LightPROF: A Lightweight Reasoning Framework for Large Language Model on Knowledge Graph

Tu Ao[1*], Yanhua Yu[1†*], Yuling Wang[2‡], Yang Deng[3], Zirui Guo[1], Liang Pang[5], Pinghui Wang[6], Tat-Seng Chua[4], Xiao Zhang[1], Zhen Cai[1]

[1]Beijing University of Posts and Telecommunications, China
[2]Hangzhou Dianzi University, China
[3]Singapore Management University, Singapore
[4]National University of Singapore, Singapore
[5]Institute of Computing Technology, Chinese Academy of Sciences, China
[6]Xi'an Jiaotong University, China
{aotu_bupt, yuyanhua, zrguo, xiao20010420, caizhen}@bupt.edu.cn,
wangyl0612@hdu.edu.cn, pangliang@ict.ac.cn, phwang@mail.xjtu.edu.cn, dcscts@nus.edu.sg

Abstract

Large Language Models (LLMs) have impressive capabilities in text understanding and zero-shot reasoning. However, delays in knowledge updates may cause them to reason incorrectly or produce harmful results. Knowledge Graphs (KGs) provide rich and reliable contextual information for the reasoning process of LLMs by structurally organizing and connecting a wide range of entities and relations. Existing KG-based LLM reasoning methods only inject KGs' knowledge into prompts in a textual form, ignoring its structural information. Moreover, they mostly rely on close-source models or open-source models with large parameters, which poses challenges to high resource consumption. To address this, we propose a novel **Light**weight and efficient **P**rompt learning-**R**eas**O**ning **F**ramework for KGQA (LightPROF), which leverages the full potential of LLMs to tackle complex reasoning tasks in a parameter-efficient manner. Specifically, LightPROF follows a "Retrieve-Embed-Reason" process, first accurately, and stably retrieving the corresponding reasoning graph from the KG through retrieval module. Next, through a Transformer-based Knowledge Adapter, it finely extracts and integrates factual and structural information from the KG, then maps this information to the LLM's token embedding space, creating an LLM-friendly prompt to be used by the LLM for the final reasoning. Additionally, LightPROF only requires training Knowledge Adapter and can be compatible with any open-source LLM. Extensive experiments on two public KGQA benchmarks demonstrate that LightPROF achieves superior performance with small-scale LLMs. Furthermore, LightPROF shows significant advantages in terms of input token count and reasoning time.

Introduction

With the emergence of more Large Language Models (LLMs), their continuously improving performance has brought substantial innovations to the field of Natural Language Processing (NLP) (Zhao et al. 2023; Touvron et al. 2023; Achiam et al. 2023; Team et al. 2023; GLM et al. 2024). The "*emergent abilities*" displayed under extensive training data and vast parameters allow LLMs to excel in complex zero-shot tasks (Wei et al. 2022a). Despite their effectiveness, LLMs often struggle with knowledge-intensive tasks due to limited task-specific prior knowledge and understanding capabilities (Sun et al. 2024). Additionally, the costly and time-consuming training process of LLMs presents considerable challenges in continuously updating and maintaining their knowledge bases.

To address the aforementioned challenges, it is crucial to enable LLMs to access a reliable and continuously updated knowledge base to support more accurate and interpretable reasoning (Pan et al. 2024). Knowledge Graphs (KGs) are ideally suited for this purpose, as they offer a structured semantic framework that delivers both accessible and timely information. Knowledge Graph Question Answering (KGQA), as a common knowledge-intensive task, existing work has explored methods for integrating LLMs with KGs to conduct KGQA reasoning (Jiang et al. 2023; Wu et al. 2023; Baek, Aji, and Saffari 2023; Wen, Wang, and Sun 2023; Sun et al. 2024; Guo et al. 2024). Broadly speaking, current KG-empowered LLM reasoning primarily involves retrieving information from KGs and incorporating the results into LLM input prompts, leveraging the LLMs' reasoning capabilities to address questions.

While LLMs reasoning on KGs holds great promise, several challenges remain: Firstly, the content of KGs is often represented directly as extensive textual content, which fails to effectively convey the rich logical relationships within their graph structure that are crucial for reasoning. In previous work, the content of KGs was presented in input prompts as multidimensional lists or in natural language form, making it difficult to clearly express the complex relationships and hierarchical structures within them. Secondly, retrieval and reasoning on KGs demand a high number of LLM calls and substantial LLM reasoning power. Previous work used

[*]These authors contributed equally.
[†]Corresponding author.
[‡]Corresponding author.

an iterative approach starting from the question entity, gradually obtaining information for reasoning. This increased the number of LLM calls, sacrificed reasoning efficiency, and diminished feasibility. The textual content describing KGs is vast, requiring not only a larger context window but also a more powerful LLM to ensure that no information is missed while avoiding the generation of incorrect answers in the redundant context.

