Language Models and Graphs

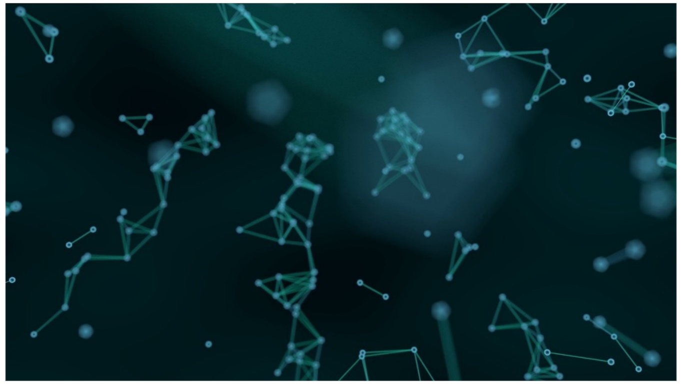
Mehwish Alam
Associate Professor
Télécom Paris
Institut Polytechnique de Paris
Winter Semester 2024-2025

Outline

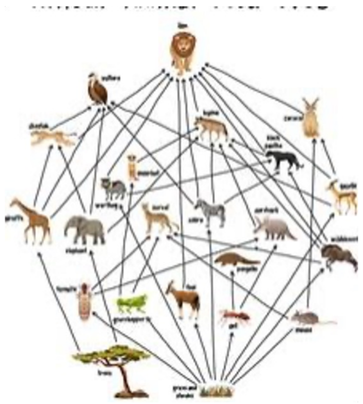
- **Foundations of Graphs**
- **Graph Representation Learning**
- **Language Models and Graphs**
  - **Graph Prompting**
  - **Aligning LLMs with Graph Representations**
  - **Fusing LLMs with Graph Representations**
  - **LLMs for Graphs**

Many Types of Data are Graphs

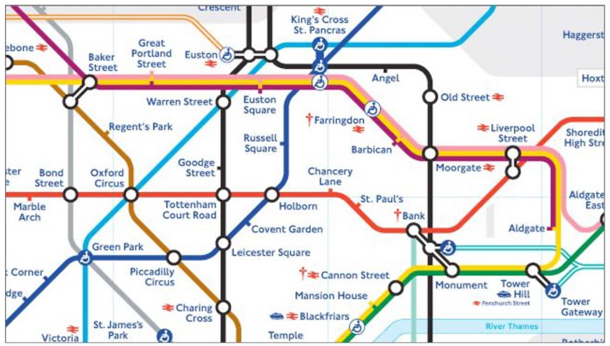
Computer Networks



Particle Networks

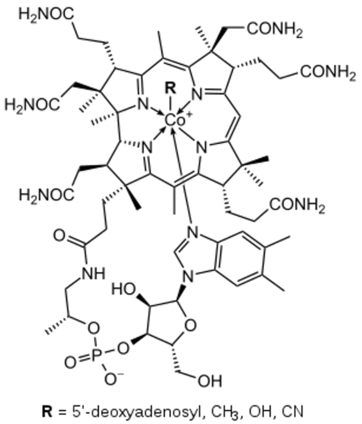


Particle Networks



Underground Networks

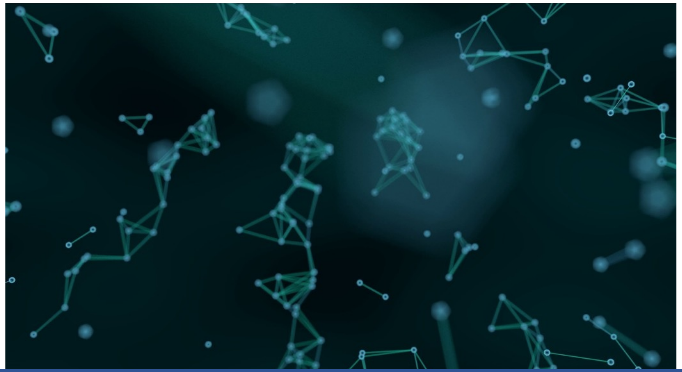
Social Networks

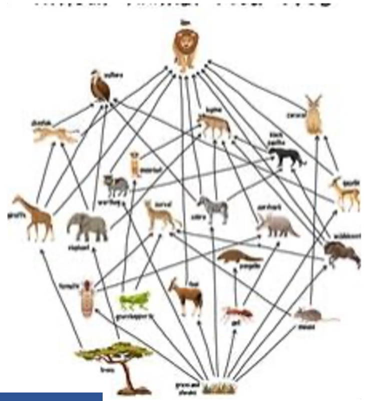


Molecule Networks

Many Types of Data are Graphs

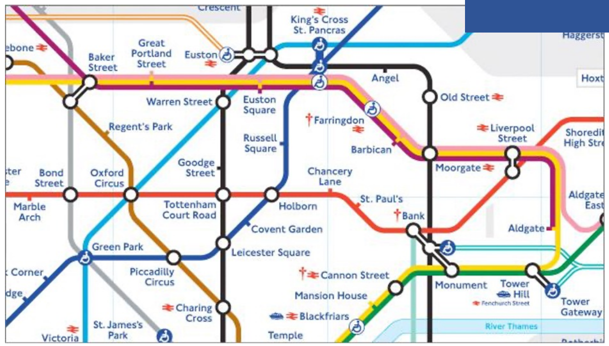
Computer Network





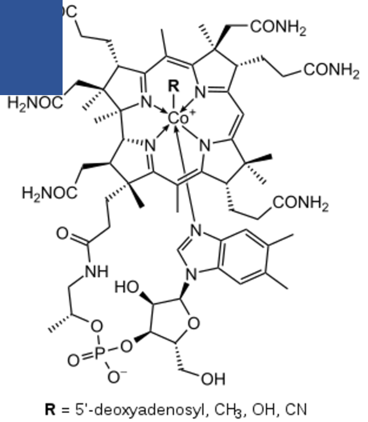
icle Networks

How do we take advantage of this structure for better prediction?



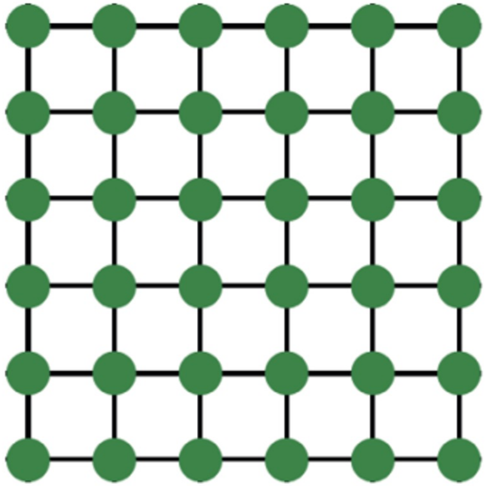
Underground Networks

Social Networks

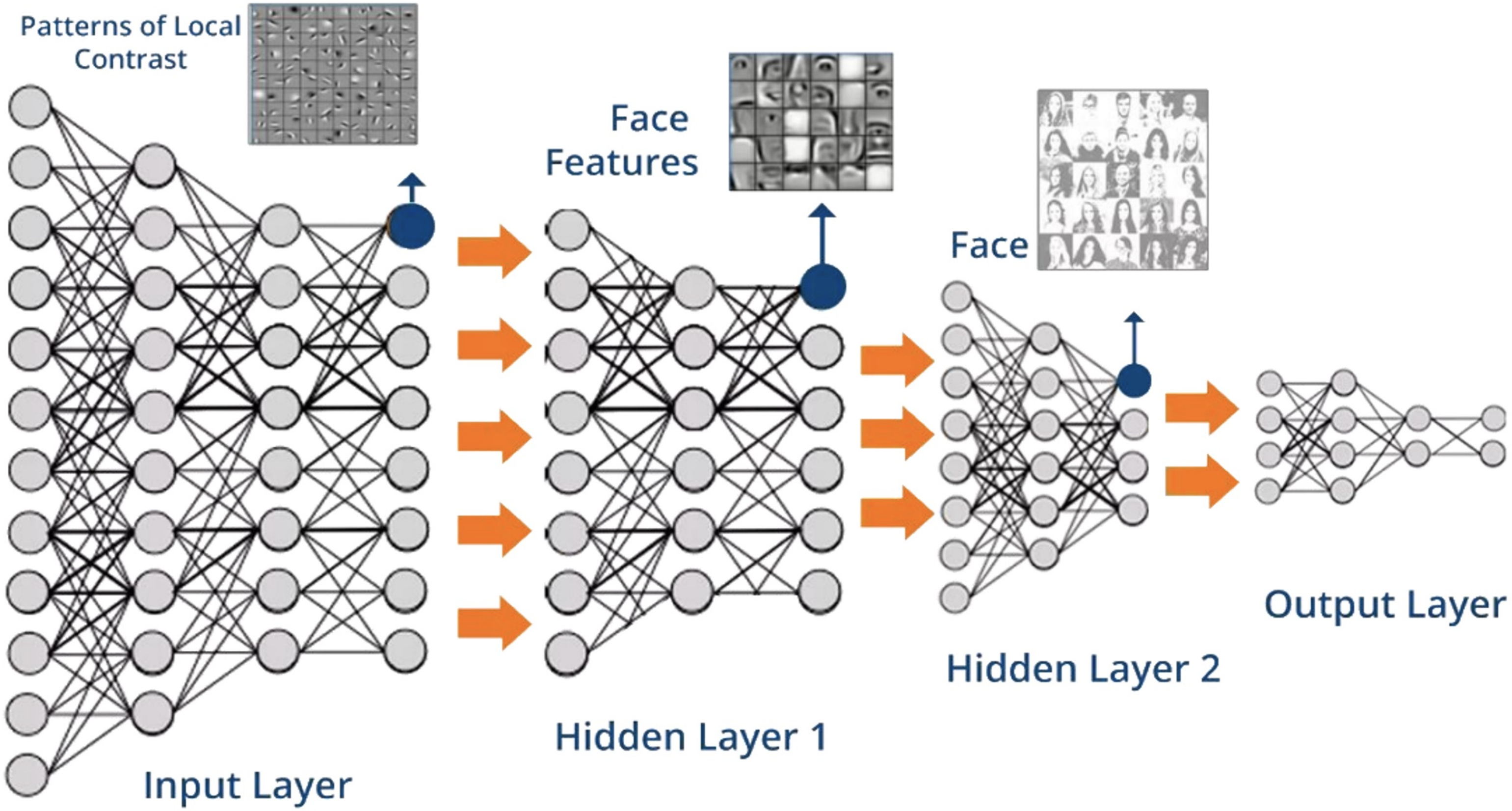


Molecule Networks

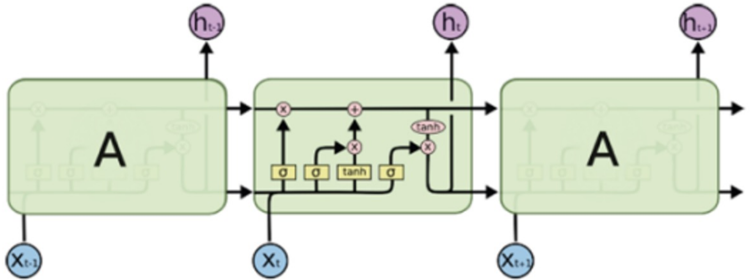
Current Machine Learning Algorithms



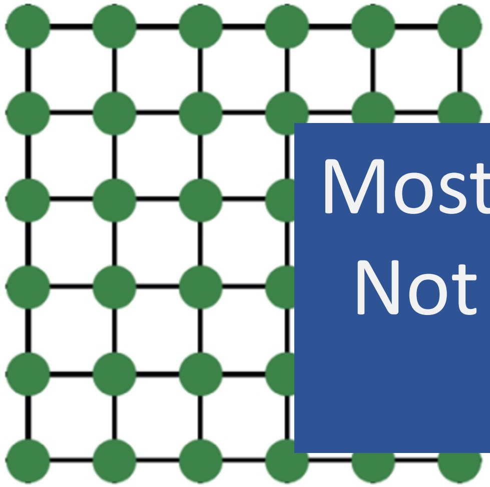
Images

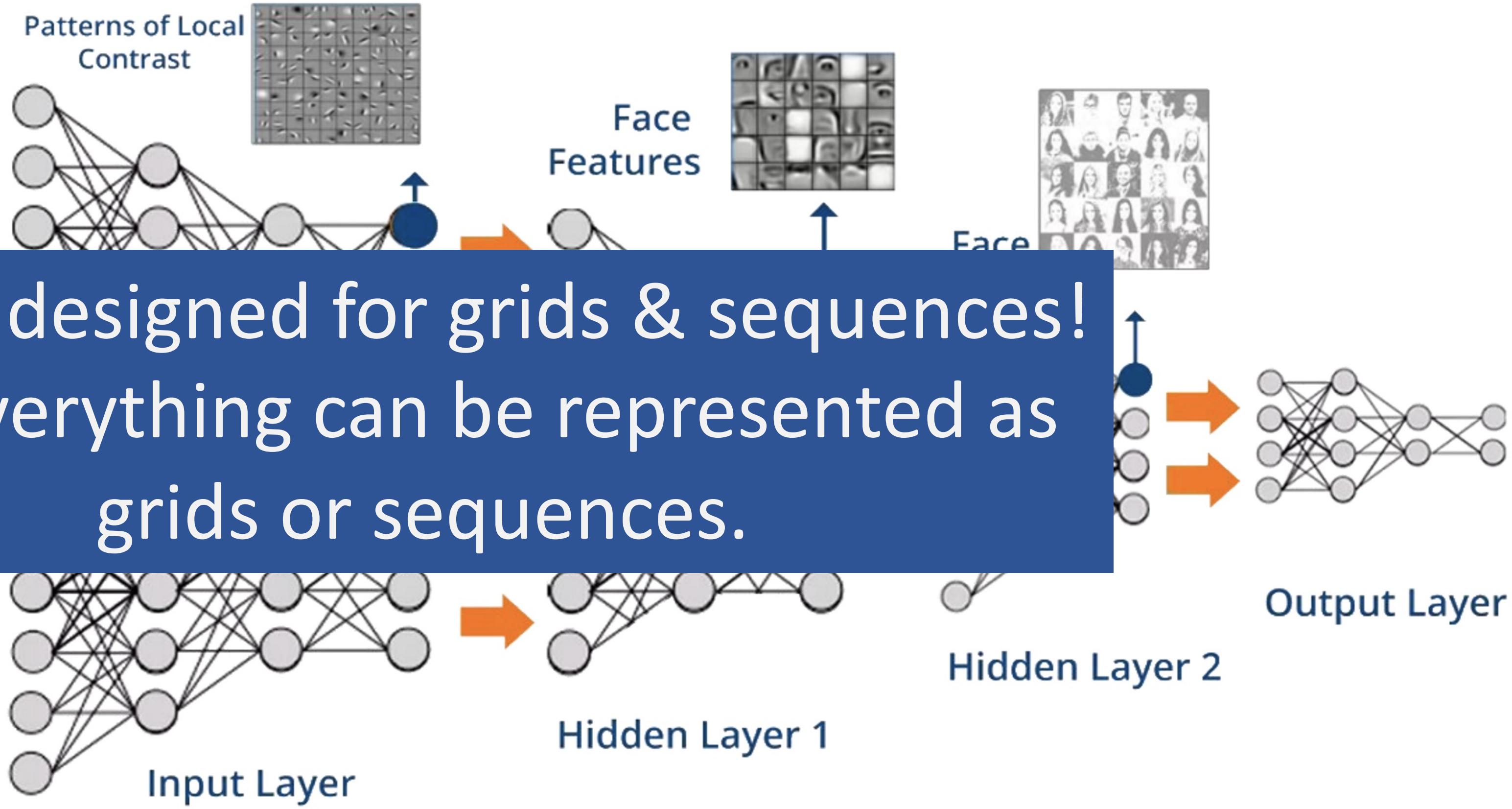


Text/Speech



Current Machine Learning Algorithms

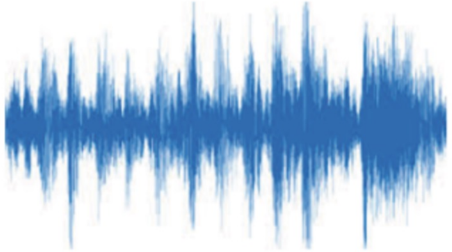


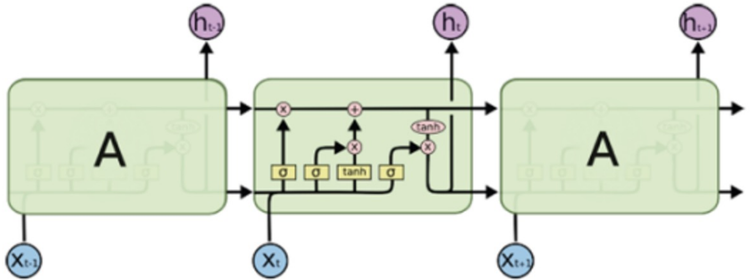


Mostly designed for grids & sequences!
Not everything can be represented as grids or sequences.

