

Towards Graph Foundation Models

WWW 2024 Tutorial

Philip S. Yu, Chuan Shi, Cheng Yang, Yuan Fang, Lichao Sun



SINGAPORE
MANAGEMENT
UNIVERSITY



Towards Graph Foundation Models

Part III: LLM & GNN+LLM Models

Presented by **Yuan Fang**, Singapore Management University

yfang@smu.edu.sg | www.yfang.site

Prepared by **Yuxia Wu**, Singapore Management University

Outline

□ **LLM based Models**

- Backbone Architectures
- Pre-training
- Adaptation

□ GNN+LLM based Models

- Backbone Architectures
- Pre-training
- Adaptation

□ Summary and outlook

LLM-based Models

□ Backbone Architectures

□ Pre-training

□ Adaptation

Model	Backbone Architecture	Pre-training	Adaptation
InstructGLM[157]	Graph-to-token + Flan-T5/LLaMA	MLM,LM	Manual Prompt Tuning
LLMtoGraph[71]	Graph-to-text + GPTs, Vicuna	LM	Manual Prompt Tuning
NLGraph[126]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
GraphText[175]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
LLM4Mol[91]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
GPT4Graph[29]	Graph-to-text + GPT-3	LM	Manual Prompt Tuning + Automatic Prompt Tuning
Graph-LLM[9]	Graph-to-text + BERT, DeBERTa, Sentence-BERT, GPTs, LLaMA	MLM,LM	Manual Prompt Tuning + Automatic Prompt Tuning

Table 3. Details of approaches involved as LLM based models

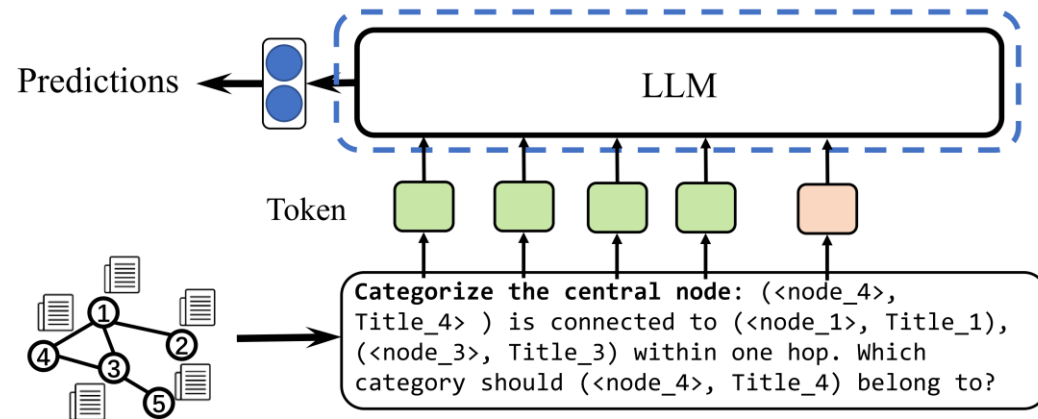
Backbone Architectures

□ Graph-to-Token

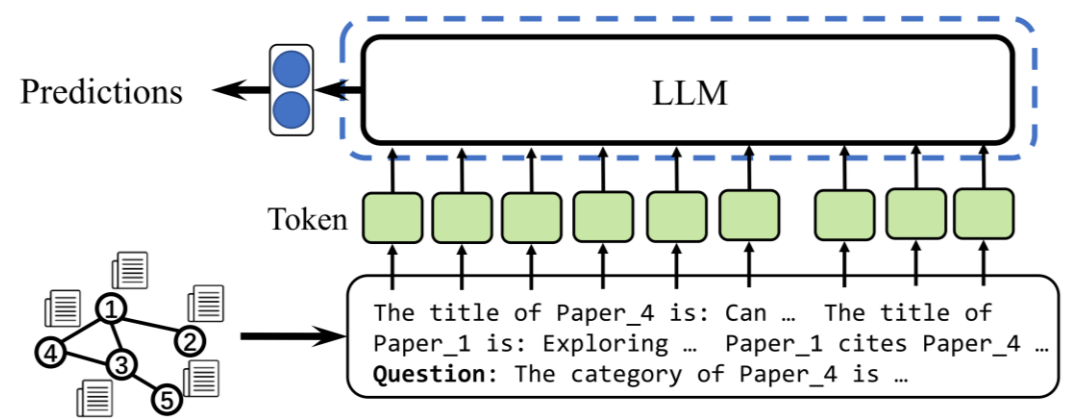
- Tokenize graph information to align it with LLM

□ Graph-to-text

- Describe graph information using natural language



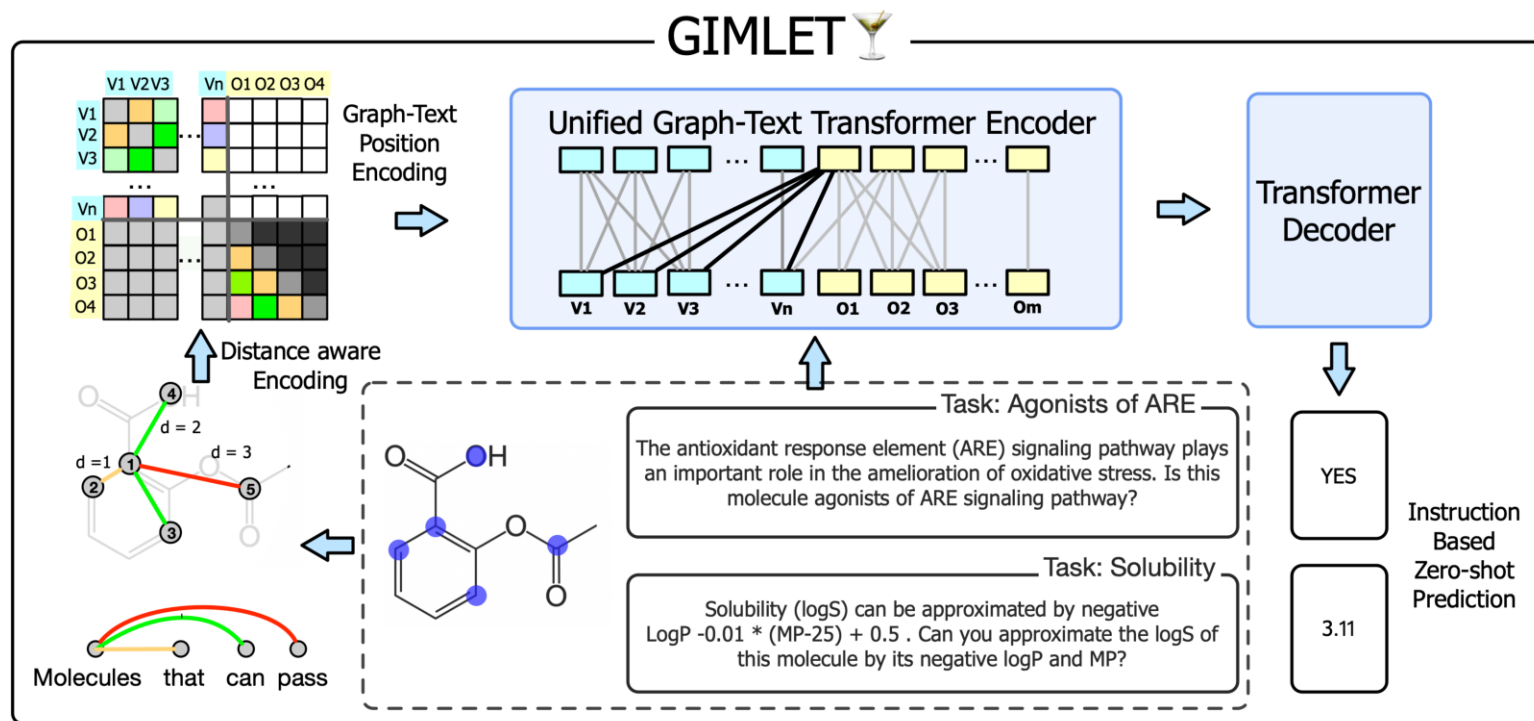
(a) Graph-to-token.



(b) Graph-to-text.

Graph-to-Token: GIMLET

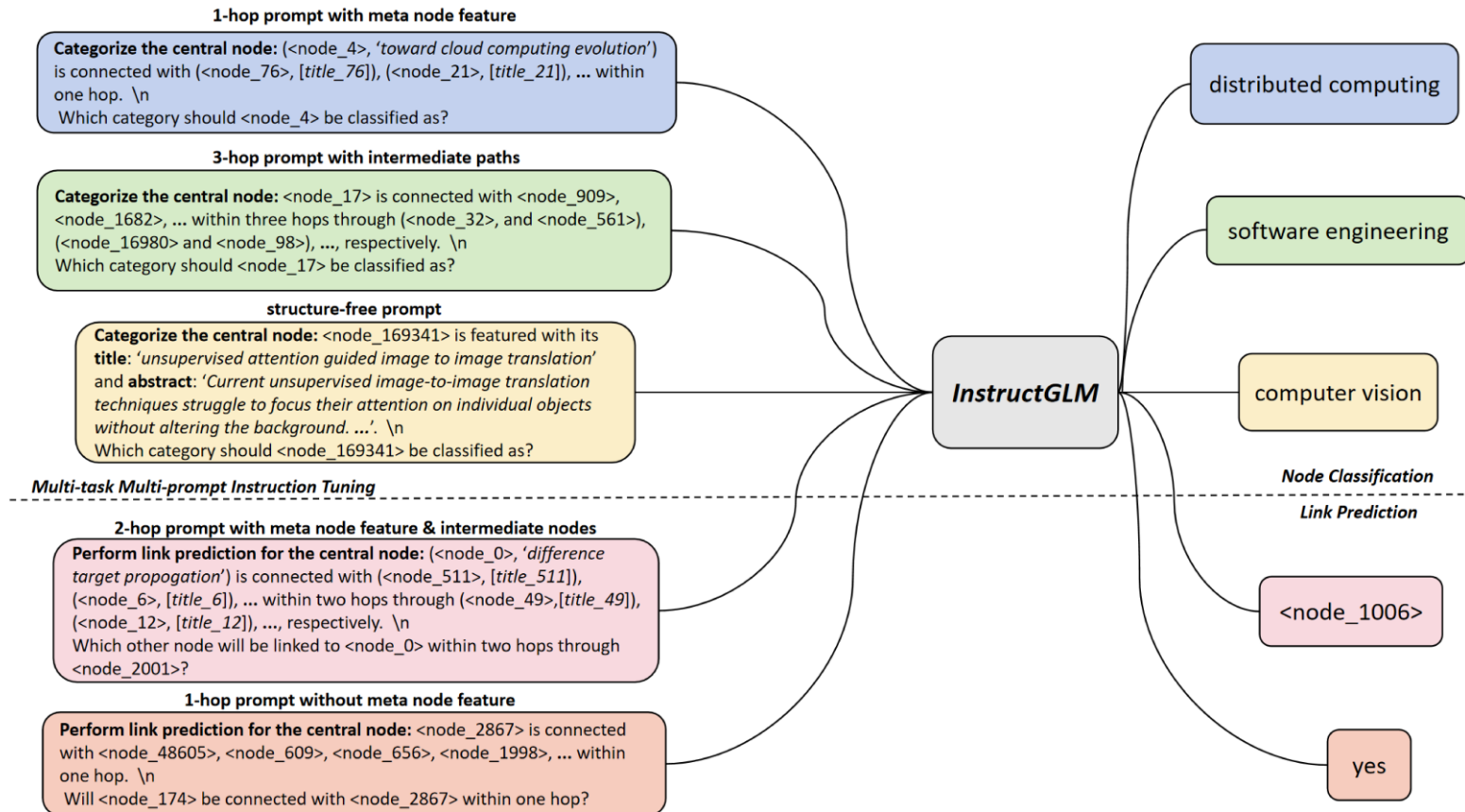
- ❑ Integrating graph data with textual data
- ❑ Encoding the graph's structural information



Zhao, et al. "GIMLET: A unified graph-text model for instruction-based molecule zero-shot learning." *NeurIPS'23*.

