Explore then Determine: A GNN-LLM Synergy Framework for Reasoning over Knowledge Graph

Guangyi Liu
EE, Tsinghua University
Beijing, China
liugy20@mails.tsinghua.edu.cn

Yongqi Zhang
DSA, HKUST (GZ)
Guangzhou, China
yongqizhang@hkust-gz.edu.cn

Yong Li
EE, Tsinghua University
Beijing, China
liyong07@mail.tsinghua.edu.cn

Quanming Yao*
EE, Tsinghua University
Beijing, China
qyaoaa@mail.tsinghua.edu.cn

Abstract

The task of reasoning over Knowledge Graphs (KGs) poses a significant challenge for Large Language Models (LLMs) due to the complex structure and large amounts of irrelevant information. Existing LLM reasoning methods overlook the importance of compositional learning on KG to supply with precise knowledge. Besides, the fine-tuning and frequent interaction with LLMs incur substantial time and resource costs. This paper focuses on the Question Answering over Knowledge Graph (KGQA) task and proposes an Explore-then-Determine (EtD) framework that synergizes LLMs with graph neural networks (GNNs) for reasoning over KGs. The Explore stage employs a lightweight GNN to explore promising candidates and relevant fine-grained knowledge to the questions, while the Determine stage utilizes the explored information to construct a knowledge-enhanced multiple-choice prompt, guiding a frozen LLM to determine the final answer. Extensive experiments on three benchmark KGQA datasets demonstrate that EtD achieves state-of-the-art performance and generates faithful reasoning results.

1 Introduction

Question Answering over Knowledge Graph (KGQA) is a task that aims at answering questions expressed in natural language from entities within a given knowledge graph (KG). KGs, storing a vast amount of facts in the form of triples (e.g., Wikidata [1], YAGO [2], and NELL [3]), are vital for a variety of applications due to their capacity to deliver explicit knowledge. This importance has led to KGQA gaining significant attention from researchers, given its crucial role in numerous intelligent applications such as Apple Siri and Microsoft Cortana [4]. As a complex and difficult task, KGQA usually requires comprehensive question analysis and multi-hop reasoning on the KG to deduce the correct answers [4].

Classical works [5–8] (e.g., NSM [7] and UniKGQA [8]), usually retrieves a question-specific graph delivering the information related to the question, and then focus on designing different modules to generate answer based on the retrieved graph. Recently, large language models (LLMs) [9–12], have shown strong power in natural language understanding and processing tasks. But in KGQA task, which predicts answers from KGs, LLMs struggle to excel due to the lack of factual knowledge and the problem of hallucinations [13, 14]. Many efforts have been made to enhance the reasoning

*Correspondence is to Q.Yao.

Preprint. Under review.

ability of LLMs over KGs. However, they all overlook the importance of compositional learning on the graph, which enables the extraction of useful patterns and relationships to supply with precise knowledge related to the question. The text-retrieval methods (e.g.,KAPING [15]), directly retrieve triplets from KGs to construct prompts for LLMs, which often results in redundant information in retrieving [16]. The step-by-step reasoning methods like StructGPT [17], ToG [18], guide LLMs to reason over KGs across multiple steps. Since LLMs lack the inherent capacity to comprehend graph structures, it is often challenging for them to perform effective topological reasoning on graphs [14]. Furthermore, the usage of LLMs, whether through training, fine-tuning, or frequent interaction, entails significant time and resource costs, especially in the KGQA task that requires multi-hop reasoning on a large KG.

In this paper, we propose an Explore-then-Determine framework, named EtD, that synergizes LLM with GNN for effective, efficient and faithful reasoning over KG. The Explore stage is composed of an LLM-empowered GNN module that explores candidates along with relevant knowledge, and adaptively filters irrelevant information in KG. The Determine stage leverages a frozen LLM to determine the final answer with a knowledge-enhanced multiple-choice prompt, which can be seamlessly integrated with any off-the-shelf LLMs during inference, without the burdensome training and fine-tuning. This approach leverages the GNN to perform compositional learning over the KG, identifying the most promising candidates and fine-grained knowledge, while the LLM aligns the question linguistically with the retrieved information to determine the final answer. Two modules are synergized together to benefit each other, as a result prompting the KGQA task. The contributions are summarized as follows:

- We propose a novel Explore-then-Determine framework, which explores promising candidates and fine-grained knowledge from KG for LLM's final answer determination, harnessing both the precision of compositional learning and the prowess of language understanding.
- We design a lightweight GNN model empowered by LLM for precise and efficient KG exploration, and propose a knowledge-enhanced multiple-choice prompt for faithful answer determination.
- Extensive experiments on KGQA show that our method effectively combines explicit knowledge on KG with implicit knowledge in LLM, outperforming state-of-the-art methods, and achieving efficient and interpretable reasoning over KG.

2 Related Work

Large Language Models (LLMs). LLMs [9–12], through pretraining on massive text data, have attained potent language processing ability. They excel in diverse question-answering tasks and demonstrate proficiency in zero-shot learning [15], generating answers based solely on textual prompts. To further improve their performance, some approaches retrieve relevant samples from the training dataset and meticulously design prompts under few-shot learning to stimulate LLMs' capabilities, such as CoT [19], Self-Consistency [20], and ToT [21]. However, these methods can not compensate for the lack of knowledge in LLMs, and performance gains are limited when applying CoT methods with smaller models ($\sim$100B parameters)[16].

KGQA. Classical *embedding-based methods* embed entities on KG and design a scoring function to rank them based on their relevance to the question [22, 23]. But these methods are too shallow to capture high-order connectivity and complex semantics. *GNN-based* works [5, 24, 7, 8] usually follow a retrieval-and-reasoning paradigm, which first retrieves a relatively small question-related subgraph [25] and then use GNNs [26, 27] on the subgraph to find the answer entities. However, they fall short in the understanding of natural language, potentially resulting in semantically incorrect answers. Furthermore, they do not truly unify the subgraph retrieval and reasoning, still separating them into two distinct stages instead of simultaneous processes.