In response to these challenges, we propose a Retrieve-Embed-Reason framework for LLMs, which is a novel **Light**weight and efficient **P**rompt learning-**R**eas**O**ning **F**ramework called **LightPROF**, designed to provide small-scale LLMs with stable retrieval and efficient reasoning capabilities. The framework is structured around three core components: the Retrieval, Embedding, and Reasoning modules. The Retrieval module utilizes relation as the fundamental retrieval unit and limits the retrieval scope based on the question's semantics to obtain the reasoning graph needed to answer the question. This approach not only boosts the accuracy and stability of retrieval but also considerably narrows the search space and reduces the need for frequent LLM invocations. Next, the Embedding module introduces a small and refined Transformer-based Knowledge Adapter that extracts and integrates the textual and structural information from the reasoning graph, generating representations perfectly suited for the LLM. This module offers an efficient and streamlined way of encoding information, addressing potential ambiguity and information redundancy while reducing the required input token count and context window size, resulting in a more accurate and efficient reasoning process. Finally, The Reasoning module combines the embedded representation vectors with carefully designed natural language prompts, allowing the LLM to derive the final answer. This design allows LightPROF to seamlessly support any open-source LLM and various KGs, requiring only the tuning of the Knowledge Adapter during training, without needing to update the costly and time-consuming LLM. Our contributions are summarized as follows:

- To the best of our knowledge, it is the first framework that transforms both the textual content and graph structure of KGs into embeddings used to prompt LLMs.
- We propose LightPROF, a lightweight and efficient prompt-learning reasoning framework that provides small-scale LLMs with stable retrieval and efficient reasoning capabilities, requiring far fewer training parameters compared to the LLM itself.
- Extensive experiments conducted on two KGQA datasets demonstrate the superiority of our proposed LightPROF, surpassing methods that use large-scale LLMs (such as LLaMa-2-70B, ChatGPT). Further analysis shows that LightPROF has significant efficiency advantages in terms of input token count and reasoning time.

Related Work

LLM Prompt Engineering. In expanding the capabilities of LLMs, prompt engineering has become a crucial technology. It maximizes the performance of LLMs across different applications and research domains by designing special task instructions (i.e., prompts) without altering model parameters (Sahoo et al. 2024; Saravia 2022). Many studies have been proposed on prompt engineering, spanning from zero-shot prompts (Radford et al. 2019) and few-shot prompts (Brown et al. 2020) to Chain-of-Thought (CoT) (Wei et al. 2022b) and its derivatives such as Tree-of-Thoughts (ToT) (Yao et al. 2024; Long 2023) and Graph-of-Thoughts (GoT) (Besta et al. 2024). Additionally, to address the issues of poor robustness and weak expressiveness in discrete prompts, many studies have explored soft prompts (Li and Liang 2021; Liu, Lee, and Yih 2022; Chen et al. 2024; Perozzi et al. 2024), demonstrating their effectiveness and feasibility in various NLP tasks and structured data representations. Proficiency in prompt engineering can enhance the understanding of the strengths and weaknesses of LLMs.

KG-based LLM Reasoning. KGs store a vast amount of explicit and structured knowledge that can effectively enhance the knowledge awareness of LLMs (Pan et al. 2024). Therefore, researchs have been conducted on using KGs to enhance LLMs' pre-training and generation techniques. Compared to natural language, KGs have clearer structured logic, which can better guide reasoning. Many studies use factual triples from KGs to construct corpora and employ various pre-training tasks to enhance the capabilities of LLMs (Zhang et al. 2023b; Dong et al. 2023a; Yu et al. 2022; Sun et al. 2021). However, this approach causes KGs to lose their advantages of interpretability and dynamism, and may also face catastrophic forgetting issues during the training process (Hu et al. 2023).

Therefore, constructing LLM prompts using factual information from KGs is a more flexible, convenient, and secure solution, and our method belongs to this kind of approach. For example, KAPING (Baek, Aji, and Saffari 2023) retrieves factual knowledge from KGs based on the semantic similarity of the question, adds it to the question as a prompt, and then uses the LLM to generate answers. KG-GPT (Kim et al. 2023) uses LLMs to perform reasoning on KG data through three steps: sentence segmentation, graph inference, and reasoning. StructGPT (Jiang et al. 2023) constructs an specialized interface for KG and proposed an Iterative Reading and Reasoning (IRR) framework for LLMs to solve KG-based tasks using this interface. ToG (Sun et al. 2024) utilizes LLMs to iteratively perform beam search on KGs, discovering reasoning paths and returning the most probable reasoning results. KnowledgeNavigator (Guo et al. 2024) enhances LLM reasoning by more efficiently and accurately retrieving external knowledge from KGs. While the aforementioned methods have demonstrated commendable performance, they uniformly represent KGs in natural language, which can introduce information redundancy and confusion, ultimately leading to incorrect reasoning.

Preliminaries

Knowledge Graph (KG) is a data structure that stores a vast quantity of knowledge in the form of triples: $\mathcal{G} = \{(h, r, t)|h, t \in \mathcal{E}, r \in \mathcal{R}\}$,where $\mathcal{E}$ and $\mathcal{R}$ denote the set of entities and relations, respectively. A triple $\langle h, r, t\rangle$ represents the existence of a relation r between the head

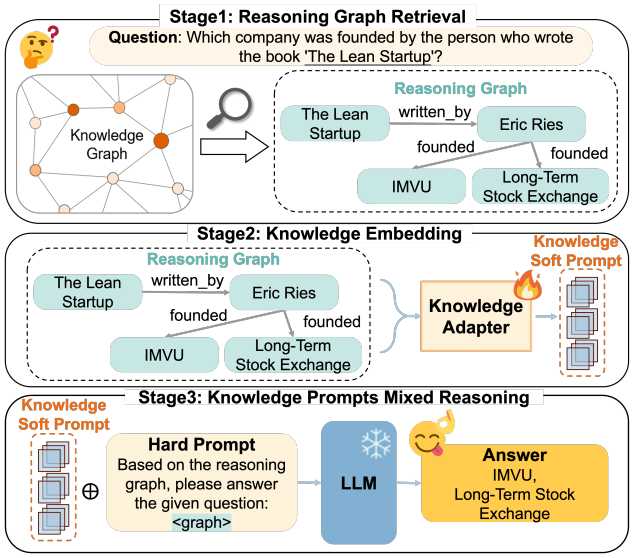

Figure 1: The architecture of our proposed Retrieve-Embed-Reason framework for knowledge graph question answer.