Graph-to-Token: InstructGLM

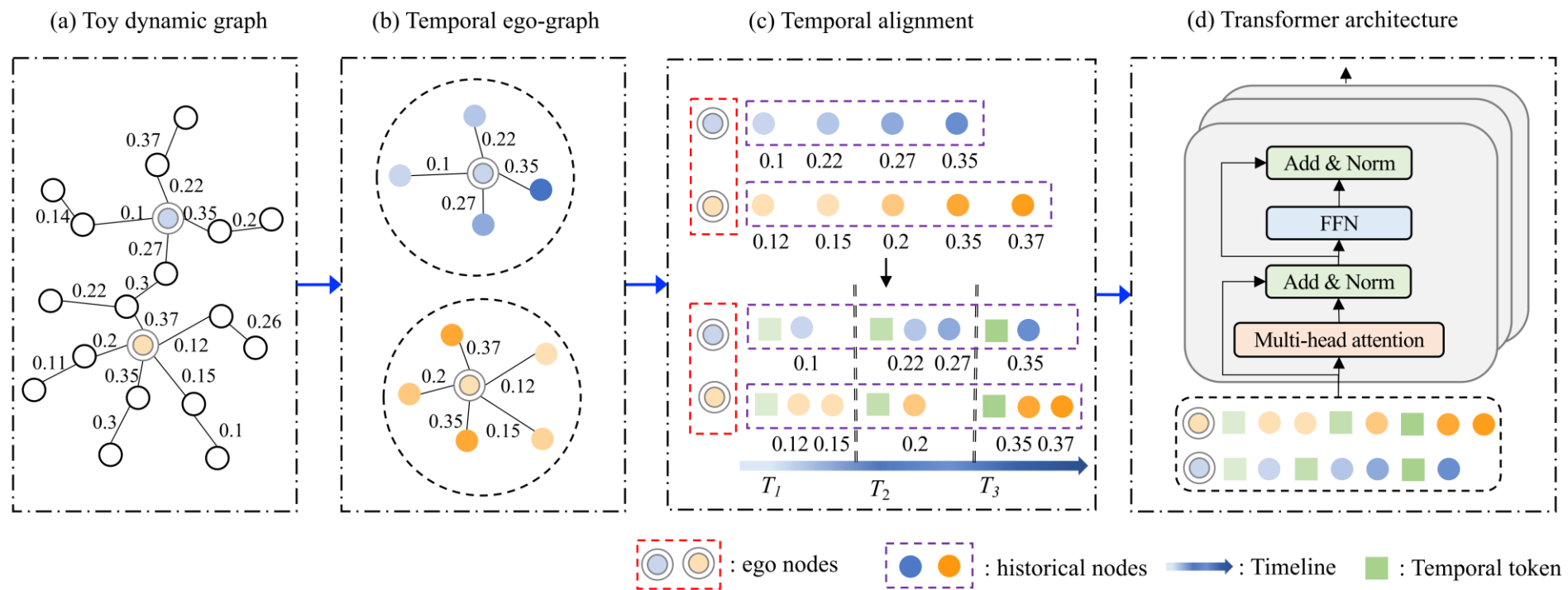
□ Expand the vocabulary of the LLM by graph node features



Ye, et al. "Language is all a graph needs." *EACL 2024*.

Graph-to-Token: SimpleDyG

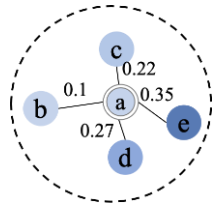
- ❑ Transformer-based approach for dynamic graphs
- ❑ Map a dynamic graph into a set of sequences



Wu, et al. "On the Feasibility of Simple Transformer for Dynamic Graph Modeling." WWW'24.

Graph-to-Token: SimpleDyG

Temporal ego-graph



$$w_i = \langle b, c, d, e \rangle$$

Temporal alignment:

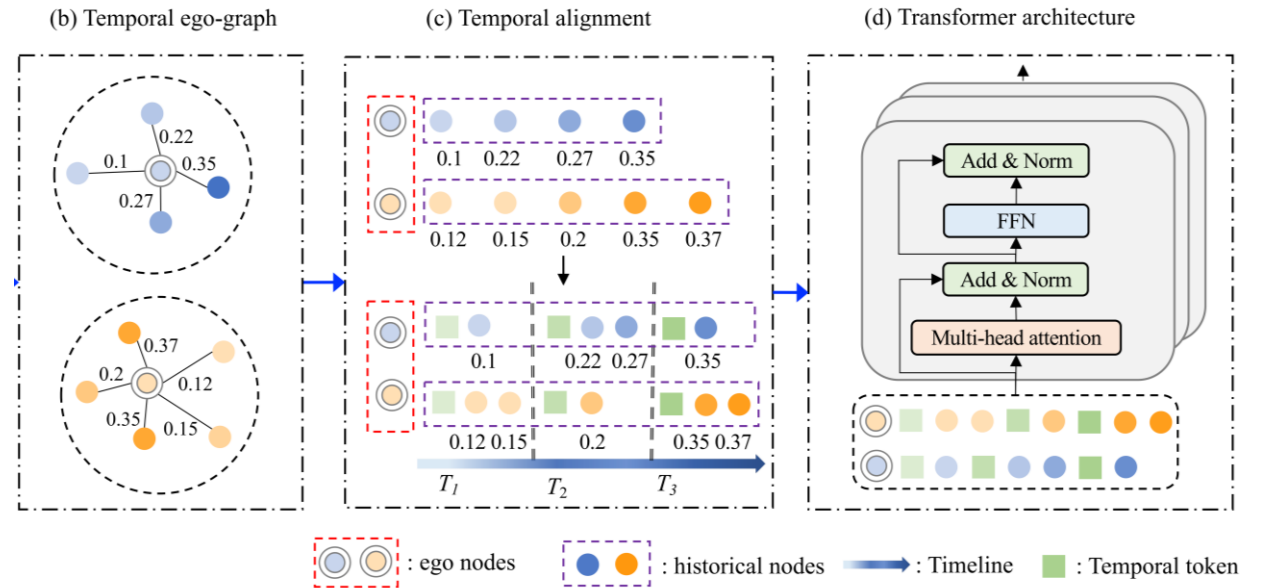
- Segment the time domain:

$$S_i^1 = \langle b \rangle \quad S_i^2 = \langle c, d \rangle \quad S_i^3 = \langle e \rangle$$

- Sequence for Transformer:

$$x'_i = \langle |hist| \rangle, a, \langle |time1| \rangle, b, \langle |time2| \rangle, c, d, \langle |time3| \rangle, e, \langle |endofhist| \rangle$$

$$y'_i = \langle |pred| \rangle \langle |time4| \rangle S_i^4 \langle |endofpred| \rangle$$



Graph-to-text

Describe graph information for various graphs and tasks

Node/edge list, graph properties

1. Connectivity

Determine if there is a path between two nodes in the graph. Note that (i,j) means that node i and node j are connected with an undirected edge. Graph: $(0,1)$ $(1,2)$ $(3,4)$ $(4,5)$
Q: Is there a path between node 1 and node 4?

2. Cycle

In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge. The nodes are numbered from 0 to 5, and the edges are: $(3,4)$ $(3,5)$ $(1,0)$ $(2,5)$ $(2,0)$
Q: Is there a cycle in this graph?

3. Topological Sort

In a directed graph with 5 nodes numbered from 0 to 4: node 0 should be visited before node 4, ...
Q: Can all the nodes be visited? Give the solution.

4. Shortest Path

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

5. Maximum Flow

In a directed graph, the nodes are numbered from 0 to 3, and the edges are: an edge from node 1 to node 0 with capacity 10, an edge from node 0 to node 2 with capacity 6, an edge from node 2 to node 3 with capacity 4.
Q: What is the maximum flow from node 1 to node 3?

6. Bipartite Graph Matching

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 job.
 Applicant 0 is interested in job 4, ...
Q: Find an assignment of jobs to applicants in such that the maximum number of applicants find the job they are interested in.

7. Hamilton Path

In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge. The nodes are numbered from 0 to 4, and the edges are: $(4,2)$ $(0,4)$ $(4,3)$ $(0,1)$ $(0,2)$ $(4,1)$ $(2,3)$
Q: Is there a path in this graph that visits every node exactly once? If yes, give the path. Note that in a path, adjacent nodes must be connected with edges.

8. GNN

In an undirected graph, the nodes are numbered from 0 to 4, and every node has an embedding. (i,j) means that node i and node j are connected with an undirected edge. Embeddings: node 0: $[1,1]$, ...
 The edges are: $(0,1)$...
 In a simple graph convolution layer, each node's embedding is updated by the sum of its neighbors' embeddings.
Q: What's the embedding of each node after one layer of simple graph convolution layer?

Graph description language

Graph Structured Data

Graph description language:

```
<?xml version='1.0' encoding='utf-8'?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns">
  <key id="relation" for="edge" attr.name="relation" attr.type="string" />
  <key id="title" for="node" attr.name="title" attr.type="string" />
  <graph edgedefault="undirected">
    <node id="P357">
      <data key="title">statistical anomaly detection via composite hypothesi models</data>
    </node>
    <node id="P79639">
      <data key="title">universal and composite hypothesis testing</data>
    </node>
    <edge source="P357" target="P79639">
      <data key="relation">reference</data>
    </edge>
  </graph>
</graphml>
```

Graph-Syntax Tree

Text Attributes:
 feature x
 label y

G-Syntax Tree

label:
 1st-hop: [A]
 2nd-hop: [B]

feature:
 center-node: [0]
 1st-hop: [1, 2]
 2nd-hop: [3, 2]

Wang, et al. "Can language models solve graph problems in natural language?." *NeurIPS'23*.

Guo, et al. "GPT4Graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking." *CoRR'23*.

Zhao, et al. "GraphText: Graph reasoning in text space." *CoRR'23*.

LLM-based Models

□ Backbone Architectures

□ Pre-training

□ Adaptation

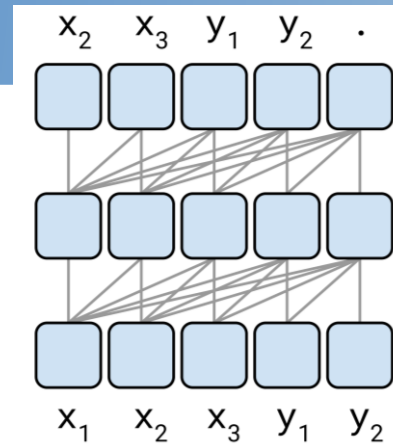
Model	Backbone Architecture	Pre-training	Adaptation
InstructGLM[157]	Graph-to-token + Flan-T5/LLaMA	MLM,LM	Manual Prompt Tuning
LLMtoGraph[71]	Graph-to-text + GPTs, Vicuna	LM	Manual Prompt Tuning
NLGraph[126]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
GraphText[175]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
LLM4Mol[91]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
GPT4Graph[29]	Graph-to-text + GPT-3	LM	Manual Prompt Tuning + Automatic Prompt Tuning
Graph-LLM[9]	Graph-to-text + BERT, DeBERTa, Sentence-BERT, GPTs, LLaMA	MLM,LM	Manual Prompt Tuning + Automatic Prompt Tuning

Table 3. Details of approaches involved as LLM based models

Pre-training

Language Modeling (LM)

➤ LLaMA, GPT-3...

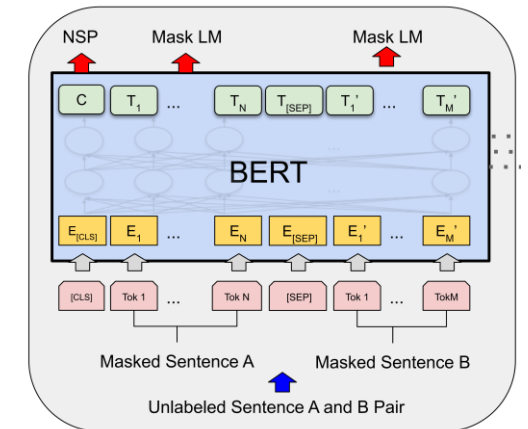


Masked Language Modeling (MLM)

➤ BERT, T5...

➤ Replace the word with the [MASK] token

e.g., my dog is hairy → my dog is [MASK]



Touvron, et al. "Llama: Open and efficient foundation language models." *CoRR*'23.

