

Graph Neural Prompting with Large Language Models

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

¹University of Notre Dame

²Amazon

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

Abstract

Large language models (LLMs) have shown remarkable generalization capability with exceptional performance in various language modeling tasks. However, they still exhibit inherent limitations in precisely capturing and returning grounded knowledge. While existing work has explored utilizing knowledge graphs (KGs) to enhance language modeling via joint training and customized model architectures, applying this to LLMs is problematic owing to their large number of parameters and high computational cost. Therefore, how to enhance pre-trained LLMs using grounded knowledge, e.g., retrieval-augmented generation, remains an open question. In this work, we propose Graph Neural Prompting (GNP), a novel plug-and-play method to assist pre-trained LLMs in learning beneficial knowledge from KGs. GNP encompasses various designs, including a standard graph neural network encoder, a cross-modality pooling module, a domain projector, and a self-supervised link prediction objective. Extensive experiments on multiple datasets demonstrate the superiority of GNP on both commonsense and biomedical reasoning tasks across different LLM sizes and settings. Code is available at <https://github.com/meettyj/GNP>.

Introduction

Large Language Models (LLMs) have demonstrated exceptional performance and general capability in various NLP tasks and use cases such as question answering (Robinson, Rytting, and Wingate 2023) and text summarization (Zhang et al. 2023). Moreover, the significant growth in model size has further endowed LLMs with emergent capabilities (Wei et al. 2022b), laying the groundwork for exploring artificial general intelligence (Bubeck et al. 2023). Accordingly, LLMs have attracted tremendous interest from academia (Wei et al. 2022a; Zhao et al. 2023) and industry (Anil et al. 2023; OpenAI 2023).

Given the broad success of LLMs, many techniques have emerged to adapt these general-purpose models to downstream tasks. Beyond the conventional approach of model fine-tuning where all model parameters are adjusted (Howard and Ruder 2018), prompt-based adaptation methods are proposed to modulate a frozen LLM’s behavior through

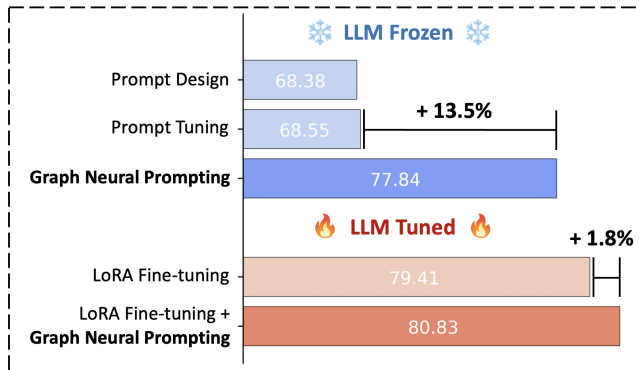


Figure 1: Result comparison across LLM Frozen (parameters unchanged) and LLM Tuned (parameters updated) settings. The proposed Graph Neural Prompting significantly improves the performance. Reported results are averaged across six datasets on two tasks for an 11B FLAN-T5 model.

prompts (Brown et al. 2020; Lester, Al-Rfou, and Constant 2021; Li and Liang 2021). Rather than adapt the parameters in LLMs, these methods freeze the LLMs and typically introduce additional trainable parameters. The idea of freezing LLMs is appealing, especially as the model size grows and the training resource dependency intensifies.

On the other hand, despite the success of LLMs in handling different real-world applications and the feasibility of adapting to specific downstream tasks, they still exhibit the inherent limitations of language modeling in accurately capturing and returning grounded knowledge (Lewis et al. 2020; Pan et al. 2023). Knowledge graphs (KGs), storing enormous facts, serve as a systematic way of representing knowledge (Ji et al. 2021). Consequently, existing methods have incorporated KGs to assist language modeling, often by designing customized model architectures to accommodate both KGs and textual data, followed by joint training sessions (Yasunaga et al. 2022; Zhang et al. 2022). Nonetheless, joint training KGs and text for LLMs is challenging due to the extensive parameters LLMs contain and the substantial computation resources they require. In addition, numerous pre-trained LLMs with exceptional capabilities are released. It becomes advantageous to employ these pre-existing LLMs, particularly beneficial if we can sidestep the need to craft a

specialized model and train it from scratch. A direct approach to employing KGs for retrieval-augmented generation (Lewis et al. 2020) is to feed the KG triples into LLMs directly (Baek, Aji, and Saffari 2023). However, this method can introduce substantial noise, given that KGs might contain various extraneous contexts. Therefore, we ask:

Can we learn beneficial knowledge from KGs and integrate them into pre-trained LLMs?

To answer the question, we propose Graph Neural Prompting (GNP), a novel plug-and-play method to assist pre-trained LLMs in learning beneficial knowledge from KGs. GNP retrieves and encodes the pertinent grounded knowledge to derive Graph Neural Prompt, an embedding vector that can be sent into LLMs to provide guidance and instructions. In particular, GNP first utilizes a graph neural network (GNN) to capture and encode the intricate graph knowledge into entity/node embeddings. Then, a cross-modality pooling module is present to determine the most relevant node embeddings in relation to the text input, and consolidate these node embeddings into a holistic graph-level embedding. After that, GNP encompasses a domain projector to bridge the inherent disparities between the graph and text domains. Finally, a self-supervised link prediction objective is introduced to enhance the model comprehension of relationships between entities and capture graph knowledge in a self-supervised manner.