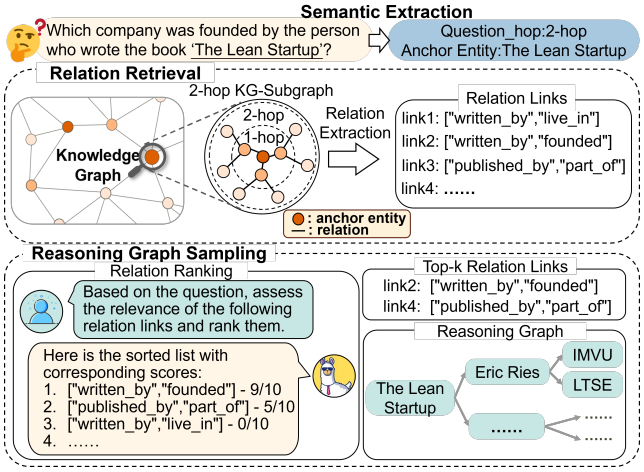

Figure 2: Three Steps Retrieval Module, including: semantic extraction, relation retrieval, and reasoning graph sampling.

entity h and the tail entity t.

Anchor Entities are a set of entities: $B = \{b_1, b_2, \ldots, b_K\}$ that are referenced in the KG-based question, where $b_k \in \mathcal{E}$ denotes the k-th entity in the question q.

Relation Link is a sequence of relations: $l = \{r_1, r_2, \ldots, r_J\}$, initiated by an anchor entity for J hop exploration, where $r_j \in \mathcal{R}$ denotes the j-th relation in the relation link.

Reasoning Path represents a concrete example of the relation link l within the KG of anchor entity $b_1 \in B$: $R_l = \{b_1, r_1, e_1, r_2, \ldots, r_M, e_M\}$, where $r_m \in l$ and $e_m \in \mathcal{E}$ denote the m-th relation and entity in R_l, respectively.

Methodology

We design the LightPROF framework, which achieves efficient complex KG problem reasoning under small-scale LLMs through precise retrieval and fine-grained structured data processing capabilities. As shown in Figure 1, our proposed Retrieve-Embed-Reason framework contains three stages: **Reasoning Graph Retrieval**, **Knowledge Embedding**, and **Knowledge Prompts Mixed Reasoning**.

Stage1: Reasoning Graph Retrieval

For the complex multi-hop KGQA task, the question *"How to efficiently, accurately, and stably retrieve information from a KG based on a question?"* is paramount. To address this critical issue, we devide the retrieval module into three steps: semantic extraction, relation retrieval, and reasoning graph sampling, as depicted in Figure 2.

Semantic Extraction. For a given question q, our goal is to extract relevant semantics (i.e., the number of hops h_q and anchor entities B) from the KG to narrow the retrieval scope while preserving the essential reasoning knowledge. This approach enables the retrieval and construction of a highly relevant and precise reasoning graph (Guo et al. 2024). Specifically, we fine-tune a pre-trained language model (PLM), such as BERT, to learn the number of hops h_q in KG required for reasoning, based on the semantic vector V_q of the query q. H is the maximum number of hops in the dataset, which can be framed as a classification task:

$$V_q = \mathrm{PLM}(q) \tag{1}$$

$$h_q = \arg\max_h P(h|V_q), h = 1, 2, \ldots, H. \tag{2}$$

Relation Retrieval. Relations in KGs describe the specific connections between two entities, providing semantic clarity for their interactions and substantially enriching the information content of KGs. Many studies currently utilize semantically rich relation links for KG reasoning tasks (Xiong, Hoang, and Wang 2017; Xu et al. 2022; Dong et al. 2023b). More crucially, relations in KGs demonstrate more stability and intuitiveness compared to the continuously changing and complex entities (Cai et al. 2023). To gather as much relevant knowledge as possible, we adopt a search for relation links in the KG based on anchor entities B and the predicted hop h_q. Specifically, the model first selects an anchor entity and then employs a constrained breadth-first search (BFS) with a depth limit of h_q. This process is designed to collect all relation links originating from the anchor entity B and extending up to a predetermined length of h_q.

Reasoning Graph Sampling. First, the retrieved relation links are fed into a LLM. Subsequently, the LLM calculates scores and ranks them according to their semantic relevance to the question q. Then, we select the top-k relevant links. Finally, we sample in KG based on the selected relation links, extracting multiple reasoning paths $\{R_1, R_2, \ldots, R_N\}$ to construct a refined reasoning graph, denoted as G_R.