Recently, many works aim to improve the reasoning ability of LLMs over KG to answer complex questions [14]. *Semantic parsing methods* use LLMs to generate query languages of the question and then execute them on KGs to obtain answers [28–31]. But they often require fine-tuning of LLMs or the search for a large number of similar examples as prompts. Moreover, the query languages generated often encounter issues of non-executability, yielding no answers [32]. *Retrieval-augmented methods* retrieve triples from KGs as extra knowledge for LLMs to output the final answers [33, 15]. These methods tend to neglect the significance of compositional knowledge for precise reasoning. As

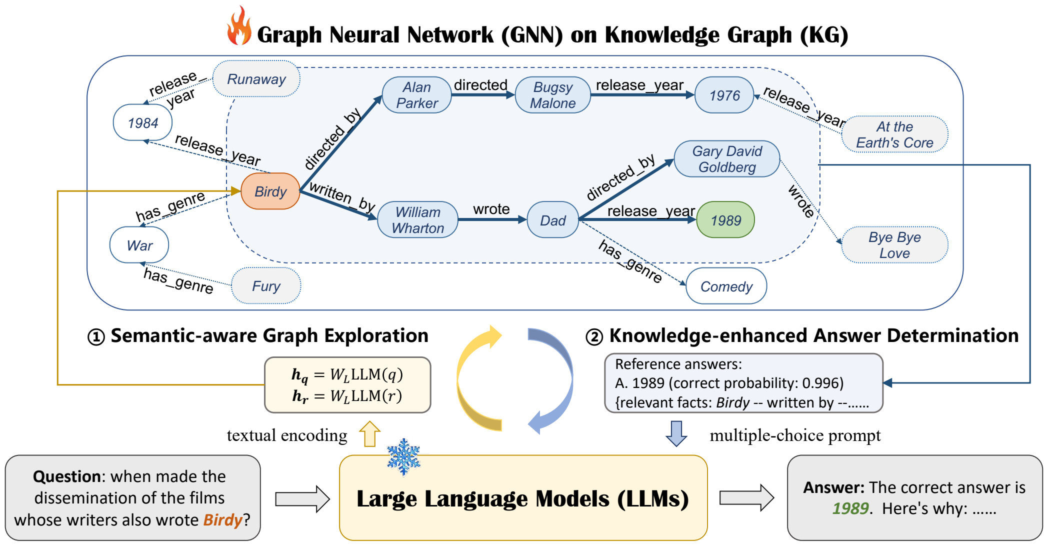


Figure 1: Illustration of the Explore-then-Determine (EtD) framework that synergizes LLMs with GNNs for reasoning over KGs.

a result, the retrieved knowledge often contains excess and irrelevant information [34]. *Agent-based methods* view LLMs as agents and establish an information interaction mechanism between KG and LLMs to iteratively deduce the answer to the question [17, 18, 35]. They attempt to guide LLMs to reason on graph, but the depth and breadth of this reasoning are limited as LLMs inherently lack the ability to comprehend graph structures. Additionally, frequent interactions with LLMs are inflexible and entail high costs.

3 Proposed Method

3.1 Overview

Given a natural language question q and a KG $\mathcal{G} = \{(e_s, r, e_o)|e_s, e_o \in \mathcal{V}, r \in \mathcal{R}\}$, where $\mathcal{V}$ is the set of entities (nodes) and $\mathcal{R}$ is the set of relation types, the task of *Question Answering over Knowledge Graph (KGQA)* is to find a function $\mathcal{F}(q, G)$ that predicts the answer entities $e_a \in \mathcal{V}$ of q over KG $\mathcal{G}$. For each question q, the involved topic entity $e_q \in \mathcal{V}$ in KG is given.

As a challenging task, KGQA typically requires precise mining of question-related compositional knowledge (such as a list of ordered triplets) in the KG, and demands text comprehension and matching capabilities between questions and entities in KG. However, LLMs is less effective in acquiring the promising candidate entities and relevant knowledge in the KG. Inspired by how humans make decisions when faced with hard tasks by identifying various possible alternatives before choosing the best one, we propose an explore-then-determine (EtD) framework, which adopts exploration module to provide promising candidates and fine-grained knowledge for LLM, and employs determination module to guide LLM to generate final answer.

The proposed framework is shown in Figure 1, which contains two components: (1) semantic-aware graph exploration; (2) knowledge-enhanced answer determination. For the first part, to precisely perceive the question-related compositional knowledge on the graph, we design an LLM-empowered GNN module to adaptively explore the related candidates and relevant knowledge of the given question from KG, i.e., $(\mathcal{C}_q, \mathcal{K}_q) = f_{\text{exp}}(q, \mathcal{G})$. For the second part, to effectively leverage the information explored in the first stage, we carefully design a knowledge-enhanced multiple-choice prompt, guiding the LLM to determine final answer based on both the explicit knowledge from KG and the implicit knowledge inside LLM, i.e., $e_a = g_{\text{det}}(q, \mathcal{C}_q, \mathcal{K}_q)$.