Images

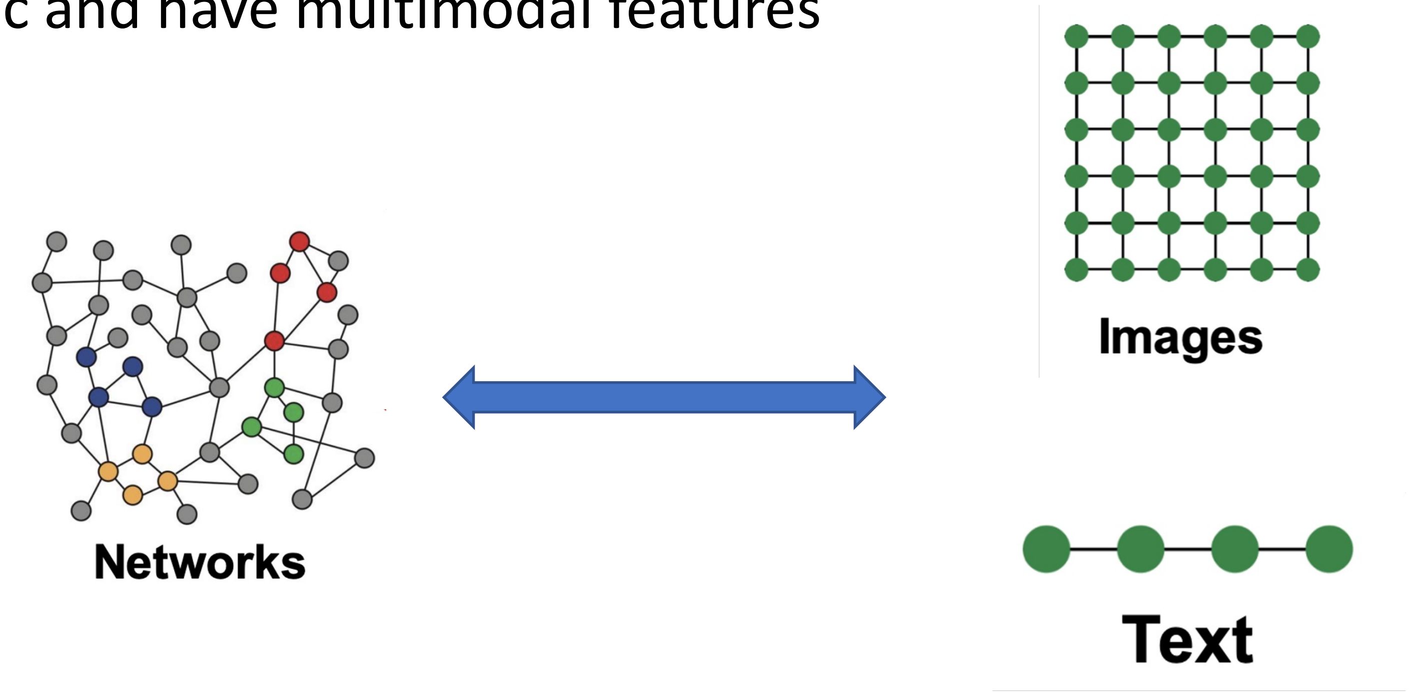
Text/Speech





Why is ML on Graphs Hard?

- Networks are complex
- Arbitrary size and complex topological structure (i.e., no spatial locality like grids)
- No fixed node ordering or reference point
- Often dynamic and have multimodal features

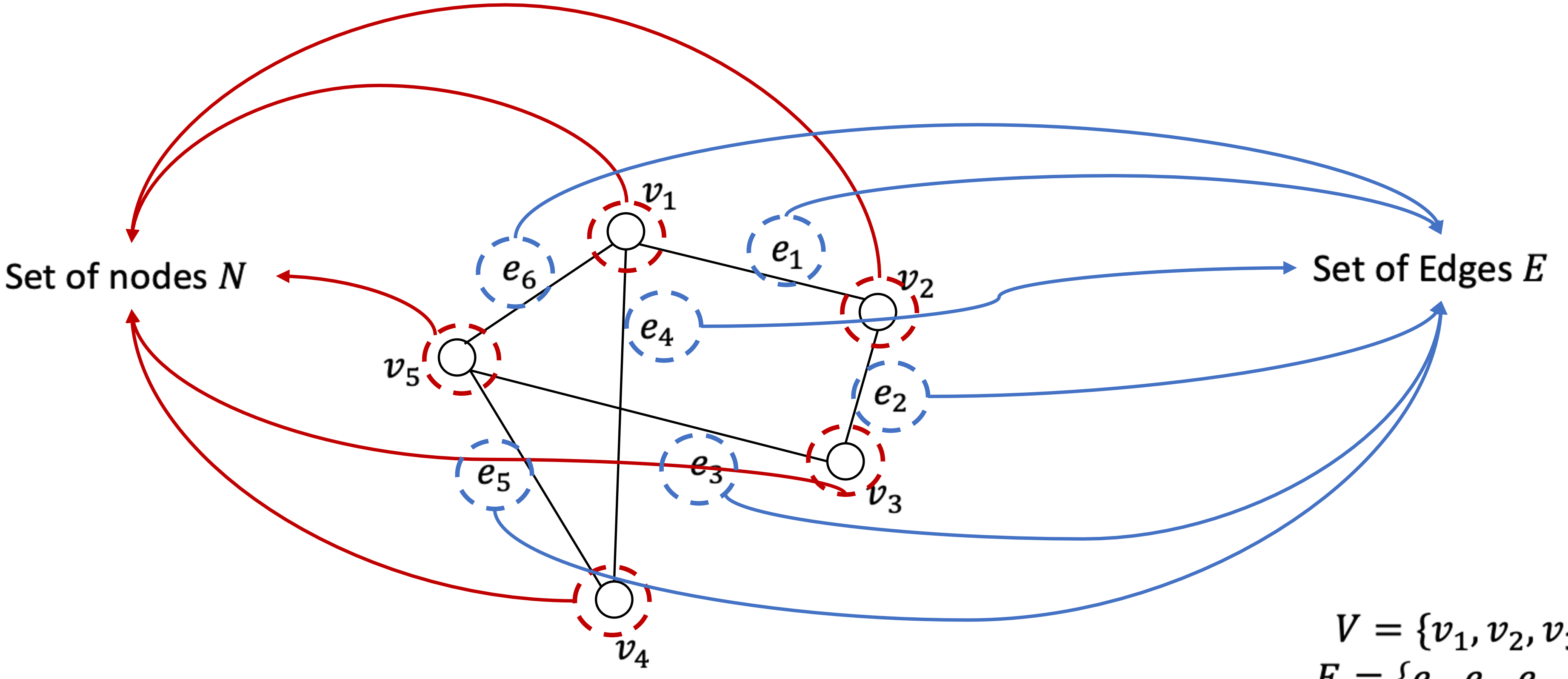


Outline

- **Foundations of Graphs**
- **Graph Representation Learning**
- **Language Models and Graphs**
  - **Graph Prompting**
  - **Aligning LLMs with Graph Representations**
  - **Fusing LLMs with Graph Representations**
  - **LLMs for Graphs**

What is Graph?

Definition (Graph). A graph can be denoted as $G = \{V, E\}$, where $V = \{v_1, v_2, \ldots, v_N\}$ is a set of $N = |V|$ nodes and $E = \{e_1, \ldots, e_M\}$ is a set of M edges.



$$V = \{v_1, v_2, v_3, v_4, v_5\}$$
$$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$$
$$N = 5$$
$$M = 6$$

More on Graphs

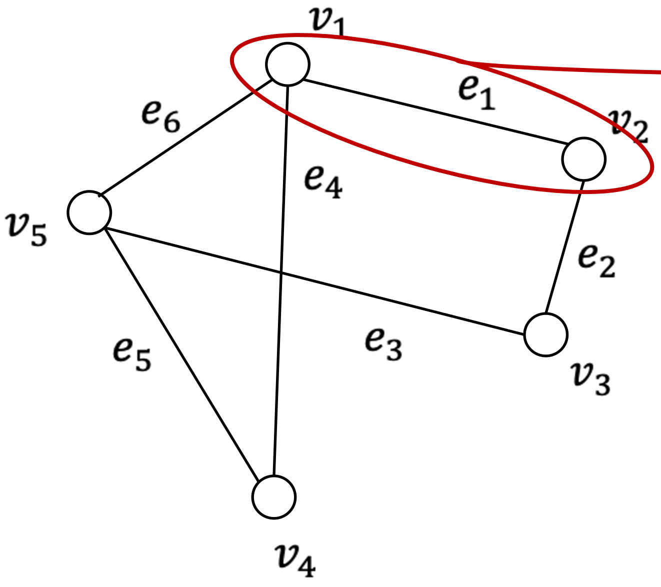
- The **size** of a given graph G is defined by **its number of nodes**, i.e., $\boldsymbol{N} = |\boldsymbol{V}|$.
- The set of edges E describes the connections between nodes.
- An edge $e \in E$ connects two nodes v_e^1 and v_e^2 ; thus, the edge e can be also represented as (v_e^1, v_e^2).
- The nodes v_e^1 and v_e^2 are **incident** to the edge e.
- A node v_i is **adjacent** to another node v_j if and only if there exists an edge between them.

Adjacency Matrix

Definition (Adjacency Matrix). For a given graph $G = \{V, E\}$, the corresponding adjacency matrix is denoted as $A \in \{0,1\}^{N\times N}$. The $i,j-th$ entry of the adjacency matrix A, indicated as $A_{i,j}$, represents the connectivity between two nodes v_i and v_j. More specifically, $A_{i,j} = 1$ if v_i is adjacent to v_j, otherwise $A_{i,j} = 1$.

$i = the\ number\ of\ row$

$j = the\ number\ of\ column$

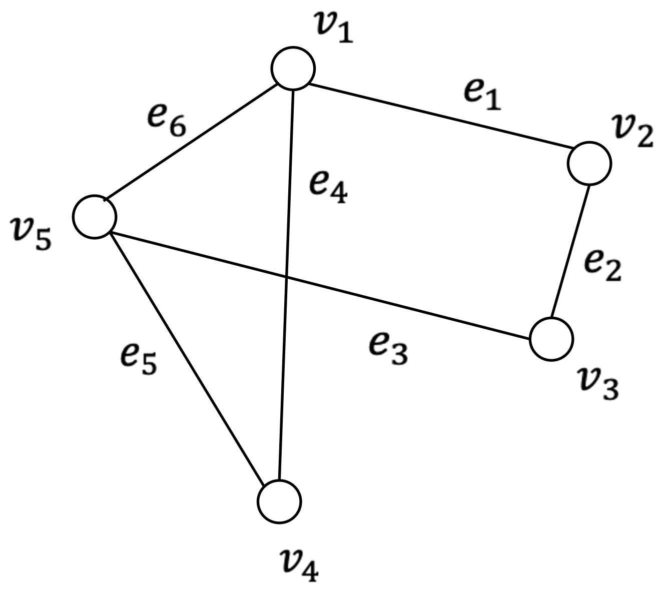


$A_{1,2}$

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

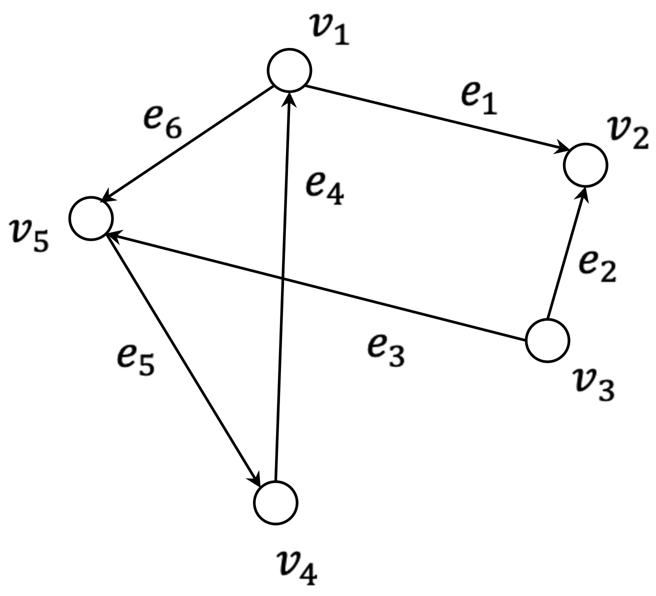
Undirected vs Directed Graphs

Undirected Graph



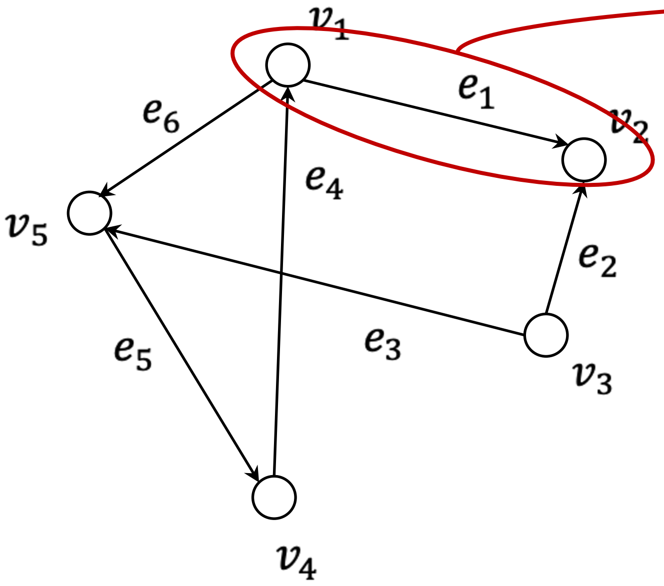
In **undirected graphs**, the order of the two nodes does not make a difference, i.e., $e_1 = (v_1, v_2) = (v_2, v_1)$. More generally, $e = (v_e^1, v_e^2) = (v_e^2, v_e^1)$.