Stage2: Knowledge Embedding

KGs typically encompass a rich array of complex structural information, including subgraph structures, relational pat-

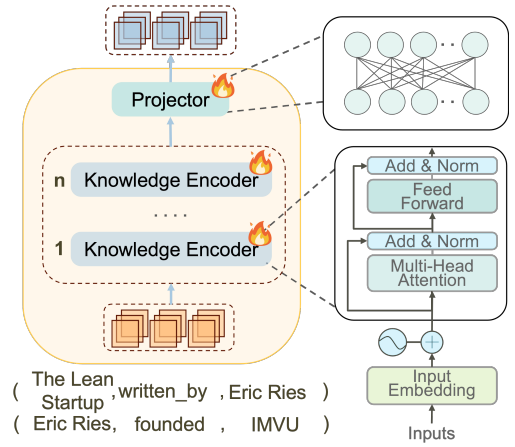

Figure 3: Illustration of the Knowledge Adapter and the schematic representation of its crucial components.

terns, and the relative relation between entities (Zhang et al. 2023a). Such structural information is essential for LLMs to gain a deep understanding of KGs. However, the natural language expression of KG structural information contains redundancy and confusion, which cannot directly reveal its inherent nature, thus impeding LLMs from effectively utilizing this information.

To address the aforementioned challenge, as inspired by (Chen et al. 2024; Perozzi et al. 2024), we propose a refined and compact Knowledge Adapter that can encode textual information in the reasoning graph while extracting its structural information, as illustrated in Figure 3. By combining textual information with structural details at a fine granularity, Knowledge Adapter aids the model in deeply comprehending the knowledge within the reasoning graph, enabling more precise reasoning.

Specifically, we assume that the reasoning graph $G_R = \{R_n\}_{n=1}^{N}$ is composed of N reasoning paths, each of which is decomposed into a set of triples $\mathcal{T}^n = \{(h_i^n, r_i^n, t_i^n)|i \in [1, h_q]\}$, where h_q is the number of reasoning hops. Subsequently, $\mathrm{Embed}(\cdot)$, i.e., BERT, is used to obtain the relational embedding e_i^r for each triple:

$$\mathbf{e}_i^r = \mathrm{Embed}(r_i^n). \tag{3}$$

We can obtain the entity embeddings e_i^h, e_i^t in the same way. Next, we aim to capture both the local and global interactions between each entity and relation. We first use $\mathrm{StructEmb}(\cdot)$ to encode the local structural information $\mathbf{s}_i$ of i-th triple in $\mathcal{T}^n$. Then, a linear layer $\mathrm{Linear}(\cdot)$ is used to aggregate the global structural information $\mathbf{z}^s$ from the entire reasoning path R_n:

$$\begin{aligned} \mathbf{s}_i &= \mathrm{StructEmb}(\mathbf{e}_i^h, \mathbf{e}_i^r, \mathbf{e}_i^t), \\ \mathbf{z}^s &= \mathrm{Linear}(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{h_q}). \end{aligned} \tag{4}$$

Additionally, to capture the textual information of the reasoning path R_n, we use $\mathrm{Fusion}(\cdot)$ to combine the text-level information of all entities and relations in R_n. We first obtain the combined text representation $\mathbf{z}^{t_h}$ of all head entities as follows:

$$\mathbf{z}^{t_h} = \mathrm{Fusion}(\mathbf{e}_1^h, \ldots, \mathbf{e}_{h_q}^h). \tag{5}$$

Then, the combined text representations of relations $\mathbf{z}^{t_r}$ and tail entities $\mathbf{z}^{t_t}$ can be obtained in the same way. Afterwards, these vectors are consolidated into a single vector $\mathbf{z}^t$ to represent the comprehensive textual information of the entire reasoning path R_n:

$$\mathbf{z}^t = f_c(\mathbf{z}^{t_h}, \mathbf{z}^{t_r}, \mathbf{z}^{t_t}), \tag{6}$$

where $f_c(\cdot)$ is the consolidation function. While $f_c(\cdot)$ can be complex neural networks or language models, to preserve the semantic integrity of the text and reduce the model's training complexity, we use a simple concatenation operation to form a composite vector that encapsulates all the textual information of the entire reasoning path.

Finally, we use $\mathrm{KnowledgeEncoder}(\cdot)$ to seamlessly integrate the obtained comprehensive textual information $\mathbf{z}^t$ and global structural information $\mathbf{z}^s$, deriving a fused representation of the reasoning path, as shown in Figure 3:

$$\mathbf{z}^f = \mathrm{KnowledgeEncoder}([\mathbf{z}^t, \mathbf{z}^s]) \tag{7}$$

In this way, the Knowledge Encoder can effectively encode each reasoning path in the reasoning graph into a single token, significantly improving the token utilisation efficiency of the LLM and enhancing the representational capacity of the reasoning paths. During the encoding process, the Knowledge Encoder captures not only rich textual information from the reasoning graph but also crucial structural information. Since the fused information $\mathbf{z}^f$ contains both textual and structural elements, the model can more fully understand the meaning embedded in each reasoning path during inference. This multidimensional information representation enhances the model's sensitivity to context , facilitating more effective deep semantic analysis and reasoning. Consequently, this information integration allows the model to more accurately capture the complex interactions between semantics and structure, thereby enhancing the accuracy and depth of reasoning.

