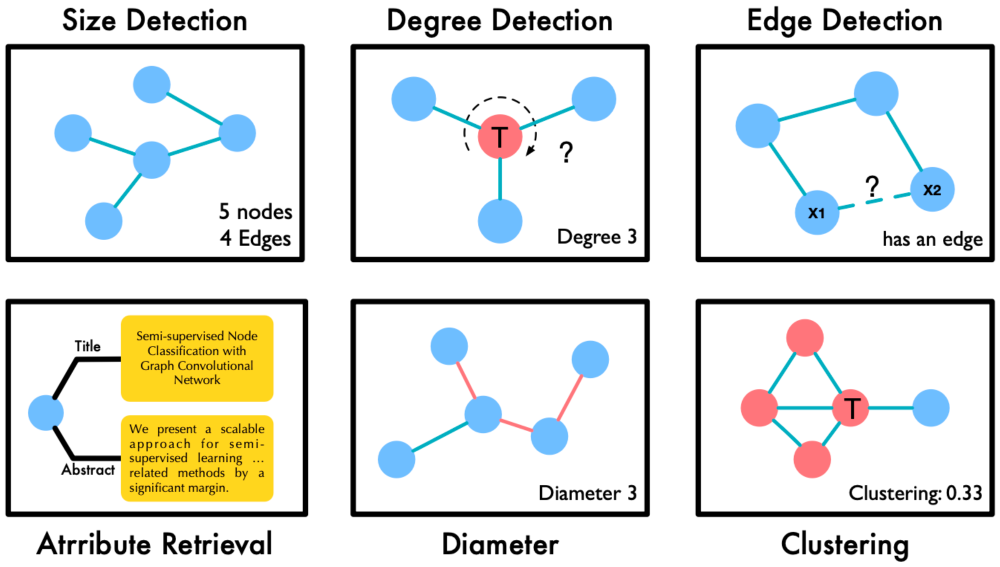


Figure 3: Structure Understanding Tasks

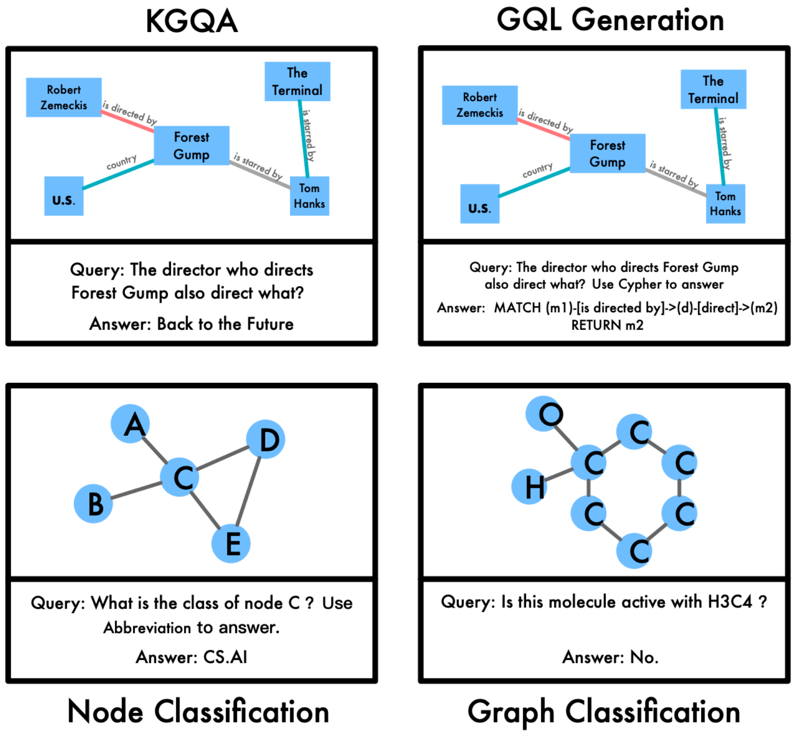


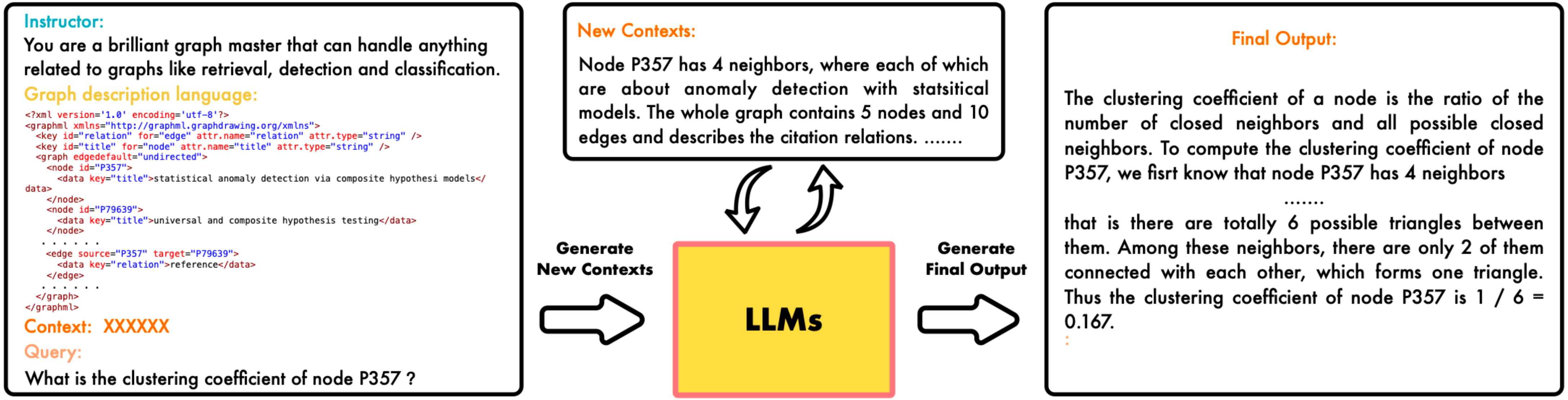
Figure 4: Semantic Understanding Tasks

[1] GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking, NeurIPS 2023

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Self-Prompting Paradigm** in GPT4Graph
  - **First Step:** Asking LLMs to automatically generate the context of the input graph.
  - **Second Step:** The generated new context is merged with the original input to give the final answer.

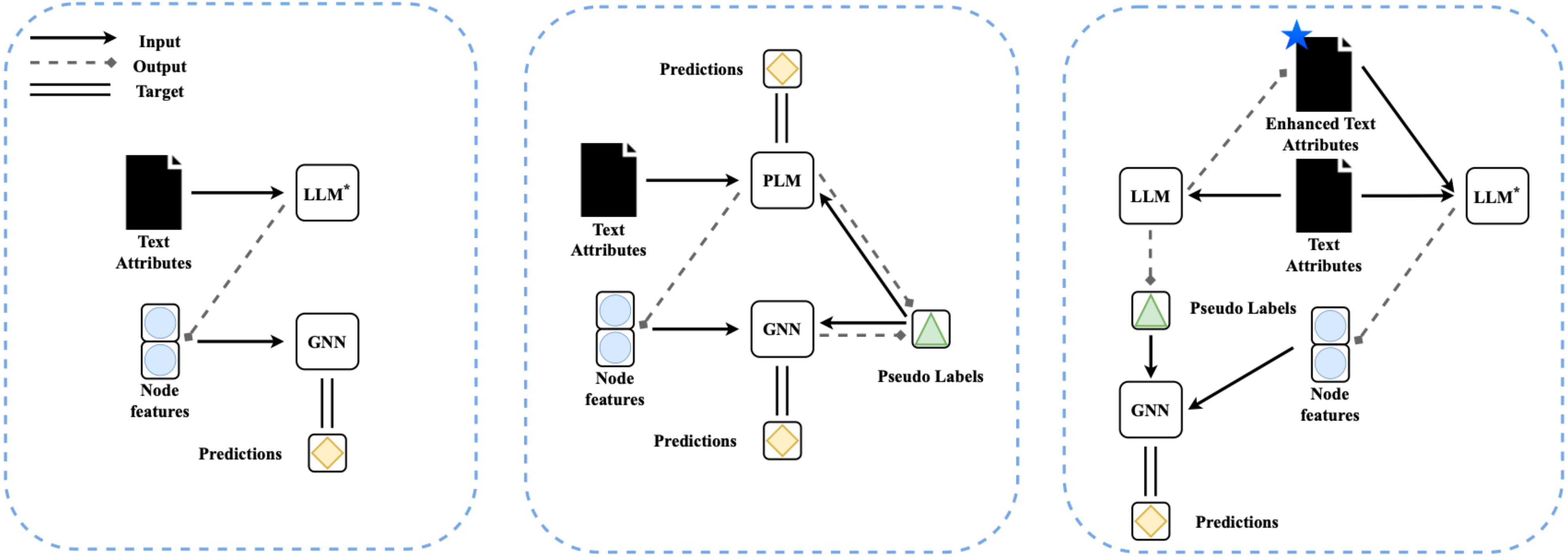


[1] GPT4Graph: Can Large Language Models Understand Graph Structured Data? An Empirical Evaluation and Benchmarking, NeurIPS 2023

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Graph-LLM** proposes two frameworks to handle the node classification task with LLMs.
  - **LLMs-as-Enhancers:** Given node features or enhanced text attributes.
  - **Three strategies to adopt LLMs as Enhancers.**

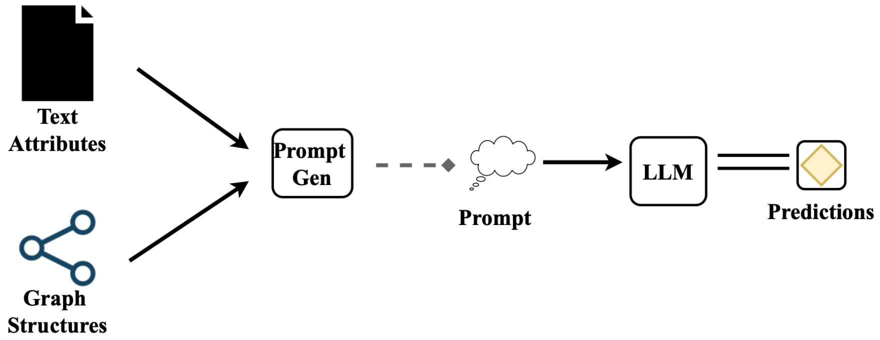


[1] Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs, KDD 2024

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **Graph-LLM** proposes <u>two frameworks</u> to handle the node classification task with LLMs.
  - **LLMs-as-Predictors:** LLMs directly give the answers based on constructed prompts
  - Design effective prompt to incorporate **structural and attribute information.**



can sample multiple times and summarize each of them to obtain more fine-grained neighborhood information.