Directed Graph



In **directed graphs**, the edge is directed from node v_e^1 to node v_e^2.

Adjacency Matrix (Directed Graph)

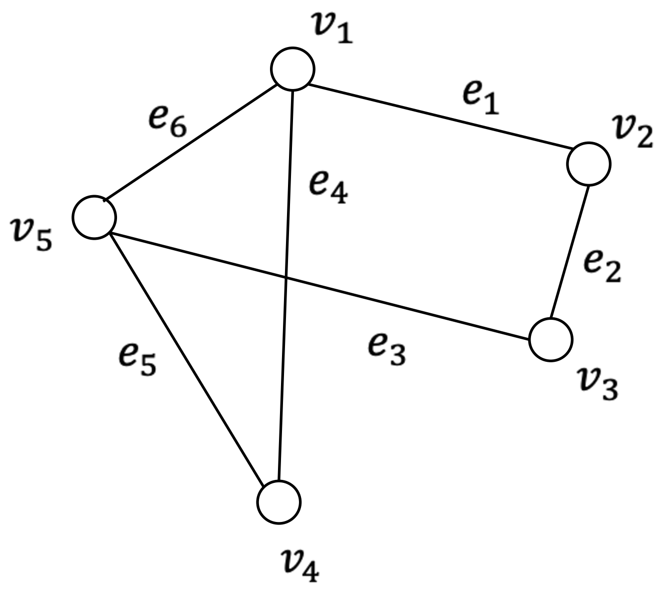


$$\boldsymbol{A} = \text{To} \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \text{From} \\ \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Unweighted and Weighted Graphs

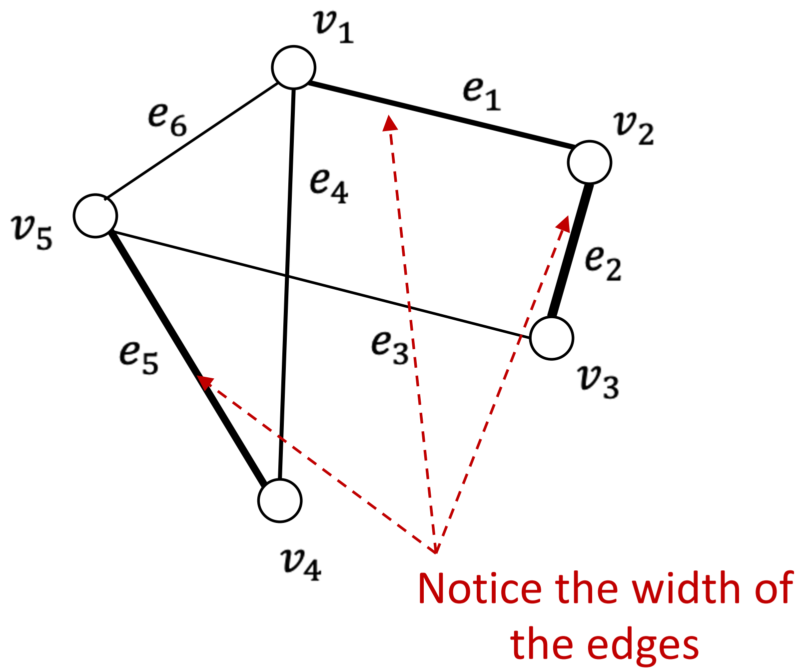
Unweighted Graphs

(undirected)

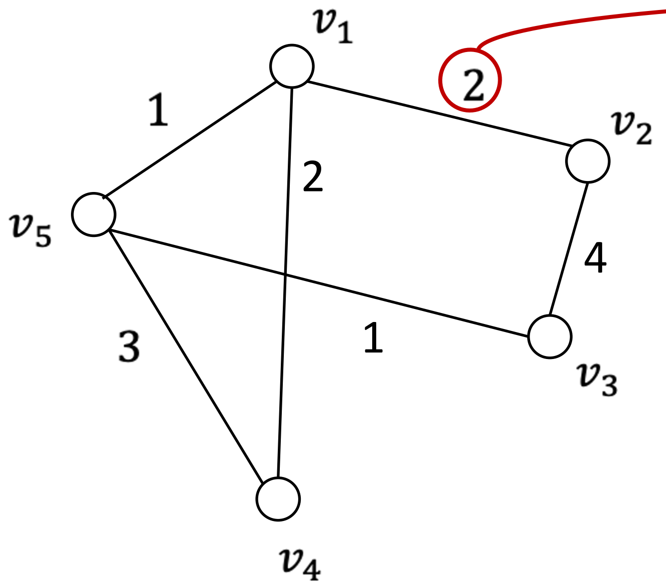


Weighted Graph

(undirected)



Adjacency Matrix (Weighted Graphs)



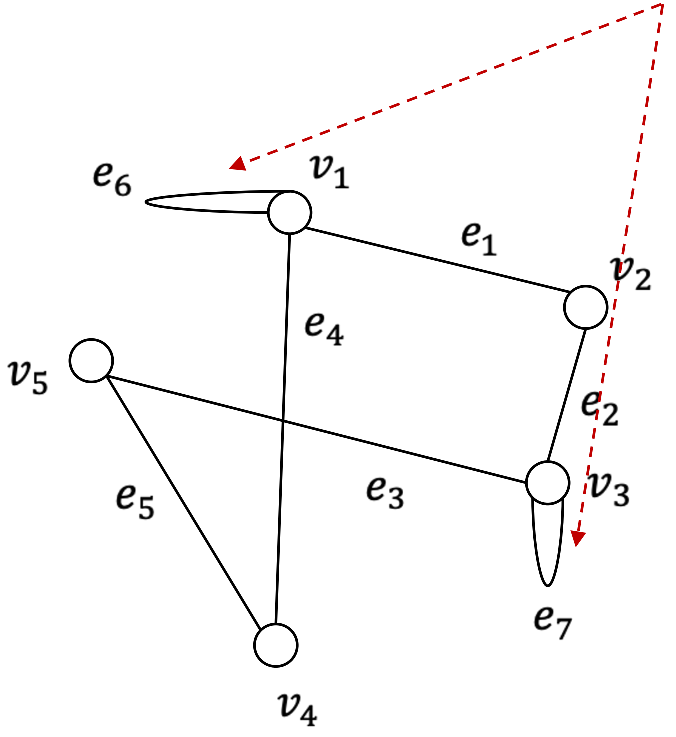
$$\boldsymbol{A} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 2 & 1 \\ 2 & 0 & 4 & 0 & 0 \\ 0 & 4 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 3 \\ 1 & 0 & 1 & 3 & 0 \end{bmatrix} \end{array}$$

Other Types of Graphs

Self-edges

(undirected)

Notice the loops



Adjacency Matrix Self-edges

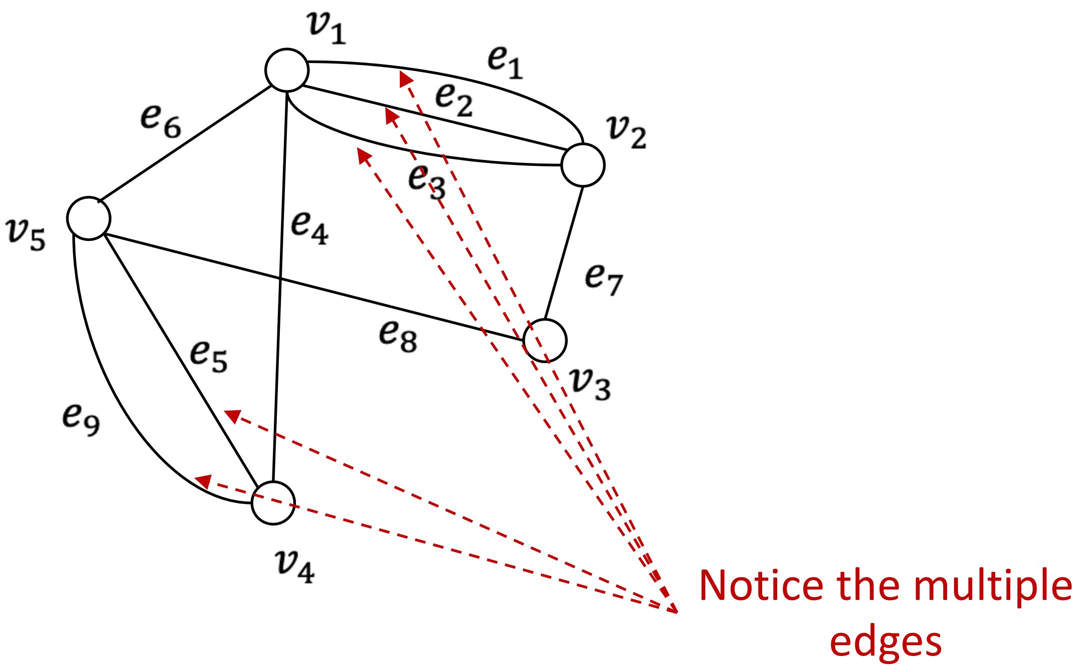
(undirected)

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Other Types of Graphs

Multigraph

(undirected)



Adjacency Matrix Multigraph

(undirected)

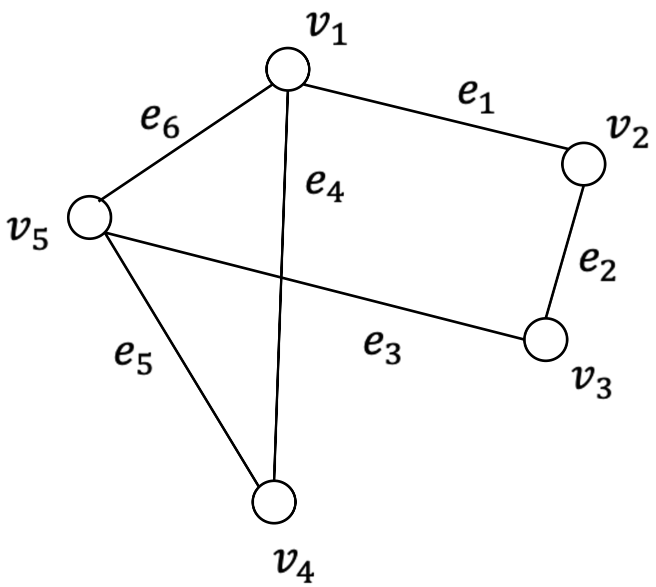
$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 3 & 0 & 1 & 1 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 1 & 0 & 1 & 2 & 0 \end{bmatrix} \end{array}$$

Graph Walk

Definition (Walk). A walk on a graph is an alternating sequence of nodes and edges, starting with a node and ending with a node where each edge is incident with the nodes immediately preceding and following it.

A walk from u to v is called a $u - v$ walk

Definition (Length of a walk). Number of edges in the walk.



$v_1 - v_2$ walk = $(v_1, e_4, v_4, e_5, v_5, e_6, v_1, e_1, v_2)$

Length of a walk = 4

Trail & Path

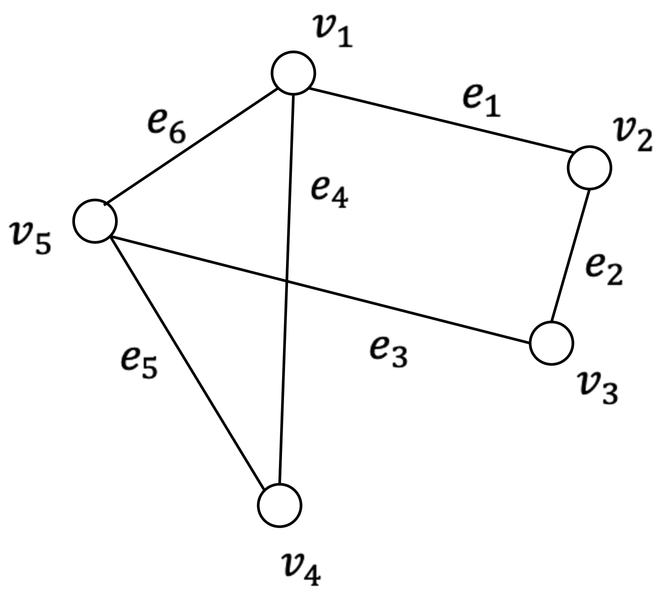
Definition (Trail). A trail is a walk whose edges are distinct.

Trail = $(v_1, e_4, v_4, e_5, v_5, e_6, v_1, e_1, v_2)$

Definition (Path). A path is a walk whose nodes are distinct.

Path = $(v_1, e_4, v_4, e_5, v_5, e_3, v_3, e_2, v_2)$

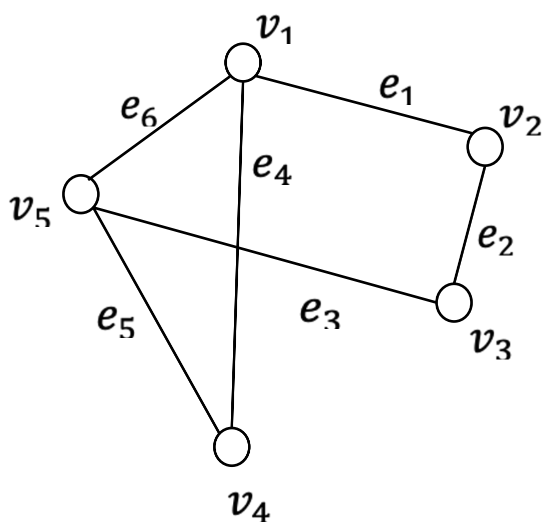
Example.



