GRAPH NEURAL PROMPTING WITH LARGE LANGUAGE MODELS

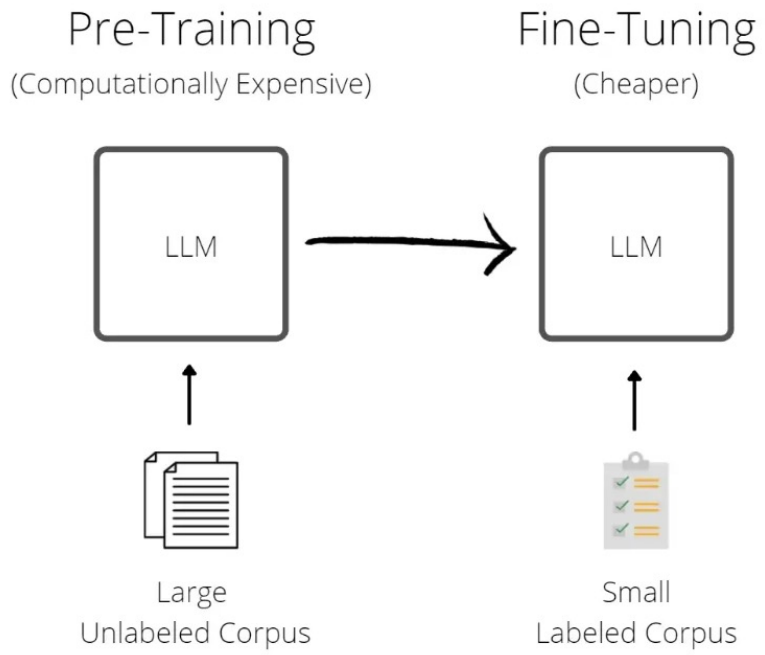
BY: TIAN ET AL.

3/25/25

Presenter: Gurjot Singh

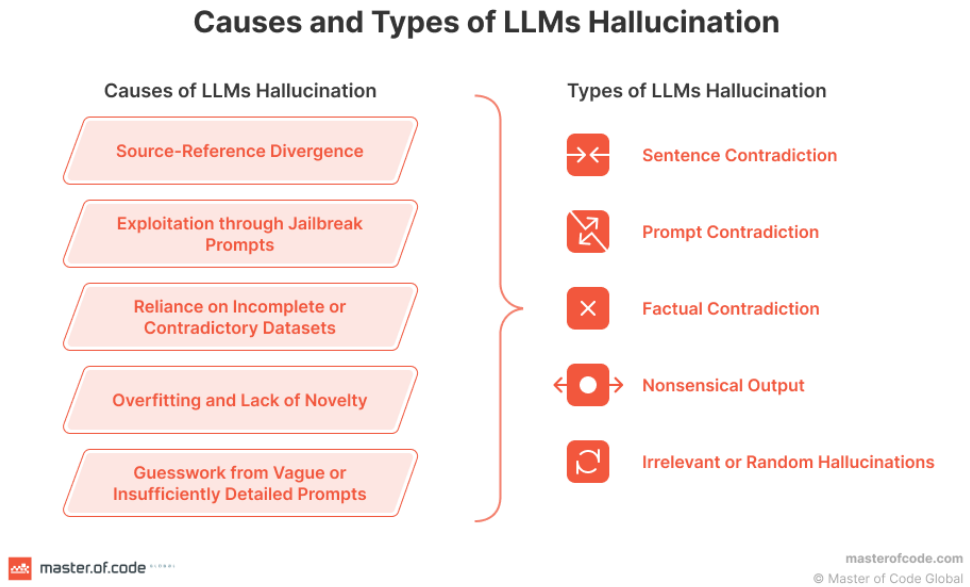
Training of Large Language Models (LLMs)

- **Massive Data & Self-supervised Learning**:
  - LLMs are trained on huge text corpora using self-supervised objectives (e. g. , predicting next tokens).
  - They leverage the transformer architecture to capture long-range dependencies.
- **Pre-training & Fine-tuning**:
  - Pre-training establishes a broad understanding of language.
  - Fine-tuning or in-context learning adapts models to specific tasks.