Observation 15. Neighborhood summarization is likely to achieve performance gain.

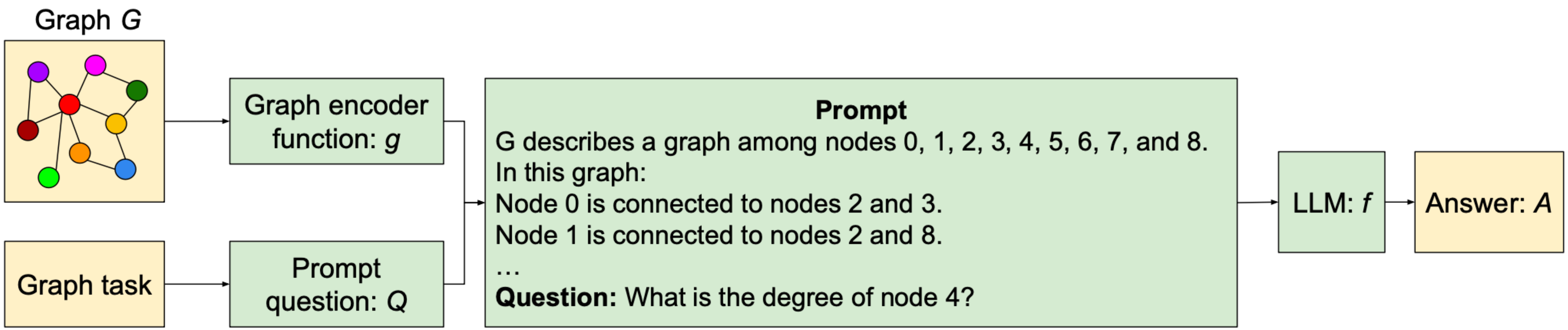
From Table 14, we note that incorporating neighborhood information in either zero-shot or few-shot approaches yields performance gains compared to the zero-shot prompt without structural information except on the PUBMED dataset. By following the "homophily" assumption [87; 39], which

Paper: Title: C-reactive protein and incident cardiovascular events among men with diabetes.
Abstract: OBJECTIVE: Several large prospective studies have shown that baseline levels of C-reactive protein (CRP)
...
Neighbor Summary: This paper focuses on different aspects of **type 2 diabetes** mellitus. It explores the levels of various markers such as tumor necrosis factor-alpha, interleukin-2 ...
Ground truth: "Diabetes Mellitus Type 1"
Structure-ignorant prompts: "Diabetes Mellitus Type 1"
Structure-aware prompt: "Diabetes Mellitus Type 2"
GNN: "Diabetes Mellitus Type 2"

[1] Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs, KDD 2024

Tuning-free LLMs

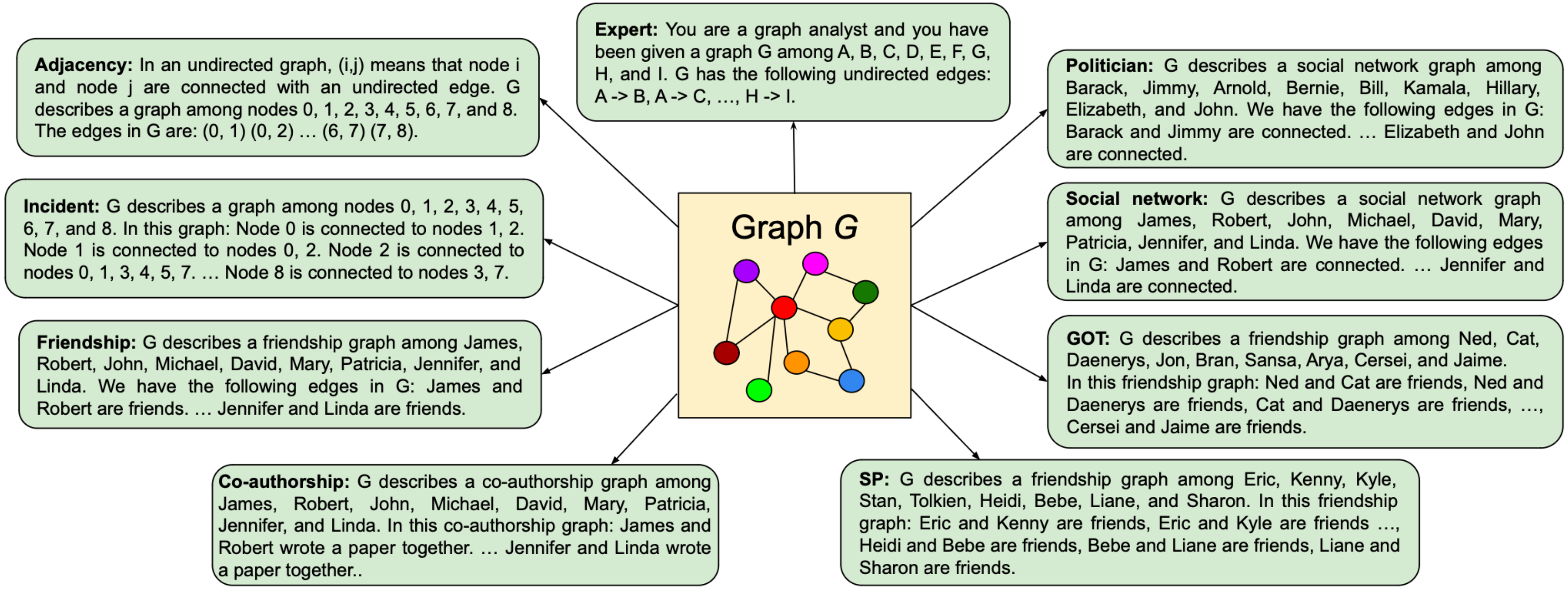
- **Talk Like a Graph** discovers different methods to encode graph into text for LLMs to solve various problems.
  - **Finding 1:** LLMs perform poorly on basic graph tasks.
  - **Finding 2:** The graph encoding function has a significant impact on LLM graph reasoning
  - **Finding 3:** Model capacity has a significant effect on performance of LLMs on graph reasoning tasks.



[1] Talk Like a Graph: Encoding Graphs for Large Language Models, arXiv 2023

Tuning-free LLMs

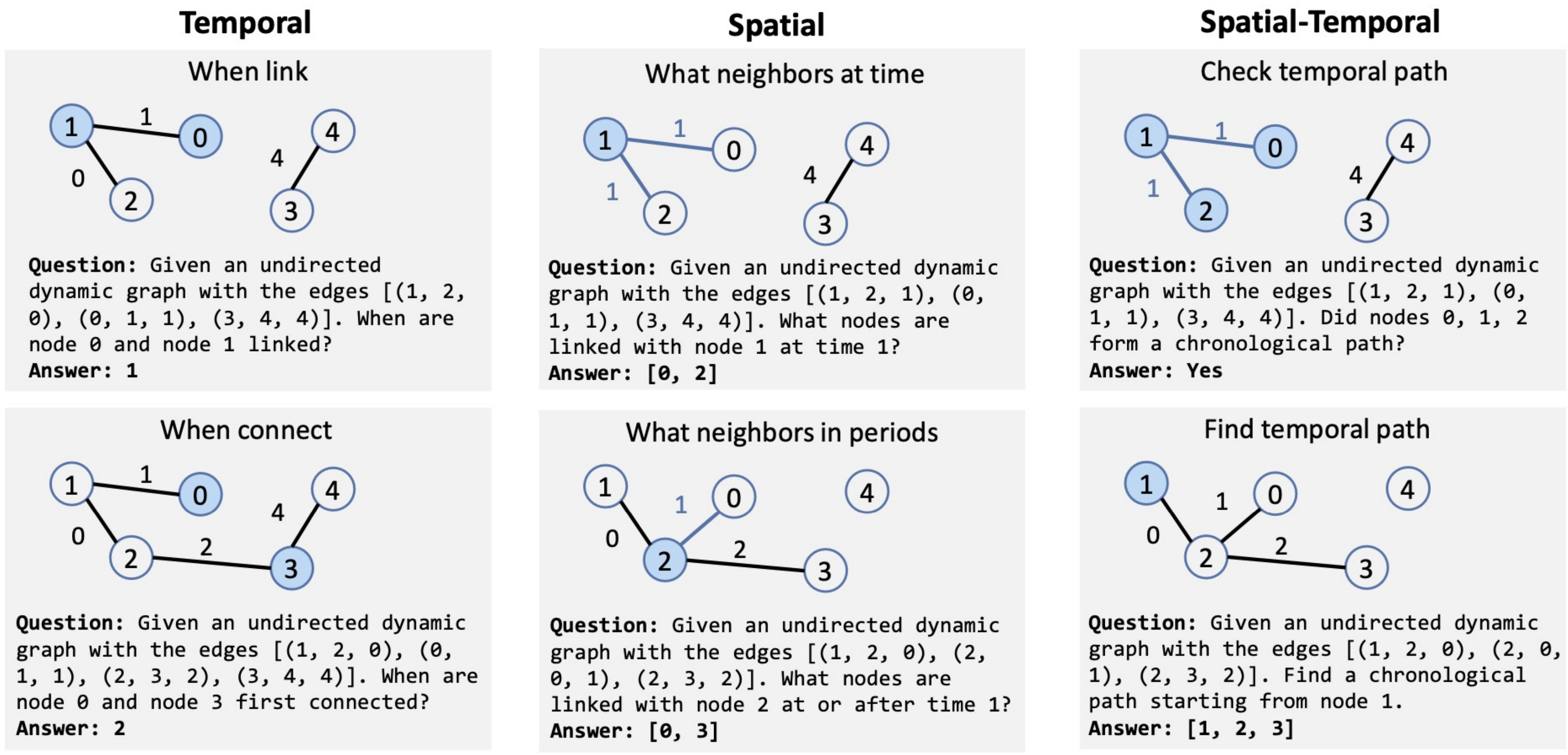
- **Various Graph Encoding methods** in Talk Like a Graph:
  - **Adjacency.** Using integer node encoding and parenthesis edge encoding.
  - **Incident**. Using integer node encoding and incident edge encoding.
  - …