By aggregating all paths $\{R_n\}_{n=1}^{N}$, we obtain the representational sequence $[\mathbf{z}_1^f, \mathbf{z}_2^f, \ldots, \mathbf{z}_N^f]$ of the reasoning graph G_R. Before inputting the sequence into the LLM, a dimension transformation is necessary. Due to the differences between the embedding space of the Knowledge Encoder and the input space of the LLM, directly using these tokens would be ineffective. Therefore, we develop a trainable projector $\Phi(\cdot)$, which maps these tokens into the token embedding space of the LLM. As a result, this process generates an input sequence suitable for the LLM, which we refer to as the knowledge soft prompt p_s:

$$p_s = \Phi([\mathbf{z}_1^f, \mathbf{z}_2^f, \ldots, \mathbf{z}_N^f]). \tag{8}$$

Here we set $\Phi(\cdot)$ as a two-layer multilayer perceptron. Following the aforementioned process, the Knowledge Adapter is able to encode the textual representation of the reasoning graph into the corresponding knowledge soft prompt. Importantly, all parameters of this adapter are derived from the parameters of the Knowledge Encoder and Projector, which are the only components requiring tuning during the LightPROF training process.

Stage3: Knowledge Prompts Mixed Reasoning

LLMs have acquired extensive knowledge through broad training on large corpora. However, despite their proficiency in general knowledge, LLMs show notable deficiencies in processing specialized knowledge, complex long logic chains, and multi-hop knowledge reasoning, which mainly stem from the limitations of their pre-training data. Additionally, although the knowledge base of LLMs can be expanded through retraining, this method is usually costly and time-consuming (Sun et al. 2024). More seriously, retraining may lead to catastrophic forgetting of existing knowledge in the model (Zhang et al. 2024). Thus, this presents certain challenges in keeping LLMs' knowledge up-to-date. To avoid the aforementioned challenges, we freeze the parameters of the LLM during the LightPROF training process and use a combination of soft prompts and hard prompts to guide the model to answer questions more precisely and efficiently, which can be seen in Figure 1.

Specifically, the input to the LLM is organized in a chat format, where instructions and questions are combined using carefully designed text templates, which we call hard prompts. During the encoding phase of the LLM, we insert the knowledge soft prompt, representing the reasoning graph, into specific locations of the hard prompt to effectively inject external knowledge, as shown in Figure 1. This approach allows the LLM to autonomously and accurately answer questions based on the given input content without the need for parameter updates. By this method, we not only maintain the stability of the model but also enhance its performance and efficiency within specific knowledge domains.

The training objective of LightPROF is to maximize the likelihood of generating correct answers $\mathcal{A}$ for all samples in the dataset $\mathcal{D}$. This can be compatible with the task of next-token prediction, a fundamental method for training generative models. The training goal can be articulated as:

$$\arg\max_{\mathcal{A}} P_{\text{llm}}(\mathcal{A}|p_{\text{p}}) = \sum_{\mathcal{D}} \sum_{t=1}^{|\mathcal{A}|} \log P_{\text{llm}}(a_t|a_{1:t-1}, p_{\text{h}}, p_{\text{s}}), \tag{9}$$

where p_{p} is the input sequence that includes both hard prompt p_{h} and soft prompt p_{s}, and $a_t(t = 1, 2, \ldots, |\mathcal{A}|)$ is the t-th token of the output sequence. Notably, when $t = 1$, $a_{1:t-1}$ is the model's beginning-of-sequence (BOS) token.

Experiments

In this experiment, we will thoroughly discuss the following questions. **Q1**: How significantly can LightPROF enhance LLMs' performance in KGQA tasks? **Q2**: Can LightPROF be integrated with different LLM backbones to enhance performance? **Q3**: Can LightPROF achieve efficient input and stable output with small-scale LLMs?

Datasets

We train and evaluate LightPROF's multi-hop reasoning capabilities on two public datasets based on the Freebase knowledge graph (Bollacker et al. 2008): WebQuestionsSP(WebQSP) (Yih et al. 2016) and ComplexWebQuestions(CWQ) (Talmor and Berant 2018). Based on previous works, we utilize match accuracy (Hits@1) to evaluate whether the model's top-1 answer is correct.

- **WebQSP** is a benchmark with fewer questions but a larger knowledge graph, consisting of 4,737 questions. Each question includes a topic entity, a reasoning chain, and a SPARQL query to find the answer. The answer entity requires up to 2-hop reasoning on the Freebase.
- **CWQ** is a benchmark specifically designed for complex knowledge graph question answering research. It includes 34,689 question-answer pairs, built upon the WebQSP dataset. It involves automatically creating more complex SPARQL queries and generating corresponding natural language questions, thereby creating a wide and diverse range of question types. These questions require up to 4-hop reasoning on Freebase.

Baselines

We consider three types of baseline methods: full fine-tuning methods, vanilla LLM methods, and LLM+KGs methods. The full fine-tuning methods include KV-Mem (Miller et al. 2016), EmbedKGQA (Saxena, Tripathi, and Talukdar 2020), TransferNet (Shi et al. 2021), NSM (He et al. 2021), KGT5 (Saxena, Kochsiek, and Gemulla 2022), GraftNet (Sun et al. 2018), PullNet (Sun, Bedrax-Weiss, and Cohen 2019), UniKGQA (Jiang et al. 2022). Vanilla LLM methods include LLaMa series models (Touvron et al. 2023). LLM+KGs methods include StructGPT (Jiang et al. 2023), ToG (Sun et al. 2024), KnowledgeNavigator (Guo et al. 2024), AgentBench (Liu et al. 2024). Notably, to ensure fair comparisons, the LLM+KGs methods we select do not involve fine-tuning the LLMs, i.e., all of them are zero-shot methods without any training of the LLM.