To fully evaluate our model, we conduct extensive experiments on multiple public benchmark datasets in the tasks of commonsense reasoning and biomedical reasoning. We further report the results across different LLM sizes and settings. We conclude that GNP can effectively encode intricate knowledge in KGs and significantly improve performance. Figure 1 shows the averaged performance improvement using our method across six datasets. Specifically, GNP improves the baseline by **+13.5%** when LLM is frozen, validating the superiority of our method in learning effective prompts. In addition, by using our method, fine-tuning LLMs with parameter-efficient approach LoRA (Hu et al. 2022) shows an improvement of **+1.8%**. More promisingly, compared to model full fine-tuning without leveraging any efficient tuning approaches, our method can achieve competitive or superior performance in **10 out of 12** evaluations, as shown in the experiment section. To summarize, our main contributions are:

- To the best of our knowledge, this is the first attempt to study the learning of beneficial knowledge from KGs for pre-trained LLMs.
- We propose GNP, a novel plug-and-play method for pre-trained LLMs to extract valuable knowledge from KGs. The proposed method contains various tailored designs, including a standard GNN, a cross-modality pooling module, a domain projector, and a self-supervised graph learning objective.
- Extensive experiments demonstrate the superiority of GNP on multiple datasets across different settings. We also present the ablation study, model design comparison, parameter sensitivity analysis, case study and visualization to validate the effectiveness of GNP.

Related Work

Large Language Models and Question Answering.

Recently, various LLMs have been proposed (Chung et al. 2022; Touvron et al. 2023; Brown et al. 2020) and have demonstrated remarkable performance across different tasks (Shi et al. 2023; Chen et al. 2023b; Wei et al. 2024; Hong et al. 2023). Question answering, as a fundamental task, demands intricate reasoning and understanding comprehension skills to interpret the text and provide appropriate responses to the posed questions (Lu et al. 2022; Zhu et al. 2021; Wang et al. 2023; Chen et al. 2023a). Although LLMs have strong learning capabilities, they have the limitation of precisely capturing accurate factual knowledge and are susceptible to generating unfounded responses (Zhao et al. 2023; Ji et al. 2023; Bang et al. 2023). In addition, the enormous number of parameters in LLMs poses difficulties in adapting LLMs for downstream tasks (Scao et al. 2022; Smith et al. 2022). Correspondingly, various approaches are presented to alleviate the intensive training dependency and reduce the computational expenses (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021; Hu et al. 2022). For instance, Prompt Tuning (Lester, Al-Rfou, and Constant 2021) introduces soft prompts to condition the pre-trained LLMs for downstream tasks. In our work, we propose to retrieve the factual knowledge from KGs to enhance LLMs, while still benefiting from circumventing the burdensome training expenses by using pre-trained LLMs.

Knowledge Graphs for Language Modeling.

Many graph learning methods are proposed to encode graphs and KGs (Ji et al. 2021; Tian et al. 2023a,b; Tang et al. 2022; Wang, Jin, and Derr 2022; Xu et al. 2023; Kou et al. 2022). Recent studies indicate that KGs can enhance language modeling by providing background knowledge (Ren et al. 2021; Wang et al. 2019). One approach to achieve this is integrating KGs into the pre-training stage of language modeling. For instance, ERNIE (Sun et al. 2021), JAKET (Yu et al. 2022), and JointGT (Ke et al. 2021) develop pre-training objectives tailored for KG triples and the paired sentences. DRAGON (Yasunaga et al. 2022) introduces a customized fusion framework to jointly pre-train the model for KGs and text. Moreover, KGs are leveraged to assist language modeling for question answering (Lin et al. 2019; Lv et al. 2020; Feng et al. 2020; Mihaylov and Frank 2018). Specifically, GreaseLM (Zhang et al. 2022) and QAGNN (Yasunaga et al. 2021) suggest that KGs can scaffold reasoning about entities with the graph structure such as negation and multi-hop reasoning to facilitate complex question answering. To encode KGs, many works study methods to learn KG entity and relation embeddings, such as TransE (Bordes et al. 2013) and DistMult (Yang et al. 2015). Recently, with the aim of integrating KGs into the emerging domain of LLMs, given existing studies pose difficulties when applying, KAPING (Baek, Aji, and Saffari 2023) employs knowledge graphs to extract relevant triples. These triples correspond to the input question, with the expectation that directly feeding them into LLMs is beneficial, despite the presence of noise. In our work, we present a learning method for identifying beneficial knowledge from KGs, offering substantial benefits to LLMs.