[1] Talk Like a Graph: Encoding Graphs for Large Language Models, arXiv 2023

Tuning-free LLMs

- **LLM4DyG** leverages LLMs to handling spatial-temporal dynamic graphs.
  - **benchmarks** the spatial-temporal comprehension of LLMs on dynamic graphs.
  - One more number in the edge indicates **the timestamp.**



[1] LLM4DyG: Can Large Language Models Solve Spatial-Temporal Problems on Dynamic Graphs?, arXiv 2023

Tuning-free LLMs

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **LLM4DyG** suggests the **Disentangled Spatial-Temporal Thoughts (DST2)**
  - General advanced prompting techniques do not guarantee a performance boost here.
  - **DST2** instructs the LLM to sequentially think about the nodes or time.

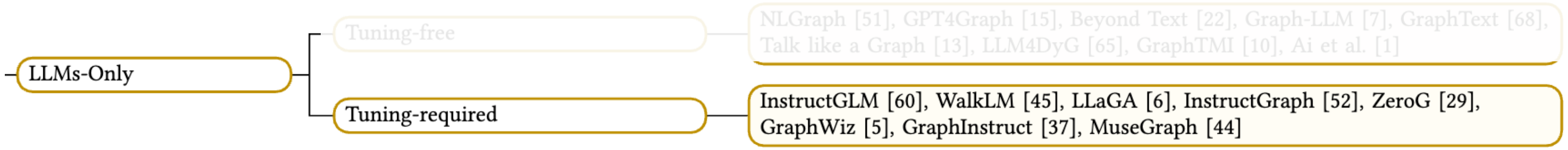
Table 5: Model performance (ACC%) on the dynamic graph tasks with one-shot prompting method and our proposed DST2 prompting methods (v1 to v4). The best and the second-best results for each task are in bold and underlined respectively.

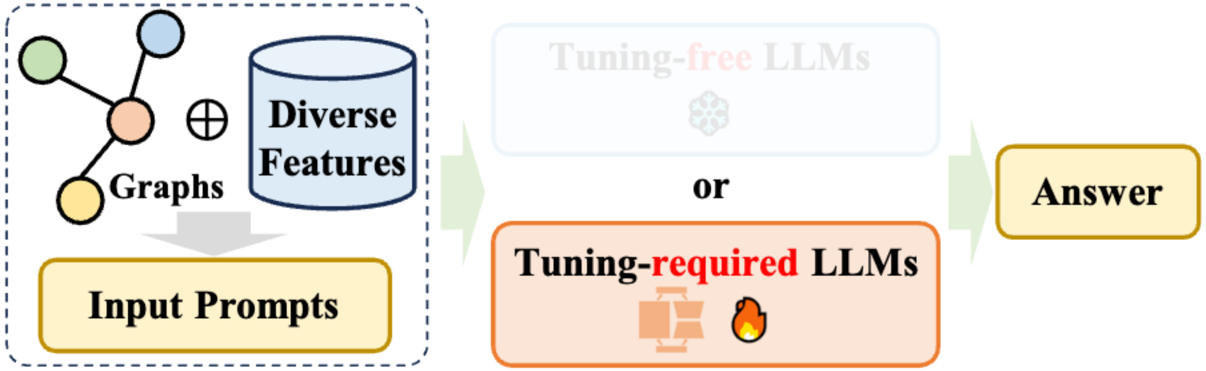
Task	Temporal			Spatial			Spatial-Temporal		
Prompting methods	when link	when connect	when tclosure	neighbor at time	neighbor in periods	check tclosure	check tpath	find tpath	sort edge
one-shot prompt	$33.7_{\pm2.1}$	$\underline{77.0}_{\pm2.9}$	$73.0_{\pm1.6}$	$34.0_{\pm1.4}$	$15.7_{\pm4.2}$	$66.7_{\pm4.5}$	$\mathbf{63.7}_{\pm2.6}$	$78.3_{\pm6.0}$	$29.3_{\pm4.0}$
v1: Think (about) nodes and then time	$40.0_{\pm1.6}$	$77.0_{\pm4.1}$	$\mathbf{74.0}_{\pm1.4}$	$34.0_{\pm0.8}$	$15.0_{\pm4.2}$	$\mathbf{69.3}_{\pm1.7}$	$61.0_{\pm3.3}$	$\mathbf{79.0}_{\pm7.5}$	$30.0_{\pm3.6}$
v2: Think (about) time and then nodes	$37.3_{\pm2.6}$	$76.7_{\pm3.4}$	$\underline{73.3}_{\pm0.5}$	$31.7_{\pm1.9}$	$\underline{15.7}_{\pm3.4}$	$\underline{67.0}_{\pm2.9}$	$61.3_{\pm1.9}$	$\mathbf{79.0}_{\pm7.5}$	$\mathbf{30.7}_{\pm3.9}$
v3: Pick nodes and then time	$\underline{59.3}_{\pm2.1}$	$\mathbf{77.0}_{\pm2.4}$	$68.0_{\pm0.8}$	$\underline{35.0}_{\pm2.9}$	$\mathbf{16.7}_{\pm4.7}$	$65.0_{\pm3.7}$	$62.3_{\pm2.9}$	$78.0_{\pm5.4}$	$\underline{30.0}_{\pm2.9}$
v4: Pick time and then nodes	$\mathbf{76.7}_{\pm1.7}$	$76.3_{\pm3.9}$	$68.7_{\pm0.9}$	$\mathbf{35.7}_{\pm2.5}$	$15.3_{\pm3.3}$	$65.3_{\pm2.9}$	$\underline{63.3}_{\pm2.6}$	$78.3_{\pm5.8}$	$29.3_{\pm2.9}$

[1] LLM4DyG: Can Large Language Models Solve Spatial-Temporal Problems on Dynamic Graphs?, arXiv 2023

Tuning-required LLMs

- **Motivation**:
  - ➢ convert graphs into sequences in a specific way and align graph token sequences and natural language token sequences using fine-tuning methods