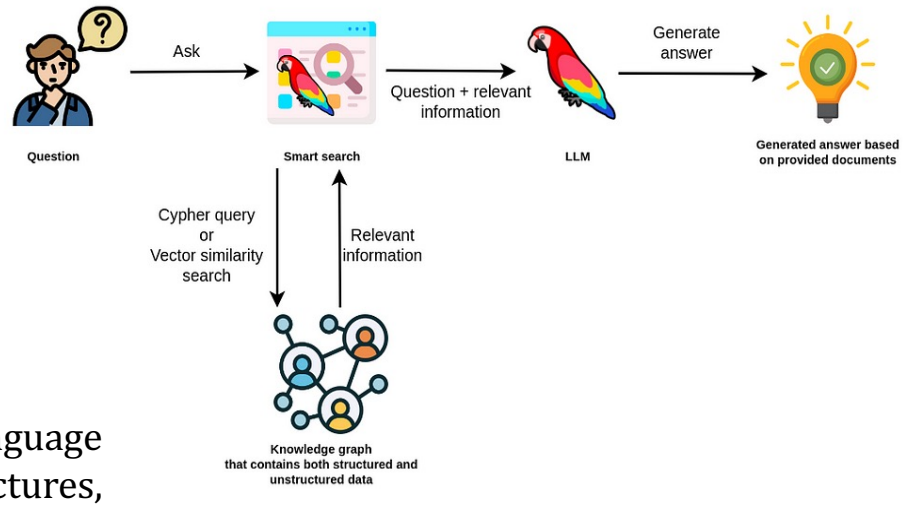
Challenges in LLMs

- However, one persistent issue is that LLMs sometimes struggle to capture and retrieve grounded factual knowledge accurately. This limitation can result in answers that are plausible but not reliably supported by external factual sources. (Hallucinations)



Solution to Hallucinations

- **Knowledge Graphs**
  - KGs systematically store a large number of facts (structured as entities and relations) that can potentially remedy this shortfall.



Prior work has attempted to merge KGs and language models via joint training or specialized architectures, but this approach can be computationally intensive and hard to scale to very large pre-trained models.

Question

- Can we enhance pre–trained LLMs using external grounded knowledge from KGs without needing to fine–tune all the parameters?

Outline

- Introduction & Background
- Contributions
- Methodology
- Results & Analysis
- Limitations
- Conclusion

Key Contributions

- **Novel Graph Neural Prompting (GNP) Method**:
  The paper introduces GNP as a plug-and-play approach to integrate structured knowledge from knowledge graphs into pre-trained LLMs without retraining the entire model.
- **Innovative Architectural Design**:
  GNP combines a graph neural network encoder, cross-modality pooling, and a domain projector—along with a self-supervised link prediction objective—to transform retrieved subgraph information into effective soft prompts for LLMs.
- **Empirical Validation Across Domains**:
  Extensive experiments on both commonsense and biomedical reasoning tasks show that GNP significantly improves performance in various settings (LLM frozen and LLM tuned), outperforming multiple baselines and even matching or surpassing full fine-tuning in many cases.

Preliminary

Definition 1 (Knowledge Graph). A knowledge graph is defined as

$$G = (E, R, T),$$

where:

- E is the set of entities,
- R is the set of relations, and
- T is the collection of fact triples, defined as

$$T = \{(e_h, r, e_t) \mid e_h, e_t \in E,\ r \in R\}.$$

Preliminary

A *fact triplet* is a structured element of a knowledge graph, typically represented as:

$$(e_h, r, e_t)$$

where:

- e_h is the *head entity* (the subject),
- r is the *relation* (or predicate) that connects the entities, and
- e_t is the *tail entity* (the object).

For example, the fact that "Barack Obama is the president of the United States" can be represented as:

$$(\text{Barack Obama}, \text{isPresidentOf}, \text{United States}).$$

These triplets are the fundamental building blocks of a knowledge graph, as they encode individual facts in a structured and consistent manner.