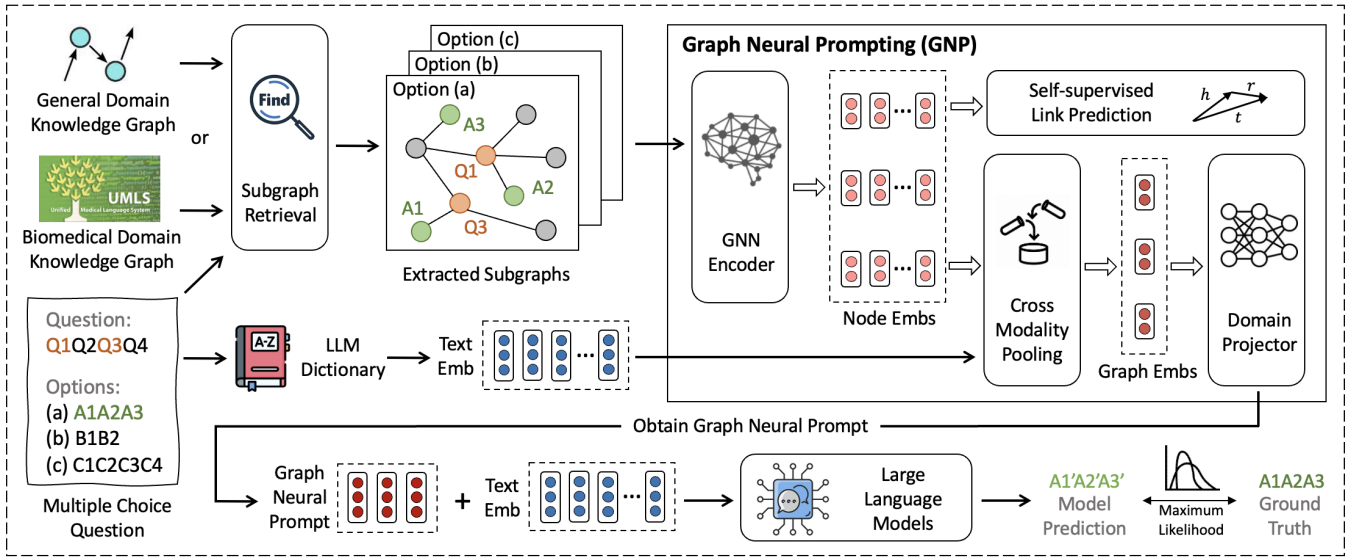


Figure 2: The overall framework. Given a multiple choice question, we first retrieve subgraphs from the knowledge graph based on the entities in the question and options. We then develop Graph Neural Prompting (GNP) to encode the pertinent factual knowledge and structural information to obtain the Graph Neural Prompt. GNP contains various designs including a GNN, a cross-modality pooling module, a domain projector, and a self-supervised link prediction objective. Later, the obtained Graph Neural Prompt is sent into LLM for inference along with the input text embedding. We utilize the standard maximum likelihood objective for downstream task adaptation, while LLM is kept frozen or tuned depending on different experimental settings.

Preliminary

In this section, we describe the knowledge graph and formally define the problem of multiple choice question answering.

Definition 1. Knowledge Graph. A knowledge graph is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where \mathcal{E} is the set of entities and \mathcal{R} is the set of relations. \mathcal{T} is the collection of fact triples $\{(e_h, r, e_t)\} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where e_h denotes the head entity, r is the relation, and e_t indicates the tail entity.

Problem 1. Multiple Choice Question Answering. Given a question Q , a set of answer options $A = \{a_k\}_{k=1}^K$, and an optional context C depending on open-book or close-book, the task is to design a machine learning model \mathcal{F}_Θ with parameters Θ that selects the best option to answer the question. Here K denotes the total number of answer options and a_k indicates the k -th answer option. The ground truth label $y \in A$ is the correct answer for Q . In addition, we use knowledge graph \mathcal{G} to provide rich knowledge and assist the model to answer the question.

Methodology

In this section, we introduce the techniques of prompting LLMs for question answering as well as subgraph retrieval. Additionally, we present Graph Neural Prompting and elaborate on its components and designs. Figure 2 illustrates the framework of our method.

Prompting LLMs for Question Answering

Prompting is the de facto approach to elicit responses from LLMs (Liu et al. 2023). The typical approach of prompting LLMs for multi-choice question answering is simple. Given a

question Q , the optional context C , and the answer options A , we first tokenize the concatenation of C, Q, A into a sequence of input text tokens X . We then design a series of prompt tokens, P , and prepend it to the input text tokens X , which is later considered as input for the LLM model to generate prediction $y' = f([P, X])$. The LLM model can be trained for downstream task adaptation using a standard maximum likelihood loss using teacher forcing (Williams and Zipser 1989) and a cross-entropy loss:

$$\mathcal{L}_{llm} = -\log p(y|X, \Theta), \quad (1)$$

where p is the probability distribution parameterized by the model. The prompt P can be either a hard prompt in the form of textual input, or a soft prompt in the form of learnable embedding vectors.

Unlike existing methods that solely use a text string as the hard prompt, our Graph Neural Prompting approach encodes structural and factual information contained in the knowledge graph \mathcal{G} into a soft prompt P , which is a sequence of trainable vectors that can be concatenated with the token embedding of X . The learning of P is encouraged to provide rich structural information and knowledge from \mathcal{G} as well as task instruction for each data instance.

Subgraph Retrieval

To semantically align the input text tokens X with the massive knowledge graph \mathcal{G} with millions of nodes, we retrieve subgraphs of \mathcal{G} that contain the relevant entities to the tokens in X . In particular, for each answer option a_k and its corresponding context C and question Q , we first obtain a set of matched entities \mathcal{E}_{match} via entity linking to match the

tokens in X to the entities in \mathcal{G} . We then retrieve a subgraph \mathcal{G}' based on the entities in \mathcal{E}_{match} by including their two-hop neighbors and the relations that connect them (Yasunaga et al. 2022). The retrieved subgraph contains the necessary content and knowledge to assist the model in answering Q .