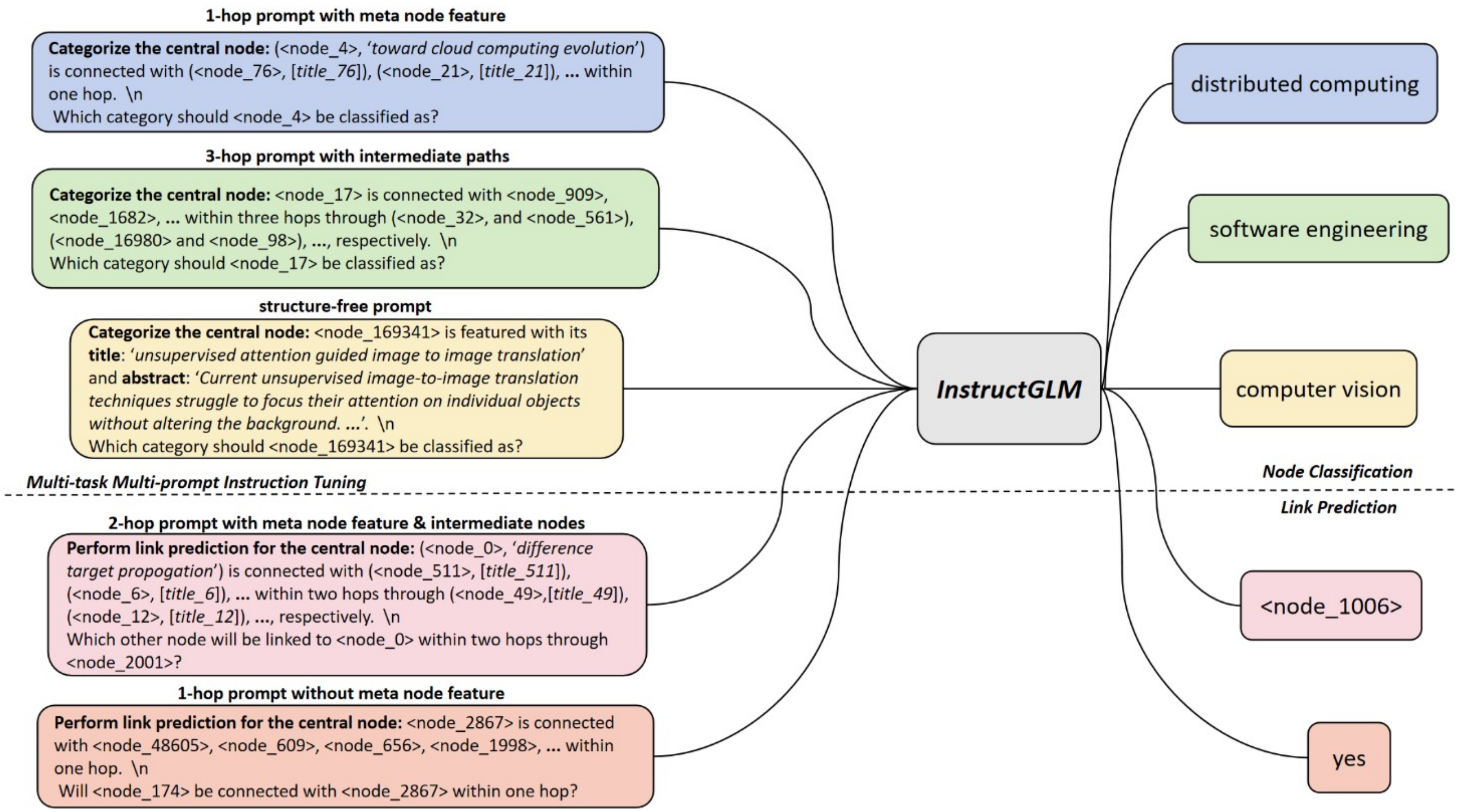




LLM-Only: InstructGLM

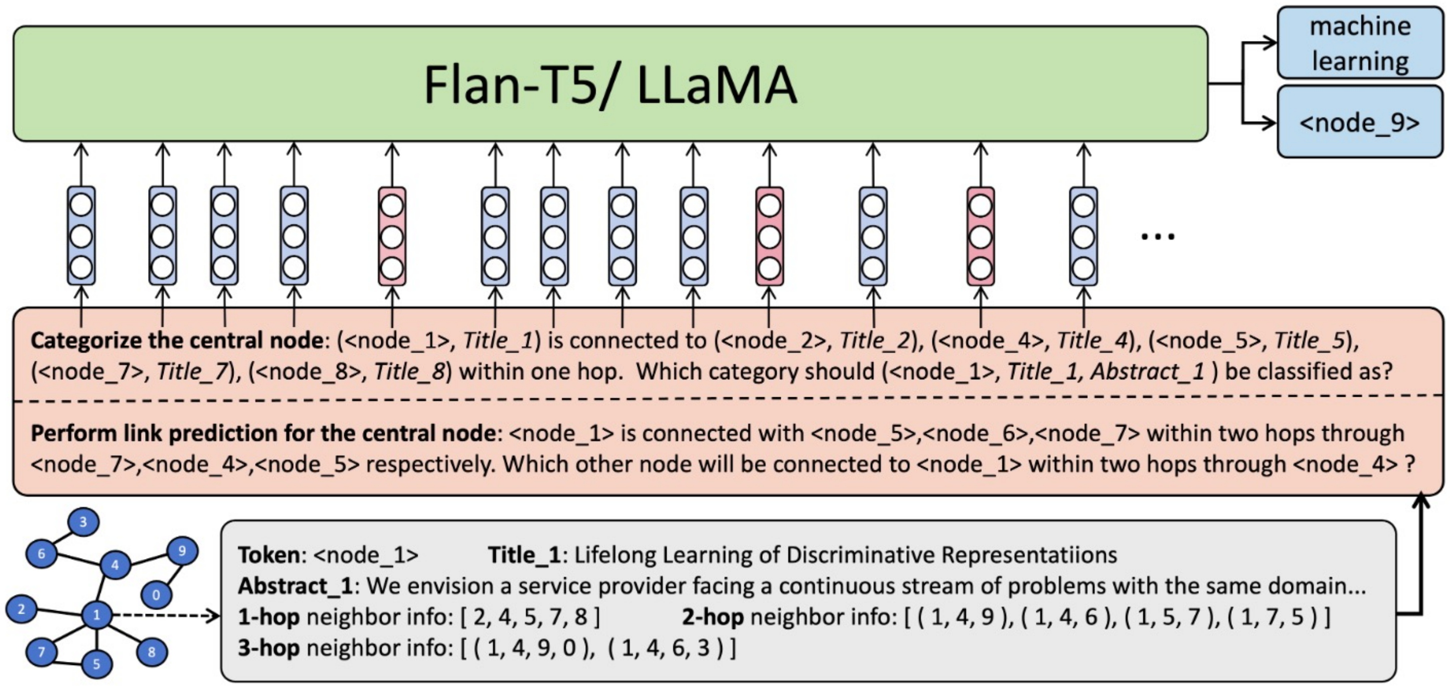
香 港 大 學
THE UNIVERSITY OF HONG KONG

- **InstructGLM:** Language is All a Graph Needs (EACL’24)



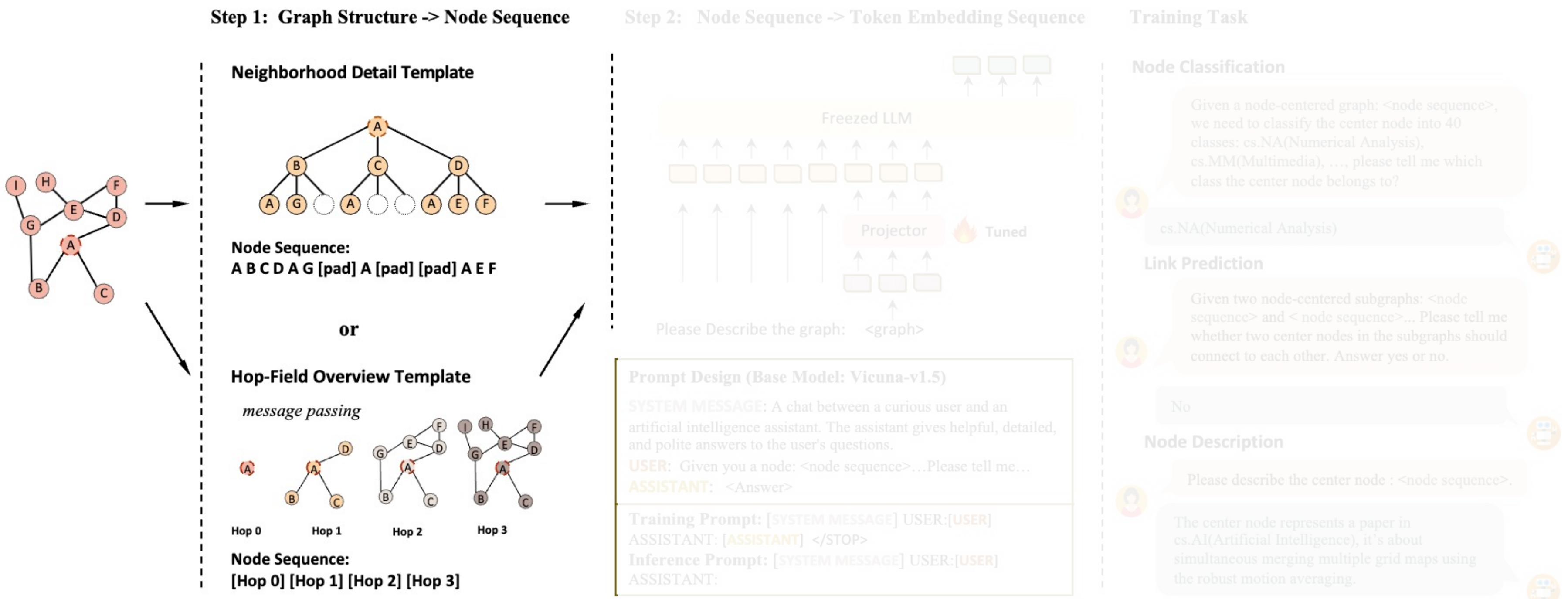
LLM-Only: InstructGLM

- **InstructGLM:** Language is All a Graph Needs (EACL'24)
  - ➢ Instruction Prompt Design
  - ➢ Generative Instruction Tuning for Node Classification
  - ➢ Auxiliary Self-Supervised Link Prediction



LLM-Only: LLaGA

- **LLaGA:** Large Language and Graph Assistant
  - ➢ Structure-Aware Graph Translation
    - ✓ Neighborhood Detail Template
    - ✓ Hop-Field Overview Template