Ouyang, et al. "Training language models to follow instructions with human feedback." *NeurIPS*'22.

Devlin, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." *CoRR*'18.

Raffel, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *JMLR*'20.

LLM-based Models

□ Backbone Architectures

□ Pre-training

□ **Adaptation**

Model	Backbone Architecture	Pre-training	Adaptation
InstructGLM[157]	Graph-to-token + Flan-T5/LLaMA	MLM,LM	Manual Prompt Tuning
LLMtoGraph[71]	Graph-to-text + GPTs, Vicuna	LM	Manual Prompt Tuning
NLGraph[126]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
GraphText[175]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
LLM4Mol[91]	Graph-to-text + GPTs	LM	Manual Prompt Tuning
GPT4Graph[29]	Graph-to-text + GPT-3	LM	Manual Prompt Tuning + Automatic Prompt Tuning
Graph-LLM[9]	Graph-to-text + BERT, DeBERTa, Sentence-BERT, GPTs, LLaMA	MLM,LM	Manual Prompt Tuning + Automatic Prompt Tuning

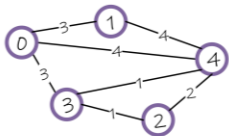
Table 3. Details of approaches involved as LLM based models

Adaptation

❑ Manual Prompting: Graph information, task descriptions

❑ Automatic Prompting: LLMs--> generate the context

Standard Prompting



<in-context exemplar>
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are: an edge between node 0 and node 4 with weight 4, an edge between node 0 and node 3 with weight 3, an edge between node 0 and node 1 with weight 3, ...
Q: Give the shortest path from node 0 to node 2.

Build-a-Graph Prompting

<in-context exemplar>
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are: an edge between node 0 and node 4 with weight 4, ...

Let's construct a graph with the nodes and edges first.

Q: Give the shortest path from node 0 to node 2.
A: All the paths from node 0 to node 2 are: 0,3,2 with a total weight of $3 + 1 = 4$, 0,1,4,2 with a total weight of $3 + 4 + 2 = 9$, 0,4,3,2 with a total weight of $4 + 1 + 1 = 6$. The weight of path 0,3,2 is the smallest, so the shortest path from node 0 to node 2 is 0,3,2 with a total weight of 4.

Algorithmic Prompting

We can use a Depth-First Search (DFS) algorithm to find the shortest path between two given nodes in an undirected graph.

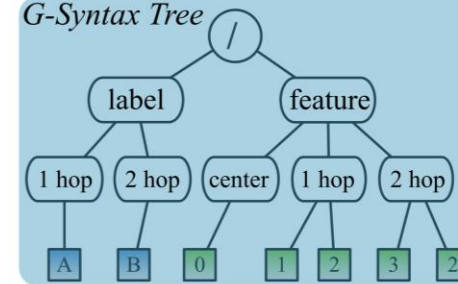
The basic idea is to start at one of the nodes and use DFS to explore all of its adjacent nodes. At each node you can keep track of the distance it takes to reach that node from the starting node.

Once you have explored all the adjacent nodes, you can backtrack and pick the node which has the shortest distance to reach the destination node.

<in-context exemplar>
In an undirected graph, the nodes are numbered from 0 to 4, and the edges are: an edge between node 0 and node 4 with weight 4, ...
Q: Give the shortest path from node 0 to node 2.
A: All the paths from node 0 to node 2 are: 0,3,2 with a total weight of $3 + 1 = 4$, 0,1,4,2 with a total weight of $3 + 4 + 2 = 9$, 0,4,3,2 with a total weight of $4 + 1 + 1 = 6$. The weight of path 0,3,2 is the smallest, so the shortest path from node 0 to node 2 is 0,3,2 with a total weight of 4.

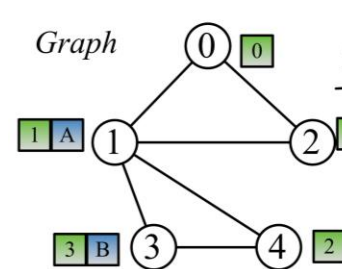
(c) GraphText

G-Syntax Tree



Traverse

Graph



Tree Construct

Text Attributes

- feature x
- label y



Task prompt and demos
Graph information:
label: G-Prompt
1st-hop: [A]
2nd-hop: [B]
feature:
center-node: [0]
1st-hop: [1, 2]
2nd-hop: [3, 2]
Question: What's the category of the node (choose from [A, B])?

According to the demos, 1st-hop labels are robust predictions. Therefore, the answer is A.

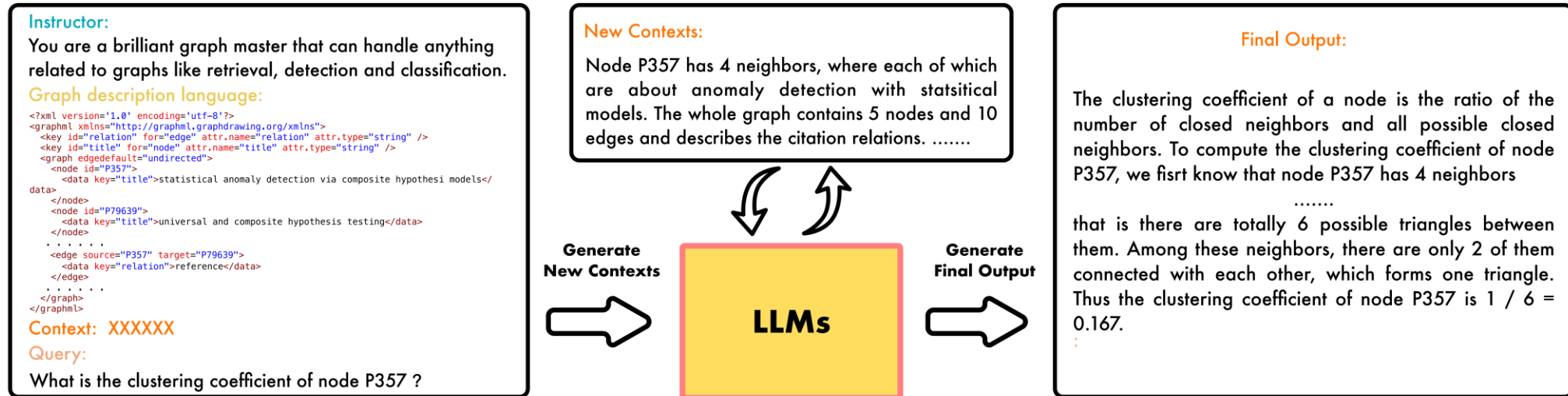


Wang, et al. "Can language models solve graph problems in natural language?." *NeurIPS'23*

Zhao, et al. "GraphText: Graph reasoning in text space." *CoRR'23*

Adaptation

- ❑ Manual Prompting: Graph information, task descriptions
- ❑ Automatic Prompting: LLMs → generate the context
 - Ask LLM generate graph/neighbor summarization



Guo, et al. "Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking." *CoRR*'23
Chen, et al. "Exploring the potential of large language models (llms) in learning on graphs." *ACM SIGKDD Explorations Newsletter* 2024

Outline

□ LLM based Models

- Backbone Architectures
- Pre-training
- Adaptation

□ **GNN+LLM based Models**

- Backbone Architectures
- Pre-training
- Adaptation

□ Summary and outlook

GNN+LLM based Models

□ Backbone Architectures

□ Pre-training

□ Adaptation

Model	Backbone Architecture	Pre-training	Adaptation
SimTeG [16]	GNN-centric	MLM, TTCL	Parameter-Efficient FT
TAPE [35]	GNN-centric	LM	Tuning-free Prompting + Parameter-Efficient FT
GIANT [11]	GNN-centric	MLM	Vanilla FT
GraD [79]	GNN-centric	MLM	Parameter-Efficient FT
GALM [147]	GNN-centric	Graph Reconstruction	Vanilla FT
GraphFormer [153]	Symmetric	MLM	Vanilla FT
GLEM [174]	Symmetric	MLM	Vanilla FT
ConGrat [4]	Symmetric	MLM + GTCL	Parameter-Efficient FT
G2P2 [136]	Symmetric	GTCL	Prompt Tuning
SAFER [6]	Symmetric	MLM	Parameter-Efficient FT
Text2Mol [18]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoMu [109]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoleculeSTM [73]	Symmetric	MLM + GTCL	Parameter-Efficient FT
CLAMP [103]	Symmetric	MLM + GTCL	Parameter-Efficient FT
Graph-Toolformer [165]	LLM-centric	LM	Tuning-free Prompting + Vanilla FT

Table 4. Details of approaches involved as GNN+LLM based models

Backbone Architectures

□ GNN-centric Methods

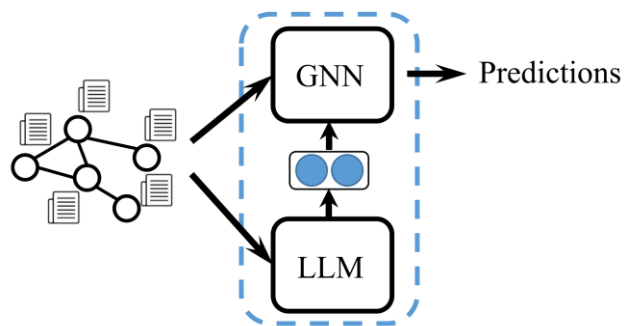
- LLMs extract node features from raw data; GNNs make predictions

□ Symmetric Methods

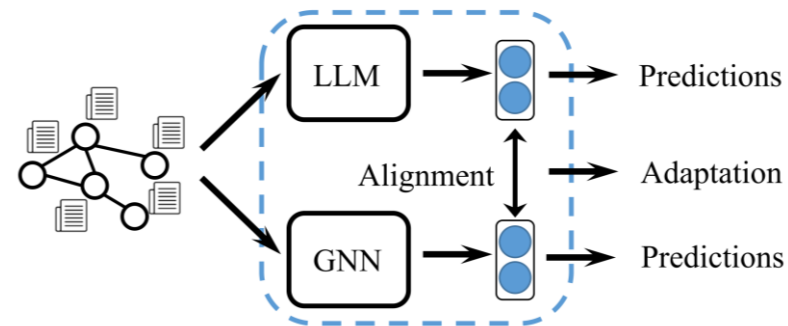
- Align the embeddings of GNN and LLM

□ LLM-centric Methods

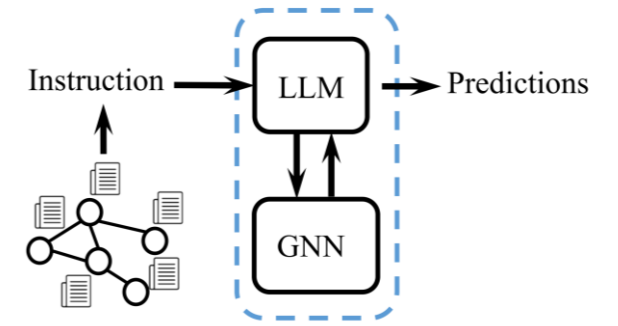
- Utilize GNNs to enhance the performance of LLM



(a) GNN-centric methods.



(b) Symmetric methods.