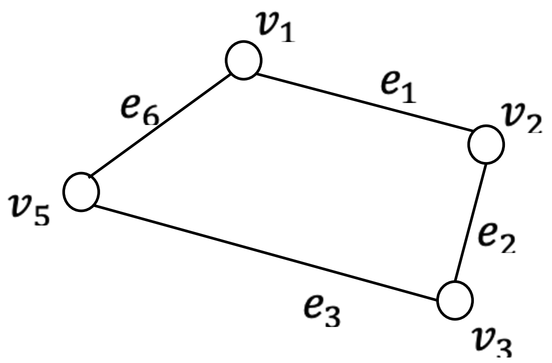
Subgraph

Definition (Subgraph). A subgraph $G' = \{V', E'\}$ of a given graph $G = \{V, E\}$ is a graph formed with a subset of nodes $V' \subset V$ and a subset of edges $E' \subset E$. Furthermore, the subset V' must contain all the nodes involved in the edges in the subset E'.

Example.



$$V = \{v_1, v_2, v_3, v_4, v_5\}$$
$$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$$
$$N = 5$$
$$M = 6$$



$$V' = \{v_1, v_2, v_3, v_5\}$$
$$E' = \{e_1, e_2, e_3, e_6\}$$
$V' \subset V$ and $E' \subset E$

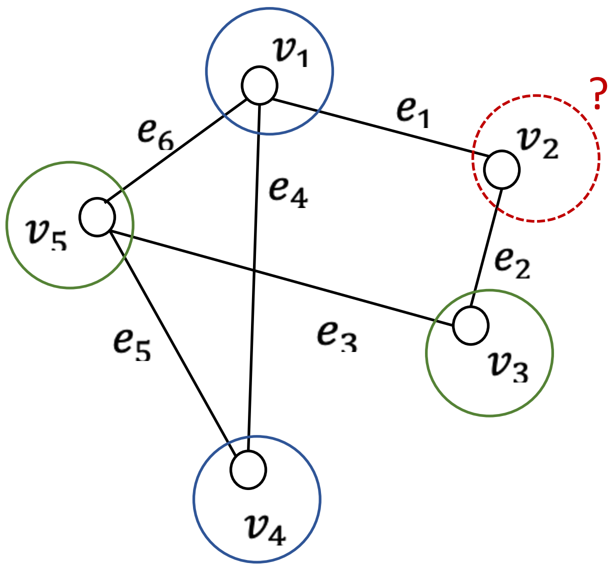
Outline

- Foundations of Graphs
- Graph Representation Learning
- Language Models and Graphs
  - Graph Prompting
  - Aligning LLMs with Graph Representations
  - Fusing LLMs with Graph Representations
  - LLMs for Graphs

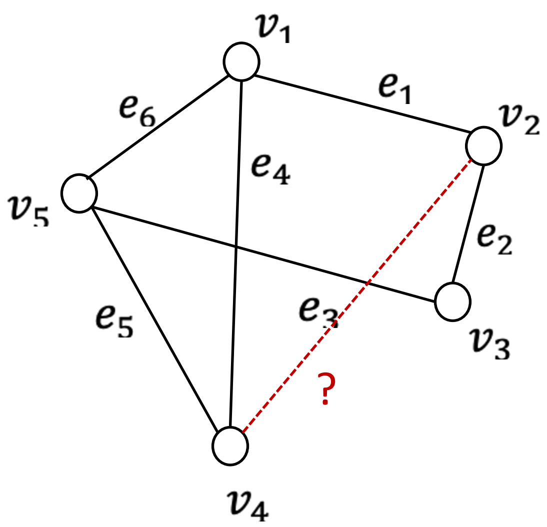
Classic Graph ML Tasks

- **Node classification.** Predict a property of a node
  - Example: Categorize online users / items
- **Link prediction.** Predict whether there are missing links between two nodes
  - Example: Knowledge graph completion
- **Graph classification.** Categorize different graphs
  - Example: Molecule property prediction
- **Clustering.** Detect if nodes form a community
  - Example: Social circle detection
- **Other tasks.**
  - Graph generation: Drug discovery

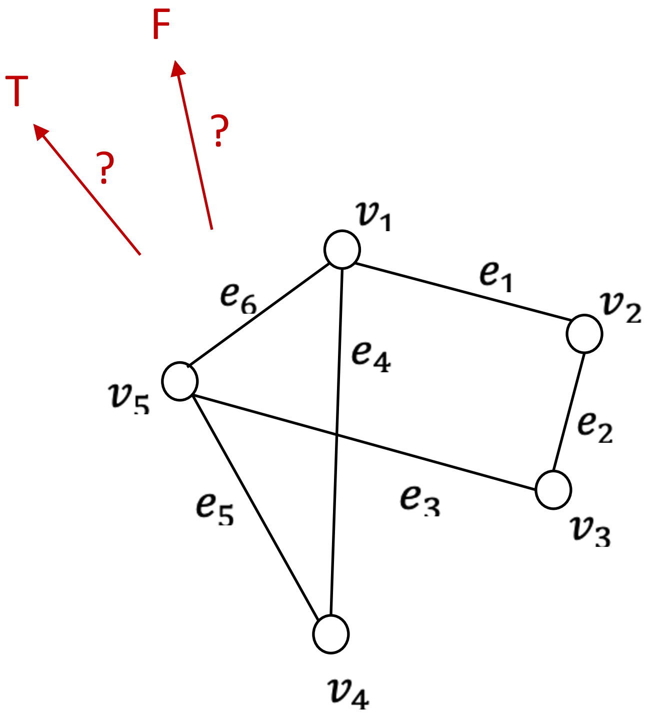
Classic Graph ML Tasks



Node Classification

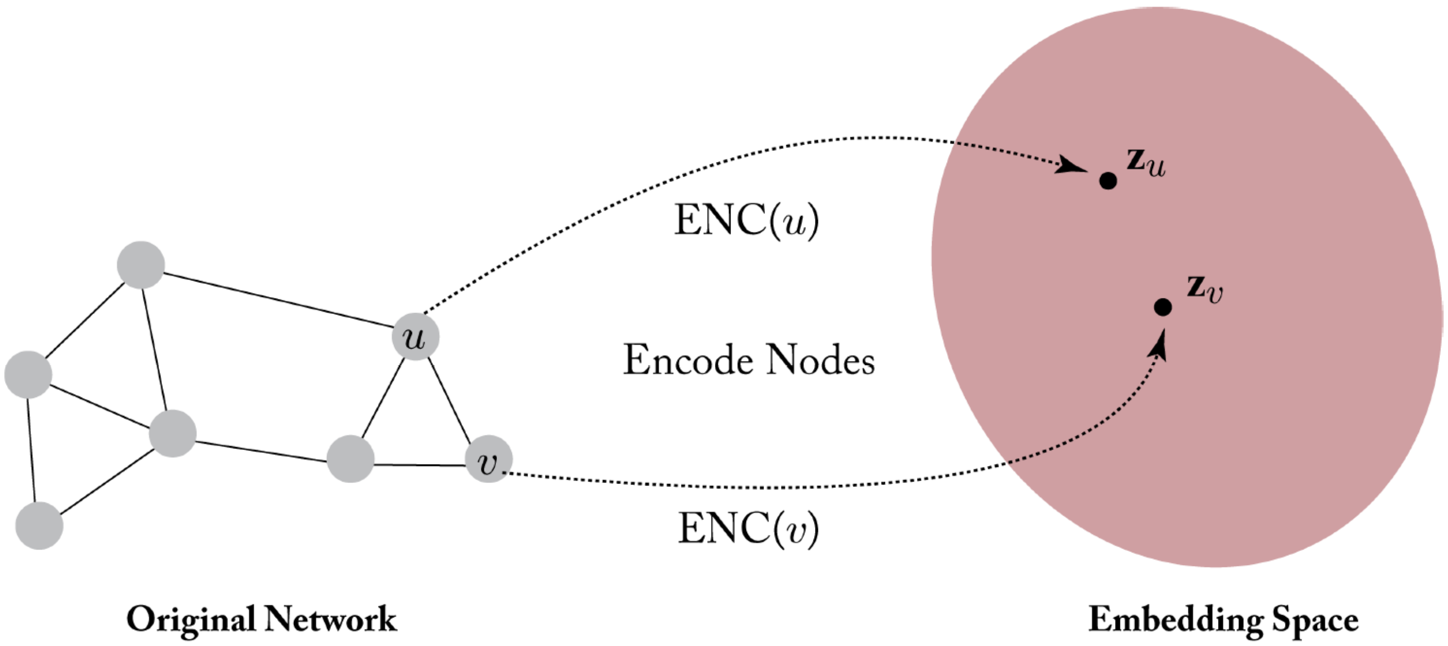


Link Prediction



Graph Classification

Graph Representation Learning



Graph Neural Networks – Quick Overview

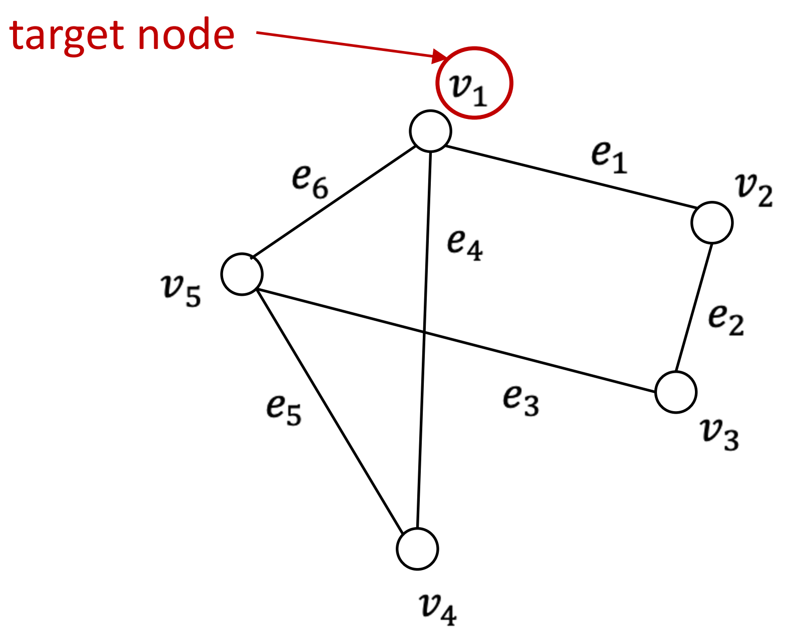
$$m_v^{(l)} = Propagate^{(l)}\left(\left\{h_u^{(l-1)} : u \in \mathcal{N}(v)\right\}\right)$$

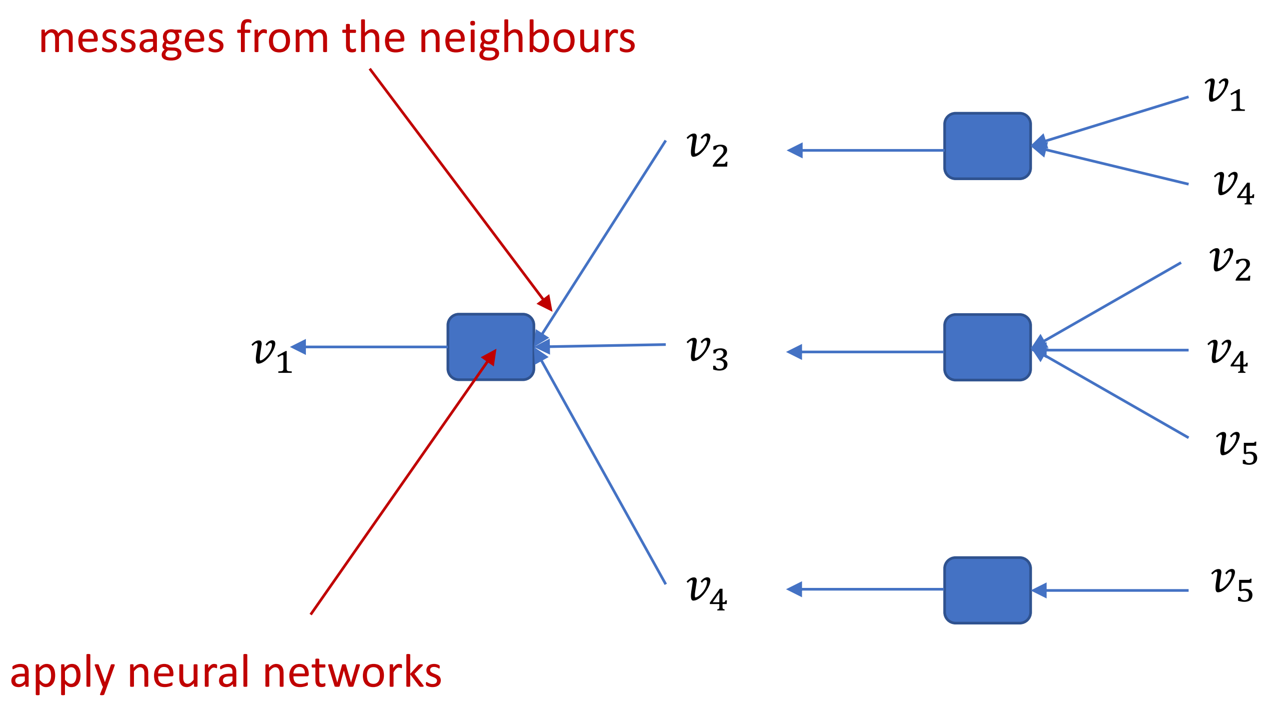
Message passing

$$h_v^{(l)} = Aggregate^{(l)}\left(h_v^{(l-1)}, m_v^{(l)}\right)$$

Feature vector of node v at l-th layer

Graph Neural Networks – Quick Overview





Outline

- **Foundations of Graphs**
- **Graph Representation Learning**
- **Language Models and Graphs**
  - **Graph Prompting**
  - **Aligning LLMs with Graph Representations**
  - **Fusing LLMs with Graph Representations**
  - **LLMs for Graphs**