Implementation

To demonstrate the plug-and-play convenience and parameter efficiency of LightPROF, we conduct experiments on two small-scale language models in the LLaMa series: LLaMa-7B-chat (Touvron et al. 2023) and LLaMa-8B-Instruct[1]. The model was optimized over one training epoch with a batch size of 4. The initial learning rate was set at 2e-3, adjusted using a cosine annealing schedule to enhance the model's learning efficiency during training. All experiments are conducted using the PyTorch toolkit on NVIDIA A800 GPU.

The Knowledge Encoder module is based on the BERT model. The module includes a two-layer MLP Projector that maps dimensions to the LLM's input dimension.

Q1: Performance Comparison

Main Result. We evaluate LightPROF against three categories of baseline methods: full fine-tuning, vanilla LLM, and LLM+KGs approaches. As illustrated in Table 1, LightPROF not only excels in simple questions but also demonstrates high performance in scenarios requiring deep reasoning and complex query handling. Specifically, LightPROF significantly surpasses the state-of-the-art model on the WebQSP dataset (83.7% vs. 75.1%) and also excels on the more

[1]https://ai.meta.com/blog/meta-llama-3/

complex CWQ dataset (59.3% vs. 57.6%). These outcomes validate our framework's excellent capability in addressing KGQA tasks, emphasizing LightPROF's efficacy in managing multi-hop and complex challenges.

Compared to vanilla LLMs and LLM+KGs methods that utilize plain text prompts, LightPROF's significant improvement indicates that soft prompts produced by the Knowledge Adapter can effectively encapsulate more complex structural knowledge than discrete text, being concise, informative, and highly expressive, thus enhancing LLM's understanding of KG information. It is noteworthy that our framework outperforms other large-scale models in all experimental conditions. For example, our framework excels, particularly in reasoning through complex problems, compared to ToG (Sun et al. 2024) with LLaMa2-70B-Chat and StructGPT (Jiang et al. 2023) with ChatGPT. Additionally, even with the smaller LLaMa2-7b version, our framework competes effectively with other large-scale models, underscoring the efficiency and optimization of our framework's design.

Methods	WebQSP	CWQ
KV-Mem	46.7	18.4
EmbedKGQA	66.6	45.9
NSM	68.7	47.6
KGT5	56.1	36.5
GraftNet	66.4	-
PullNet	68.1	-
TransferNet	71.4	48.6
UniKGQA	<u>75.1</u>	50.7
LLaMa2-7B-Chat	61.4	31.5
LLaMa2-70B-Chat	57.4	39.1
ToG (LLaMa2-70B)	68.9	<u>57.6</u>
StructGPT (ChatGPT)	72.6	54.3
AgentBench	47.8	24.8
KnowledgeNavigator(LLaMa2-70B)	71.8	-
LightPROF (LLaMa3-8B)	**83.8**	**59.3**
LightPROF (LLaMa2-7B)	71.2	48.5

Table 1: Performance comparison of LightPROF with baselines on the two datasets. Bold and underlined typefaces indicate optimal and sub-optimal methods, respectively.

Methods	WebQSP	CWQ
LightPROF	**83.77**	**59.26**
LightPROF w/o Struct	82.36	58.05
LightPROF w/o Train	80.37	55.63
LightPROF w/ Random Retrieve	53.44	46.84

Table 2: Model ablation study of our LightPROF framework.

Ablation Study. An ablation study is performed on LightPROF to investigate the specific effects of the Knowledge Adapter on KGQA task performance. We examine three variants: (1) *w/o Struct*, removing the structural information included in the knowledge embedding process, (2) *w/o Train*, without training the Knowledge Encoder, and (3) *w/ Random Retrieve*, randomly retrieve reasoning paths from KGs. The results are displayed in Table 2.

The results indicate that the integration of structural information is crucial for the model's understanding and handling of entities and relationships in complex queries. The incorporation of structural information significantly enhances the model's utilization efficiency of data in the knowledge graph. Continuous training of the Knowledge Encoder is also essential for enhancing the model's comprehension and generation of knowledge representations. This training process notably improves the model's capability to encode complex structural knowledge, allowing it to more accurately respond to queries rooted in deep knowledge. Moreover, randomly retrieved reasoning paths can cause significant damage to performance, highlighting the importance of an accurate and stable retrieval module.

Additionally, we explore different structural encoders. The structural encoder used in our framework encodes triples as Head (H) + Relation (R) - Tail (T). Results in Table 3 show that the performance of the H+R+T encoding method slightly declines due to its inability to distinguish the order of the triples, *e.g.*, the structural information derived from (Eric Ries, founded, IMVU) and (IMVU, founded, Eric Ries) is identical, reducing the model's capacity to understand structural information. In contrast, LightPROF can better capture structural information within the reasoning graph and integrate it at a finer granularity, enhancing the model's understanding, particularly in scenarios involving complex structured data reasoning.

Methods	WebQSP	CWQ
LightPROF(H+R+T)	83.68	58.32
LightPROF(H+R-T)	**83.77**	**59.26**

Table 3: Performance impact of different structure encoder in LightPROF.