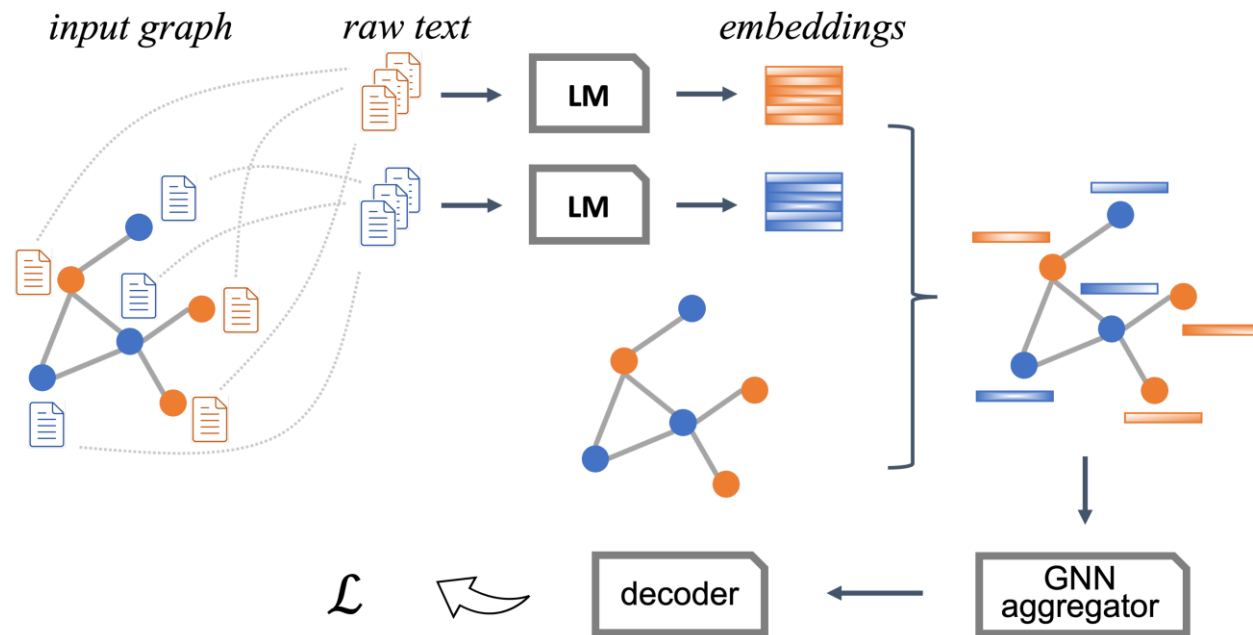


(c) LLM-centric methods.

GNN-centric Methods: GaLM

□ The backbone model:

Raw text \rightarrow LMs \rightarrow GNN aggregator \rightarrow decoder



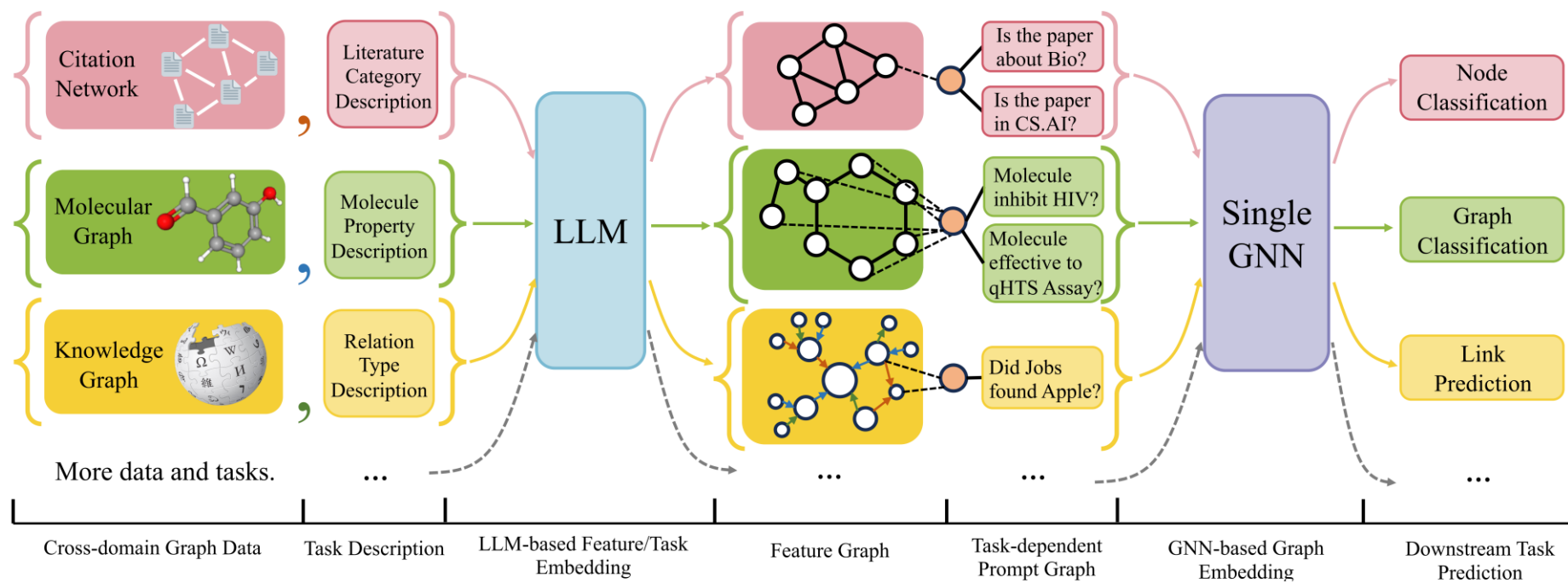
Xie, et al. "Graph-aware language model pre-training on a large graph corpus can help multiple graph applications." *KDD'23*.

GNN-centric Methods: One for all

□ The backbone model:

Text-attributed graph
Task description

LLMs → Prompted graph → GNN → Downstream tasks

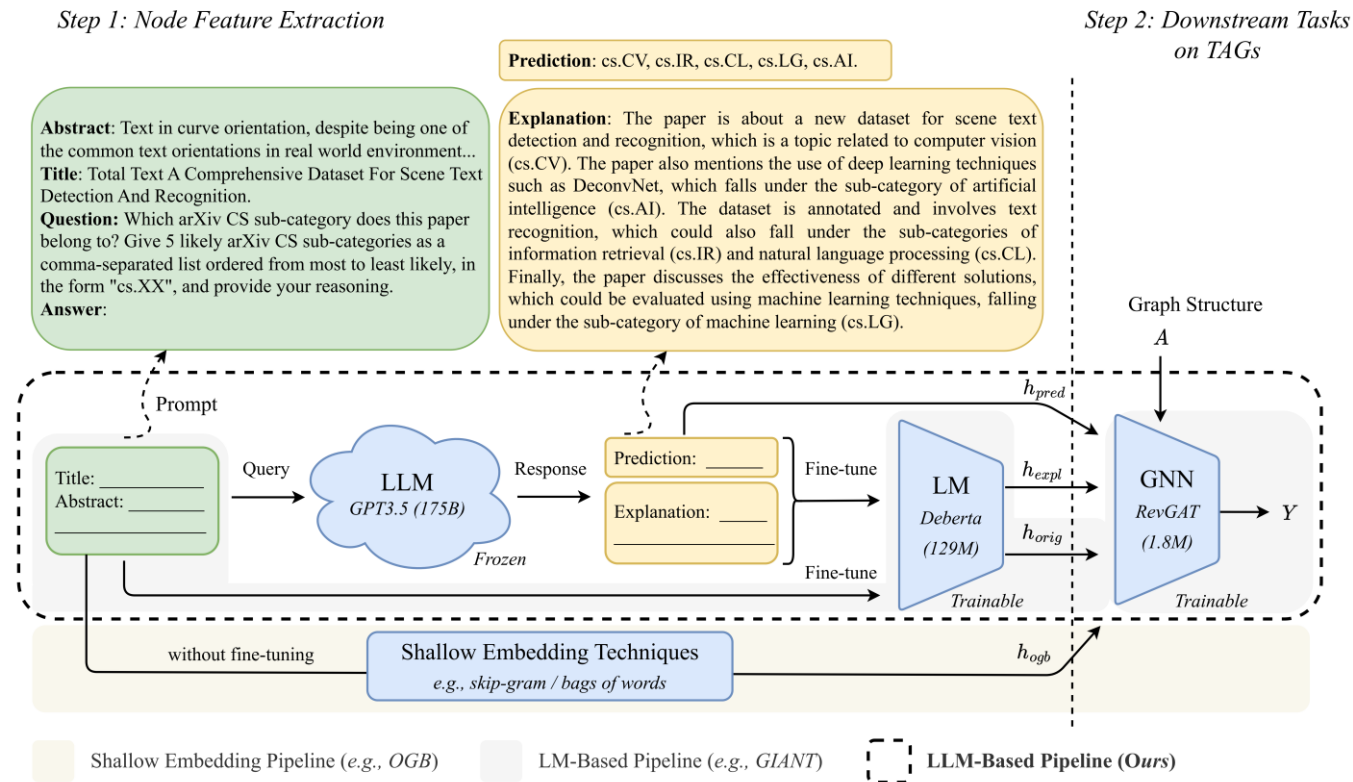


Liu, et al. "One for all: Towards training one graph model for all classification tasks." *ICLR'24*

GNN-centric Methods: TAPE

□ The backbone model:

Textual attributes \rightarrow LLM \rightarrow Prediction & Explanation \rightarrow Fine-tune LM \rightarrow Node features \rightarrow GNN

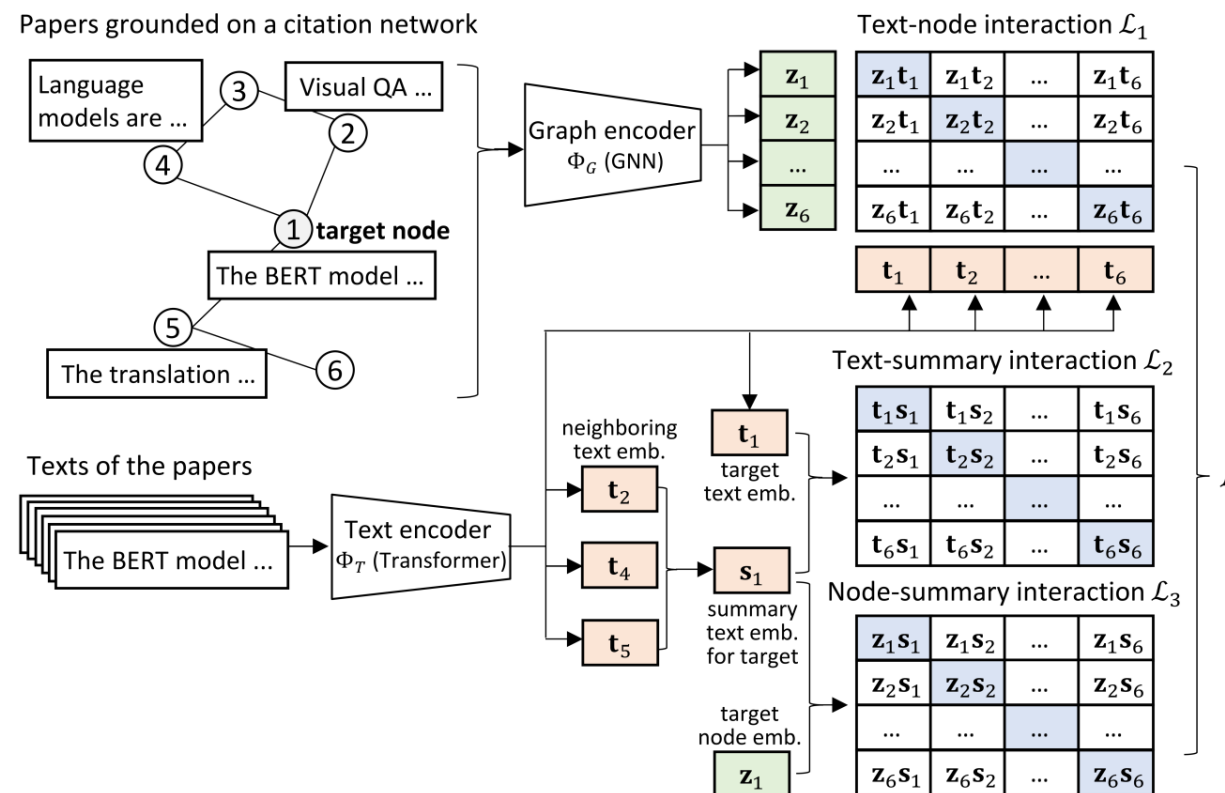
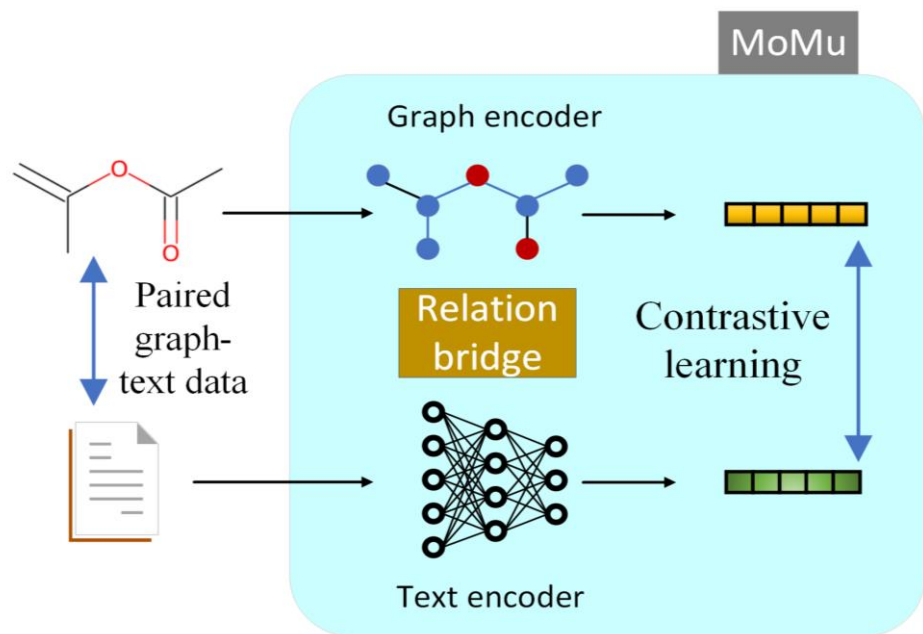