Graph Neural Prompting

Graph Neural Prompting contains various designs, including a GNN encoder that embeds the knowledge graph, a cross-modality pooling module that determines the pertinent node embeddings, a domain projector that bridges the discrepancies between graph and text, and a self-supervised link prediction objective that encourages the model to recognize structural information.

GNN Encoder. Although the retrieved subgraph \mathcal{G}' contains rich contextual information regarding the question and answer choices, some entities and relations are not relevant to the actual question. Directly feeding every fact triples in \mathcal{G}' can introduce noise and prevent the LLM model from concentrating on the critical information. Therefore, we introduce a GNN to encode the most relevant knowledge and further integrate the complex relationships among the entities. In particular, we first initialize the node embeddings using pre-trained entity embeddings (Feng et al. 2020; Yasunaga, Leskovec, and Liang 2022). Next, we employ a standard graph attention network (Veličković et al. 2018) as our GNN encoder for the retrieved subgraph \mathcal{G}' .

$$H_1 = f_{GNN}(\mathcal{G}'), \quad (2)$$

where $H_1 \in \mathbb{R}^{d_g}$ represents the node embeddings learned by GNN for every node in \mathcal{G}' , and d_g denotes the output dimension of the GNN encoder.

Cross-modality Pooling. With the aim of identifying the most pertinent nodes in relation to the question, and consolidating the node embeddings into a holistic graph-level representation for subsequent use, we design the cross-modality pooling module. In particular, we first introduce a self-attention layer to dynamically identify node significance using the internal graph characteristics and the implicit interactions among nodes:

$$H_2 = \text{Self-Attn}(H_1), \quad (3)$$

where H_2 is the obtained node embeddings and Self-Attn indicates the self-attention component. Then, we leverage the textual prompt to calculate the importance of nodes within the graph. To ensure uniformity, we utilize the dictionary in the LLM to obtain the text embeddings $\mathcal{T} \in \mathbb{R}^{d_t}$ for every token in the input text, where d_t denotes the dimension of the LLM dictionary. Next, we apply a transformation to the text embeddings \mathcal{T} and obtain the transformed text embedding \mathcal{T}' , ensuring that the dimension of \mathcal{T}' matches the dimension d_g of node embeddings H_2 . After that, we calculate the cross-modality attention using H_2 and \mathcal{T}' . We use H_2 as the query and the \mathcal{T}' as the key and value. The procedure is as follows:

$$\begin{aligned} \mathcal{T}' &= \text{FFN}_1(\sigma(\text{FFN}_2(\mathcal{T}))), \\ H_3 &= \text{softmax}[H_2 \cdot (\mathcal{T}')^T / \sqrt{d_g}] \cdot \mathcal{T}', \end{aligned} \quad (4)$$

where σ is the GELU activation function, FFN_1 and FFN_2 are feed-forward neural networks, and H_3 is the final

node embeddings obtained with cross-modality attention considered. Next, we generate the graph-level embedding by average pooling the node embeddings H_3 in \mathcal{G}' :

$$H_4 = \text{POOL}(H_3), \quad (5)$$

where H_4 represents the graph-level embedding that takes into account the node significance in \mathcal{G}' .

Domain Projector. In order to create a mapping between the graph-level embeddings and the text domain to facilitate comprehension by the LLM, we design a domain projector to align them. This projector aims to bridge the inherent disparities between the graph and text, allowing for more seamless integration. In addition, the projector maps the graph-level embeddings to the same dimension d_t of LLM, which ensures compatibility and consistency when interfacing with the LLM’s inherent structures. We design the projector as follows:

$$Z = \text{FFN}_3(\sigma(\text{FFN}_4(H_4))), \quad (6)$$

where Z denotes Graph Neural Prompt, the final output of GNP, and FFN_3 , FFN_4 are feed-forward neural networks.

Self-supervised Link Prediction. While the downstream cross-entropy objective enables the model to learn and adapt to the target dataset, we design a link prediction task to further refine its understanding of relationships between entities and capture graph knowledge in a self-supervised manner. Specifically, we mask out some edges in \mathcal{G}' and enforce the model to predict them. This encourages the model to learn to use the partial graph content and structure to reason about the missing links. Concretely, we denote the set of masked-out edges as $\mathcal{E}_{mask} \subseteq \mathcal{E}$. Given the learned node embeddings of the head entity and tail entity in a triplet $\{\mathbf{h}_3, \mathbf{t}_3\} \in H_3$, we adopt a widely-used knowledge graph embedding method DistMult (Yang et al. 2015) to map the entity embeddings and relation in the KG to vectors, \mathbf{h} , \mathbf{r} , \mathbf{t} . We then define the scoring function $\phi(e_h, e_t) = \langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$ to generate the scores for each triple, where $\langle \cdot, \cdot, \cdot \rangle$ denotes the trilinear dot product, and \mathbf{r} represents the relations in KGs. A higher ϕ indicates a higher chance of (e_h, r, e_t) being a correct positive triple instead of an incorrect negative triple. We enforce the model to predict the masked edges in \mathcal{E}_{mask} as positive and other random edges as negative. The link prediction loss \mathcal{L}_{lp} is defined as follows:

$$\mathcal{L}_{lp} = \sum_{(e_h, r, e_t) \in \mathcal{E}_{mask}} (S_{pos} + S_{neg}), \quad (7)$$

where $S_{pos} = -\log \sigma_s(\phi(e_h, e_t) + \gamma)$ indicates the score for correct positive triples, γ is the margin, σ_s is the sigmoid function, $\{(e'_h, r, e'_t)\}$ are n negative triples corresponding to the positive triplet (e_h, r, e_t) , and $S_{neg} = \frac{1}{n} \sum_{(e'_h, r, e'_t)} \log \sigma_s(\phi(e'_h, e'_t) + \gamma)$ is the score for incorrect negative triples. The final objective function \mathcal{L} is defined as the weighted combination of \mathcal{L}_{llm} and \mathcal{L}_{lp} :

$$\mathcal{L} = \mathcal{L}_{llm} + \lambda \mathcal{L}_{lp}, \quad (8)$$

where λ is a trade-off weight for balancing two losses.