3.2 Semantic-Aware Graph Exploration

In this section, we aim to extract the promising candidates and relevant compositional knowledge of given question from KG, such that we can provide the LLM with supportive information at once,

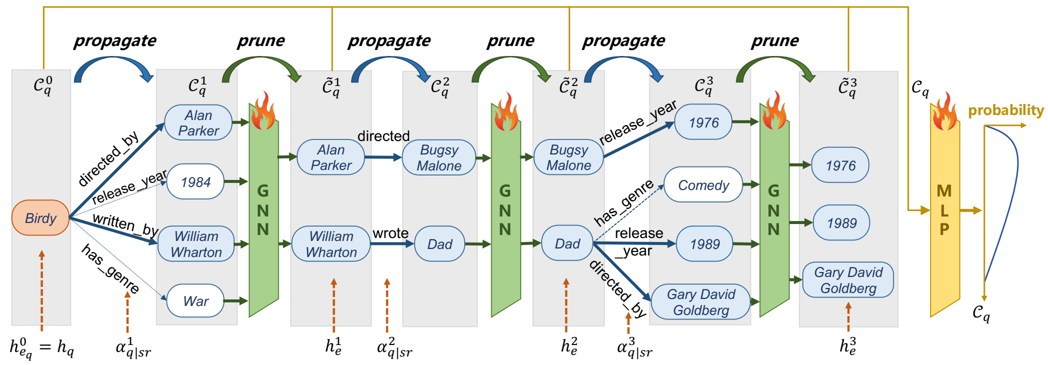


Figure 2: Illustration of semantic-aware graph exploration. We start exploration from topic entity *Birdy*. In each step, we firstly get the unpruned candidate set $\mathcal{C}_q^\ell$, calculate the attention weights $\alpha_{q|sr}^\ell$ of different edges, prune several irrelevant entities (in white), and update candidate set $\widetilde{\mathcal{C}}_q^\ell$ (in blue). The representations are propagated from entities in $\widetilde{\mathcal{C}}_q^{\ell-1}$ to $\widetilde{\mathcal{C}}_q^\ell$ through an one-layer GNN.

avoiding the high costs associated with multiple interactions. Due to a large amount of irrelevant information present in the KG, we design an adaptive propagation GNN that, with the semantic representation capabilities provided by the LLM, can automatically prune irrelevant information in KG, enabling precise, efficient and interpretable reasoning.

Semantic-aware pruning. Given the topic entity e_q involved in the question q, we aim to gradually explore the potential answers starting from e_q on the KG. Instead of directly using LLM to explore the candidates, we design a lightweight neural network empowered by LLM, serving as a graph explorer. Specifically, we initialize the candidate set $\mathcal{C}_q^0 \equiv \{e_q\}$. The representation of e_q is initialized as the question encoding, i.e., $\boldsymbol{h}_{e_q}^0 = \boldsymbol{h}_q = \boldsymbol{W}_L \cdot \text{LLM}(q)$, where $\text{LLM}(\cdot)$ computes the average embedding in the last layer of Llama2-13B, and $\boldsymbol{W}_L \in \mathbb{R}^{d \times d_L}$ is a learnable weighting matrix mapping the representation to a lower dimension d. The representations of other entities are initialized as $\mathbf{0}$. Assume in the ℓ-th step ($\ell = 1, 2, \ldots, L$), we have explored a set $\mathcal{C}_q^{\ell-1}$ of current candidates. The set is then expanded by propagating to the neighbors of entities in $\mathcal{C}_q^{\ell-1}$, resulting in an updated set $\mathcal{C}_q^\ell = \{e_o : (e_s, r, e_o) \in \mathcal{G}, e_s \in \mathcal{C}_q^{\ell-1}\}$.

Since KG contains a lot of information irrelevant to the question, we should filter the irrelevant edges during exploring such that the size of $\mathcal{C}_q^\ell$ will not grow exponentially. In the ℓ-th step, we calculate the attention weight $\alpha_{q|sr}^\ell$, to measure the importance of each edge (e_s, r, e_o) with $e_s \in \mathcal{C}_q^{\ell-1}$ as:

$$\alpha_{q|sr}^\ell = \sigma\left(\boldsymbol{W}_s^\ell \boldsymbol{h}_s^{\ell-1} + \boldsymbol{W}_r^\ell \boldsymbol{h}_r + \boldsymbol{W}_q^\ell \boldsymbol{h}_q + \boldsymbol{W}_{qr}^\ell (\boldsymbol{h}_r \odot \boldsymbol{h}_q)\right), \tag{1}$$

where σ is the sigmoid function, $\boldsymbol{W}^\ell$'s in $\mathbb{R}^{1 \times d}$ are learnable weight matrices, and $\odot$ is the Hadamard product of vectors. The representations $\boldsymbol{h}_q$ and $\boldsymbol{h}_r$ are textual encodings mapped from an LLM (e.g., Llama2-13B) such that the semantic relevance of question with the current edge can be measured. Note that the LLM used as text encoding will not be updated. The representation $\boldsymbol{h}_s^{\ell-1} \in \mathbb{R}^d$ of head entity e_s contains the knowledge learned in the $(\ell-1)$-th step.

By incorporating the representations of question q, relations, and entities within the triplets, we can utilize the powerful semantic modeling capabilities of LLM. Meanwhile, setting different weight matrices in each step allows us to capture essential information from different sections of the question, thereby preserving the triplets that are semantically relevant to the question. For each head node e_s, we select the top-K edges (e_s, r, e_o) based on $\alpha_{q|sr}^\ell$ of different edges and prune the others, resulting in a smaller candidate set $\widetilde{\mathcal{C}}_q^\ell$. Here, K is a hyperparameter based on the characteristics of the KG. For example, Figure 2 illustrates a simple example with $K = 2$ and $L = 3$, where blue entities are retained while white entities irrelevant to the question are pruned. This way, we can adaptively explore the related information on the graph while reducing computation costs.

GNN encoding through propagation. To learn representations of each entity $e_o \in \widetilde{\mathcal{C}}_q^\ell$, we use a lightweight network, i.e., a 1-layer GNN, to propagate the information from entities $e_s \in \widetilde{\mathcal{C}}_q^{\ell-1}$ one step further to entities e_o with

$$\boldsymbol{h}_o^\ell = \delta\Big(\sum\nolimits_{(e_s,r,e_o)\in\widetilde{\mathcal{N}}_{e_o}^\ell} \alpha_{q|sr}^\ell \boldsymbol{W}^\ell (\boldsymbol{h}_s^{\ell-1} \odot \boldsymbol{h}_r)\Big), \tag{2}$$

where $\delta(\cdot)$ is the activation function, $\widetilde{\mathcal{N}}^{\ell}_{e_o}$ is the set of preserved neighbor edges of tail entity e_o, $\boldsymbol{W}^{\ell} \in \mathbb{R}^{d\times d}$ is a learnable weight matrix in the ℓ-th step, and the attention weight $\alpha^{\ell}_{q|sr}$ is computed in Eq.(1). In this way, the compositional information of edges relevant to question q connecting from topic entity e_q to e_o can be propagated into $\boldsymbol{h}^{\ell}_{o}$.