Graph GPT

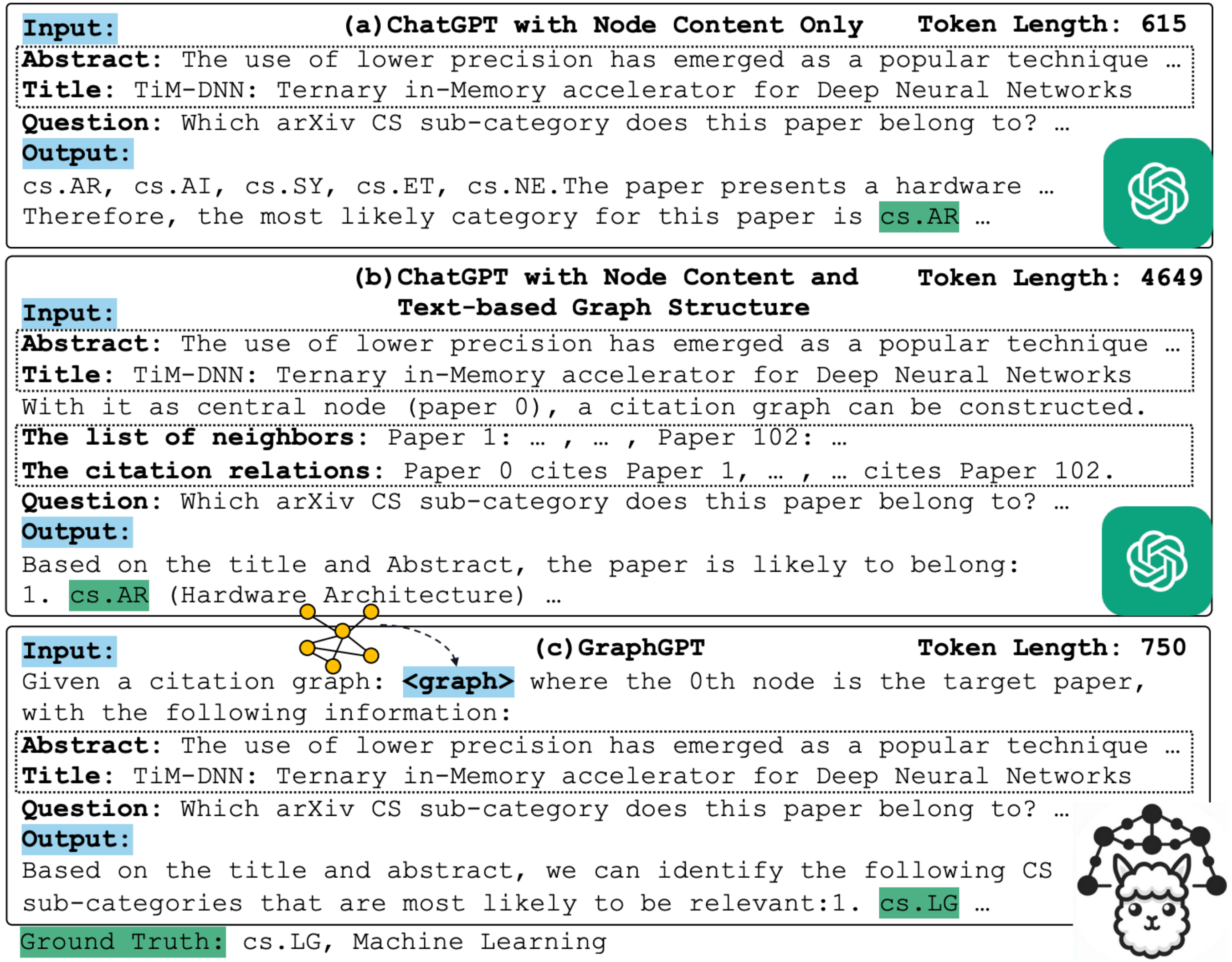
Graph Instruction Tuning for Large Language Models

1. **Feed** graph structures into LLMs.
2. Empower LLMs to **understand** graph structures?
3. Provide LLMs with the ability to **reason** step-by-step for zero-shot complex graph learning tasks.

Graph GPT (1)

Feed graph structures into LLMs?

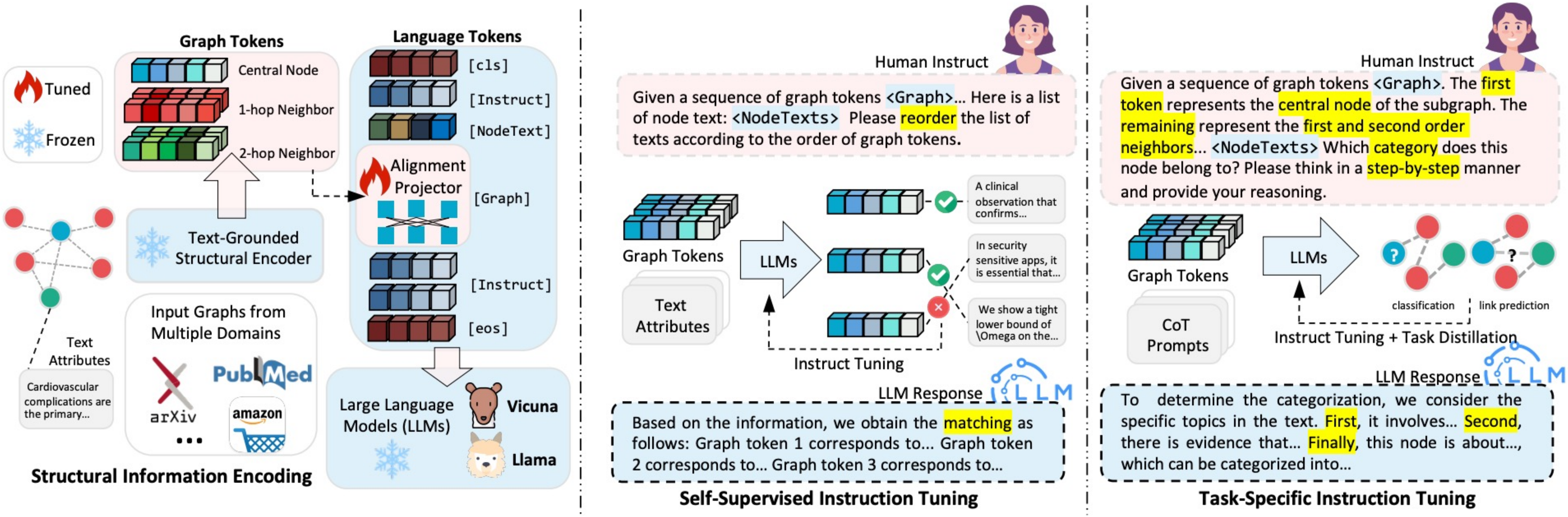
- Without Graph Structure
  - fail when interdisciplinary field
- Text-based Graph
  - fail when interdisciplinary field
    unacceptably long token length
- GraphGPT effectively learning from graph controllable token length



Graph GPT: Overall Architecture

Feed graph structures into LLMs

Graph as a sequence of “Graph Tokens”



Text – Graph Grounding (1)

Graph Encoder

Non-linear activation function

$$\boldsymbol{H}^{(l)} = \sigma\left(\widetilde{\boldsymbol{A}}\boldsymbol{H}^{(l-1)}\boldsymbol{W}\right)$$

Graph representation of the l-th layer

Parameter matrix

Self-loop adjacency matrix

Text – Graph Grounding (1)

Text Structure Alignment

Graph encoder

$$\boldsymbol{H} = f_G(\boldsymbol{X})$$
$$\widehat{\boldsymbol{H}} = norm(\boldsymbol{H})$$

Encoded graph representation

Text encoder

Raw textual content

$$\boldsymbol{T} = f_T(\boldsymbol{C})$$
$$\widehat{\boldsymbol{T}} = norm(\boldsymbol{T})$$

Encoded text representations

Alignment Across Modalities

Trainable temperature parameter

$$\Gamma_1 = (\widehat{\boldsymbol{H}}\,\widehat{\boldsymbol{T}}^T).\exp(\tau)$$

$$\Gamma_2 = (\widehat{\boldsymbol{H}}\,\widehat{\boldsymbol{T}}'^T).\exp(\tau)$$

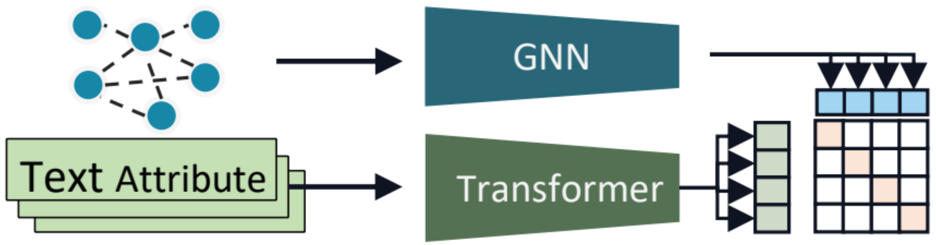
$$\Gamma_3 = (\widehat{\boldsymbol{T}}^T\widehat{\boldsymbol{T}}'^T).\exp(\tau)$$

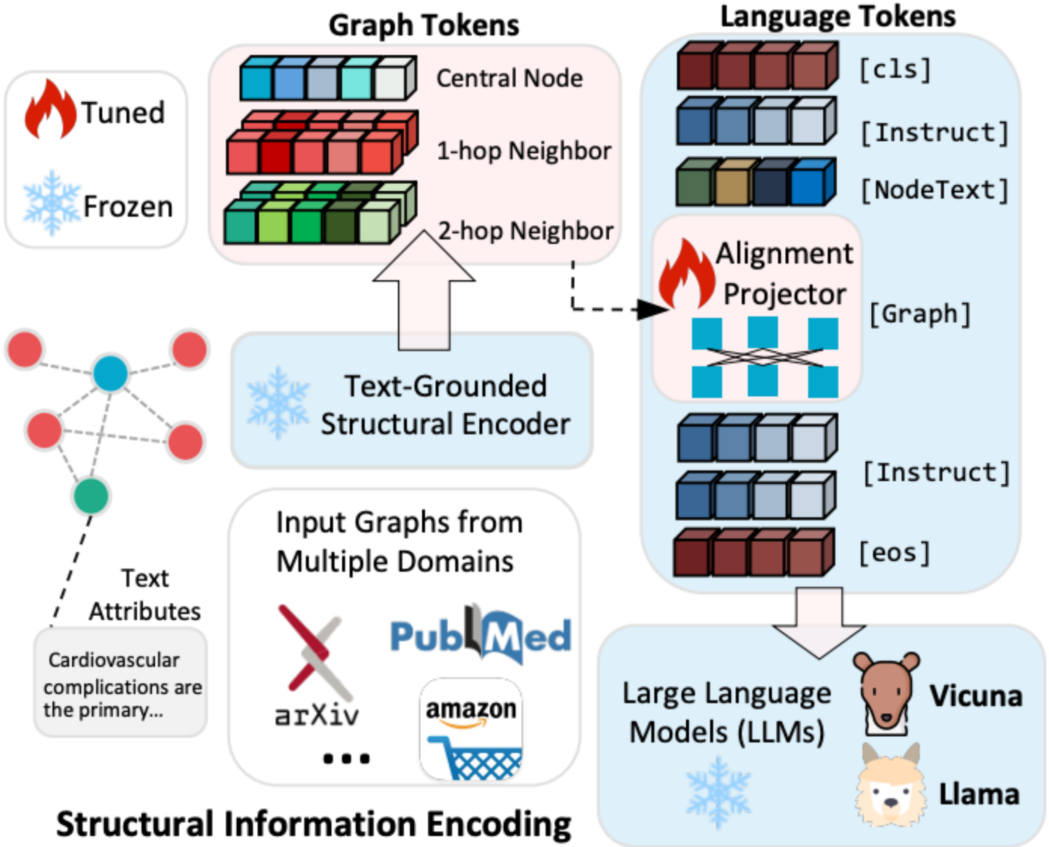
Cross-entropy loss

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

Text – Graph Grounding (1)

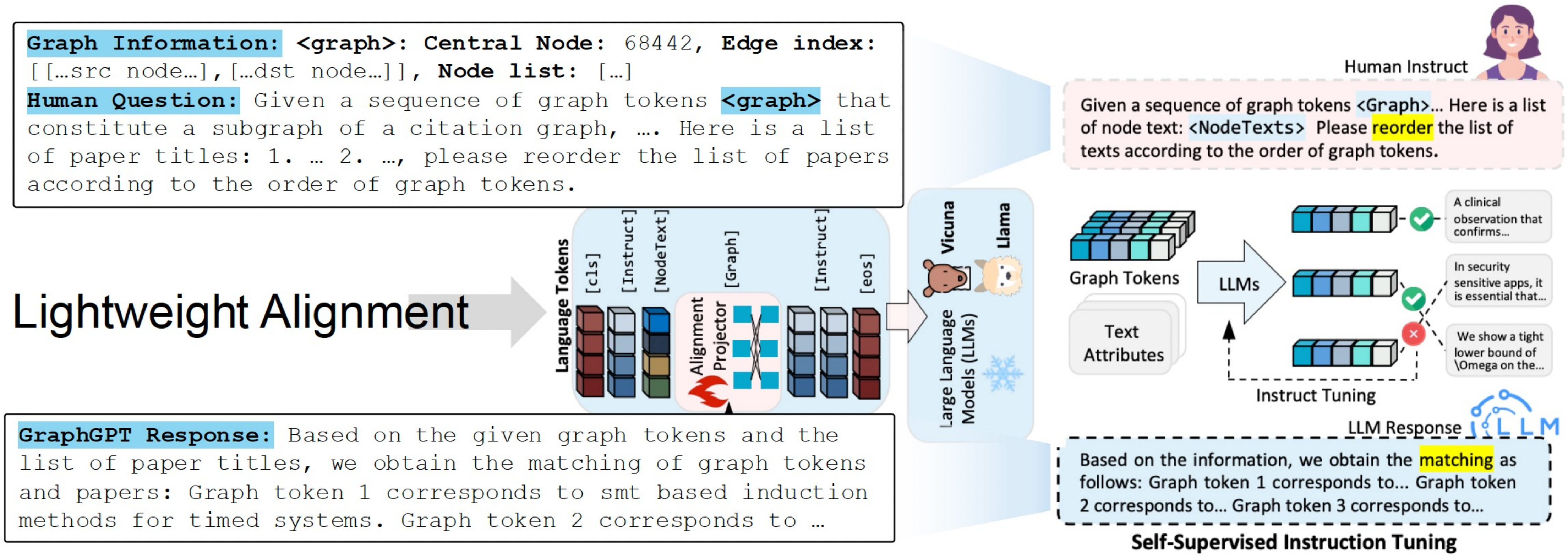
Initializing graph encoder with natural language alignment





Self - Supervised Instruction Tuning (2)

Let LLM match the graph tokens with the corresponding natural language content in the prompt



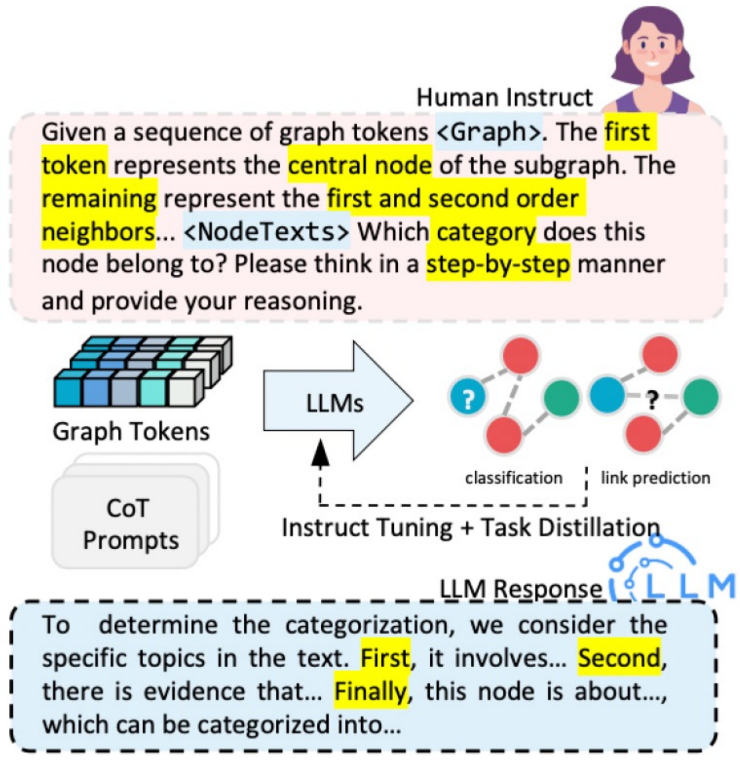
Task Specific Instruction Tuning (2)

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".