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

Table 1: Overall experimental results on commonsense reasoning and biomedical reasoning tasks. The best results across different LLM sizes and settings are highlighted in bold. Δ_{PT} and Δ_{LoRA} represent the relative performance improvement of our method to Prompt Tuning and LoRA, respectively. We also include the full fine-tuning result in gray color for further reference. * means multiple prompt design methods are evaluated while only the best result is reported. Accuracy is used as the evaluation metric.

Experiments

In this section, we conduct extensive experiments to compare the performances of different models. We also show ablation study, model design comparison, and parameter sensitivity analysis to demonstrate the effectiveness of GNP. Moreover, we present case study and visualization to provide an intuitive understanding and illustrate how KGs benefit.

Experiment Setup

Knowledge Graphs and Datasets. We conduct experiments on both the general domain (commonsense reasoning) and the biomedical domain (biomedical reasoning). For the used knowledge graphs, we consider ConceptNet (Speer, Chin, and Havasi 2017) that contains rich commonsense knowledge regarding the daily concepts, and Unified Medical Language System (UMLS) (Bodenreider 2004) that involves well-structured health and biomedical information. For datasets, we use four commonsense reasoning datasets, including OpenBookQA (OBQA) (Mihaylov et al. 2018), AI2 Reasoning Challenge (ARC) (Clark et al. 2018), Physical Interaction Question Answering (PIQA) (Bisk et al. 2020), and RiddleSense (Riddle) (Lin et al. 2021). In addition, we consider PubMedQA (Jin et al. 2019) and BioASQ (Tsatsaronis et al. 2015) for biomedical reasoning.

Two Settings: LLM Frozen vs. LLM Tuned. To fully

evaluate the model, we employ two settings: LLM Frozen and LLM Tuned. For LLM Frozen, we keep the parameters in LLM unchanged and only adapt the prompt. For LLM Tuned, the original LLM parameters are updated for downstream tasks by utilizing LoRA or full fine-tuning.

Baselines. In the setting of LLM Frozen, we compare with nine baselines, including LLM-only that uses no prompt, three prompt design methods that use different instructions as hard prompts, KG Flattening that flattens the nodes in the graph into a sequence via relevance score (REL) ranking (Yasunaga et al. 2022) or breadth-first search (BFS), KAPING (Baek, Aji, and Saffari 2023) that injects the important KG triples within one-hop (OH) and two-hop (TH) neighborhoods, and Prompt Tuning (Lester, Al-Rfou, and Constant 2021) that introduces soft prompts. In the setting of LLM Tuned, we compare with LoRA that updates partial LLM parameters. In addition, we include full model fine-tuning results as the referencing benchmark.

Implementation Details. For the proposed model, we set the learning rate to $1e-4$, batch size to 8, hidden dimension of GNN to 1024, and training epochs to 50. In order to adapt the model effectively to each dataset, we search the GNN layers from 2 to 5, cross-modality pooling layers from 1 to 3, trade-off weight λ from $\{0.1, 0.5\}$, and link drop rate from $\{0.1, 0.3, 0.7\}$. We choose FLAN-T5 xlarge (3B parameters)

and xlarge (11B parameters) as the LLMs used in this paper. We adjust the maximum sequence length of LLMs to best fit the question length for each dataset. We run all experiments on four NVIDIA Tesla V100 GPUs with 24GB RAM.

Performance Comparison

To comprehensively evaluate our model, we conduct rigorous experiments using various LLMs across two reasoning tasks under different settings. The results are reported in Table 1. According to the table, in the setting of LLM Frozen, we observe that the utilization of the prompt design instructions often yields performance improvement, compared to LLM-only that uses no instructions, though the enhancement is mostly marginal. Interestingly, the baseline methods that inject KG information directly (KG Flattening and KAPING) can significantly hurt the model performance. This aligns with our motivation that KGs contain irrelevant contexts for the downstream tasks that could introduce noises or even alter the semantics if not handled carefully. While Prompt Tuning shows improved outcomes using the trainable soft prompts, their improvement is trivial. In contrast, our GNP exhibits significant and notable performance improvements across various datasets, settings, and LLMs. For example, for the commonsense reasoning task, GNP provides +25.37% improvement on Riddle for 3B LLM, and +15.66% improvement for 11B LLM. In addition, for the biomedical reasoning task, GNP improves the performance by +34.14% on BioASQ for 3B LLM and +38.75% for 11B LLM. In general, GNP achieves an improvement of +12.76% and +13.54% for 3B and 11B LLM, respectively.