As shown in Figure 2, after L steps of propagation, we can form the final candidate set $\mathcal{C}_q = \widetilde{\mathcal{C}}^0_q \cup \cdots \cup \widetilde{\mathcal{C}}^L_q$ and obtain their representations $\boldsymbol{h}^L_e$. Finally, we use a multi-layer perceptron (MLP) and softmax function on the entity representation $\boldsymbol{h}^L_{e_i}$ and the question representation $\boldsymbol{h}_q$ to obtain the probability of entity e_i being the correct answer:

$$p(q, e_i) = e^{\mathsf{MLP}([\boldsymbol{h}^L_{e_i};\boldsymbol{h}_q])} / \sum\nolimits_{\forall e_j \in \mathcal{C}_q} e^{\mathsf{MLP}([\boldsymbol{h}^L_{e_j};\boldsymbol{h}_q])}. \tag{3}$$

We optimize the neural network with supervision given by the question-answer pairs. Specifically, we use the CrossEntropy Loss [36]:

$$\mathcal{L} = \sum\nolimits_{(q,e_a)\in\mathcal{F}_{tra}} -\log(p(q, e_a)), \tag{4}$$

where $\mathcal{F}_{tra}$ is the training set of question-answer pairs. The set of model parameters are randomly initialized and optimized by minimizing $\mathcal{L}$ with Adam stochastic gradient descent algorithm [37].

Through adaptive propagation with GNN encoding, the exploration module can achieve effective compositional learning to identify promising candidates. Moreover, with different edge weights, we can extract the relevant triplets as fine-grained knowledge $\mathcal{K}_q$, which are organized into relational paths in Section 3.3, effectively illustrating the reasoning process of our model.

3.3 Knowledge-Enhanced Answer Determination

Although the exploration module can indicate the probabilities of candidates from the output scores, leveraging the powerful language understanding capabilities of the LLM for aligning the question linguistically with the explored information can provide greater benefits. Therefore, after exploring a set $\mathcal{C}_q$ of promising candidate answers with the relevant knowledge $\mathcal{K}_q$, we leverage LLM to determine the final answer. In this section, we design a knowledge-enhanced multiple-choice prompt to direct the LLM to generate final answers by leveraging both explicit knowledge from KG and implicit knowledge inside LLM.

Multiple-choice prompt. With the explored candidates $\mathcal{C}_q$, we design a multiple-choice format for LLM, which allows the LLM to consider and compare multiple given reference answers at once. The multiple-choice prompt is constructed using a task description, the provided question, and reference answers that contains the top-N candidate entities in $\mathcal{C}_q$ identified with the highest probabilities during exploration. Meanwhile, we consider two kinds of knowledge aligned with each candidate to enhance the determination. First, we use the correct probability (as per Eq.(3)) of each reference answer to provide the confidence returned by the exploration module for LLM. Second, we extract the relevant path connecting from the topic entity e_q to each candidate entity as an evidence chain.

These paths reveal the compositional associations between topic entities and candidate answers, providing fine-grained knowledge and faithful evidence for the final answer determination. Overall, the input prompt is formulated as on the right.

<Task Description>
<Question>
<Reference Answers>:
A. candidate 1 (correct probability) {evidence chain}
B. candidate 2 (correct probability) {evidence chain} ……

Evidence extraction. To extract evidence chain as fine-grained compositional knowledge for LLM, we employ a greedy algorithm to trace back paths. Specifically, we backtrack from each reference candidate answer e_c, select the edge from $\widetilde{\mathcal{N}}^{L}_{e_c}$ with the highest attention weight (as per Eq.(1)), and set the head entity as a new starting point in the next step. By conducting L-steps backtracking, it will eventually trace back to the initial topic entity e_q and obtain an evidence path between e_q and e_c (the detailed algorithm is shown in Appendix B.3). For example, in Figure 1, the evidence chain connecting the topic entity *Birdy* and the reference answer *1989* is: *Birdy* $\xrightarrow{written\_by}$ William Wharton $\xrightarrow{wrote}$ Dad $\xrightarrow{release\_year}$ *1989*. Furthermore, LLMs can conduct semantic checks based on evidence chains to determine the final answer. For instance, it can recognize that the second reference answer (*1976*) in Figure 1 does not align with the semantic context of the question based on the paths between *Birdy* and *1976*.

Moreover, to avoid the LLM fully relying on the information given by the exploration module, and utilize the implicit knowledge within the LLM, we design the prompt of the task description in a way that encourages the LLM to generate answers based on its own knowledge. The complete input-output examples of the final answer determination are provided in the Appendix D, which demonstrates our method's capability for accurate and interpretable reasoning. In addition, to avoid distractions from irrelevant information, we do not offer any few-shot examples, providing a clear structure for the LLM to follow, while maintaining the benefits of zero-shot learning.

3.4 Training Strategy

During training, we freeze the LLM in both text encoding and answer determination steps to avoid the expensive training cost. Instead, we only update the lightweight GNN and the scoring MLP in exploration module by minimizing the loss in Eq.(4). During reasoning, unlike traditional approaches [5, 7, 8], which first retrieve a subgraph and then perform reasoning on it, our exploration module can unify and simultaneously perform these two steps, adaptively filter out irrelevant information on the graph, and achieve more efficient reasoning. Additionally, it is noteworthy that the answer determination module can be adapted to any pre-trained LLM without the need for retraining or fine-tuning, thereby avoiding time and resource overhead.