Preliminary

Problem 1 (Multiple Choice Question Answering). Given:

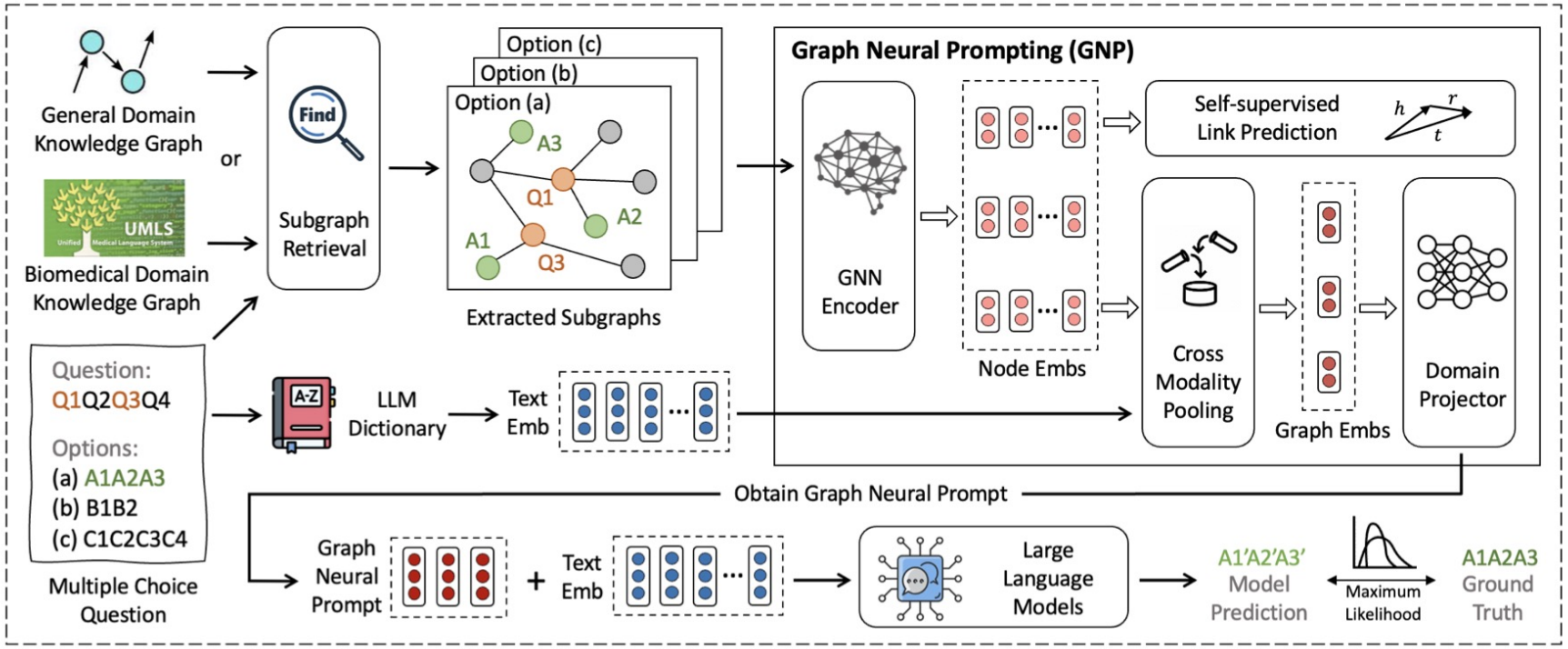
- a question Q,
- a set of answer options
$$A = \{a_1, a_2, \ldots, a_K\},$$
- and an optional context C (depending on whether it is an open-book or close-book setting),

the task is to design a model F_θ with parameters θ that selects the correct answer y from the set A. This can be expressed as:

$$y = F_\theta(Q, C, A),$$

where the ground truth y is the correct answer among the K options.