In the setting of LLM Tuned, we first study the performance in comparison with LoRA and then report the model full fine-tuning for additional reference. As shown in the table, LoRA is a significantly more powerful approach than Prompt Tuning due to the direct update of the LLM internal parameters. Combining with the proposed GNP, the performance can be further improved. For example, GNP achieves 3.73% improvement on OBQA for 3B LLM, and 3.57% improvement on BioAQS for 11B LLM. Moreover, model full fine-tuning is an important reference to study the performance gap since LoRA only updates a small fraction of the model parameters. Surprisingly, we find that the incorporation of GNP can surpass the results of full fine-tuning. In contrast, relying solely on LoRA shows difficulties in achieving a comparable performance of full fine-tuning. In total, our final performance matches or surpasses model full fine-tuning in **10 out of 12** evaluations across different LLM sizes and datasets, as shown in Table 1.

Ablation Study

Since GNP contains various model components (i.e., cross-modality pooling (CMP), self-supervised link prediction (SLP), and domain projector (DP)), we conduct ablation studies to analyze the contributions of different components by removing each of them independently (see Table 2). Specifically, removing DP significantly affects the performance, showing that DP has a large contribution to the proposed method. In addition, the decreasing performances of removing CMP and SLP demonstrate the effectiveness

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

Table 2: Results of ablation study.

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

Table 3: Results of integrating different model designs.

of CMP and SLP in enhancing the model. In most cases, SLP yields greater significance compared to CMP, while in BioASQ, CMP plays a more important role. Finally, the proposed GNP achieves the best results in all cases, indicating the strong capability of different components in our model.

Model Design Comparison

A salient property of GNP is the learning of Graph Neural Prompt for each data instance, i.e., various questions yield different retrieved subgraphs, resulting in unique prompts. Given its distinction to the dataset-level prompt (DLP) from Prompt Tuning that learns prompt for each dataset, we present the outcomes of integrating DLP for further investigation. As shown in Table 3, incorporating DLP cannot further boost the performance and might even diminish it in certain cases. This indicates that our instance-level prompt provides adequate guidance for LLM to perform well. In addition, we validate the importance of explicitly modeling relations using a widely-used Relational GNN (RGNN) (Zhang et al. 2022). The observed decline in performance suggests that a standard GNN is sufficient to capture the graph information, and explicitly modeling the relations might increase the difficulty of generating suitable guidance for the task.

Parameter Sensitivity

Next, we perform sensitivity analysis focusing on the following parameters: the number of GNN layers and the number of layers in the cross-modality pooling component.

Impact of GNN Layers. We evaluate the influence of GNN layers for both 3B and 11B models in Figure 3. According to the figure, we have the following observations. First, various datasets have different optimal numbers of GNN layers. To illustrate, for ARC, 3 layers can achieve the optimal performance while 4 layers perform the best for PubMedQA. Second, the optimal number of GNN layers for 3B and 11B

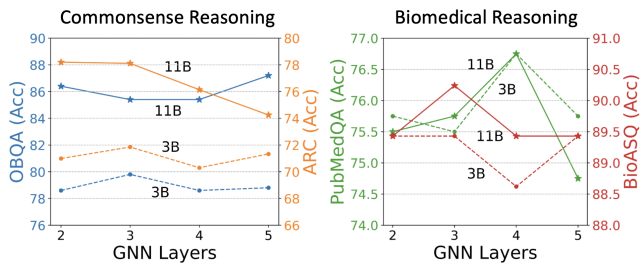


Figure 3: Performance w.r.t. different number of GNN layers.

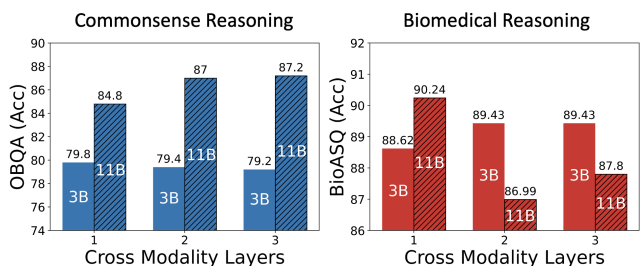


Figure 4: Performance w.r.t. different number of cross-modality pooling layers.

LLMs differs. For example, for OBQA, 3 layers work best for 3B LLM, while 11B LLM reaches its top performance when using 5 layers. Third, choosing different GNN layers can have a weak impact on some datasets while can also drastically affect the performance on other datasets. To demonstrate, increasing from 3 layers to 5 layers for 11B LLM can decrease the performance on ARC by a large margin (from 78.1 to 74.3), while adjusting the layers for BioASQ may not lead to a big change in the performance.

Impact of Cross-modality Pooling Layers. We report the performance of different cross-modality pooling layers in Figure 4. As shown in the figure, we observe that the commonsense reasoning dataset OBQA and biomedical reasoning dataset BioASQ demonstrate different reactions to layer numbers. Specifically, for OBQA, the performance of the larger 11B LLM increases with more layers, while the performance of the smaller 3B LLM decreases. On the other hand, for BioASQ, the larger 11B LLM tends to show a degraded performance when adding more layers, while the smaller 3B model presents an improved performance. This indicates that suitable cross-modality pooling layers can lead to the best model performance.

Case Study and Visualization

For a more intuitive understanding and comparison, we randomly select two examples from the OBQA dataset and visualize the retrieved subgraphs in Figure 5. For visualization clarity, we only show question entities and a limited number of their neighbors. We remarkably notice that the retrieved subgraphs encompass certain entities for the correct answer, and there exist edges connecting the question and answer entities, which makes the task of question answering easier by leveraging this information.