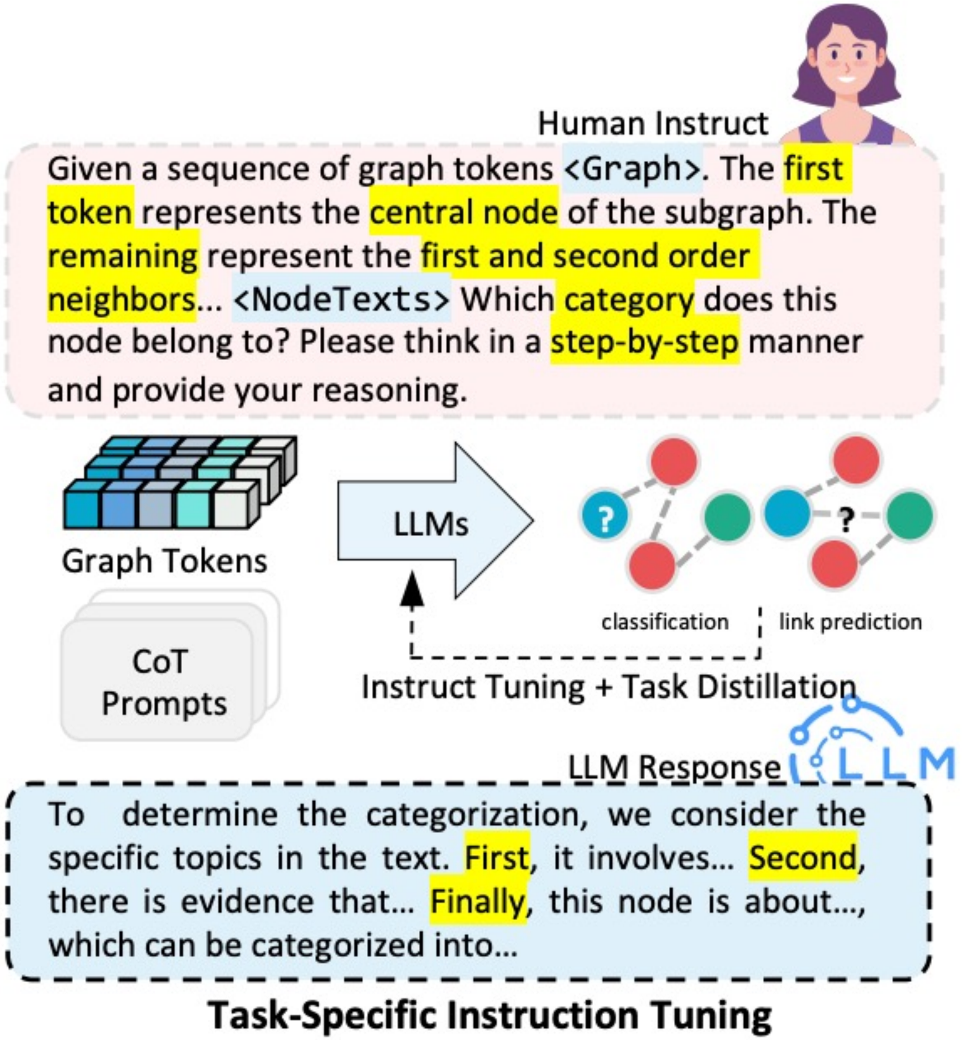
GraphGPT Response: cs.IT, cs.LG, cs.SP, cs.CV, cs.NA. The paper discusses the Restricted Isometry …. So, it is likely to belong to cs.IT…



Task-Specific Instruction Tuning

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

Distilling reasoning capabilities from a powerful model (ChatGPT) through the CoT



Experimental Results

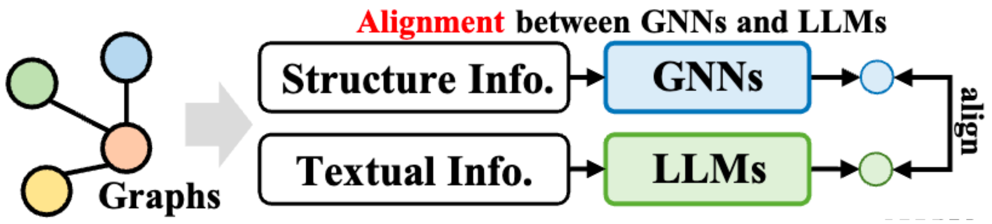
Dataset	Arxiv-Arxiv		Arxiv-PubMed		Arxiv-Cora		(Arxiv+PubMed)-Cora		(Arxiv+PubMed)-Arxiv	
Model	Accuracy	Macro-F1	acc	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
MLP	0.5179	0.2536	0.3940	0.1885	0.0258	0.0037	0.0220	0.0006	0.2127	0.0145
GraphSAGE	0.5480	0.3290	0.3950	0.1939	0.0328	0.0132	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	$2.26e^{-9}$	$1.56e^{-10}$	$2.22e^{-7}$	$1.55e^{-9}$	$1.04e^{-9}$	$9.96e^{-6}$	$7.62e^{-8}$	$1.97e^{-7}$	$1.5e^{-13}$	$4.63e^{-6}$

Alignment b/w GNN & LLMs

- Align the feature spaces of GNNs and LLMs
- GNNs and LLMs process different types of data
- Distinct feature spaces in GNNs and LLMs

Methods

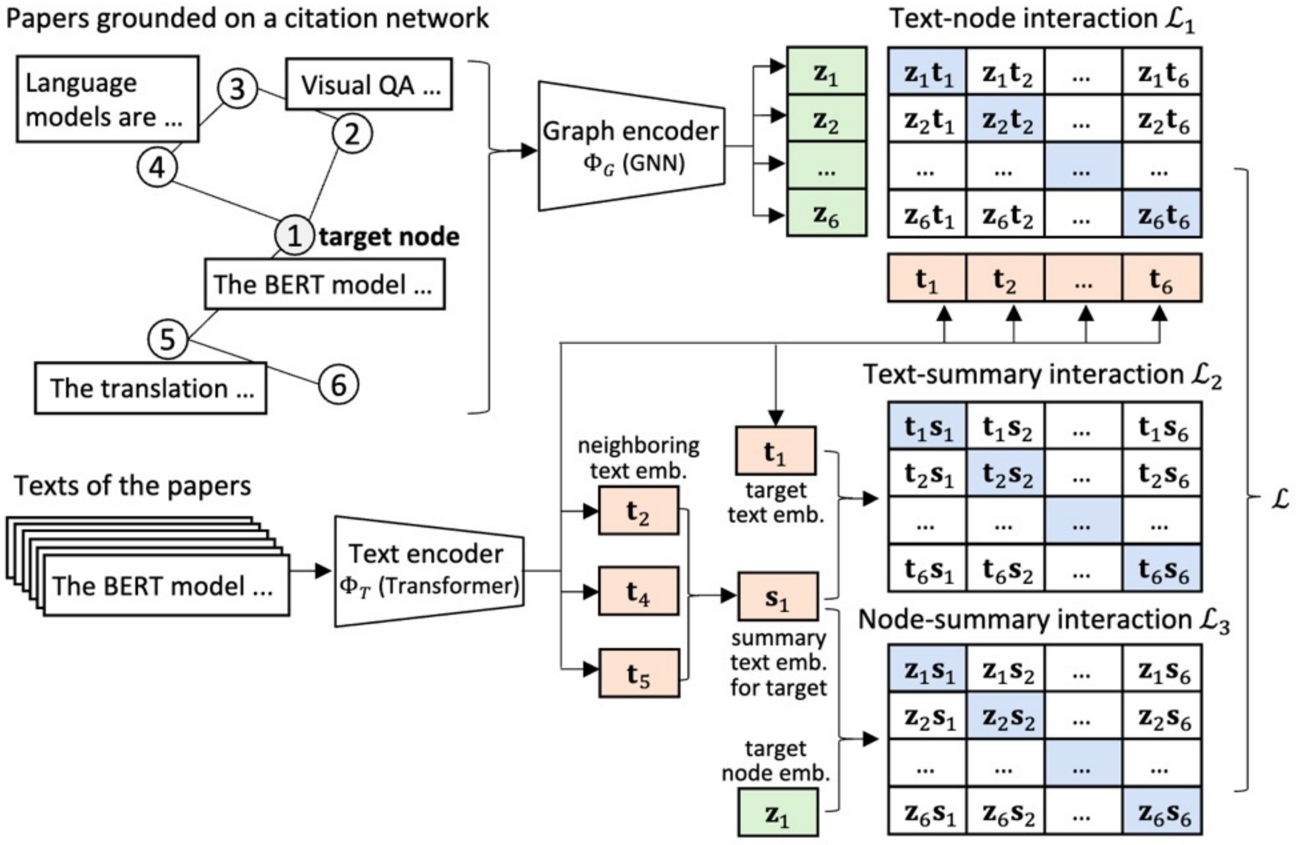
- Contrastive Learning
- EM Iterative Training



Alignment b/w GNN & LLMs

Methods considering contrastive Alignment.

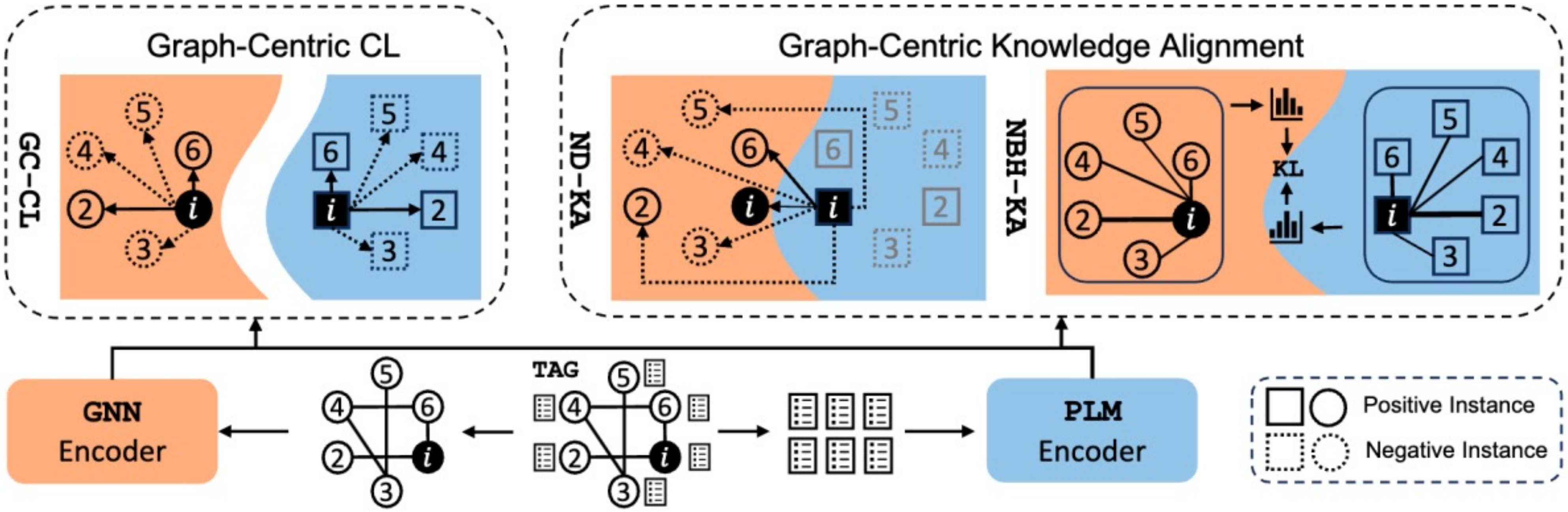
G2P2 proposes text-node, text-summary, and node-summary alignment and improves the text classification performance in low-resource environments



Source: https://arxiv.org/pdf/2305.03324

Alignment b/w GNN & LLMs

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



Source: https://aclanthology.org/2023.findings-emnlp.181/

Alignment b/w GNN & LLMs

- **Graph and Language Learning by Expectation Minimization**

- **Language Models** for Node Classification

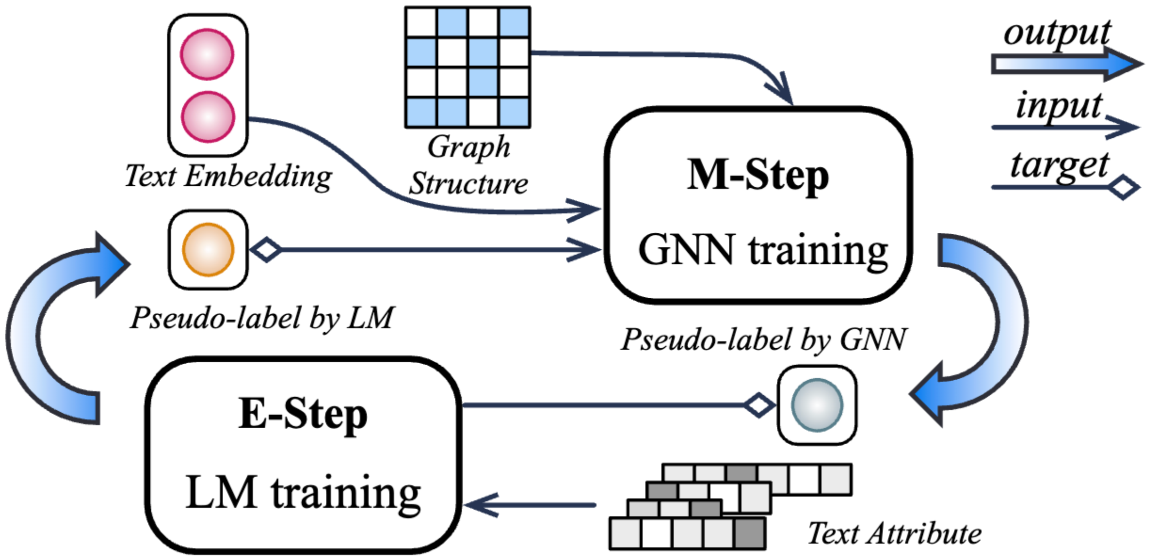
$$\mathbf{h}_n = \mathrm{SeqEnc}_{\theta_1}(\mathbf{s}_n)$$

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

- Graph Neural Networks for Node Classification

$$\mathbf{h}_n^{(l)} = \sigma(\mathrm{AGG}_\phi(\mathrm{MSG}_\phi(\mathbf{h}_{\mathrm{NB}(n)}^{(l-1)}), A))$$

$$p_\phi(\mathbf{y}_n|A) = \mathrm{Cat}(\mathbf{y}_n \mid \mathrm{softmax}(\mathbf{h}_n^{(L)}))$$