Methodology



Methodology

This section details the process by which the Graph Neural Prompting (GNP) is constructed and integrated into a pre-trained LLM. We describe each component step-by-step.

Prompting LLMs for Question Answering

Given a question Q, an optional context C, and answer options

$$A = \{a_1, a_2, \ldots, a_K\},$$

the input text tokens are constructed by concatenating C, Q, and A into a sequence X. A prompt P (which can be a soft trainable prompt) is then prepended:

$$\text{Input to LLM} = [P, X].$$

The model generates the output y as:

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

The training objective (using maximum likelihood) is:

$$L_{\text{llm}} = -\log p(y \mid X, \theta).$$

Methodology

Question: "What is the best way to guess a baby's eye color?"

Answer Options:
- (a) Use a random guess
- (b) Consult an optometrist
- (c) Use genetic testing
- (d) Look at family history

The goal is for the model to choose option (d), which is correct because family history often plays a key role in determining eye color.

Methodology

- **Sequence X**:
  "Question: What is the best way to guess a baby's eye color? Options: (a) Use a random guess, (b) Consult an optometrist, (c) Use genetic testing, (d) Look at family history. "
- **Input to LLM**:
  [P,X] where P is the learned soft prompt that will carry structured knowledge from the graph.

The LLM then produces an output y based on this input. The training uses a maximum likelihood loss:

$$L_{llm} = -\log p(y|X,\theta).$$

Methodology

Subgraph Retrieval

For each question and its corresponding answer options, the following steps are performed:

1. **Entity Linking:** Match tokens in X with entities in the knowledge graph G to obtain a set of entities E_{match}.
2. **Subgraph Extraction:** Retrieve a subgraph $\vec{G}$ from G that includes E_{match} and their two-hop neighbors.

Methodology

- **Identified Entities**:
  From the question, entities such as "baby", "eye color", and "family history" are extracted.

- **Retrieved Subgraph $G^{\rightarrow}G$**:
  The subgraph might include:
  - An edge: ("baby","relatedTo","family")
  - An edge: ("family","influences","eye color")

This subgraph contains relevant factual knowledge that can support the reasoning process.

Methodology

Graph Neural Prompting

The goal of this module is to encode the retrieved subgraph $\vec{G}$ into a graph neural prompt Z that guides the LLM.

GNN Encoder

Initialize node embeddings using pre-trained embeddings, then process the subgraph $\vec{G}$ using a Graph Neural Network (e.g., a Graph Attention Network):

$$H_1 = f_{\text{GNN}}(\vec{G}),$$

where $H_1 \in R^{n \times d_g}$ represents the embeddings for n nodes with dimension d_g.

Methodology

Cross-Modality Pooling

To fuse graph information with textual features:

1. **Self-Attention on Graph Nodes:**

$$H_2 = \text{SelfAttn}(H_1).$$

2. **Textual Transformation:** Obtain the text embeddings T from the LLM's dictionary and transform them:

$$T' = \text{FFN}_1\big(\omega(\text{FFN}_2(T))\big),$$

ensuring $T' \in R^{m \times d_g}$ where m is the number of tokens.

3. **Cross-Attention:** Compute cross-modality attention:

$$H_3 = \text{softmax}\left(\frac{H_2(T')^\top}{\sqrt{d_g}}\right)T'.$$

4. **Graph-Level Representation:** Pool the node embeddings:

$$H_4 = \text{POOL}(H_3).$$

Methodology

- **Self-Attention**:
  Enhances the representation of each node by considering the influence of other nodes in the subgraph.

- **Cross-Attention**:
  Allows the model to figure out which graph nodes (e. g. , "family") are most relevant given the text X. The pooled vector H_4 now summarizes the key knowledge.

Methodology

Domain Projector

Map the graph-level embedding H_4 to the same dimension as the LLM's text embeddings:

$$Z = \mathrm{FFN}_3\big(\omega(\mathrm{FFN}_4(H_4))\big),$$

where Z is the final Graph Neural Prompt.

The final vector Z is a compact representation that encapsulates the essential knowledge (e. g. , that "family history" is linked to eye color) and is in the same dimension as the text embeddings of the LLM.