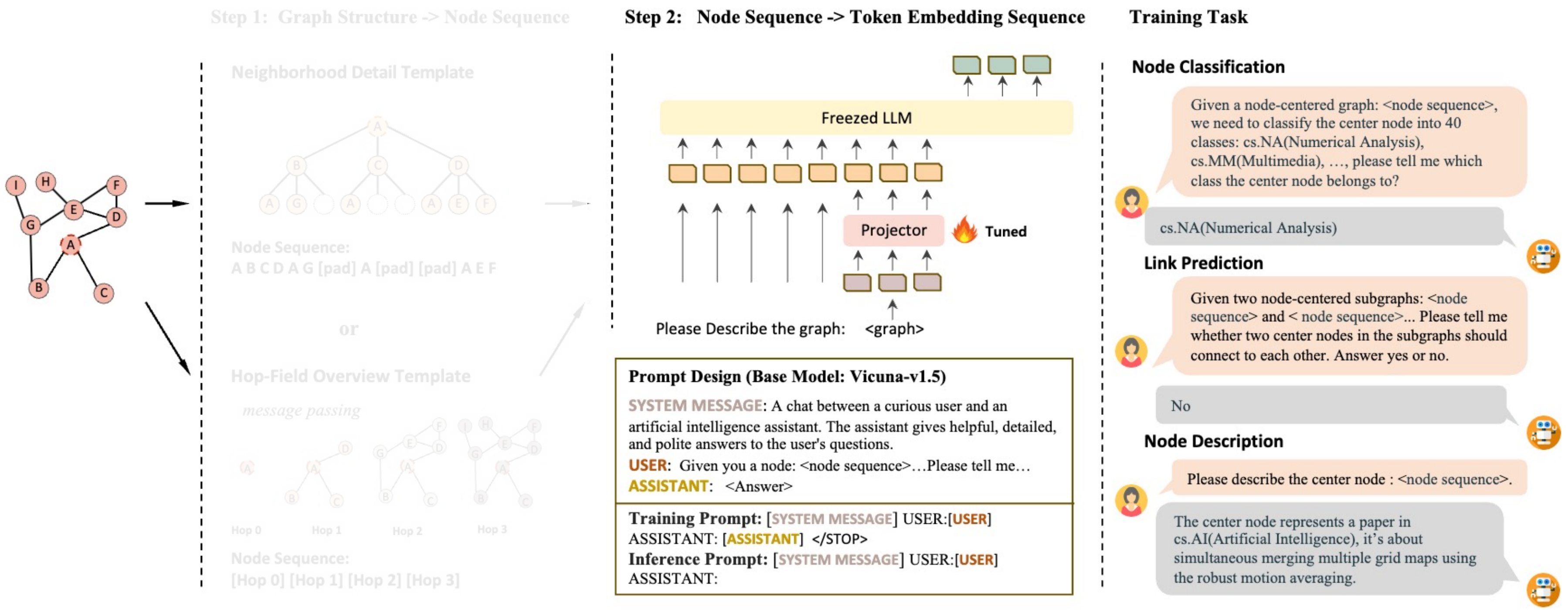


LLM-Only: LLaGA

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **LLaGA:** Large Language and Graph Assistant
  - ➢Alignment Tuning

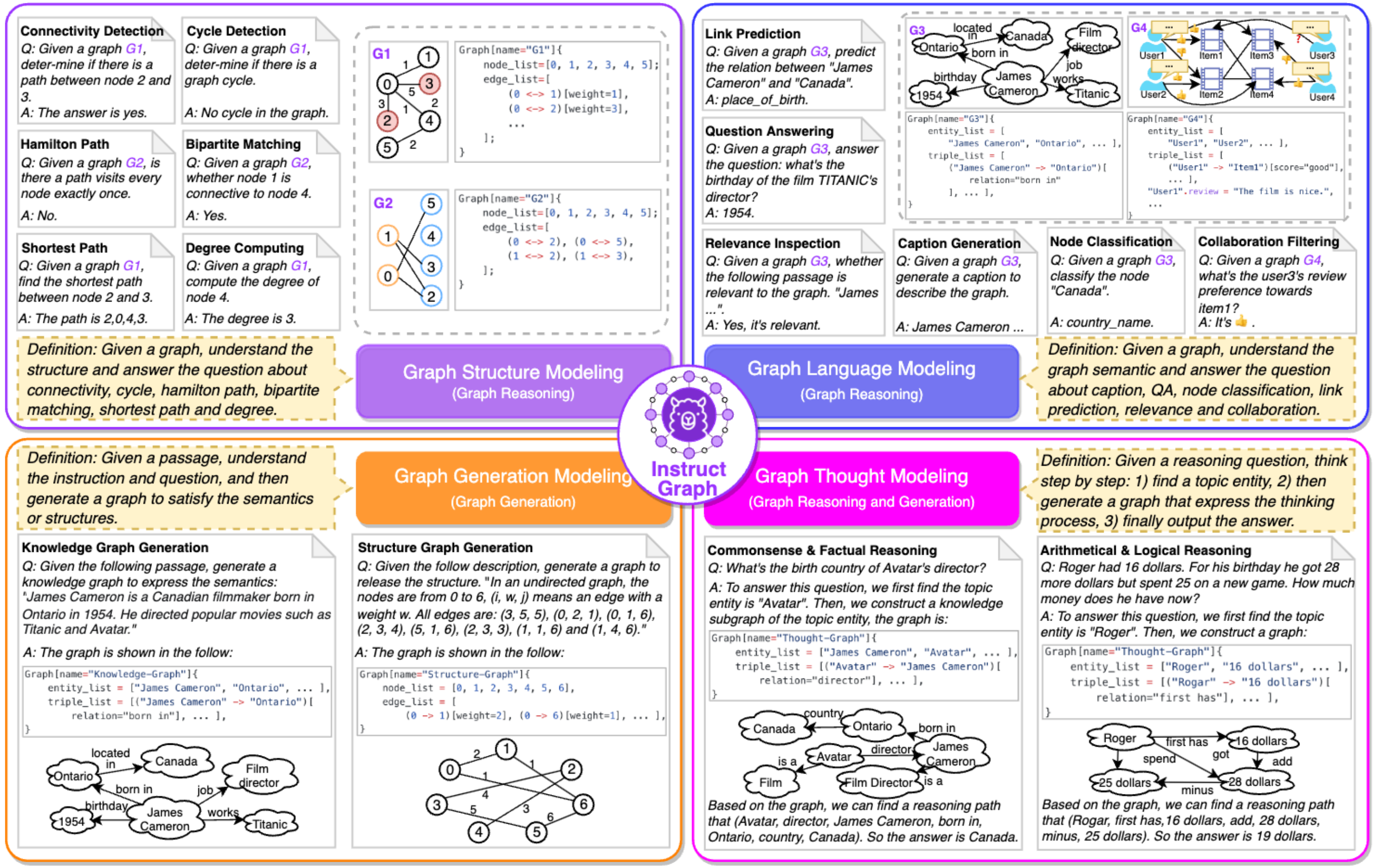
$$\underset{\theta}{\text{maximize}}\; p(X_{answer} | X_{graph}, X_{question}, X_{system}).$$



LLM-Only: InstructGraph

香 港 大 學
THE UNIVERSITY OF HONG KONG

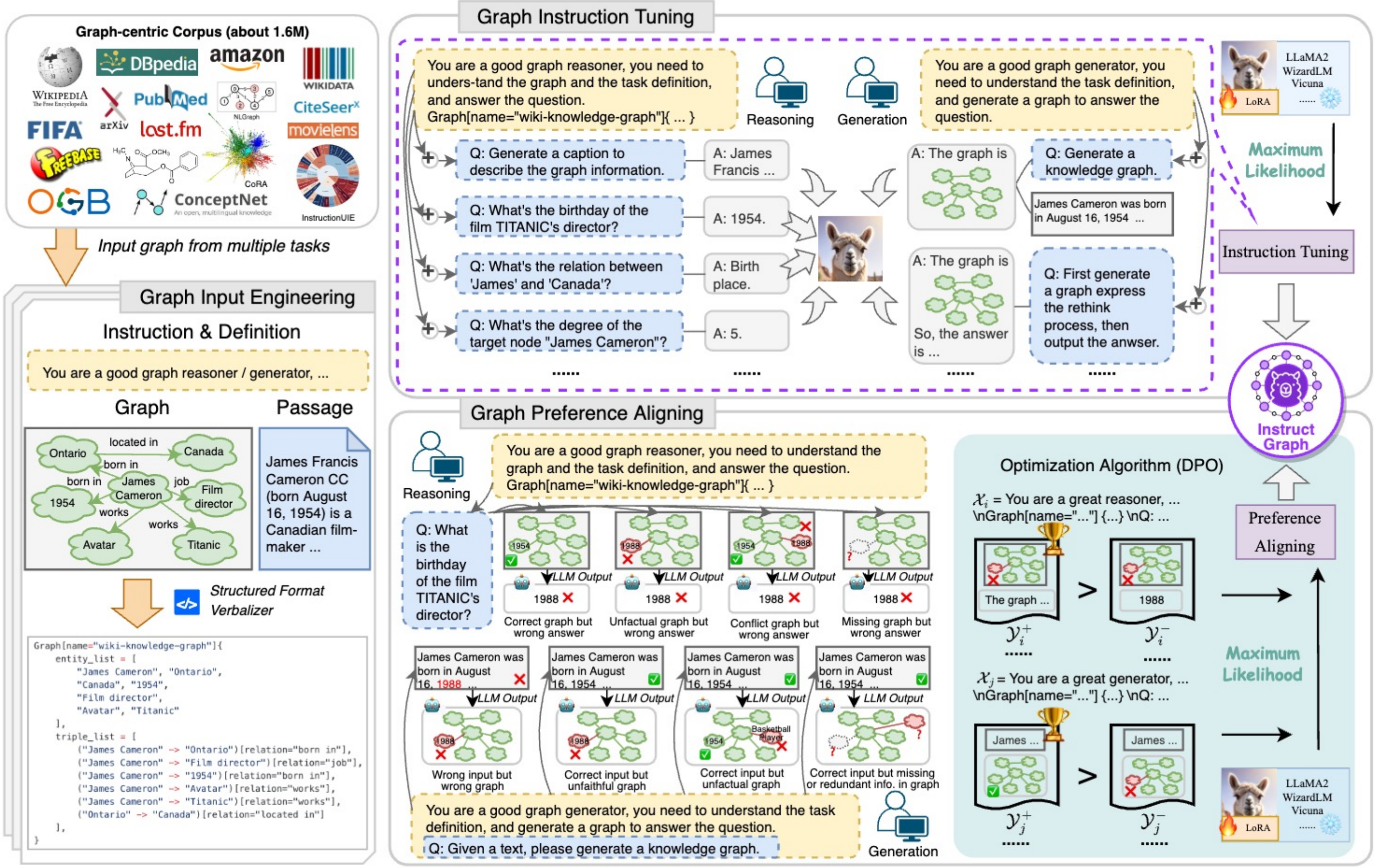
- **InstructGraph:** Boosting Large Language Models via Graph-centric Instruction Tuning and Preference Alignment