He, et al. "Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning." *ICLR'24*

Symmetric Methods: MoMu, G2P2

□ The backbone model:

- Dual encoders: Graph & Text encoder
- Contrastive Learning



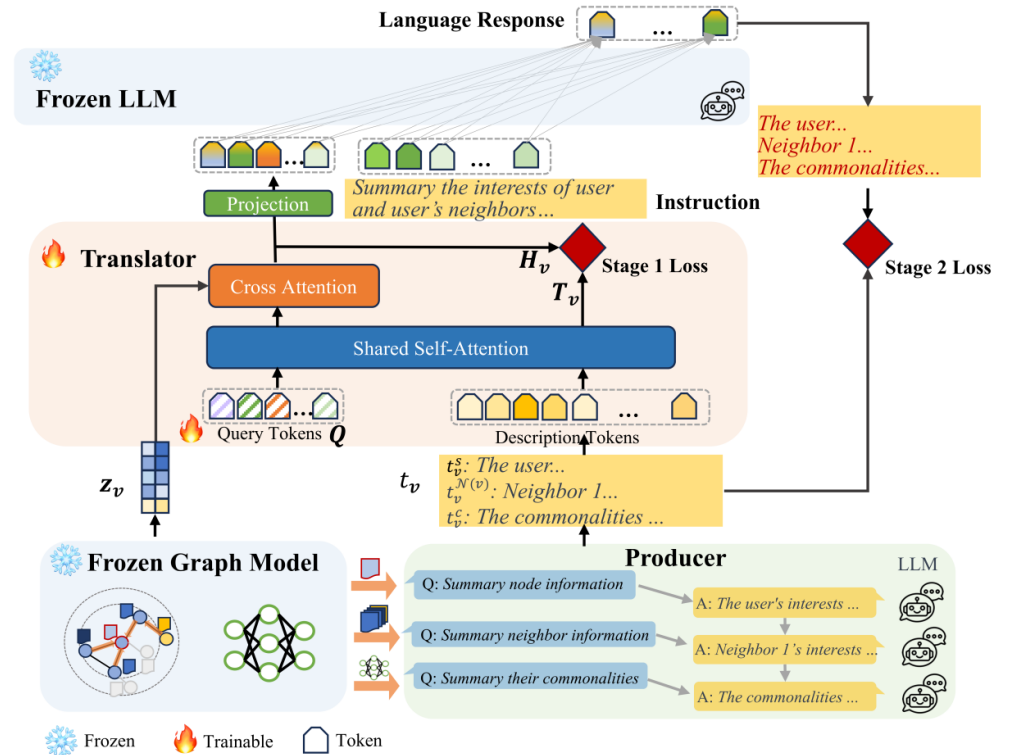
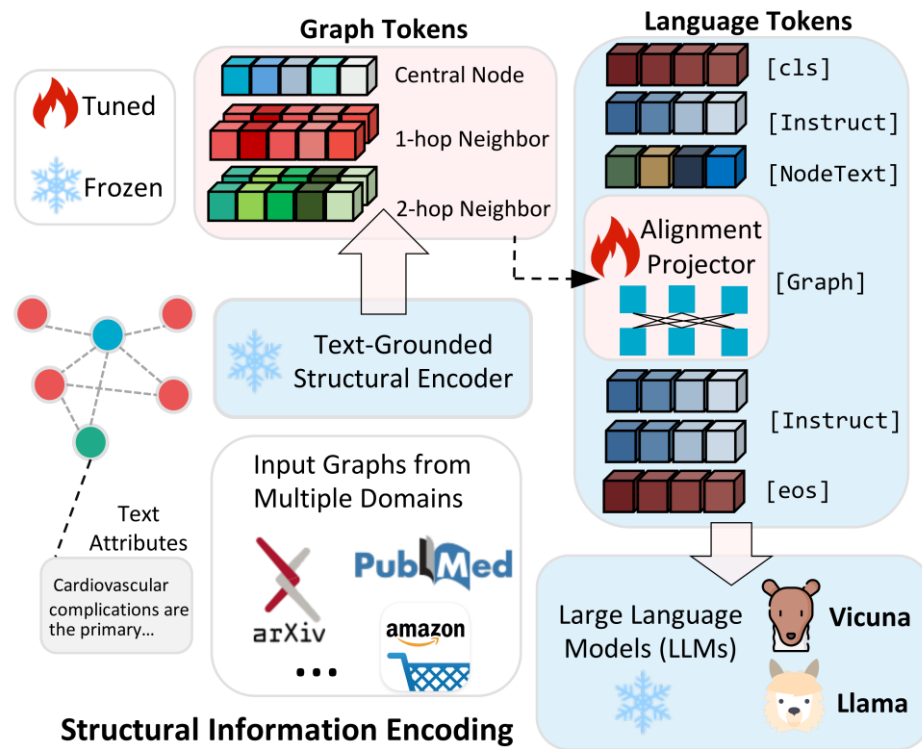
Su, et al. "A molecular multimodal foundation model associating molecule graphs with natural language." *CoRR*'22.

Wen, et al. "Augmenting low-resource text classification with graph-grounded pre-training and prompting." *SIGIR*'23.

LLM-centric Methods: GraphGPT, GraphTranslator

□ The backbone model:

Graph \rightarrow GNN \rightarrow Projection \rightarrow LLM



Tang, et al. "GraphGPT: Graph instruction tuning for large language models." *SIGIR'24*

Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." *WWW'24*

GNN+LLM based Models

□ Backbone Architectures

□ Pre-training

□ Adaptation

Model	Backbone Architecture	Pre-training	Adaptation
SimTeG [16]	GNN-centric	MLM, TTCL	Parameter-Efficient FT
TAPE [35]	GNN-centric	LM	Tuning-free Prompting + Parameter-Efficient FT
GIANT [11]	GNN-centric	MLM	Vanilla FT
GraD [79]	GNN-centric	MLM	Parameter-Efficient FT
GALM [147]	GNN-centric	Graph Reconstruction	Vanilla FT
GraphFormer [153]	Symmetric	MLM	Vanilla FT
GLEM [174]	Symmetric	MLM	Vanilla FT
ConGrat [4]	Symmetric	MLM + GTCL	Parameter-Efficient FT
G2P2 [136]	Symmetric	GTCL	Prompt Tuning
SAFER [6]	Symmetric	MLM	Parameter-Efficient FT
Text2Mol [18]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoMu [109]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoleculeSTM [73]	Symmetric	MLM + GTCL	Parameter-Efficient FT
CLAMP [103]	Symmetric	MLM + GTCL	Parameter-Efficient FT
Graph-Toolformer [165]	LLM-centric	LM	Tuning-free Prompting + Vanilla FT

Table 4. Details of approaches involved as GNN+LLM based models

Pre-training

□ GNN or LLM-based

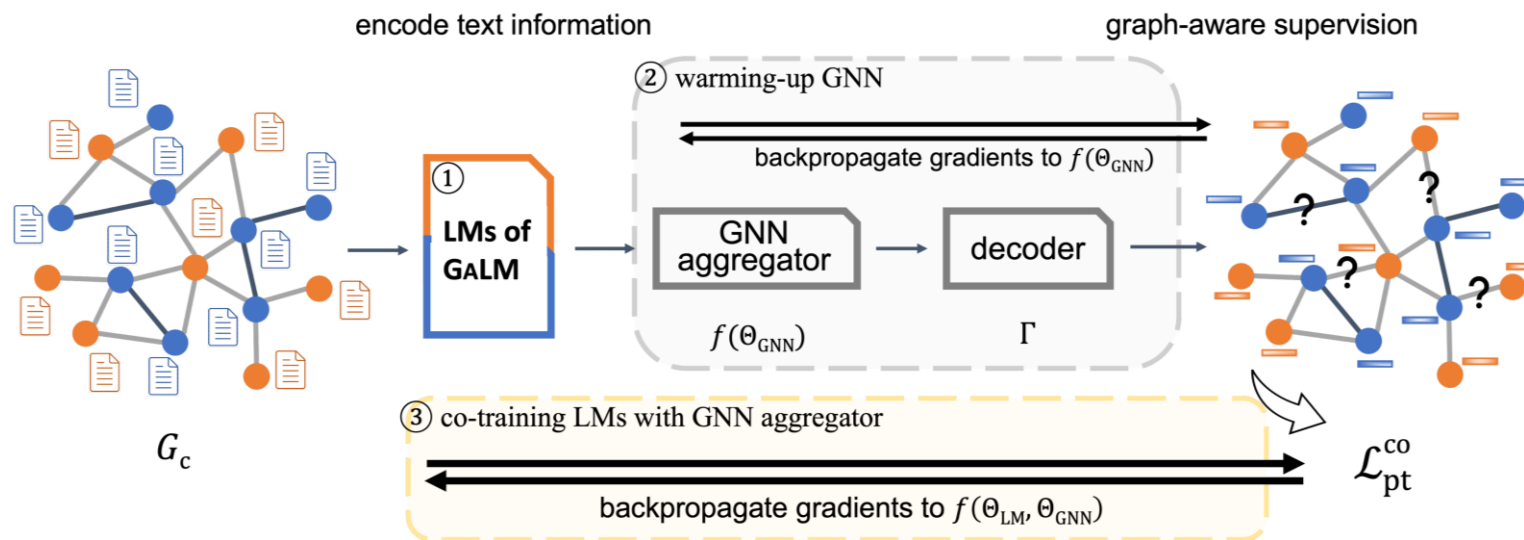
- Masked Language Modeling
- Language Modeling
- Text-Text Contrastive Learning
- Graph reconstruction

□ Alignment-based

- Graph-Text Contrastive Learning

GNN or LLM-based: GaLM

- GaLM (Graph-aware Language Model pre-training):
 - Fine-tuning existing general LMs by graph-aware supervision
 - Warming up the GNN aggregator by fixing the pre-trained LMs
 - Co-training GNN+LMs