Considering there are common compositional relationships between questions and the concepts in the graphs, the lightweight network in exploration module can be further benefited from pre-training strategies. Specifically, we pre-train the networks on two comprehensive KGQA datasets WebQSP [38] and CWQ [39], and then fine-tune them on target datasets. This approach enables the exploration module to better learn the compositional relationships in KGQA, enhancing question understanding and generalization abilities.

4 Experiments

4.1 Experimental Setup

Datasets. Following existing KGQA works [7, 8], we use three benchmark datasets, namely WebQSP [38], CWQ [39], and MetaQA [40], to evaluate different methods. The MetaQA dataset is divided into three versions based on the number of hops required in KG, namely 1-hop, 2-hop, and 3-hop. Table 1 displays the statistics of these three datasets.

Table 1: Statistics of KGQA datasets.

Datasets	#Train	#Valid	#Test	Max #hop
WebQSP	2,848	250	1,639	2
CWQ	27,639	3,519	3,531	4
MetaQA-1	96,106	9,990	9,947	1
MetaQA-2	118,980	14,872	14,872	2
MetaQA-3	114,196	14,274	14,274	3

Evaluation Metrics. Following [17, 18], we focus on generating the answer with the highest confidence, and use Hits@1 to evaluate whether the top-ranked predicted answer is correct.

Experiment Details. In the pre-training stage, we set the dimension d as 256 for the exploration module, learning rate as 1e-4, batch size as 20, number of layers L as 3 and number of sampling K as 200. As for the fine-tuning stage, we adjust the L and K based on the performance on validation set, and details are described in Appendix B.4. Considering the plug-and-play convenience of EtD, we use two LLMs for answer determination in experiments: Llama2-13B-chat [11], and ChatGPT[2]. We typically set number of reference answers N as 3, and the influence of N is shown in Appendix C.2. The pre-training and fine-tuning of exploration module are conducted on a RTX 3090-24GB GPU, and the inference of Llama2-13B-chat is running on two RTX 3090-24GB GPUs.

Baseline Methods. We consider the following baseline methods for performance comparison: (1) KG-based methods without using LLMs: KV-Mem [22], GraftNet [5], EmbedKGQA [23], NSM [7], SR+NSM [25], UniKGQA [8]; (2) LLM-based methods: KB-Binder [30] based on Codex [41], KAPING [15] based on GPT-3 [9], RoG [32] that can be plug-and-play with different LLMs, KD-CoT [35] based on ChatGPT, StructGPT [17] based on ChatGPT, ToG [18] that can be plug-and-play with different LLMs.

[2] `https://openai.com/`

Table 2: Performance comparison of different methods for KGQA (Hits@1 in percent).

Type	Methods	WebQSP	CWQ	MetaQA-1	MetaQA-2	MetaQA-3
KG-based	KV-Mem	46.7	18.4	96.2	82.7	48.9
	GraftNet	66.4	36.8	97.0	94.8	77.7
	EmbedKGQA	66.6	-	97.5	98.8	94.8
	NSM	68.7	47.6	97.1	**99.9**	98.9
	SR+NSM	69.5	50.2	-	-	-
	UniKGQA	75.1	50.7	97.5	99.0	99.1
GPT3-based	KB-Binder	74.4	-	92.9	**99.9**	99.5
	KAPING	73.9	-	-	-	-
Llama2-based	Llama2-13B	40.9	22.1	31.9	15.8	34.9
	RoG-Llama2-7B	74.2	56.4	-	-	-
	ToG-Llama2-70B	68.9	57.6	-	-	-
	EtD-Llama2-13B	**77.4**	**57.7**	**97.9**	99.1	**99.6**
ChatGPT-based	ChatGPT	61.2	38.8	61.9	31.0	43.2
	RoG-ChatGPT	81.5	52.7	-	-	-
	KD-CoT	68.6	55.7	-	-	-
	StructGPT	72.6	-	94.2	93.9	80.2
	ToG-ChatGPT	76.2	58.9	-	-	-
	EtD-ChatGPT	**82.5**	**62.0**	**98.1**	99.7	**99.7**

4.2 Performance Comparison

From the results in Table 2, it can be observed that our method EtD, whether combined with the Llama2-13B-chat or ChatGPT, outperforms traditional methods without LLMs, except for the MetaQA-2hop dataset, where many methods achieve over 99% accuracy. From the last two groups of the table, it can be seen that incorporating KG can effectively enhance the performance of LLMs, as they often lack the knowledge relevant to the questions. Within methods using Llama2, our approach combined with the 13B model outperforms ToG which is combined with the 70B model, and also outperforms some methods using ChatGPT. Similarly, within methods using ChatGPT, our approach demonstrates significant advantages. This shows the superiority of proposed explore-then-determine framework, effectively harnessing the precision of compositional learning and the prowess of language understanding. Additionally, it is evident that the performance of our method improves with the integration of more powerful LLM, e.g. ChatGPT. This demonstrates that our method effectively combines explicit structural knowledge from KGs with implicit knowledge from LLM, leading to more accurate reasoning and answering.

We also compare the inference time and number of interaction with LLM of several representative LLM-based methods, specifically executing Llama2-13B within our local environment. As can be seen in Table 3, RoG requires two interactions with LLM, while step-by-step reasoning methods StructGPT and ToG require even more frequent interactions, thus incurring a high cost. However, in our approach, using LLM for question encoding and graph exploration is highly efficient, and we ultimately requires only one-step inference of LLM, thereby achieving faster speeds.

4.3 Ablation Study

4.3.1 Effectiveness of Graph Exploration

In this section, we analyze the effectiveness of our exploration module, focusing on the Hits@1 of its independent output answer without the answer determination by LLM (i.e., EtD-w.o.-AD). Here, we choose UniKGQA, the state-of-the-art KG-based model without LLMs for comparison. As can be seen in Table 5, our exploration module outperforms UniKGQA on all datasets, demonstrating the effectiveness of the designed network for topology and semantic aware reasoning on the graph.