LLM-Only: InstructGraph

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **InstructGraph:** Boosting Large Language Models via Graph-centric Instruction Tuning and Preference Alignment



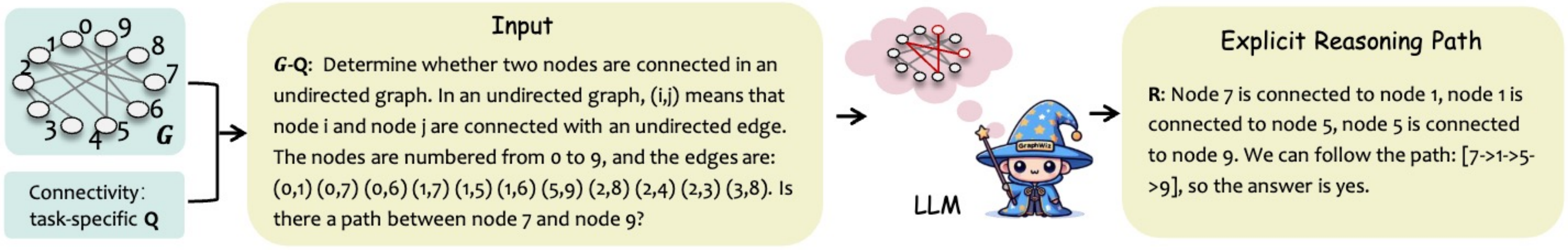
LLM-Only: InstructGraph

- **InstructGraph:** Boosting Large Language Models via Graph-centric Instruction Tuning and Preference Alignment

Task Groups	Task Clusters	Task Definition	Task Input	Task Output
Graph Structure Modeling	Connection Detection, Cycle Detection, Hamilton Path, Bipartite Matching, Shortest Path, Degree Computing	The tasks in this group aim to make LLMs better understand some basic graph structures. The input only contains nodes, directed or un-directed edges, and optional weights.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i]$	$\mathcal{Y}_i = \mathcal{A}_i$
Graph Language Modeling	Graph Caption Generation	The task aims to generate a caption passage $\mathcal{P}_i$ to describe the graph $\mathcal{G}_i$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i]$	$\mathcal{Y}_i = \mathcal{P}_i$
	Graph Question Answering	The task aims to reason on the whole graph $\mathcal{G}_i$ and find an entity as the final answer $\mathcal{A}_i \in \mathcal{E}_i$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{A}_i$
	Graph Node Classification	The task aims to classify the target node into pre-defined classes based on $\mathcal{G}_i$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{A}_i$
	Graph Link Prediction	The task aims to predict the relation between two given nodes based on $\mathcal{G}_i$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{A}_i$
	Graph Relevance Inspection	The task aims to detect whether the graph $\mathcal{G}_i$ is relevant to the passage $\mathcal{P}_i$, we have $\mathcal{A}_i \in \{\text{relevant}, \text{irrelevant}\}$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{A}_i$
	Graph Collaboration Filtering	The task aims to predict whether the target user prefers the target item based on the whole graph $\mathcal{G}_i$, the answer $\mathcal{A}_i$ can be set as a score.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{C}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{A}_i$
Graph Generation Modeling	Knowledge Graph Generation	The task aims to given a passage $\mathcal{P}_i$ that describes a piece of factual or commonsense information, the task aims to extract entities and relations from $\mathcal{P}_i$ to generate a graph $\mathcal{G}_i$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{C}_i$
	Structure Graph Generation	The task aims to generate a graph to meet the structure information described in the passage $\mathcal{P}_i$.	$\mathcal{X}_i = [\mathcal{I}_i, \mathcal{P}_i]$	$\mathcal{Y}_i = \mathcal{C}_i$
Graph Thought Modeling	Arithmetic Symbolic Robotic Logic	The task aims to solve the general reasoning task in three think steps: 1) first find the question subject, 2) then generate a thought graph $\mathcal{G}_i$ to express the rationale and 3) finally output the result $\mathcal{A}_i$ based on the graph.	$\mathcal{X}_i = \mathcal{I}_i$	$\mathcal{Y}_i = [\mathcal{C}_i; \mathcal{A}_i]$

LLM-Only: GraphWiz

- **GraphWiz:** An Instruction-Following Language Model for Graph Problems
  - ➢ Graph Reasoning Problems



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.

LLM-Only: GraphWiz

香 港 大 學
THE UNIVERSITY OF HONG KONG

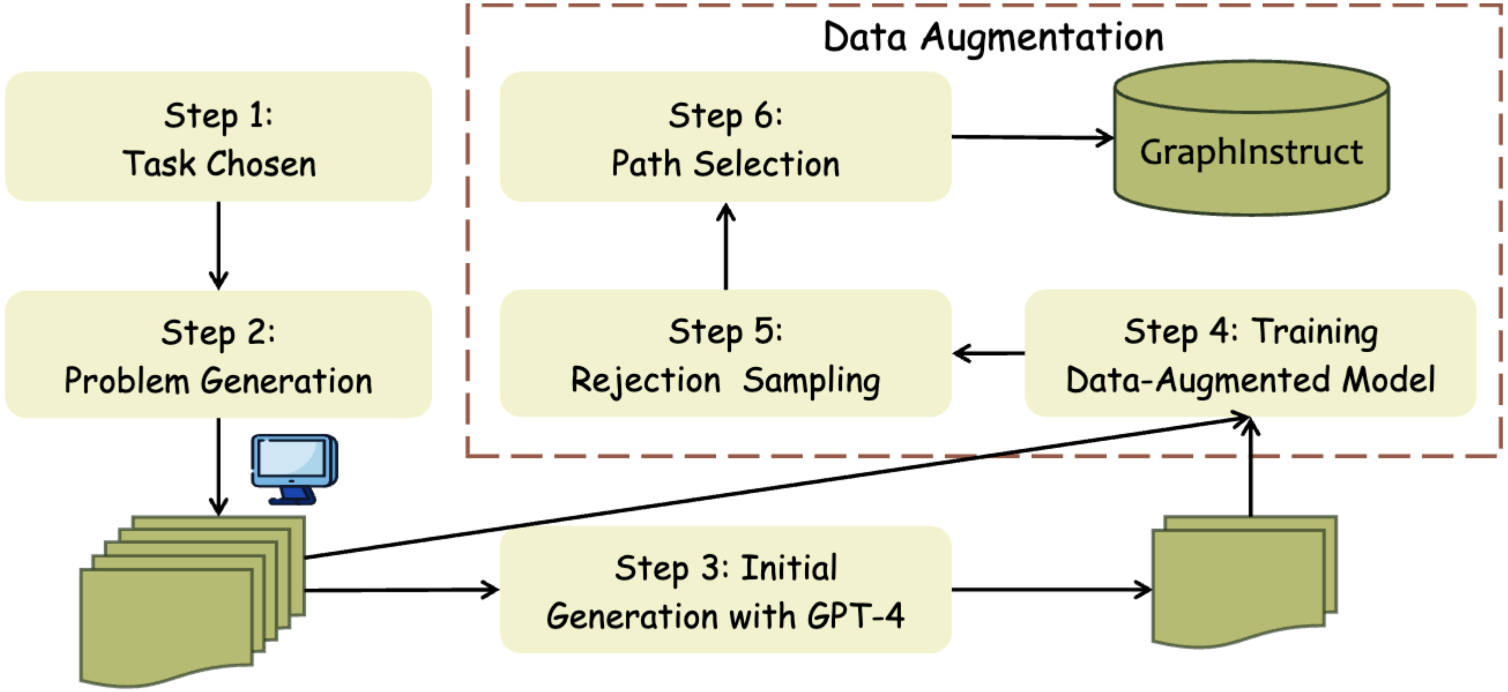
➢ Overview of nine graph tasks in our GraphInstruct benchmark

Problem	Definition	Time Complexity	Weighted?	Directed?	Node Range	Difficulty
Cycle Detection	Detect if a given graph $\mathcal{G}$ contains any cycles.	$O(\|E\|)$	✗	✗	[2, 100]	Easy
Connectivity	Assess if two nodes u and v in a given graph $\mathcal{G}$ are connected via a path.	$O(\|V\| + \|E\|)$	✗	✗	[2, 100]	Easy
Bipartite Graph Check	Judge if a given graph $\mathcal{G}$ is bipartite.	$O(\|V\| + \|E\|)$	✗	✓	[2, 100]	Easy
Topological Sort	Find a topological ordering of vertices in a directed acyclic graph $\mathcal{G}$.	$O(\|V\| + \|E\|)$	✗	✓	[2, 50]	Easy
Shortest Path	Compute the shortest path between two specific nodes u and v in a given graph $\mathcal{G}$.	$O(\|E\| + \|V\| \log \|V\|)$	✓	✗	[2, 100]	Medium
Maximum Triangle Sum	Find the maximum sum of weights for any connected triplet of vertices in a given graph $\mathcal{G}$.	$O(\|V\|^3)$	✓	✗	[2, 25]	Medium
Maximum Flow	Calculate the maximum flow from a source node s to a sink node t in a directed graph $\mathcal{G}$.	$O(\|V\|^2\sqrt{\|E\|})$	✓	✓	[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