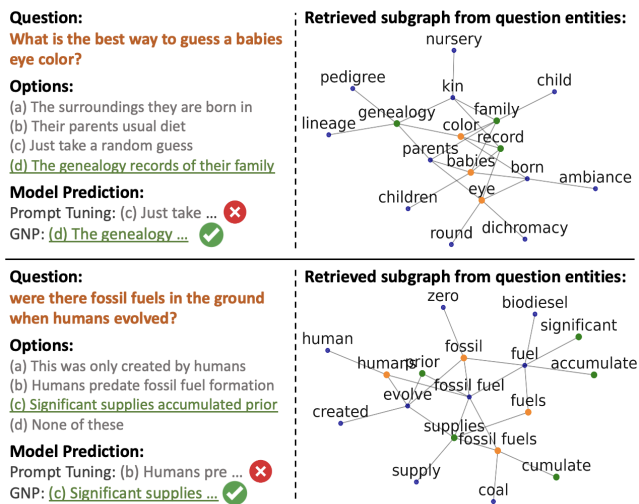


Figure 5: Case study on two QA examples from OBQA dataset. Question entities are marked in orange and their subsampled neighbors in the KG are marked in blue. The entities appearing in the correct answer are marked in green.

To answer the question “What is the best way to guess a babies eye color?”, Prompt Tuning makes the wrong generation “Just take a random guess”. On the other hand, our retrieved subgraph offers the links that directly relate the entity “babies” to “family”, “record”, and further to “genealogy”, which all appear in the correct option (d). This important context provides valuable insights for the model. Note that the subgraph also contains irrelevant entities such as “round” and “nursery”. This explains why directly using the knowledge graph can introduce noise. However, our GNP method possesses the capability to collect the most critical information in the graph to determine the correct answer.

The second question “were there fossil fuels in the ground when humans evolved?” requires correctly identifying the historical sequencing order between the entity “humans” and “fossil fuels”. The retrieved subgraph contains the critical relation, i.e., “humans”, “evolve”, “prior”, “fossil fuel”. Nevertheless, the subgraph also contains the entity “created” that could confuse the model into selecting option (a). GNP is able to capture the structural proximity among the key entities and select the correct answer (c).

Conclusion

In this paper, we address the limitations of LLMs in precisely capturing and returning grounded knowledge. In particular, we propose Graph Neural Prompting (GNP), a novel plug-and-play method to assist pre-trained LLMs in learning beneficial knowledge from KGs. Extensive experiments on commonsense and biomedical reasoning tasks demonstrate that GNP can improve the performance by +13.5% when LLM is frozen, and +1.8% when LLM is tuned. In addition, we present ablation studies, model design comparison, parameter sensitivity, case study and visualization to validate the effectiveness of the proposed method.

References

- Anil, R.; Dai, A. M.; Firat, O.; Johnson, M.; Lepikhin, D.; Passos, A.; Shakeri, S.; Taropa, E.; Bailey, P.; Chen, Z.; et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Baek, J.; Aji, A. F.; and Saffari, A. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In *ACL Workshop on Matching Entities*.
- Bang, Y.; Cahyawijaya, S.; Lee, N.; Dai, W.; Su, D.; Wilie, B.; Lovenia, H.; Ji, Z.; Yu, T.; Chung, W.; et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.
- Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.
- Bodenreider, O. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*.
- Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NeurIPS*.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *NeurIPS*.
- Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.
- Chen, X.; Jiang, J.-Y.; Chang, W.-C.; Hsieh, C.-J.; Yu, H.-F.; and Wang, W. 2023a. MinPrompt: Graph-based Minimal Prompt Data Augmentation for Few-shot Question Answering. *arXiv preprint arXiv:2310.05007*.
- Chen, X.; Liu, Y.; Yang, Y.; Yuan, J.; You, Q.; Liu, L.-P.; and Yang, H. 2023b. Reason out Your Layout: Evoking the Layout Master from Large Language Models for Text-to-Image Synthesis. *arXiv preprint arXiv:2311.17126*.
- Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Feng, Y.; Chen, X.; Lin, B. Y.; Wang, P.; Yan, J.; and Ren, X. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *EMNLP*.
- Hong, J.; Wang, J. T.; Zhang, C.; Li, Z.; Li, B.; and Wang, Z. 2023. DP-OPT: Make Large Language Model Your Privacy-Preserving Prompt Engineer. *arXiv preprint arXiv:2312.03724*.
- Howard, J.; and Ruder, S. 2018. Universal language model fine-tuning for text classification. In *ACL*.
- Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*.
- Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*.
- Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*.
- Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W. W.; and Lu, X. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*.
- Ke, P.; Ji, H.; Ran, Y.; Cui, X.; Wang, L.; Song, L.; Zhu, X.; and Huang, M. 2021. Jointgt: Graph-text joint representation learning for text generation from knowledge graphs. In *ACL-IJCNLP*.
- Kou, Z.; Zhang, Y.; Zhang, D.; and Wang, D. 2022. CrowdGraph: A Crowdsourcing Multi-Modal Knowledge Graph Approach to Explainable Fauxtography Detection. *Proceedings of the ACM on Human-Computer Interaction*.
- Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. In *EMNLP*.
- Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*.
- Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL-IJCNLP*.
- Lin, B. Y.; Chen, X.; Chen, J.; and Ren, X. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP*.
- Lin, B. Y.; Wu, Z.; Yang, Y.; Lee, D.-H.; and Ren, X. 2021. RiddleSense: Reasoning about Riddle Questions Featuring Linguistic Creativity and Commonsense Knowledge. In *ACL-IJCNLP*.
- Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*.
- Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*.
- Lv, S.; Guo, D.; Xu, J.; Tang, D.; Duan, N.; Gong, M.; Shou, L.; Jiang, D.; Cao, G.; and Hu, S. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *AAAI*.
- Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.
- Mihaylov, T.; and Frank, A. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *ACL*.

- OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.
- Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv preprint arXiv:2306.08302*.
- Ren, H.; Dai, H.; Dai, B.; Chen, X.; Yasunaga, M.; Sun, H.; Schuurmans, D.; Leskovec, J.; and Zhou, D. 2021. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In *ICML*.
- Robinson, J.; Rytting, C. M.; and Wingate, D. 2023. Leveraging large language models for multiple choice question answering. In *ICLR*.
- Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; Gallé, M.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Shi, Y.; Xu, S.; Liu, Z.; Liu, T.; Li, X.; and Liu, N. 2023. Mededit: Model editing for medical question answering with external knowledge bases. *arXiv preprint arXiv:2309.16035*.
- Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhunoye, S.; Zerveas, G.; Korthikanti, V.; et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.
- Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.
- Sun, Y.; Wang, S.; Feng, S.; Ding, S.; Pang, C.; Shang, J.; Liu, J.; Chen, X.; Zhao, Y.; Lu, Y.; et al. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.
- Tang, Z.; Pei, S.; Zhang, Z.; Zhu, Y.; Zhuang, F.; Hoehndorf, R.; and Zhang, X. 2022. Positive-unlabeled learning with adversarial data augmentation for knowledge graph completion. In *IJCAI*.
- Tian, Y.; Dong, K.; Zhang, C.; Zhang, C.; and Chawla, N. V. 2023a. Heterogeneous Graph Masked Autoencoders. In *AAAI*.
- Tian, Y.; Zhang, C.; Guo, Z.; Zhang, X.; and Chawla, N. 2023b. Learning MLPs on Graphs: A Unified View of Effectiveness, Robustness, and Efficiency. In *ICLR*.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Tsatsaronis, G.; Balikas, G.; Malakasiotis, P.; Partalas, I.; Zschunke, M.; Alvers, M. R.; Weissenborn, D.; Krithara, A.; Petridis, S.; Polychronopoulos, D.; et al. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.
- Wang, X.; Kapanipathi, P.; Musa, R.; Yu, M.; Talamadupula, K.; Abdelaziz, I.; Chang, M.; Fokoue, A.; Makni, B.; Mattei, N.; et al. 2019. Improving natural language inference using external knowledge in the science questions domain. In *AAAI*.
- Wang, Y.; Jin, W.; and Derr, T. 2022. Graph neural networks: Self-supervised learning. *Graph Neural Networks: Foundations, Frontiers, and Applications*.
- Wang, Y.; Lipka, N.; Rossi, R. A.; Siu, A.; Zhang, R.; and Derr, T. 2023. Knowledge Graph Prompting for Multi-Document Question Answering. *arXiv preprint arXiv:2308.11730*.
- Wei, J.; Bosma, M.; Zhao, V. Y.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned language models are zero-shot learners. In *ICLR*.
- Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.
- Wei, W.; Ren, X.; Tang, J.; Wang, Q.; Su, L.; Cheng, S.; Wang, J.; Yin, D.; and Huang, C. 2024. LLMRec: Large Language Models with Graph Augmentation for Recommendation. In *WSDM*.
- Williams, R. J.; and Zipser, D. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*.
- Xu, Z.; Zeng, H.; Tan, J.; Fu, Z.; Zhang, Y.; and Ai, Q. 2023. A Reusable Model-agnostic Framework for Faithfully Explainable Recommendation and System Scrutability. *ACM Transactions on Information Systems*.
- Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.
- Yasunaga, M.; Bosselut, A.; Ren, H.; Zhang, X.; Manning, C. D.; Liang, P. S.; and Leskovec, J. 2022. Deep bidirectional language-knowledge graph pretraining. In *NeurIPS*.
- Yasunaga, M.; Leskovec, J.; and Liang, P. 2022. Linkbert: Pretraining language models with document links. In *ACL*.
- Yasunaga, M.; Ren, H.; Bosselut, A.; Liang, P.; and Leskovec, J. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *NAACL*.
- Yu, D.; Zhu, C.; Yang, Y.; and Zeng, M. 2022. Jacket: Joint pre-training of knowledge graph and language understanding. In *AAAI*.
- Zhang, T.; Ladhak, F.; Durmus, E.; Liang, P.; McKeown, K.; and Hashimoto, T. B. 2023. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.
- Zhang, X.; Bosselut, A.; Yasunaga, M.; Ren, H.; Liang, P.; Manning, C. D.; and Leskovec, J. 2022. GreaseLM: Graph REASONing Enhanced Language Models for Question Answering. In *ICLR*.
- Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Zhu, F.; Lei, W.; Wang, C.; Zheng, J.; Poria, S.; and Chua, T.-S. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.