4.3.2 Effectiveness of LLM's Answer Determination

Furthermore, Table 5 demonstrates the effectiveness of using LLM for answer determination. Although the exploration module itself (EtD-w.o.-AD) achieves decent performance, utilizing the

Table 3: Comparison of inference time (seconds) and number of interaction with LLM (Llama2-13B) per question of different methods.

Methods	WebQSP		CWQ	
	time	#interaction	time	#interaction
RoG	1.98	2	3.04	2
StructGPT	3.37	3	4.22	4
ToG	16.7	15	20.5	22
EtD	**1.29**	**1**	**1.99**	**1**

Table 4: Comparison of different variants of EtD-Llama2-13B in Hits@1(%).

Methods	WebQSP	CWQ
EtD	77.4	57.7
w.o.-mcp	72.7	54.0
w.o.-cand	77.1	50.0
w.o.-prob	75.2	50.4
w.o.-path	76.8	57.0

Table 5: Comparison of Different Variants of EtD in Hits@1(%).

Methods	WebQSP	CWQ	MetaQA-1	MetaQA-2	MetaQA-3
UniKGQA	75.1	50.7	97.5	99.0	99.1
EtD-w.o.-AD	76.4	55.6	97.6	99.1	99.5
EtD-Llama2-13B	77.4	57.7	97.9	99.1	99.6
EtD-ChatGPT	82.5	62.0	98.1	99.7	99.7

powerful language processing capability of LLM for answer determination brings significant gains. With its powerful language processing capabilities, the LLM can analyze the semantic relationships between a given question, the candidate answers and the relevant knowledge. By leveraging the implicit knowledge embedded within itself, it achieves more accurate reasoning.

4.3.3 Influence of Prompt Form for Answer Determination

We design different forms of prompts to evaluate their influence in guiding the final answer determination of EtD-Llama2-13B. The variants include: (1) not using multiple-choice prompt (EtD-w.o.-mcp), (2) not using candidate entities (EtD-w.o.-cand), (3) not using correct probabilities (EtD-w.o.-prob), and (4) not using evidence paths (EtD-w.o.-path), which are shown in Appendix E.

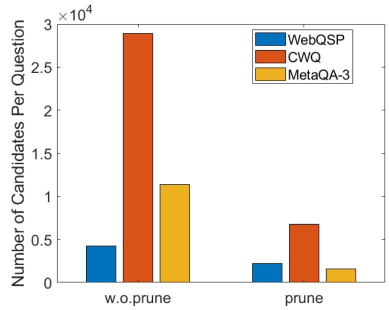
As shown in Table 4, compared with EtD-w.o.-mcp, which inputs all candidate answers continuously without specific labels, our designed multiple-choice prompt can achieve better performance, since it offers a clearer structure and strength the connection of candidate answers with related knowledge. Moreover, all three factors in the prompt, i.e., candidate entity, correct probability and evidence path, have a positive impact on guiding LLM to make correct choices. Providing candidate answers can refine the target for LLM (compared with EtD-w.o.-cand), and correct probabilities prompts LLM to engage in reasoning with attention (compared with EtD-w.o.-prob). Evidence paths are crucial as they provide fine-grained knowledge for LLM reasoning (compared with EtD-w.o.-path), ensuring the transparency and reliability of the output results, as can be seen in Section 4.4.

4.3.4 Influence of Pruning

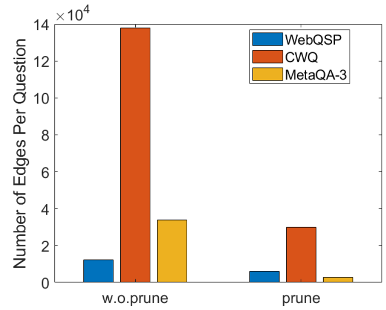
In this section, we discuss the influence of the pruning for candidate set. As can be seen in Figure 3, the size of the unfiltered candidate set and the number of edges involved in each question are typically immense due to the large-scale KG. In comparison, with pruning, both of them have significantly decreased. Therefore, we can filter out a large amount of irrelevant information, significantly reducing the computational costs, achieving more efficient reasoning.

4.3.5 Effect of Pre-training Strategy

In Section 3.4, we have mentioned that the networks in exploration module can be benefited from pre-training strategies. We compare the performance of individual training strategy and pretrain-finetune strategy in Table 6. It is evident that the strategy of pre-training significantly outperforms training separately on distinct datasets. This underscores the effectiveness of pre-training in enabling our exploration module to better grasp the common compositional relationships between questions and the concepts in the graphs, thereby enhancing the question-comprehension abilities empowered by LLM. We also compare the impact of training strategy on the learning curves and conduct experiments on a new dataset in sports domain in the Appendix C.1.



(a) Number of candidates.



(b) Number of edges.

Figure 3: Influence of pruning on three datasets.

Table 6: Influence of pretraining of exploration module.

Methods	WebQSP	CWQ	MetaQA-1	MetaQA-2	MetaQA-3
w.o.-pretrain	72.6	53.4	97.8	99.0	99.4
pretrain-finetune	76.4	55.6	97.8	99.1	99.5

4.4 Case Study

We present two case studies in Figure 4, which display the question, reference answers of the input prompt, and the final result of the LLM's answer determination. The shared task description can be found in Appendix D. It can be observed that the LLM makes correct determinations based on the candidate answers, correct probabilities, and relevant path facts, even when the right answer is not ranked at the top, indicating the importance of determination stage. Moreover, the LLM provides its reasoning process, enhancing the credibility of the output. These examples illustrate how our method combines explicit knowledge from the exploration stage with the implicit knowledge of the LLM, achieving accurate and faithful results.