LLM-Only: GraphWiz

香 港 大 學
THE UNIVERSITY OF HONG KONG

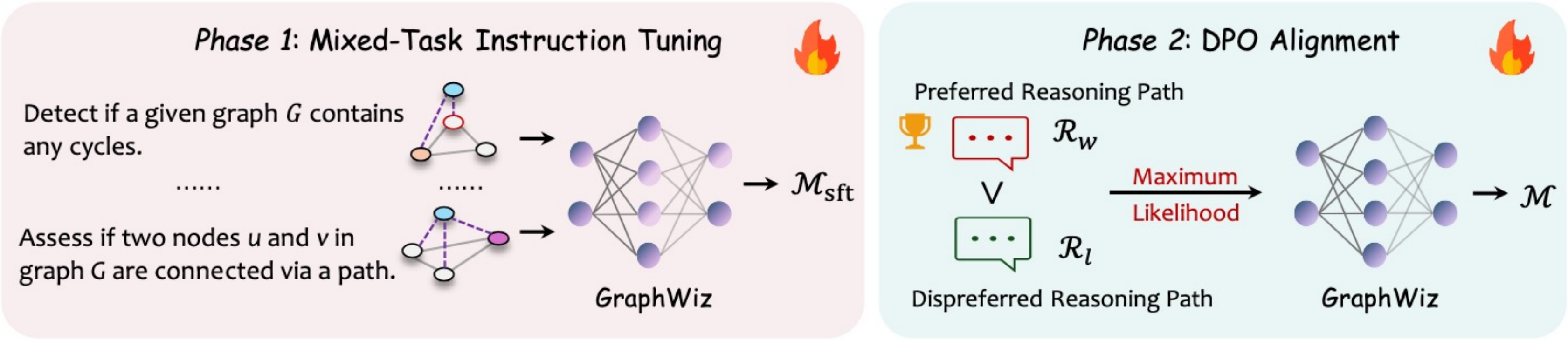
- Graph Problem Generation
  - ✓ Diverse Distributions
  - ✓ Length Constraints
  - ✓ Unique Instances
  - ✓ Scalable Graph Sizes
- Explicit Reasoning Paths Generation
  - ✓ Data Augmentation with Rejection Sampling



LLM-Only: GraphWiz

香 港 大 學
THE UNIVERSITY OF HONG KONG

- GraphWiz
  - ✓ Phase 1: Mixed-Task Instruction Tuning
  - ✓ Phase 2: DPO Alignment of Reasoning Abilities

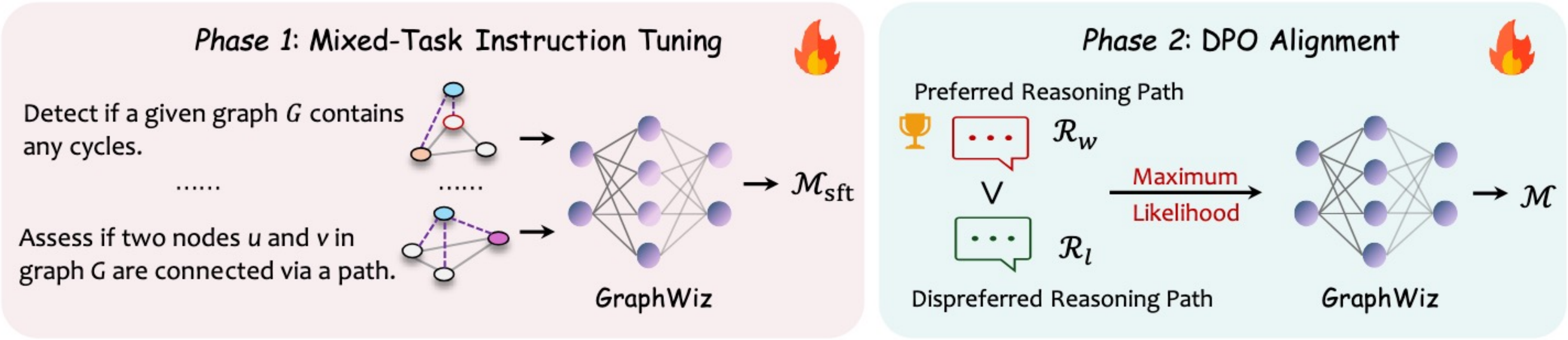


$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{N}\sum_{j=1}^{M} \log \mathrm{P}(\mathcal{R}_{i,j} | \mathcal{G}_i, Q_i; \theta)$$

$$\mathcal{L}_{DPO}(\mathcal{M}(\theta); \mathcal{M}(\theta)_{sft}) = -\mathbb{E}_{(x,\mathcal{R}_w,\mathcal{R}_l)\sim D} \left[\log \sigma \left(\beta \log \frac{\mathcal{M}(\theta)(\mathcal{R}_w|x)}{\mathcal{M}(\theta)(\mathcal{R}_l|x)} - \beta \log \frac{\mathcal{M}(\theta)_{sft}(\mathcal{R}_w|x)}{\mathcal{M}(\theta)_{sft}(\mathcal{R}_l|x)} \right) \right]$$

LLM-Only: GraphWiz

- GraphWiz
  - ✓ Phase 1: Mixed-Task Instruction Tuning
  - ✓ Phase 2: DPO Alignment of Reasoning Abilities

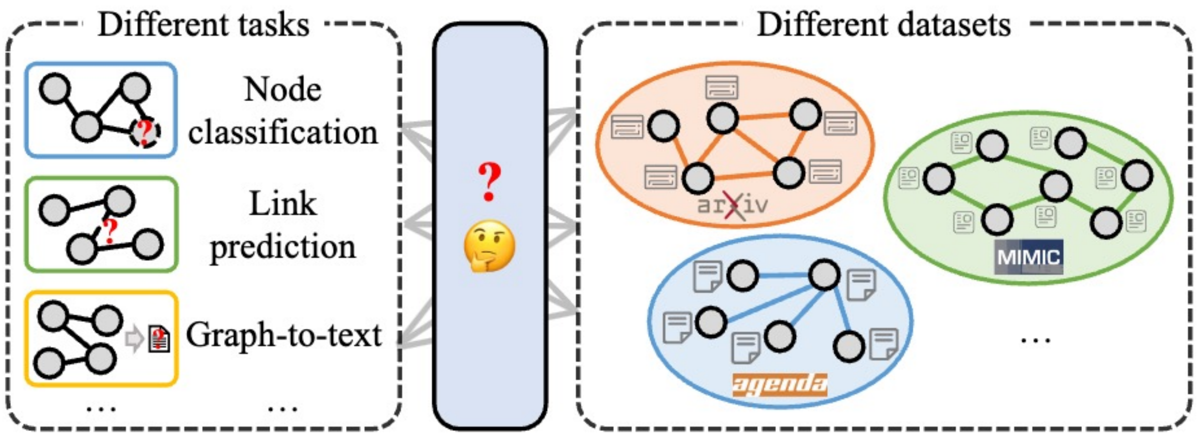


$$\mathcal{L}_{\text{LM}} = -\sum_{i=1}^{N}\sum_{j=1}^{M} \log \mathrm{P}(\mathcal{R}_{i,j} | \mathcal{G}_i, Q_i; \theta)$$

$$\mathcal{L}_{DPO}(\mathcal{M}(\theta); \mathcal{M}(\theta)_{sft}) = -\mathbb{E}_{(x,\mathcal{R}_w,\mathcal{R}_l)\sim D} \left[\log \sigma \left(\beta \log \frac{\mathcal{M}(\theta)(\mathcal{R}_w|x)}{\mathcal{M}(\theta)(\mathcal{R}_l|x)} - \beta \log \frac{\mathcal{M}(\theta)_{sft}(\mathcal{R}_w|x)}{\mathcal{M}(\theta)_{sft}(\mathcal{R}_l|x)}\right)\right]$$

LLM-Only: MuseGraph

- **MuseGraph:** Graph-oriented Instruction Tuning of Large Language Models for Generic Graph Mining