Methodology

Self-Supervised Link Prediction

To further reinforce the learning of graph structures, a link prediction task is added:

1. **Masking:** Mask out a set of edges E_{mask} from $\vec{G}$.
2. **Scoring:** For each triplet (e_h, r, e_t), compute a score using a function (e.g., DistMult):

$$\varepsilon(e_h, e_t) = \langle h, r, t \rangle.$$

3. **Loss:** Define the link prediction loss as:

$$L_{\text{lp}} = \sum_{(e_h, r, e_t) \in E_{\text{mask}}} \left(-\log \sigma(\varepsilon(e_h, e_t) + \vartheta) + \frac{1}{n} \sum_{(e'_h, r, e'_t)} -\log \sigma(\varepsilon(e'_h, e'_t) + \vartheta) \right),$$

where σ is the sigmoid function, ϑ is a margin, and the second term averages over negative samples.

Methodology

Missing Links:

Suppose the edge ("family","influences","eye color") is masked. The model must use the surrounding graph context to predict this relation. By learning to predict these missing edges, the GNP module improves its ability to represent the graph structure, which benefits the overall prompt generation.

Methodology

Overall Training Objective

The final training loss is a combination of the LLM loss and the link prediction loss:

$$L = L_{\text{llm}} + \varpi L_{\text{lp}},$$

where ϖ balances the contribution of the link prediction task.

- **LLM Frozen Setting**:
The LLM's parameters are fixed; only the GNP module (including the GNN encoder, cross–modality pooling, domain projector, and link prediction head) is trained.

- **LLM Tuned Setting**:
A variant (e. g. , using LoRA) might allow limited updating of the LLM parameters, but the primary focus is still on the GNP module.

Results

LLM	Setting	Method	Commonsense Reasoning				Biomedical Reasoning		Total
			OBQA	**ARC**	**PIQA**	**Riddle**	**PubMedQA**	**BioASQ**	
FLAN-T5 xlarge (3B)	LLM Frozen	LLM-only	69.20	68.24	58.43	53.73	71.50	65.85	64.49
		Prompt Designs*	72.20	70.99	60.94	52.75	70.50	67.48	65.33
		KG Flattening REL	61.80	64.12	57.56	43.33	69.25	65.04	60.18
		KG Flattening BFS	62.80	63.86	56.69	44.12	69.25	65.04	60.29
		KAPING TH	58.80	63.52	52.34	40.78	70.00	65.04	58.41
		KAPING OH	60.00	63.09	51.69	41.37	70.00	65.04	58.53
		Prompt Tuning	72.20	70.64	60.83	53.33	72.00	66.67	65.95
		GNP	**79.80**	**71.85**	**61.48**	**66.86**	**76.75**	**89.43**	**74.36**
		Δ_{PT}	↑ 10.53%	↑ 1.71%	↑ 1.07%	↑ 25.37%	↑ 6.60%	↑ 34.14%	↑ 12.76%
	LLM Tuned	Full Fine-tuning	82.80	73.30	63.55	74.12	76.25	91.06	76.85
		LoRA	80.40	71.33	63.76	72.94	76.25	92.68	76.23
		LoRA + GNP	**83.40**	**72.45**	**64.31**	**75.49**	76.25	92.68	**77.43**
		Δ_{LoRA}	↑ 3.73%	↑ 1.57%	↑ 0.86%	↑ 3.50%	↑ 0.00%	↑ 0.00%	↑ 1.58%
FLAN-T5 xxlarge (11B)	LLM Frozen	LLM-only	76.80	68.93	56.58	61.37	71.75	65.85	66.88
		Prompt Designs*	79.60	74.16	58.00	60.59	71.25	66.67	68.38
		KG Flattening REL	72.80	66.78	56.80	53.53	69.50	66.67	64.35
		KG Flattening BFS	72.40	66.95	56.37	54.90	68.75	65.85	64.20
		KAPING TH	60.60	57.25	53.21	48.43	68.75	66.67	59.15
		KAPING OH	60.00	56.65	52.99	47.65	69.25	66.67	58.87
		Prompt Tuning	78.80	74.85	61.26	61.37	70.00	65.04	68.55
		GNP	**87.20**	**78.20**	**63.66**	**70.98**	**76.75**	**90.24**	**77.84**
		Δ_{PT}	↑ 10.66%	↑ 4.48%	↑ 3.92%	↑ 15.66%	↑ 9.64%	↑ 38.75%	↑ 13.54%
	LLM Tuned	Full Fine-tuning	89.40	76.82	65.61	80.78	78.00	92.68	80.55
		LoRA	88.60	78.54	65.61	74.90	77.75	91.06	79.41
		LoRA + GNP	**89.60**	**78.71**	**65.94**	**76.67**	**79.75**	**94.31**	**80.83**
		Δ_{LoRA}	↑ 1.13%	↑ 0.22%	↑ 0.50%	↑ 2.36%	↑ 2.57%	↑ 3.57%	↑ 1.79%