Question–Answer	When made the dissemination of the films whose writers also wrote Birdy? - 1989.
Input Question and Reference Answers	Question: when made the dissemination of the films whose writers also wrote Birdy? Reference answers: A. 1989 (correct probability: 0.996) {relevant facts: (Birdy, written_by, William Wharton), (Dad, written_by, William Wharton), (Dad, release_year, 1989)} B. 1998 (correct probability: 0.004) {relevant facts: (Birdy, has_tags, nicolas cage), (Snake Eyes, has_tags, nicolas cage), (Snake Eyes, release_year, 1998)} C. 1976 (correct probability: 0.0) {relevant facts: (Birdy, directed_by, Alan Parker), (Bugsy Malone, directed_by, Alan Parker), (Bugsy Malone, release_year, 1976)}
Output by EtD-Llama2-13B	Based on the reference answers and my own knowledge, I can answer the question. The correct answer is A. 1989. Here's why: The reference answers provide three relevant facts: 1. (Birdy, written_by, William Wharton) 2. (Dad, written_by, William Wharton) 3. (Dad, release_year, 1989) Based on these facts, we can infer that the film "Dad" was written by William Wharton and released in 1989. Therefore, the correct answer is A. 1989.

(a) One example from MetaQA-3hop.

Question-Answer	What's the home field for the sports team owned by Mark Attanasio? - Miller Park.
Input Question and Reference Answers	Question: What's the home field for the sports team owned by Mark Attanasio? Reference answers: A. Seattle Pilots (correct probability: 0.7) {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.previously_known_as, Seattle Pilots)} B. Miller Park(correct probability: 0.285) {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park)} C. Milwaukee (correct probability: 0.015) {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.location, Milwaukee)}
Output by EtD-Llama2-13B	Based on the information provided, the correct answer is B. Miller Park. Here's why: The correct reference answer with the correct probability is B. Miller Park. The relevant facts provided are: * (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio) * (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park). These facts indicate that Miller Park is the home field of the Milwaukee Brewers, which is owned by Mark Attanasio. Therefore, the answer is B. Miller Park.

(b) One example from CWQ.

Figure 4: Examples of accurate and faithful output by EtD-Llama2-13B.

5 Conclusion

In this paper, we present a novel Explore-then-Determine (EtD) framework that synergizes LLMs with GNNs for reasoning over KG. During the Explore stage, we design an LLM-empowered GNN module to perform compositional learning on the KG, exploring the promising candidates and fine-grained compositional knowledge of the question. In the Determine stage, the information gathered from the exploration is organized to create a knowledge-enhanced multiple-choice prompt, which a frozen LLM then uses to determine the final answer. This two-step process leverages the precision of GNN-based compositional learning and the language understanding prowess of LLMs, effectively combining explicit knowledge from KGs with implicit knowledge in LLMs. Extensive experiments demonstrate the superiority of EtD, achieving accurate, efficient and faithful reasoning over KGs.

References

[1] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[2] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.

[3] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1306–1313, 2010.

[4] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. Complex knowledge base question answering: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[5] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 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*, pages 4231–4242, 2018.

[6] Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 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*, pages 4149–4158, 2021.

[7] Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 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*, pages 553–561, 2021.

[8] Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*, 2022.

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[11] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[12] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[13] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[14] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[15] Jinheon Baek, Alham Aji, and Amir Saffari. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[16] Garima Agrawal, Tharindu Kumarage, Zeyad Alghami, and Huan Liu. Can knowledge graphs reduce hallucinations in llms?: A survey. *arXiv preprint arXiv:2311.07914*, 2023.

[17] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*, 2023.

[18] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2023.

[19] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[20] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2022.

[21] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, 2016.

[23] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 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*, pages 4498–4507, 2020.

[24] Haitian Sun, Tania Bedrax-Weiss, and William Cohen. 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)*, pages 2380–2390, 2019.

[25] Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784, 2022.

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2016.

[27] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

[28] Yawei Sun, Lingling Zhang, Gong Cheng, and Yuzhong Qu. Sparqa: skeleton-based semantic parsing for complex questions over knowledge bases. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8952–8959, 2020.

[29] Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The Eleventh International Conference on Learning Representations*, 2022.

[30] Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980, 2023.

[31] Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Lidong Bing, Shafiq Joty, and Soujanya Poria. Chain of knowledge: A framework for grounding large language models with structured knowledge bases. *arXiv preprint arXiv:2305.13269*, 2023.

[32] LINHAO LUO, Yuan-Fang Li, Reza Haf, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*, 2023.

[33] Shiyang Li, Yifan Gao, Haoming Jiang, Qingyu Yin, Zheng Li, Xifeng Yan, Chao Zhang, and Bing Yin. Graph reasoning for question answering with triplet retrieval. *arXiv preprint arXiv:2305.18742*, 2023.

[34] Parishad BehnamGhader, Santiago Miret, and Siva Reddy. Can retriever-augmented language models reason? the blame game between the retriever and the language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15492–15509, 2023.

[35] Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*, 2023.

[36] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*, pages 2863–2872. PMLR, 2018.

[37] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. Technical report, arXiv:1412.6980, 2014.

[38] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206, 2016.

[39] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, 2018.

[40] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[41] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[42] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[43] Liwen Zhang, John Winn, and Ryota Tomioka. Gaussian attention model and its application to knowledge base embedding and question answering. *arXiv preprint arXiv:1611.02266*, 2016.

[44] Iyad AlAgha. Multihop question answering by using sequential path expansion with backtracking. *IEEE Access*, 10:76842–76854, 2022.

A Discussion

Limitations. In our method, the exploration depth L is preset based on the characteristics of the dataset. Therefore, in real-world applications, there is a possibility that the correct answer might not be found within step L. It is worth exploring how to use LLM to determine the depth of exploration in the future.

Broader Impacts. The method could facilitate real-world applications, such as mentioned in Section 1. The Explore-then-Determine framework not only improves the accuracy and efficiency of KGQA tasks, but also has the potential to be adapted for a wide range of reasoning tasks involving LLMs.