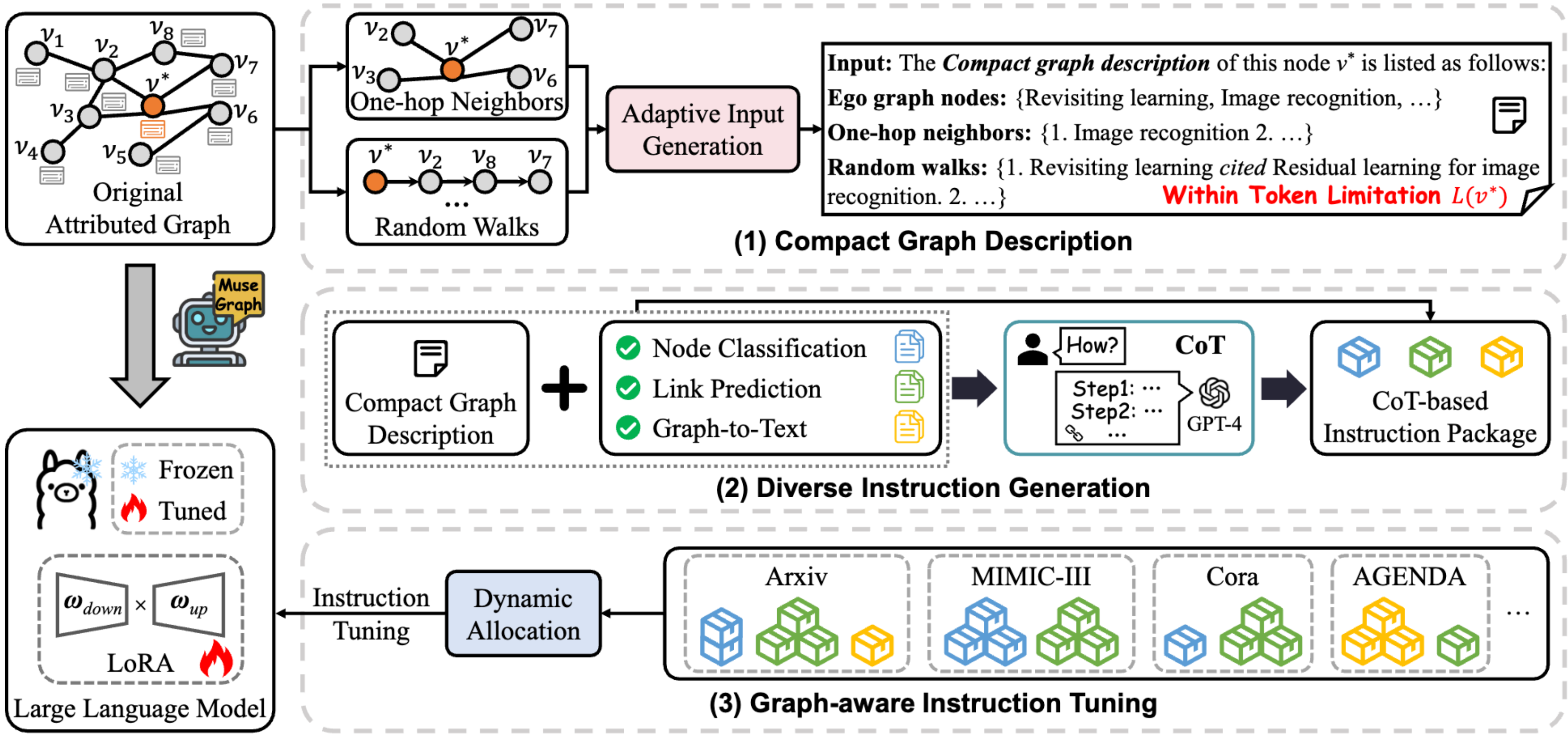


	GNN*	LLM*	Cross-task	Cross-dataset
ExpAsFeat [18]	✓			
PRODIGY [22]	✓			
LLMForGraph [5]	✓			
All-in-One [50]	✓		✓	
OFA [34]	✓		✓	✓
NLGraph [57]		✓	✓	~~✓~~
GPT4GRAPH [15]		✓	✓	~~✓~~
GraphGPT [15]	✓	~~✓~~		✓
InstructGLM [71]		✓	✓	~~✓~~
MuseGraph		✓	✓	✓

LLM-Only: MuseGraph

香 港 大 學
THE UNIVERSITY OF HONG KONG

- **MuseGraph:** Graph-oriented Instruction Tuning of Large Language Models for Generic Graph Mining



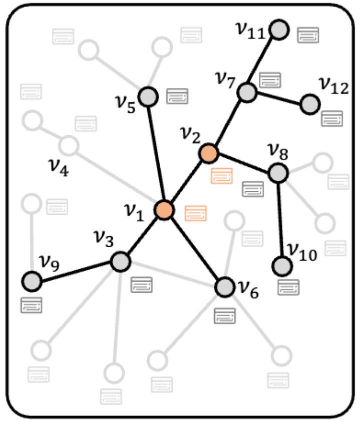
LLM-Only: MuseGraph

Algorithm 1: Adaptive Input Generation

Input: Attributed graph $\mathcal{G}$ with N nodes, token count set $\mathcal{T}$, node energy set $\mathcal{H}$, target node v^*, token limitation $L(v^*)$

Output: Key neighbor set $\mathcal{N}(v^*)$, key walk set $\mathcal{W}(v^*)$

1: Initialize $\mathcal{N}(v^*)$, $\mathcal{W}(v^*)$ as empty
2: Select $v_i \in G(v^*)$ with $H(v_i) \geq H(v^*)$ and $L(v^*) \geq T(v_i)$ for $\mathcal{N}(v^*)$, where $G(v^*)$ is v^*'s one-hop neighbors
3: Expand $\mathcal{W}(v^*)$ starting from v^* based on $\mathcal{G}$ within $L(v^*)$ and $H(v^*)$ constraints



Name	# T	# D	H
v_1: Deepwalk: Online learning of social representations	8	5	16
v_2: Inductive Representation Learning on Large Graphs	11	3	22
v_3: Gcc: Graph contrastive coding for graph neural network pre-training	14	5	28
v_4: Ensemble deep learning	4	2	8
…		…	

Target	*Key Neighbors*: $H(v_i) \geq H(v^*)$	*Key Walks*: $H(v_j) \geq H(v^*)$
$v^* = v_1$	$\{v_2, v_3, v_5, v_6\}$	① $v_1 \rightarrow v_2 \rightarrow v_7 \rightarrow v_{12} \rightarrow \cdots$ ② $v_1 \rightarrow v_3$
$v^* = v_2$	$\{v_7, v_8\}$	① $v_2 \rightarrow v_7 \rightarrow v_{11}$ ② $v_2 \rightarrow v_7 \rightarrow v_{12}$ ③ $v_2 \rightarrow v_8 \rightarrow v_{10} \rightarrow \cdots$ ④ $v_2 \rightarrow v_8 \rightarrow \cdots$

Given the classification of target **PAPER {*title*} with <*category*>** in the *Arxiv* dataset, give your explanation based on the provided ***Compact graph description***.

Focus your analysis on elucidating the reasons behind this classification in a clear **Chain of Thought**. Keep the analysis brief and to the point.

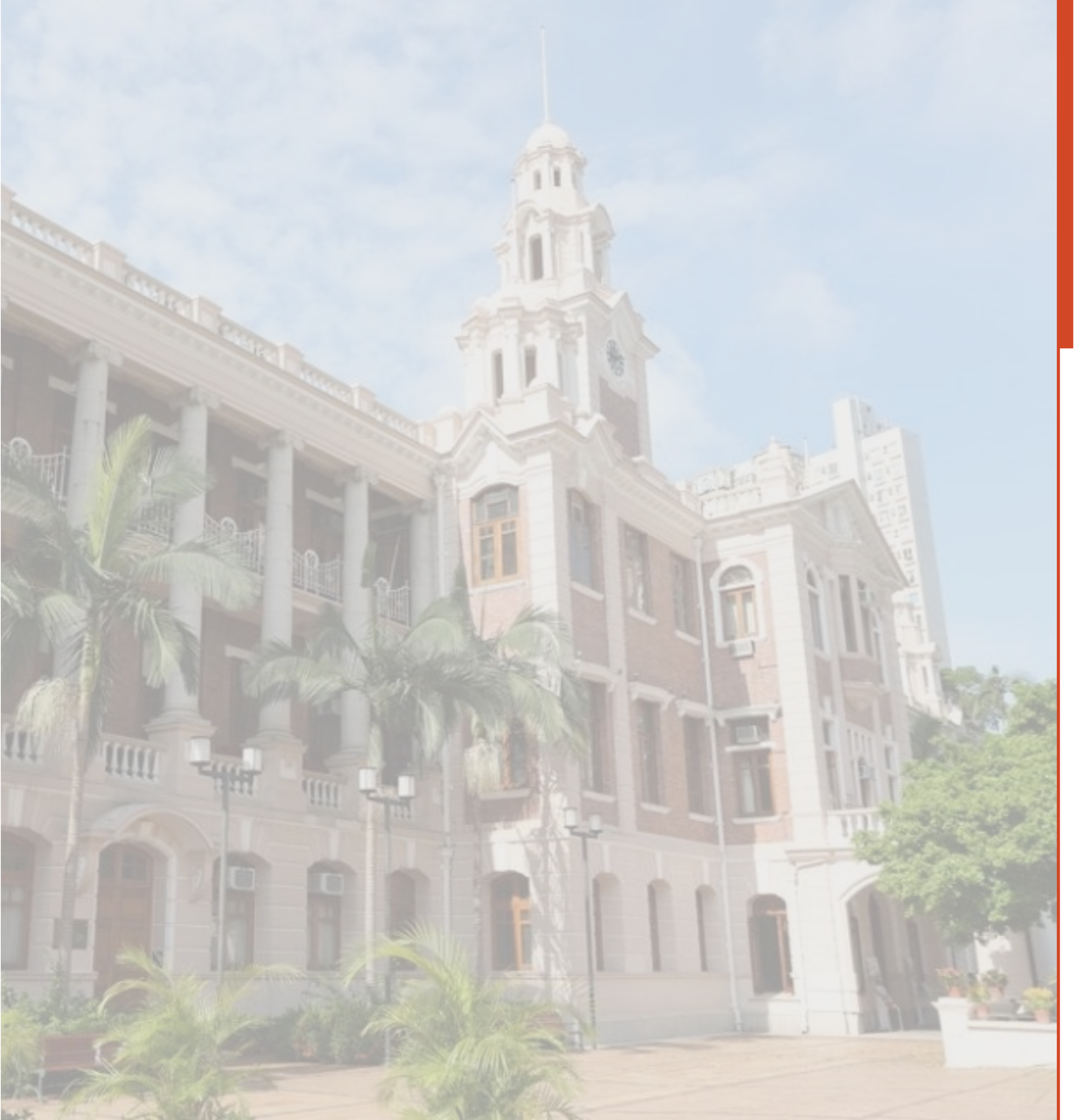
The ***Compact graph description*** of this PAPER is listed as follows: {***Ego graph node:*** …; ***One-hop neighbors:***…; ***Random walks:***…}

The classification of PAPER as <*category*> is accurate considering its …

*1. **The title of the PAPER**...which is a common topic in...*
*2. **The abstract** makes numerous references to commonly studied topics in ...*
*3. **The ego graph nodes**... also predominantly fall under the thematic domain of ...*
*4. **The one-hop neighbors**... further solidify the notion that...*
*5. **The data from random walks**, which represents a wider view of the PAPER's citation network...*

In conclusion, considering the PAPER's ***Compact graph description***, its classification is valid under **<*category*>**, **because of ...**

Q & A



Thanks!

Data Intelligence Lab @ HKU