Results

A. LLM Frozen Setting

•Baseline Comparisons:
In this setting, the LLM's parameters remain unchanged and only the prompt is adapted. Several baselines were compared, including:

- **LLM-only**: No prompt or additional guidance.
- **Prompt Tuning**: Uses soft prompts but without KG-based guidance.
- **KG Flattening and KAPING**: Directly inject KG triples, which often introduce noise

•GNP's Improvement:
The Graph Neural Prompting (GNP) method significantly outperforms these baselines. For example, on commonsense reasoning tasks:

- On datasets like RiddleSense, GNP shows an improvement of around +25. 37**%** for a 3B LLM.
- For a larger 11B LLM, the improvement is around +15. 66**%**.

Results

- **Parameter Updates with LoRA**:
  In the LLM tuned setting, a small subset of the LLM parameters is updated (using techniques such as LoRA).

- **Combined Performance**:
  Even with these updates, the addition of the GNP module further boosts performance:
  - Improvements are modest (around +1. 8**%** overall), but in some cases, the combination of LoRA and GNP even surpasses full fine–tuning.

Ablation Study

The paper also investigates what happens when individual components of GNP are removed:

- **Without the Domain Projector** (**DP**): Removing DP led to a significant drop in performance, highlighting its critical role in aligning the graph features with the LLM's token space.
- **Without Cross-Modality Pooling** (**CMP**) **or Self-Supervised Link Prediction** (**SLP**): Similar performance degradations were observed, indicating that both components are essential for effectively fusing the graph and text information.

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

Model Design Comparison

Instance-Level Prompting (**GNP**) **vs**. **Dataset-Level Prompting** (**DLP**): The instance-specific prompts of GNP outperform dataset-level prompts.

Standard GNN vs. **RGNN**: The standard GNN is sufficient—and even preferable—for extracting graph representations, while more complex relational models (RGNN) can complicate prompt generation without added benefit.

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

Parameter Sensitivity

- **GNN Layers**:
  - The optimal number of layers can differ by dataset and LLM size.
  - For instance, a 3–layer GNN might work best for a 3B model on one dataset, while a 5–layer GNN might be optimal for an 11B model on another task.

- **Cross–Modality Pooling Layers**:
  Similarly, the number of pooling layers impacts performance:
  - Some datasets see improved performance with more pooling layers, while others might experience a drop.

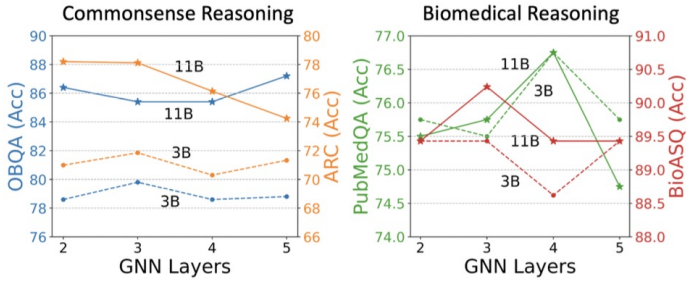


Figure 3: Performance *w.r.t.* different number of GNN layers.