Q2: Plug-and-Play

For our framework, any open-source LLM capable of accepting token embedding inputs is suitable. In this section, we evaluate the effectiveness of integrating different LLMs within LightPROF. As illustrated in Table 5, the results demonstrate that the LightPROF framework significantly enhances the performance of integrated LLMs, regardless of the baseline performance of the original models. LightPROF enhances the model's capability to address complex KG questions through effective integration and optimization of structured data. This plug-and-play integration strategy does not require costly fine-tuning of LLMs, making it particularly suitable for quickly enhancing existing models' performance on KGQA task.

Question	what drugs lindsay lohan abuse?
Answer	["Alcoholic beverage", "Cocaine"]
StructGPT	The relevant relation: celebrities.celebrity.substance_abuse_problems The possible constraints: celebrities.substance_abuse_problem.substance: Alcoholic beverage The final answers: Alcoholic beverage
LightPROF	Number of Hops: 2 Relation Links: ['base.popstra.celebrity.substance_abuse', 'base.popstra.substance_abuse.substance'] - 9/10 ['base.popstra.celebrity.substance_abuse', 'base.popstra.substance_abuse.abuser'] - 9/10 Based on the knowledge graphs, please answer the given question. Please keep the answer as simple as possible and return all the possible answers as a list. knowledge graphs: $< graph >$ ["Cocaine", "Alcoholic beverage"]

Table 4: Case Study of LightPROF and StructGPT on the WebQSP Dataset.

Methods	WebQSP	CWQ
Llama2-7b	61.36	31.49
LightPROF (Llama2-7b)	71.19	48.48
Llama3-8b	66.83	48.87
LightPROF (Llama3-8b)	**83.77**	**59.26**

Table 5: The performance of integrating various LLMs into the LightPROF framework.

Q3: Efficient Input and Stable Output

Efficiency Results. A series of efficiency tests are conducted to compare the performance of LightPROF and StructGPT (Jiang et al. 2023) when processing the WebQSP dataset. Specifically, the models' runtime, the total number of input tokens, and the average Number of tokens Per Request (NPR) are measured, with results presented in Table 6. The table shows that LightPROF is more time-efficient when processing the same dataset, with a 30% reduction in time cost (1:11:49 vs. 1:42:12). Regarding the total number of input tokens, LightPROF and StructGPT show a significant difference (365,380 vs. 24,750,610), demonstrating that LightPROF is more economical in input processing, reducing token usage by approximately 98%. Furthermore, LightPROF's NPR value is 224, significantly lower than StructGPT's 6400. This comparison further highlights LightPROF's advantage in the number of tokens needed per request, showcasing its more precise and resource-efficient handling of each request, validating LightPROF's effectiveness when integrating small-scale LLMs.

Methods	TimeCost	TokenUsed	NPR
LightPROF	**1:11:49**	**365,380**	**224**
StructGPT	1:42:12	24,750,610	6400

Table 6: Efficiency performance of LightPROF and StructGPT on Llama-3-8b. NPR represents the average number of tokens per request.

Case Study. As shown in Table 4, we validate LightPROF's efficient input and stable output capabilities when using small-scale LLMs by comparing its performance with StructGPT to answer complex queries about Lindsay Lohan's drug abuse. The results show that LightPROF not only accurately identify and comprehensively answer the query, but also demonstrate deeper reasoning pathways and overall scoring. In contrast, although StructGPT handled the relevant questions, it failed to fully capture all related answers. Interestingly, we found that LightPROF can consistently generate output that includes only the answers and uses fewer input tokens and less reasoning time. This suggests that LightPROF can effectively integrate and precisely output complex information from knowledge graphs, demonstrating its reliability and practicality in efficiently and accurately handling complex KGQA tasks.

Conclusion

In this paper, we introduce the LightPROF framework, which accurately retrieves and efficiently encodes KGs to enhance LLM reasoning. To effectively narrow the retrieval scope, LightPROF incrementally samples the KG using stable relationships as units. To achieve efficient reasoning on LLMs with fewer parameters, we develop a delicate Knowledge Adapter that can effectively parse graph structures and perform fine-grained information integration, thus condensing the reasoning graph into a smaller number of tokens and achieving comprehensive alignment with the LLM's input space through the Projector. Experimental results show that our framework outperforms other baseline methods, particularly those involving large-scale language models. In comparison to other methods based exclusively on text, our knowledge soft prompts integrate a more comprehensive range of structural and textual information, making them more easily understood by LLMs. In future work, we plan to explore 1) KG encoders with stronger generalization and compatibility, and design an encoder that can be applied to unseen KG data without retraining. 2) A unified cross-modal encoder capable of encoding multimodal KGs.

Acknowledgments

The research was supported by the National Natural Science Foundation of China (Grant No. U22B2019).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Baek, J.; Aji, A. F.; and Saffari, A. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *arXiv preprint arXiv:2306.04136*.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 17682–17690.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, 1247–1250. New York, NY, USA: Association for Computing Machinery. ISBN 9781605581026.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cai, B.; Xiang, Y.; Gao, L.; et al. 2023. Temporal Knowledge Graph Completion: A Survey. In Elkind, E., ed., *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 6545–6553. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Chen, R.; Zhao, T.; Jaiswal, A.; Shah, N.; and Wang, Z. 2024. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*.