Source: https://openreview.net/pdf?id=q0nmYciuuZN

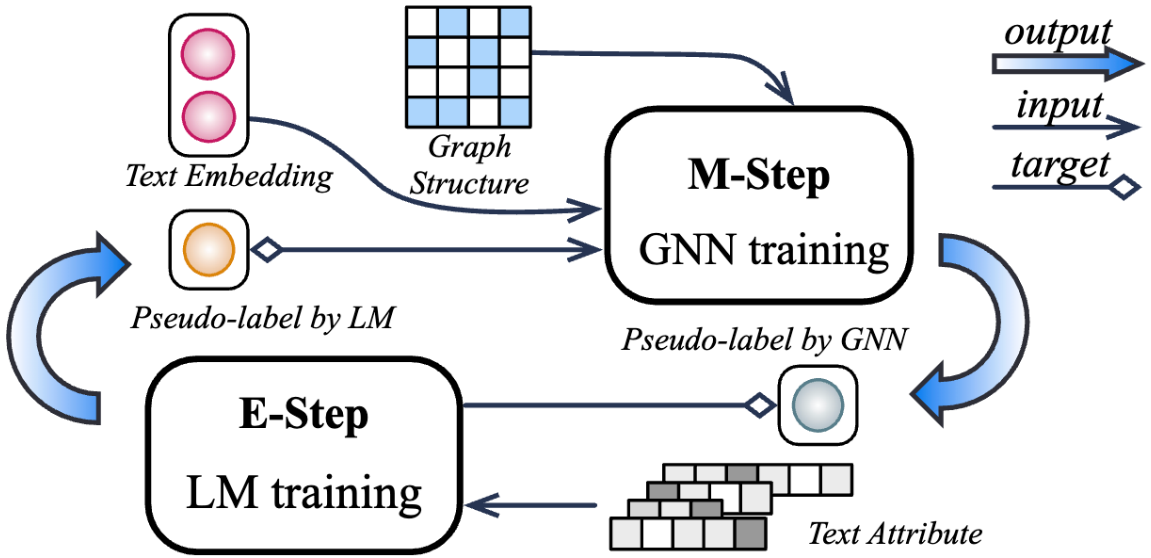
Alignment b/w GNN & LLMs

- **Graph and Language Learning by Expectation Minimization**

- **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



Alignment b/w GNN & LLMs

- **Graph and Language Learning by Expectation Minimization**

- **GLEM** achieves better performance improvements with various GNN backbones

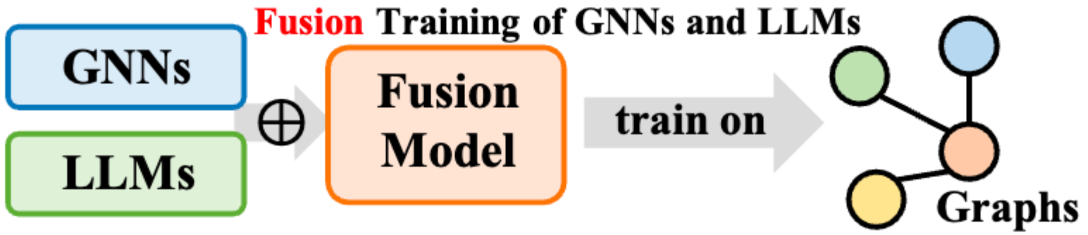
Datasets	Methods		GNN X_{OGB}	X_{GIANT}	X_{PLM}	GLEM-GNN	G ↑	LM LM-Ft	GLEM-LM	L↑
Arxiv	GCN	*val*	73.00 ± 0.17	74.89 ± 0.17	47.56 ± 1.91	76.86 ± 0.19	3.86	75.27 ± 0.09	76.17 ± 0.47	0.90
		test	71.74 ± 0.29	73.29 ± 0.10	48.19 ± 1.47	<u>75.93 ± 0.19</u>	4.19	74.13 ± 0.04	75.71 ± 0.24	1.58
	SAGE	*val*	72.77 ± 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

LLM & GNN Fusion

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

Methods

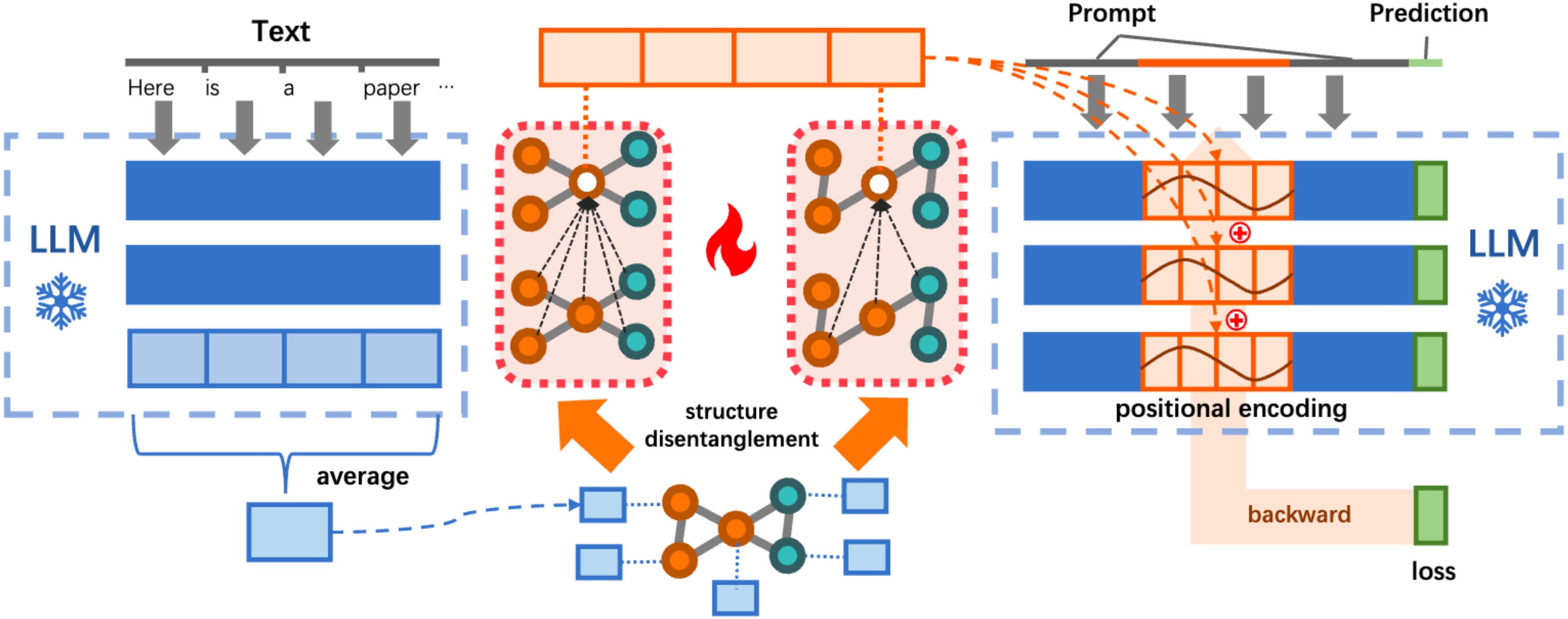
Fuse Transformer Layer with Graph Neural Network Layer



LLM & GNN Fusion

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 citation network and e-commerce graph tasks.



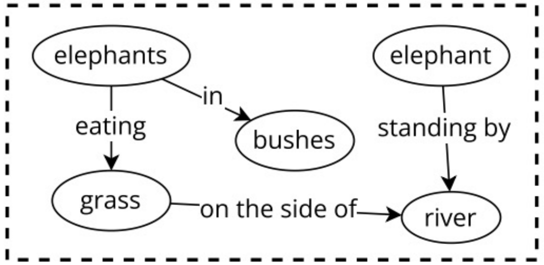
Source: https://arxiv.org/abs/2310.18152

LLMs as Agents for Graphs

Build LLM based Agents for Graph Tasks

- Model is capable of solving problems step by step by utilizing external tools and interacting with the graph

G-Retriever – Q&A over graphs



What kind of animal is in the bushes?
Answer the question and list the nodes or edges from the graph that support your answer

User Query

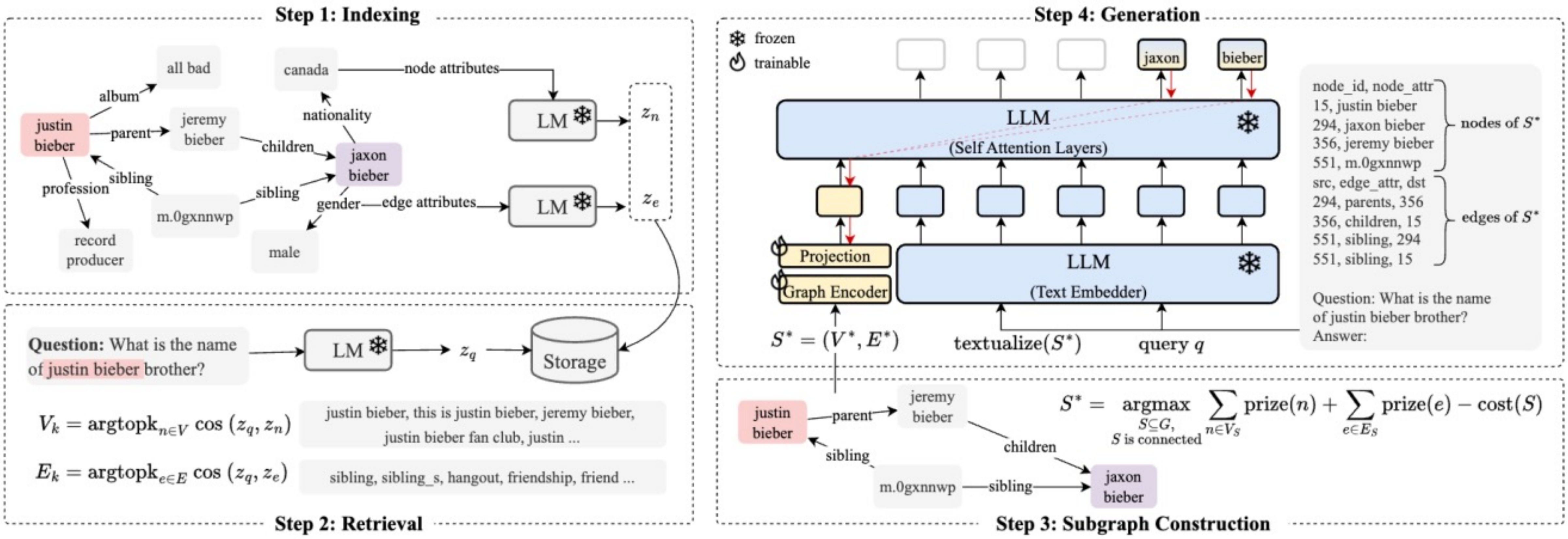
[Incorrect response with hallucinated nodes and edges]
The animal in the bushes is a deer.
Nodes:
* Deer (node 1)
* Bushes (node 2)
Edges:
* Deer → Bushes (edge 1)
* Deer → Grass (edge 2)
* Bushes → Grass (edge 3)

LLM w/ Graph Prompt Tuning

[Correct response with accurate node and edge references]
Based on the information provided in the graph, the animal in the bushes is an elephant. The nodes that support this answer are:
* Node 1: The elephant is mentioned in the first line of the graph.
The edges that support this answer are:
* The edge connects the elephant to the bushes, indicating that the elephant is in the bushes.
Therefore, the answer to the question is: The animal in the bushes is an elephant.

G-Retriever

G-Retriever



Step 1: Indexing
all bad
canada
node attributes
album
nationality
justin bieber
parent
jeremy bieber
children
jaxon bieber
LM
z_n
sibling
profession
sibling
m.0gxnnwp
gender
edge attributes
z_e
record producer
male
Question: What is the name of justin bieber brother?
z_q
Storage
$V_k = \text{argtopk}_{n \in V} \cos(z_q, z_n)$
justin bieber, this is justin bieber, jeremy bieber, justin bieber fan club, justin ...
$E_k = \text{argtopk}_{e \in E} \cos(z_q, z_e)$
sibling, sibling_s, hangout, friendship, friend ...
Step 2: Retrieval
Step 4: Generation
frozen
trainable
jaxon
bieber
LLM
(Self Attention Layers)
Projection
Graph Encoder
LLM
(Text Embedder)
$S^* = (V^*, E^*)$
textualize(S^*)
query q
node_id, node_attr
15, justin bieber
294, jaxon bieber
356, jeremy bieber
551, m.0gxnnwp
nodes of S^*
src, edge_attr, dst
294, parents, 356
356, children, 15
551, sibling, 294
551, sibling, 15
edges of S^*
Question: What is the name of justin bieber brother?
Answer:
$S^* = \underset{S \subseteq G, \ S \text{ is connected}}{\text{argmax}} \sum_{n \in V_S} \text{prize}(n) + \sum_{e \in E_S} \text{prize}(e) - \text{cost}(S)$
Step 3: Subgraph Construction