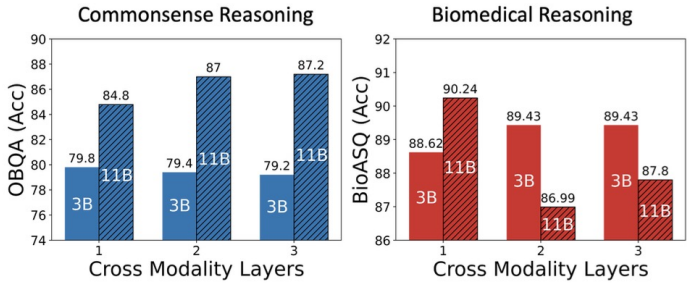


Figure 4: Performance *w.r.t.* different number of cross-modality pooling layers.

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Case Study

The visualization shows that while the subgraph contains some irrelevant nodes (e. g. , extraneous attributes like "round" or "nursery"), it also captures a critical link between "baby", "family", and "eye color". This link—illustrated by a clear connection in the graph—demonstrates why the correct answer (option d) is the most appropriate choice.

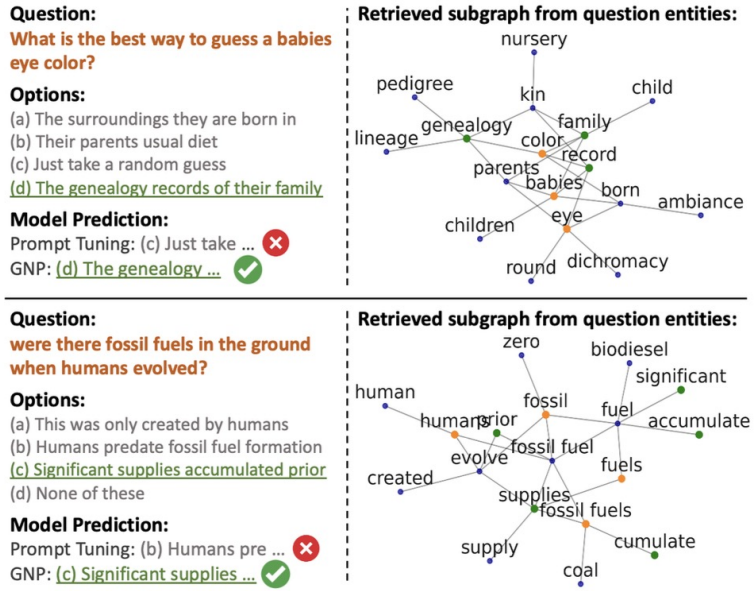


Figure 5: Case study on two QA examples from OBQA dataset. Question entities are marked in green and their subsampled neighbors in the KG are marked in blue. The entities appearing in the correct answer are marked in orange.

Limitation

1. **Subgraph Noise**:
   The process of retrieving subgraphs from a large knowledge graph may introduce noisy or irrelevant information. Although the GNP module is designed to filter and distill the important parts, extraneous or inaccurate relations can still negatively impact the quality of the generated prompt.

2. **Hyperparameter Sensitivity**:
   The model's performance is highly sensitive to the choice of hyperparameters—such as the number of GNN layers, cross-modality pooling layers, and the trade-off weight for the link prediction loss. This sensitivity can require extensive tuning for different datasets and LLM sizes, potentially limiting the method's ease of deployment.

Conclusion

- **Effective Knowledge Integration**:
  GNP significantly boosts LLM performance by integrating structured, KG–derived knowledge through a soft prompt.

- **Versatile and Robust**:
  The method works well across different domains and LLM sizes, whether the LLM is frozen or partially tuned.

- **Promising Future Directions**:
  Despite some limitations, this approach lays a solid foundation for further improvements in fusing external knowledge with LLMs.

UNIVERSITY OF
WATERLOO

FACULTY OF MATHEMATICS

Thank you!