Dong, G.; Zhao, J.; Hui, T.; Guo, D.; et al. 2023a. Revisit input perturbation problems for llms: A unified robustness evaluation framework for noisy slot filling task. In *CCF International Conference on Natural Language Processing and Chinese Computing*, 682–694. Springer.

Dong, W.; Sun, S.; Zhao, J.; and Zhang, N. 2023b. Knowledge graph relation reasoning with variational reinforcement network. *Information Fusion*, 100: 101900.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; et al. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*.

Guo, T.; Yang, Q.; Wang, C.; Liu, Y.; Li, P.; et al. 2024. Knowledgenavigator: Leveraging large language models for enhanced reasoning over knowledge graph. *Complex & Intelligent Systems*, 1–14.

He, G.; Lan, Y.; Jiang, J.; Zhao, W. X.; and Wen, J.-R. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM international conference on web search and data mining*, 553–561.

Hu, L.; Liu, Z.; Zhao, Z.; Hou, L.; Nie, L.; and Li, J. 2023. A survey of knowledge enhanced pre-trained language models. *IEEE Transactions on Knowledge and Data Engineering*.

Jiang, J.; Zhou, K.; Ye, K.; Zhao, X.; Wen, J.-R.; et al. 2023. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jiang, J.; Zhou, K.; Zhao, X.; and Wen, J.-R. 2022. UniKGQA: Unified Retrieval and Reasoning for Solving Multi-hop Question Answering Over Knowledge Graph. In *The Eleventh International Conference on Learning Representations*.

Kim, J.; Kwon, Y.; Jo, Y.; and Choi, E. 2023. Kg-gpt: A general framework for reasoning on knowledge graphs using large language models. *arXiv preprint arXiv:2310.11220*.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4582–4597.

Liu, C.-L.; Lee, H.-y.; and Yih, W.-t. 2022. Structured prompt tuning. *arXiv preprint arXiv:2205.12309*.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; et al. 2024. AgentBench: Evaluating LLMs as Agents. In *The Twelfth International Conference on Learning Representations*.

Long, J. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; et al. 2016. Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1400–1409.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; et al. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Perozzi, B.; Fatemi, B.; Zelle, D.; Tsitsulin, A.; Kazemi, M.; Al-Rfou, R.; and Halcrow, J. 2024. Let your graph do the talking: Encoding structured data for llms. *arXiv preprint arXiv:2402.05862*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Sahoo, P.; Singh, A. K.; Saha, S.; Jain, V.; et al. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Saravia, E. 2022. Prompt Engineering Guide. *https://github.com/dair-ai/Prompt-Engineering-Guide*.

Saxena, A.; Kochsiek, A.; and Gemulla, R. 2022. Sequence-to-Sequence Knowledge Graph Completion and Question Answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2814–2828.

Saxena, A.; Tripathi, A.; and Talukdar, P. 2020. Improving multi-hop question answering over knowledge graphs using

knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 4498–4507.

Shi, J.; Cao, S.; Hou, L.; Li, J.; and Zhang, H. 2021. TransferNet: An Effective and Transparent Framework for Multi-hop Question Answering over Relation Graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4149–4158.

Sun, H.; Bedrax-Weiss, T.; and Cohen, W. 2019. PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2380–2390.

Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; et al. 2018. Open Domain Question Answering Using Early Fusion of Knowledge Bases and Text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4231–4242.

Sun, J.; Xu, C.; Tang, L.; Wang, S.; Lin, C.; et al. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*.

Sun, Y.; Wang, S.; Feng, S.; Ding, S.; Pang, C.; et al. 2021. ERNIE 3.0: LARGE-SCALE KNOWLEDGE ENHANCED PRE-TRAINING FOR LANGUAGE UNDERSTANDING AND GENERATION. *arXiv preprint arXiv:2107.02137*.

Talmor, A.; and Berant, J. 2018. The Web as a Knowledge-Base for Answering Complex Questions. In Walker, M.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 641–651. New Orleans, Louisiana: Association for Computational Linguistics.

Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wen, Y.; Wang, Z.; and Sun, J. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *arXiv preprint arXiv:2308.09729*.

Wu, Y.; Hu, N.; Qi, G.; Bi, S.; Ren, J.; et al. 2023. Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering. *arXiv preprint arXiv:2309.11206*.

Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 564–573. Association for Computational Linguistics.

Xu, X.; Zhang, P.; He, Y.; Chao, C.; and Yan, C. 2022. Subgraph Neighboring Relations Infomax for Inductive Link Prediction on Knowledge Graphs. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 2341–2347. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; et al. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Yih, W.-t.; Richardson, M.; Meek, C.; Chang, M.-W.; and Suh, J. 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In Erk, K.; and Smith, N. A., eds., *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 201–206. Berlin, Germany: Association for Computational Linguistics.

Yu, D.; Zhu, C.; Yang, Y.; and Zeng, M. 2022. Jaket: Joint pre-training of knowledge graph and language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11630–11638.

Zhang, M.; Sun, M.; Wang, P.; Fan, S.; Mo, Y.; Xu, X.; et al. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. In *Proceedings of the ACM on Web Conference 2024*, 1003–1014.

Zhang, Y.; Chen, Z.; Zhang, W.; and Chen, H. 2023a. Making Large Language Models Perform Better in Knowledge Graph Completion. *CoRR*, abs/2310.06671.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; et al. 2023b. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.