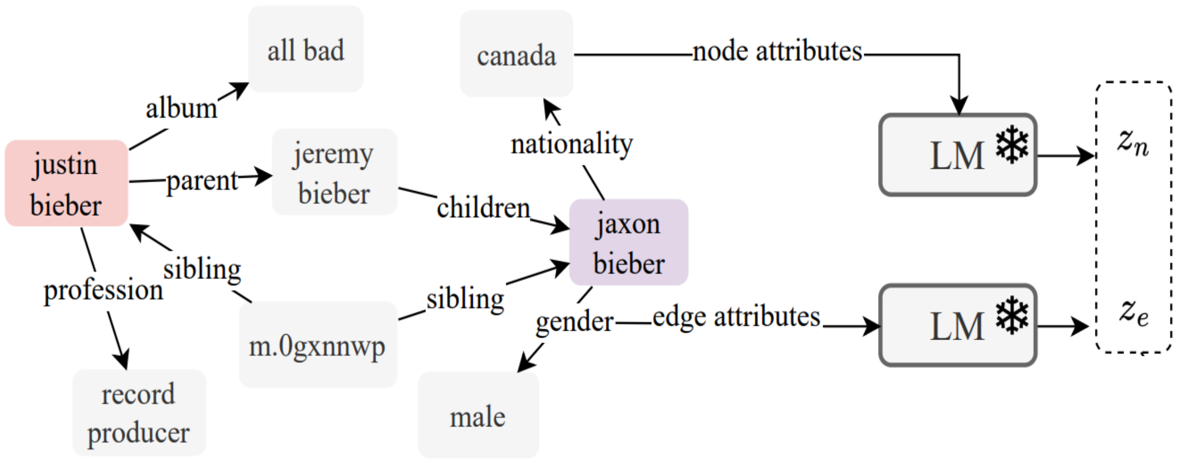
G-Retriever - Indexing

Sequential text of node n

Length of text associated with n

$$x_n \in D^{L_n}$$

Vocabulary



Representations of node n

$$z_n = LM(x_n) \in \mathbb{R}^d$$

Dimension of output vector

Pre-trained Language Model

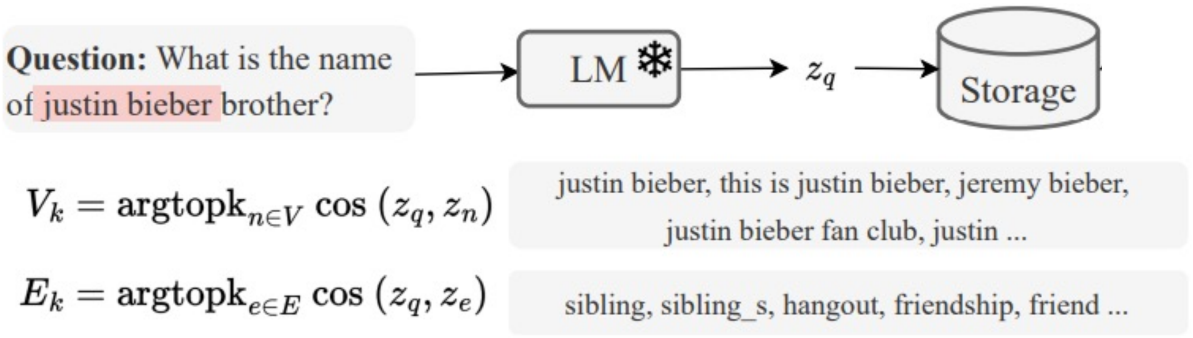
G-Retriever - Retrieval

Representations of the question q

$$z_q = LM(x_q) \in \mathbb{R}^d$$

Dimension of output vector

Pre-trained Language Model



$$V_k = argtopk_{n \in V} \cos(z_q, z_n)$$

$$E_k = argtopk_{e \in E} \cos(z_q, z_e)$$

G-Retriever – Subgraph Construction

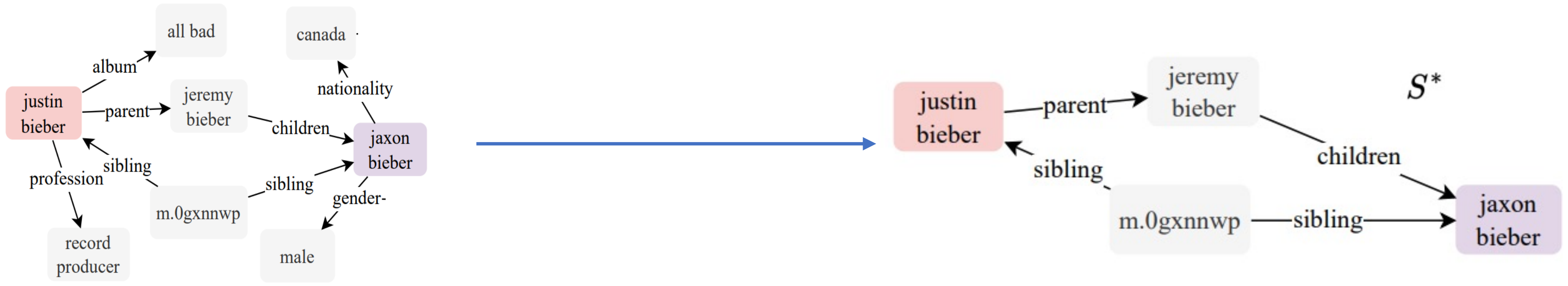
Prize-Collecting Steiner Tree (PCST)

Find a connected subgraph that

- maximizes the total prize values of its nodes
- While minimizing the total costs of its edges
- higher prize values to nodes and edges more relevant to the query

$$prize(n) = \begin{cases} k - i, & if\ n\ \in\ V_k\ and\ n\ is\ the\ top\ i\ node, \\ 0, & otherwise \end{cases}$$

$$S^* =\ argmax_{s \subseteq G} \sum_{n \in V_S} prize(n) + \sum_{e \in E_S} prize(e)\ - cost(S)$$



G-Retriever – Answer Generation

Graph Encoder.

$$h_g = POOL\left(GNN_{\phi_1}(S^*)\right) \in \mathbb{R}^{d_g}$$

Mean pooling operation

retrieved subgraph

dimension of graph encoder

Projection Layer.

$$\hat{h}_g = MLP_{\phi_2}(h_g) \in \mathbb{R}^{d_l}$$

dimension LLM's hidden embedding

Align graph token with the vector space of LLM

G-Retriever – Answer Generation

Text Embedder.

$$h_t = TextEmbedder\left([textualize\ (S^*); x_q]\right) \in \mathbb{R}^{L\ x\ d_l}$$

concatenation

number of tokens

LLM Generation with Graph Prompt Tuning.

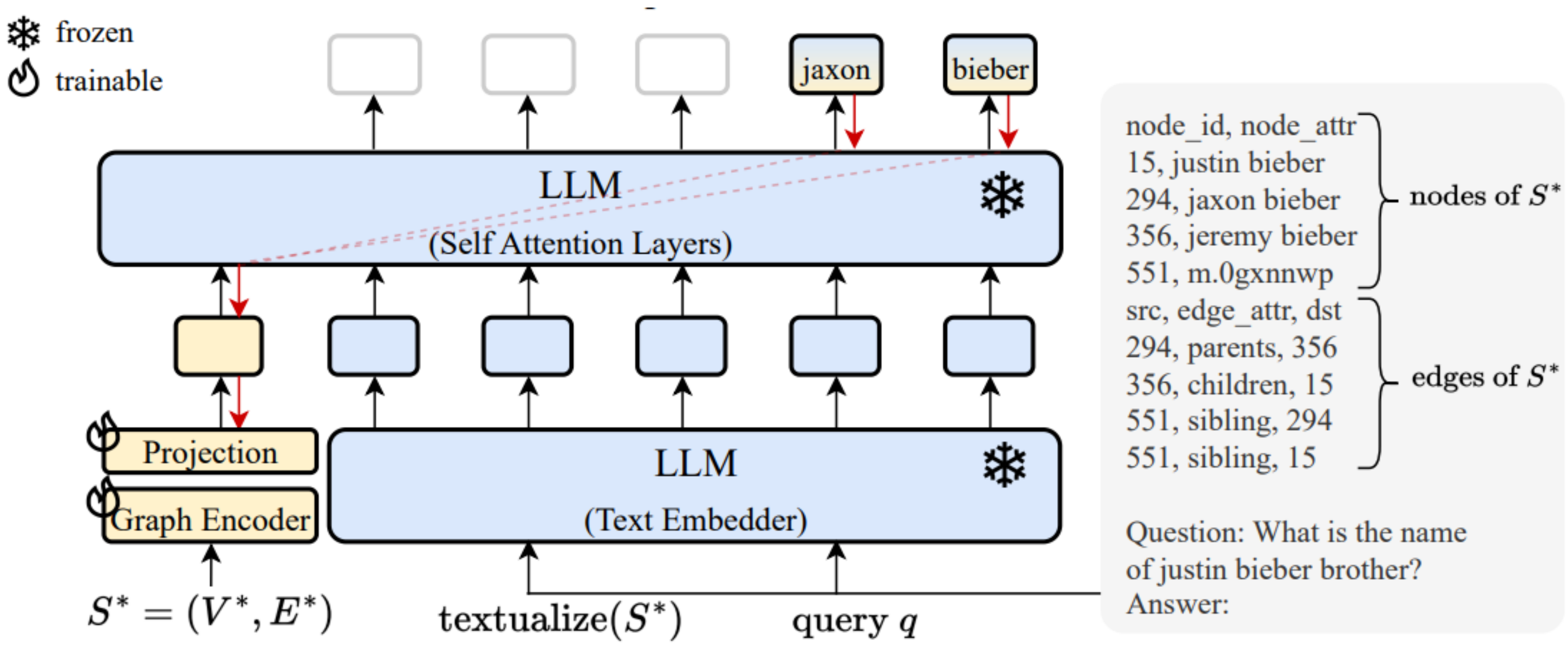
$$p_{\theta,\phi_1,\phi_2}\left(Y|S^*, x_q\right) = \prod_{i=1}^{r} p_{\theta,\phi_1,\phi_2}\ (y_i|y_{<i}, [\hat{h}_g; h_t])$$

output

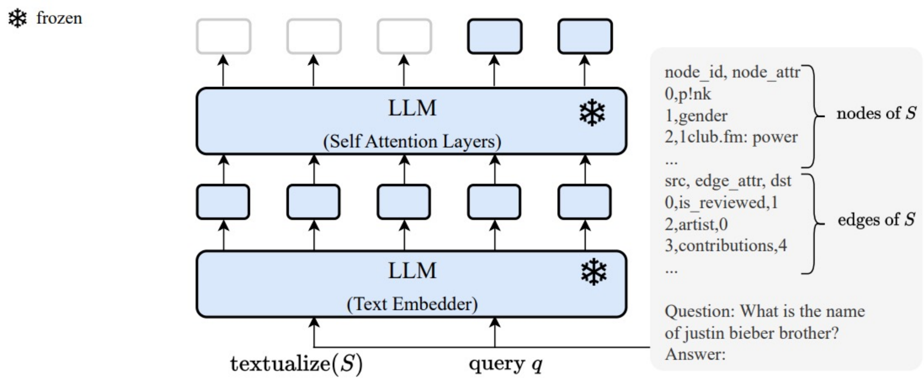
parameter

Soft prompt

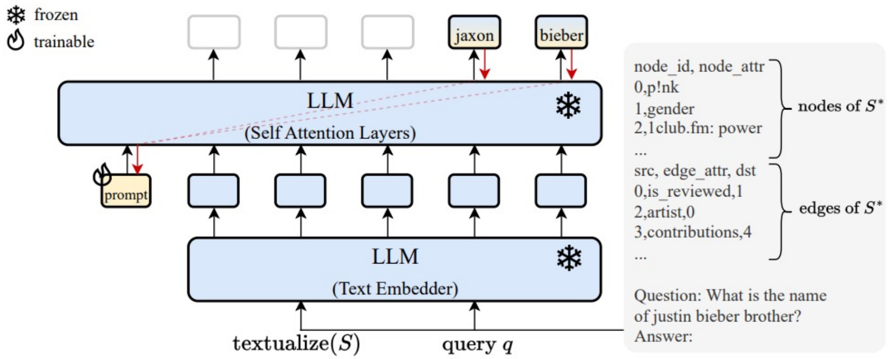
G-Retriever – Answer Generation



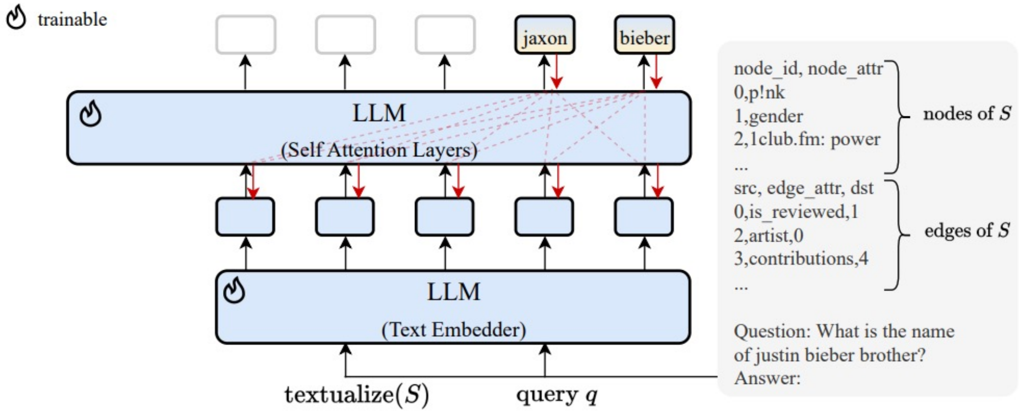
G-Retriever – Variations



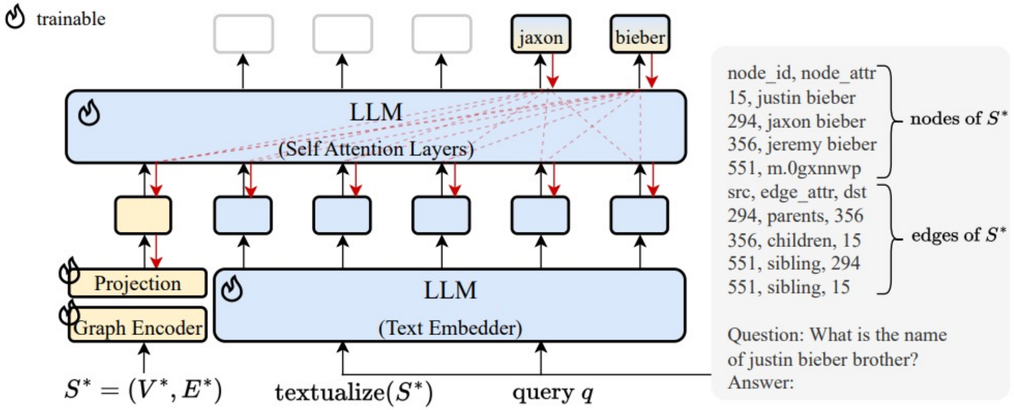
Inference-Only



Prompt Tuning



LoRA



G-retriever with LoRA

G-Retriever – Results

Setting	Method	ExplaGraphs	SceneGraphs	WebQSP
Inference-only	Zero-shot	0.5650	0.3974	41.06
	Zero-CoT [18]	0.5704	0.5260	51.30
	CoT-BAG [44]	0.5794	0.5680	39.60
	KAPING [1]	0.6227	0.4375	52.64
Frozen LLM w/ PT	Prompt tuning	0.5763 ± 0.0243	0.6341 ± 0.0024	48.34 ± 0.64
	GraphToken [31]	0.8508 ± 0.0551	0.4903 ± 0.0105	57.05 ± 0.74
	G-Retriever	0.8516 ± 0.0092	<u>0.8131 ± 0.0162</u>	<u>70.49 ± 1.21</u>
	$\Delta_{\text{Prompt tuning}}$	↑ 47.77%	↑ 28.23%	↑ 45.81%
Tuned LLM	LoRA	<u>0.8538 ± 0.0353</u>	0.7862 ± 0.0031	66.03 ± 0.47
	G-Retriever w/ LoRA	**0.8705 ± 0.0329**	**0.8683 ± 0.0072**	**73.79 ± 0.70**
	Δ_{LoRA}	↑ 1.95%	↑ 11.74%	↑ 10.44%

Outline

- **Foundations of Graphs**
- **Graph Representation Learning**
- **Language Models and Graphs**
  - **Graph Prompting**
  - **Aligning LLMs with Graph Representations**
  - **Fusing LLMs with Graph Representations**
  - **LLMs for Graphs**