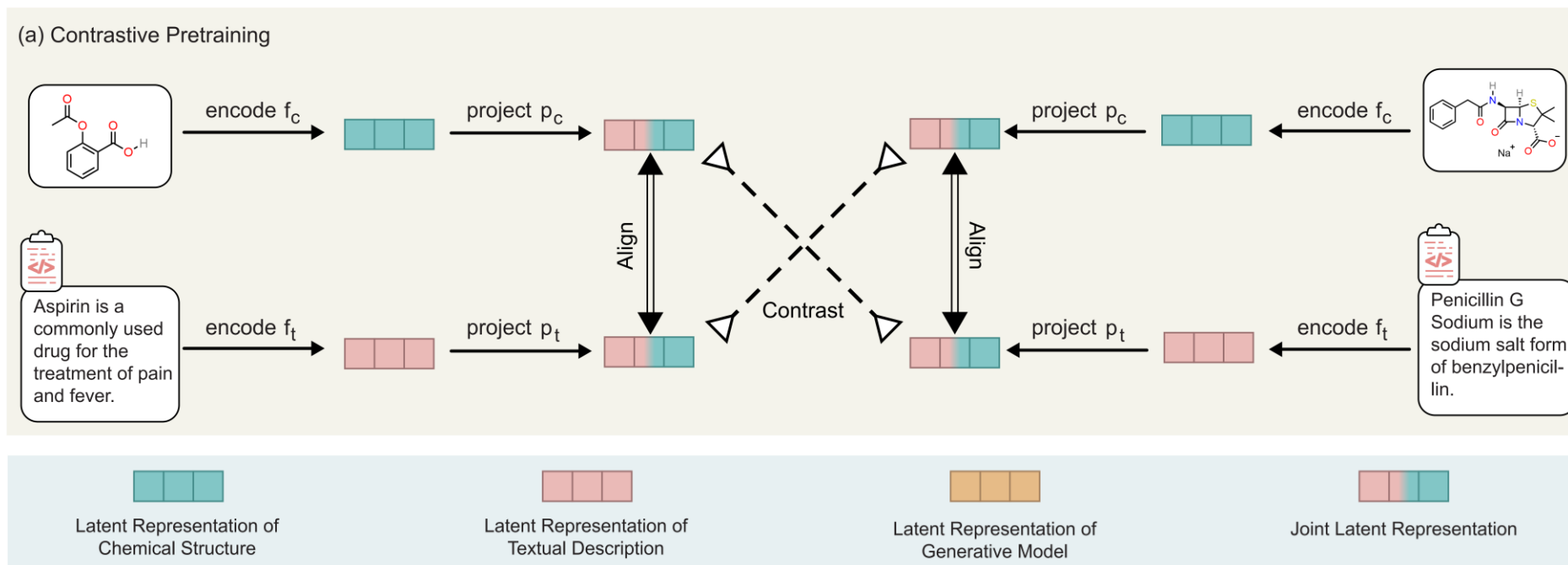


Xie, et al. "Graph-aware language model pre-training on a large graph corpus can help multiple graph applications." *KDD'23*.

Alignment-based: MoleculeSTM

□ Graph-Text Contrastive Learning (GTCL)

- Map the graph and text representations extracted to a joint space using two projectors (p_c and p_t) via contrastive learning

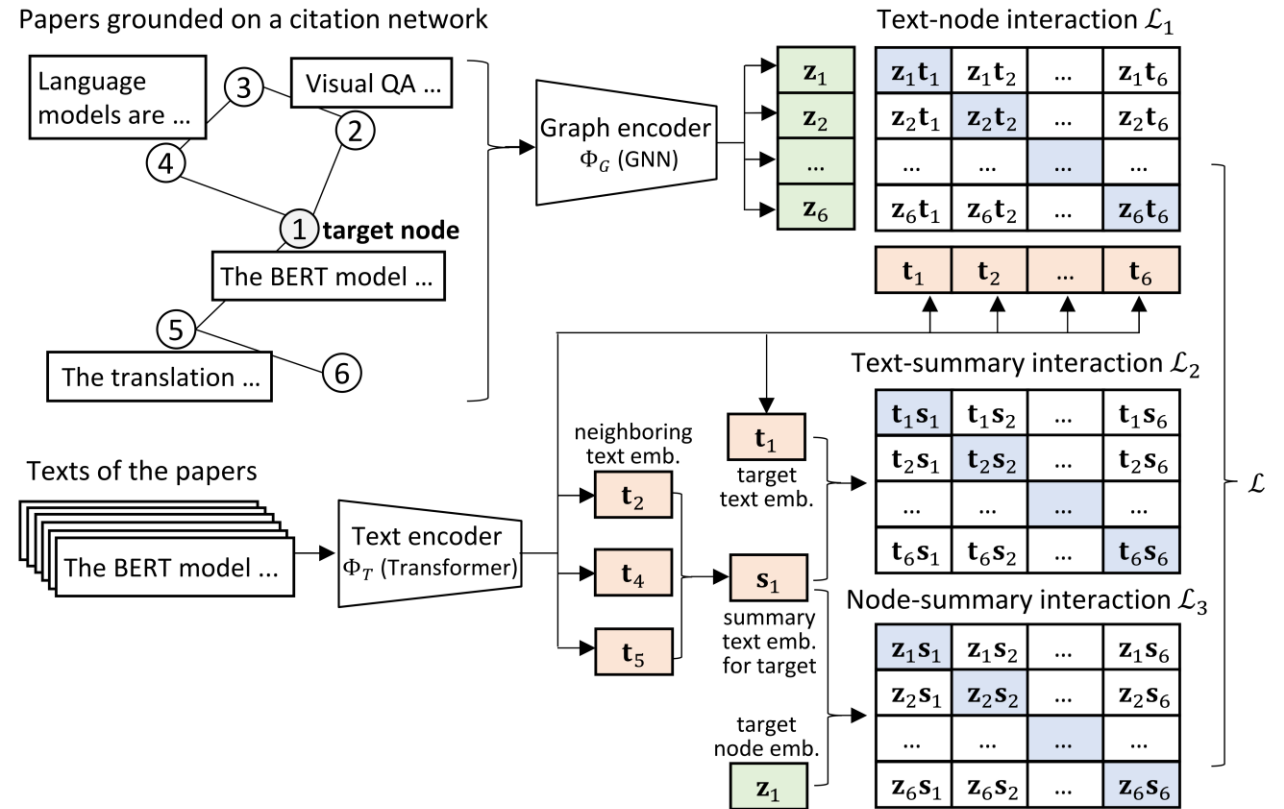


Liu, et al. "Multi-modal molecule structure–text model for text-based retrieval and editing." *Nature Machine Intelligence* 2023

Alignment-based: G2P2

- ❑ Dual encoders
- ❑ Three kinds of alignments
 - Text-Node: L1
 - Text summary-Text: L2
 - Text summary-Node: L3
- Text-summary: text of neighbors

$$s_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{t}_j$$



GNN+LLM based Models

□ Backbone Architectures

□ Pre-training

□ Adaptation

Model	Backbone Architecture	Pre-training	Adaptation
SimTeG [16]	GNN-centric	MLM, TTCL	Parameter-Efficient FT
TAPE [35]	GNN-centric	LM	Tuning-free Prompting + Parameter-Efficient FT
GIANT [11]	GNN-centric	MLM	Vanilla FT
GraD [79]	GNN-centric	MLM	Parameter-Efficient FT
GALM [147]	GNN-centric	Graph Reconstruction	Vanilla FT
GraphFormer [153]	Symmetric	MLM	Vanilla FT
GLEM [174]	Symmetric	MLM	Vanilla FT
ConGrat [4]	Symmetric	MLM + GTCL	Parameter-Efficient FT
G2P2 [136]	Symmetric	GTCL	Prompt Tuning
SAFER [6]	Symmetric	MLM	Parameter-Efficient FT
Text2Mol [18]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoMu [109]	Symmetric	MLM + GTCL	Parameter-Efficient FT
MoleculeSTM [73]	Symmetric	MLM + GTCL	Parameter-Efficient FT
CLAMP [103]	Symmetric	MLM + GTCL	Parameter-Efficient FT
Graph-Toolformer [165]	LLM-centric	LM	Tuning-free Prompting + Vanilla FT

Table 4. Details of approaches involved as GNN+LLM based models

Adaptation

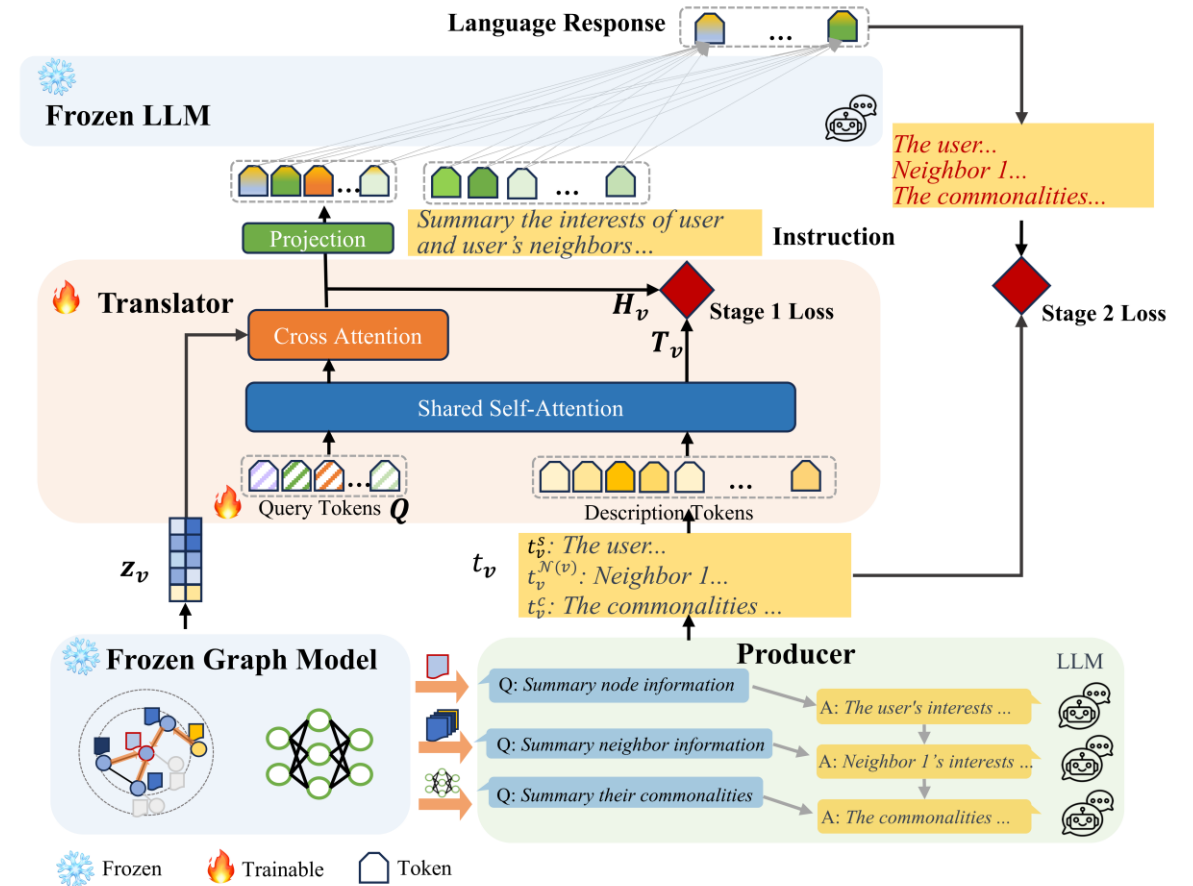
□ Fine-tuning

- Vanilla tuning: tune all the parameters
 - computationally intensive, resource-demanding
- Parameter-efficient fine-tuning (PEFT): tune a subset of parameters
 - more efficient, resource-friendly

□ Prompt-Tuning: design and tune external prompts

PEFT: GraphTranslator

- ❑ Frozen:
 - Graph Model
 - Large Language Model
- ❑ Tunable:
 - Producer Module
 - Construct alignment data
 - Translator Module
 - Convert node representations into tokens for LLM prediction

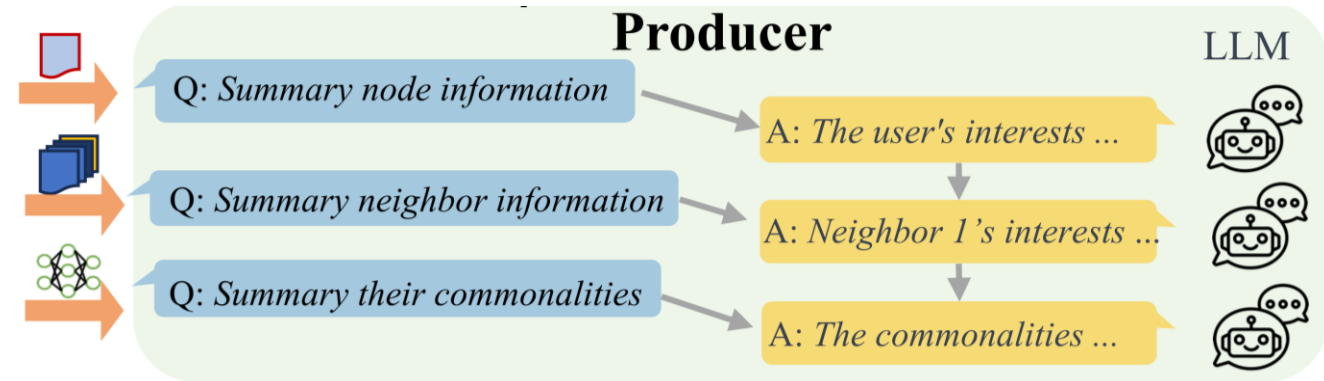


PEFT: GraphTranslator

□ Producer:

➤ “Chain of Thought” (CoT) ->LLM->high-quality description

- node information
- neighbor information
- commonalities



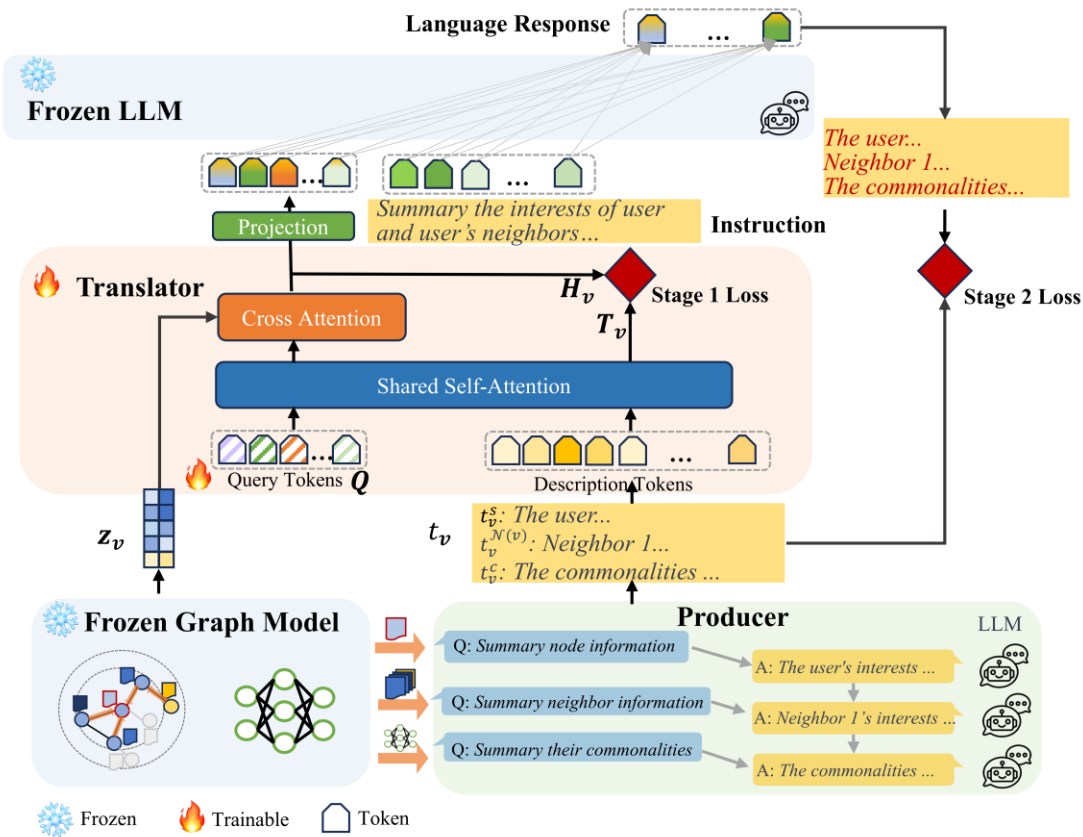
□ Prompt template:

Dataset	Step	Prompt
Taobao	User behavior summary	User Behavior Description: <User Behavior Description> . Please summarize the characteristics of this user according to the product behavior information. The answer format is: What kind of characteristics does the user have in terms of interests, hobbies, personality traits, and life needs
	Neighbor behavior summary	Neighbor Behavior Description: <Neighbor Behavior Description> . Please summarize most of the similarities that this user's friends have based on the product behavior information. The answer format is: What do several friends of this user have in common in interests, hobbies, personality traits, and life needs?

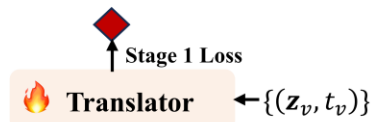
Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." WWW'24

PEFT: GraphTranslator

□ Training: Only fine-tune Translator and Projection

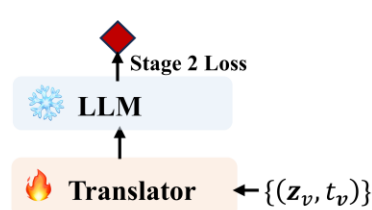


Stage 1 Training Phase



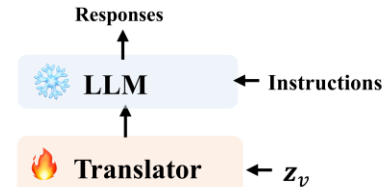
➤ Stage1: Align graph-text

Stage 2 Training Phase



➤ Stage2: Align graph-LLM

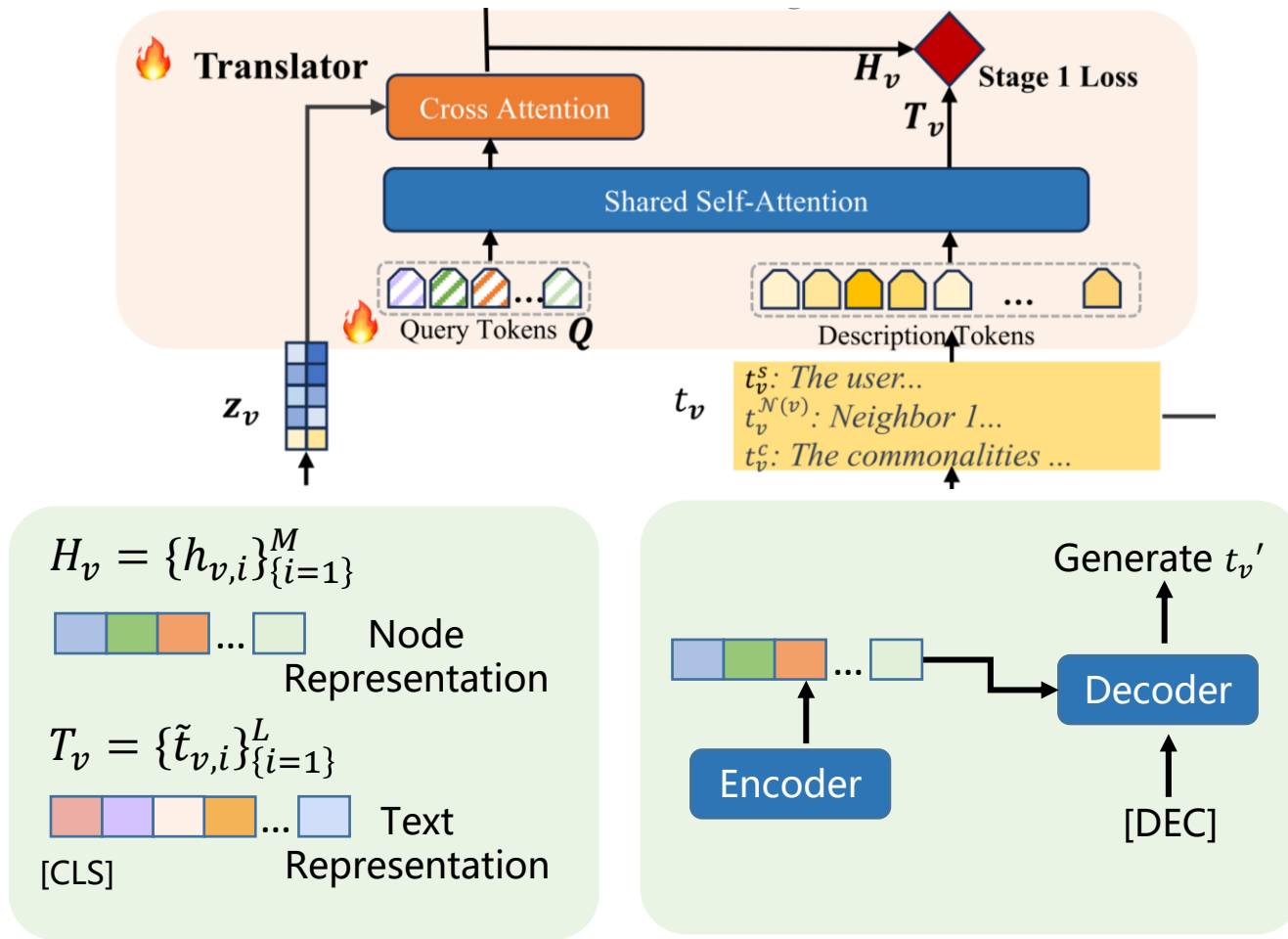
Inference Phase



Zhang, et al. "GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks." WWW'24

PEFT: GraphTranslator

Training: Stage 1



➤ Contrastive Objective

- Node \leftrightarrow Text
- High-level alignment

➤ Matching Objective

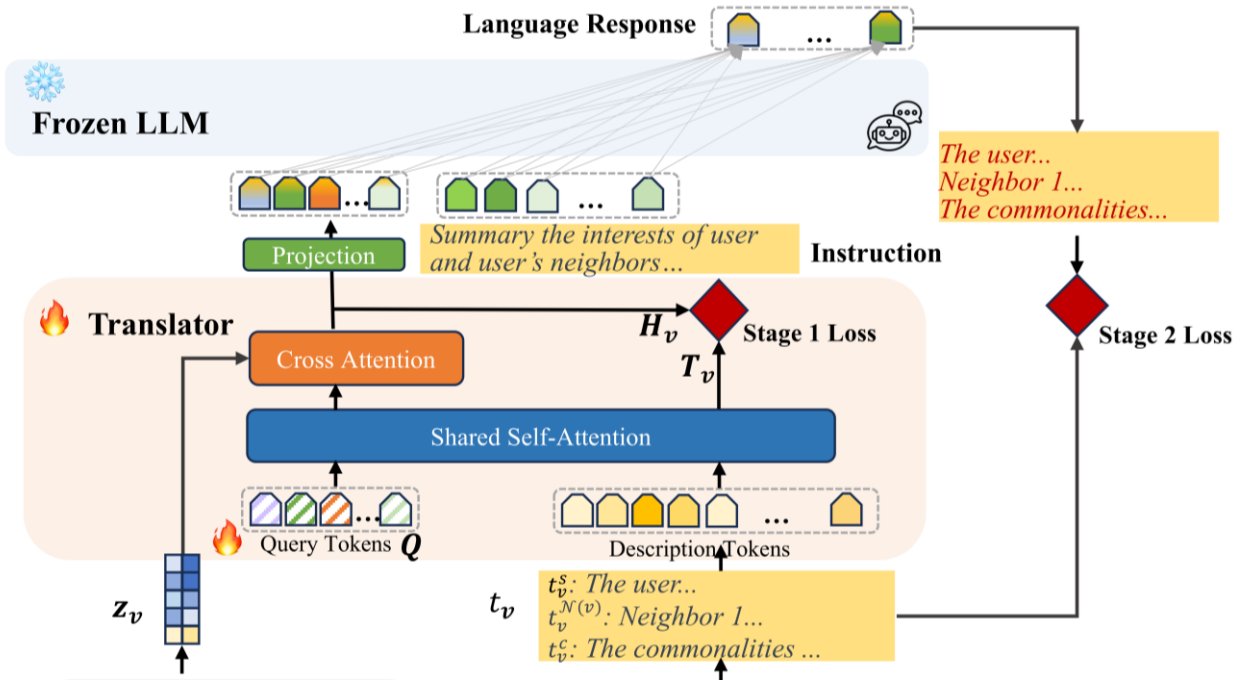
- Node \leftrightarrow Text
- Fine-grained alignment

➤ Generation Objective

- Node \rightarrow Text
- Replace the [CLS] token with a new [DEC] token as the first text token to signal the decoding task

PEFT: GraphTranslator

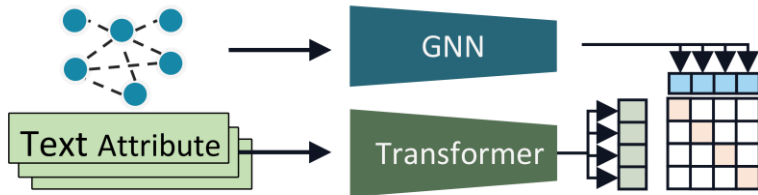
□ Training: Stage 2



- Projection:
 - A linear layer: project H_v to token representation space of LLM
- Concatenate:
 - Connect the projected representation with the human instruction and feed into LLM
- Fine-tune Translator
 - Align the response text of LLM with the actual descriptive text

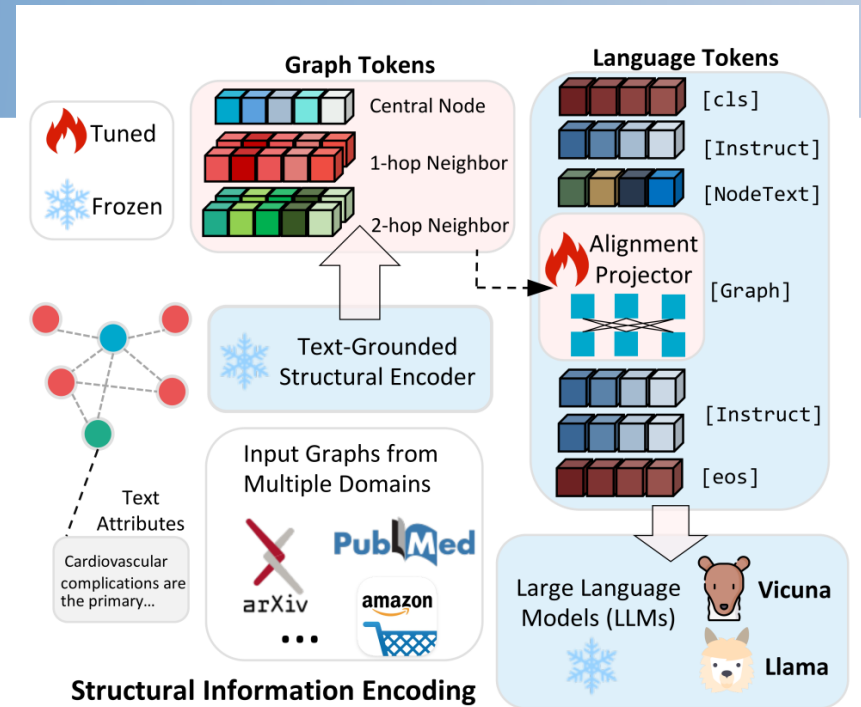
PEFT: GraphGPT

Graph: Text-Grounded Structural Encoder



Projector: Map graph representation to LLM

Instruction Tuning: Only fine-tune projector



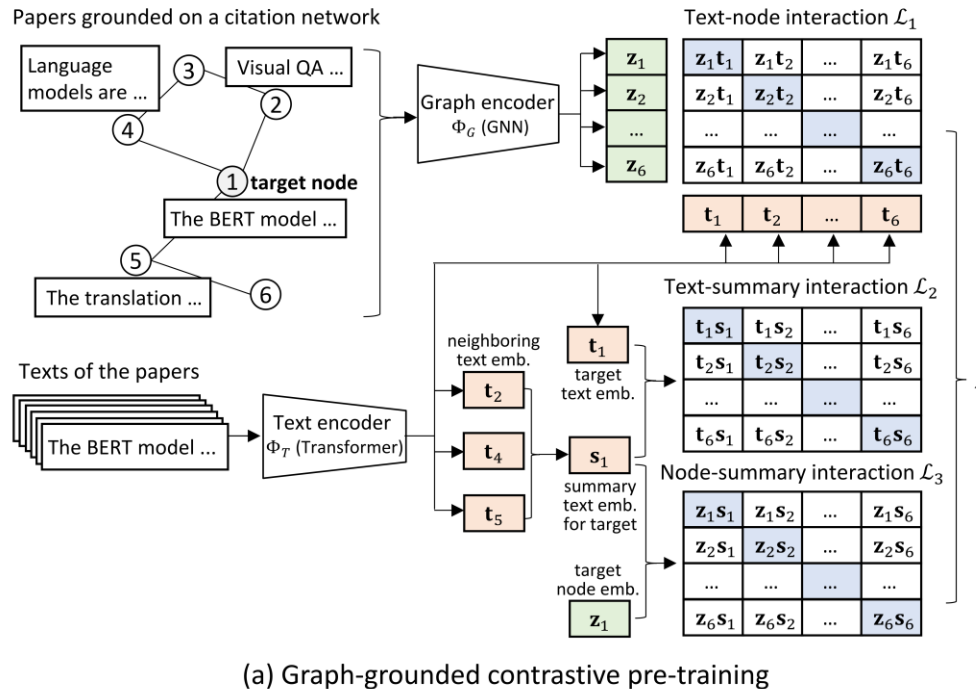
Graph Information: `<graph>`: Central Node: 68442, Edge index: [[...src node...],[...dst node...]], Node list: [...] **Graph Matching**
Human Question: Given a sequence of graph tokens `<graph>` that constitute a subgraph of a citation graph, ... Here is a list of paper titles: 1. ... 2. ..., please reorder the list of papers according to the order of graph tokens.
GraphGPT Response: Based on the given graph tokens and the list of paper titles, we obtain the matching of graph tokens and papers: Graph token 1 corresponds to smt based induction methods for timed systems. Graph token 2 corresponds to ...

Graph Information: `<graph>`: Central Node: 2, Edge index: [[...src node...],[...dst node...]], Node list: [...] **Node Classification**
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? ...
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...

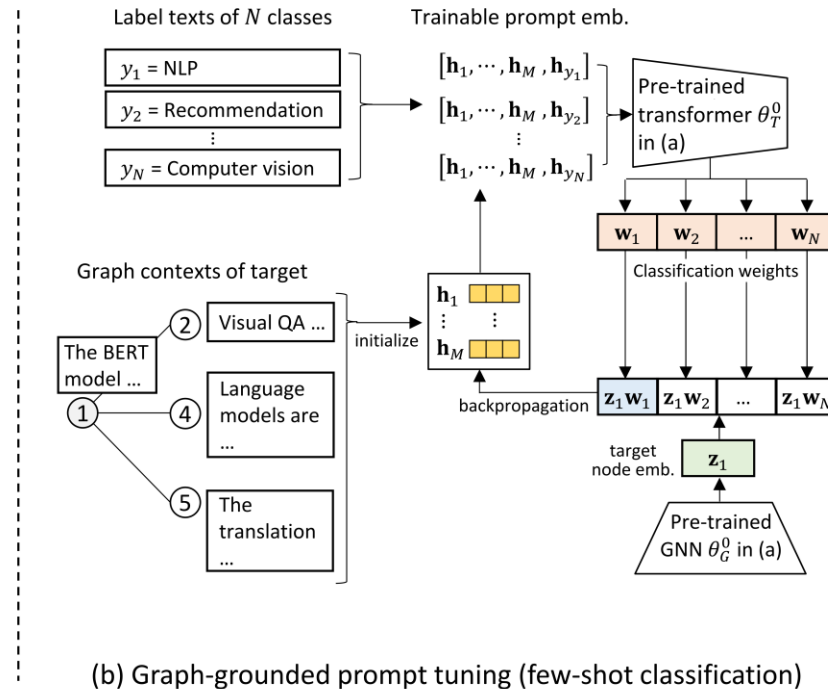
Graph Information: `<graph>`: Central Node 1: 8471, Edge index 1: [[...src node...],[...dst node...]], Node list 1: [...]
`<graph>`: Central Node 2: 19368, Edge index 2: [[...src node...],[...dst node...]], Node list 2: [...] **Link Prediction**
Human Question: Given a sequence of graph tokens: `<graph>`, ... Abstract: ... Title: ... 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: Yes, they are connected. Based on the first paper, ... And the second paper proposes ...

Prompt-Tuning: G2P2

- Learnable prompts: $[h_1, \dots, h_M, h_{CLASS}]$
- Tuning prompts with limited labeled data for efficient adaptation



(a) Graph-grounded contrastive pre-training



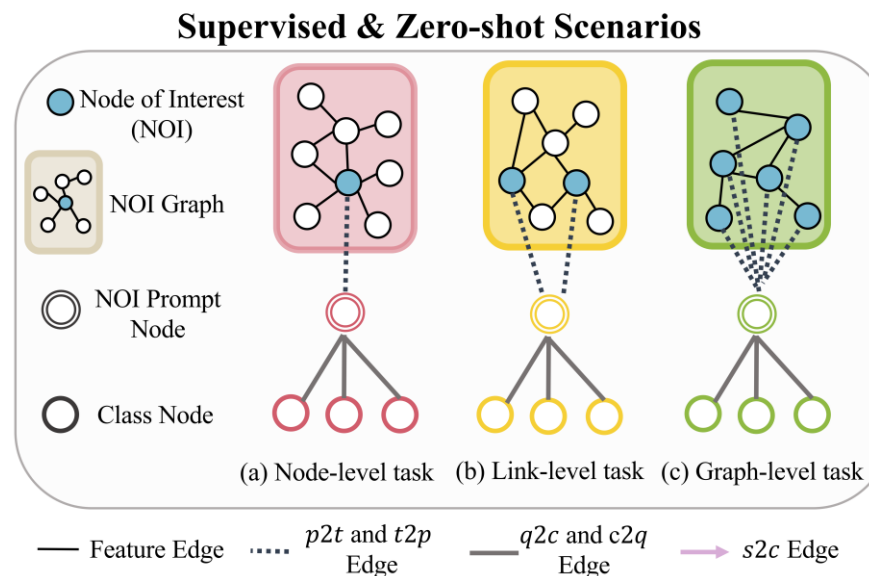
(b) Graph-grounded prompt tuning (few-shot classification)

Wen, et al. "Augmenting low-resource text classification with graph-grounded pre-training and prompting." *SIGIR'23*.

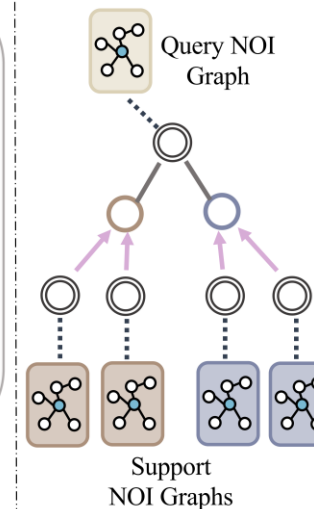
Prompt-Tuning: One for all

□ NOI (Node of Interest):

- Node-level: node
- Link-level: node pair
- Graph-level: subgraph



Few-shot Scenario



□ NOI Prompt Node

Text feature of the NOI prompt node: Prompt node. $\langle task\ description \rangle$.

Example: Prompt node. Graph classification on molecule properties.

Example: Prompt node. Node classification on the literature category of the paper.

□ Class Node

Text feature of class node: Prompt node. $\langle class\ description \rangle$.

Example: Prompt node. Molecule property. The molecule is effective in: ...

Example: Prompt node. Literature Category. cs.AI (Artificial Intelligence). Covers all areas of AI except Vision ...

Liu, et al. "One for all: Towards training one graph model for all classification tasks." *ICLR'24*.

Outline

□ LLM based Models

- Backbone Architectures
- Pre-training
- Adaptation

□ GNN+LLM based Models

- Backbone Architectures
- Pre-training
- Adaptation

□ **Summary and outlook**

Summary and outlook

□ Summary

- Leveraging LLMs facilitates a unified approach to various graph tasks by describing them in natural language.
- Merging graph data, text, and other modalities into LLMs creates a promising path for graph foundation models.
- Combining GNNs and LLMs leads to improved performance in graph-related tasks.

Summary and outlook

□ Outlook

- Focus on resolving LLMs' limitations: multi-hop reasoning, graph topology, and diverse graph data.
- Explore efficient training methods to manage the high computational costs and data requirements.
- Explore applications of GNN+LLM models in multimodal and cross-modal tasks.