Safeguards Statement. In this paper, we primarily focus on the KGQA task. We recognize that future research on border applications of reasoning with LLMs may carry the risk of misuse. Therefore, we recommend thoroughly considering all aspects of safety before applying these techniques in real-world scenarios.

B Implementation Details

B.1 Question and Relation Encoding

Considering the remarkable modeling capacity of the LLM, we first employ the average embedding in the last layer of the LLM (e.g., Llama2-13B) to produce text encoding as the representations of question q and relation r in the KG:

$$\bar{\boldsymbol{h}}_q = \text{LLM}(q), \bar{\boldsymbol{h}}_r = \text{LLM}(r). \tag{5}$$

Note that the parameter dimension d_L of the LLM is typically high, so we use a weight matrix $\boldsymbol{W}_L$ to reduce the dimension from d_L to d, i.e., $\boldsymbol{h}_q = \boldsymbol{W}_L \bar{\boldsymbol{h}}_q, \boldsymbol{h}_r = \boldsymbol{W}_L \bar{\boldsymbol{h}}_r$.

It is worth noting that the reverse relations $(-r)$ play an important role in graph reasoning, but there is no golden rule for obtaining the textual representation of them. So we use a linear layer to map $\boldsymbol{h}_r$, generating the representation $\boldsymbol{h}_{-r}$ of reverse relation $-r$, i.e., $\boldsymbol{h}_{-r} = \boldsymbol{W}_{-r}\boldsymbol{h}_r + \boldsymbol{b}_{-r}$. Additionally, we separately learn a representation $\boldsymbol{h}_{id}$ for the identity relation. In this way, we can leverage LLM's capabilities to achieve unified representation, effectively mining semantic information from the KG.

B.2 Exploration Algorithm

Algorithm 1 Semantic-aware graph exploration.

Require:
question q, topic entity e_q, KG G, question encoding $\boldsymbol{h}_q$, relations encoding $\boldsymbol{h}_r$'s, depth L, model parameters $\boldsymbol{\Theta}$.

initialize $\boldsymbol{h}^0_{e_q} = \boldsymbol{h}_q$ and $\widetilde{\mathcal{C}}^0_q = \{e_q\}$;
for $l = 1, 2 \cdots L$ **do**
  get the candidate set $\mathcal{C}^\ell_q = \{e_o : (e_s, r, e_o) \in \mathcal{G}, e_s \in \widetilde{\mathcal{C}}^{\ell-1}_q\}$
  calculate attention weights $\alpha^\ell_{q|sr}$ (by Eq.(1));
  get the pruned candidate set $\widetilde{\mathcal{C}}^\ell_q$ based on different $\alpha^\ell_{q|sr}$;
  for $e \in \widetilde{\mathcal{C}}^\ell_q$ (in parallel) **do**
    $\boldsymbol{h}^\ell_o := \delta(\sum_{(e_s,r,e_o)\in\widetilde{\mathcal{N}}^\ell_{e_o}} \alpha^\ell_{q|sr} \boldsymbol{W}^\ell(\boldsymbol{h}^{\ell-1}_s \odot \boldsymbol{h}_r))$ (by Eq.(2)).
  end for
end for
return $\boldsymbol{h}^L_e$ for all $e \in \mathcal{C}_q = \widetilde{\mathcal{C}}^0_q \cup \cdots \cup \widetilde{\mathcal{C}}^L_q$.

B.3 Path Backtracking Algorithm

Algorithm 2 Path Backtracking.

Require:
question q, KG $\mathcal{G}$, depth L, model parameters $\mathbf{\Theta}$, hyperparameter N.

run Algorithm 1, obtain attention weights $\{\alpha^{\ell}_{q|sr}\}_{\ell=1,\dots,L}$ of different edges in each step ℓ;
select top-N candidates entities to form $\widetilde{\mathcal{C}}_q$
for $e_c \in \widetilde{\mathcal{C}}_q$ **do**
    set $e_t = e_c$;
    for $l = L, L-1 \cdots 1$ **do**
        obtain attention weights $\{\alpha^{\ell}_{q|sr}\}$ of edges whose tail node is e_t;
        $\mathcal{E}^{\ell}_{q|e_c} = \{(e_h, r, e_t) : h = \arg\max_s \alpha^{\ell}_{q|sr}\}$;
        $e_t = e_h$;
    end for
end for
return $\mathcal{P}_{q|e_c} = \mathcal{E}^1_{q|e_c} \cup \cdots \cup \mathcal{E}^L_{q|e_c}$ for $e_c \in \widetilde{\mathcal{C}}_q$.

B.4 Details of Experiments.

Datasets. We adopt three benchmark KGQA datasets: WebQuestionSP (WebQSP)[38], Complex WebQuestions (CWQ) [39] and MetaQA [40] in this work. For WebQSP and CWQ, the corresponding KGs are Freebase [42]. Following previous works [7], we reduce the size of Freebase by extracting all triples that contain within the respective max reasoning hops of the topic entities for each question. For MetaQA, we directly use the original WikiMovies KG [3]. The statistics of three KGs are presented in Table 7.

Table 7: Statistics of three knowledge graphs.

KG	#Entities	#Relations	#Triples
Freebase-WebQSP	1,441,421	6,102	20,111,715
Freebase-CWQ	2,429,346	6,649	138,785,703
WikiMovies-MetaQA	43,234	9	134,741

Hyperparameters. During the pre-training stage of exploration module, (with the maximum number of training epochs set to 30), we set the learning rate as 1e-4, weight decay as 1e-3, batch size as 20, dimension d as 256, number of layers L as 3 and number of sampling K as 200. As for the fine-tuning stage, we tune the learning rate in $[10^{-6}, 10^{-3}]$, weight decay in $[10^{-5}, 10^{-2}]$. We also adjust the L and K based on the performance on validation set, which is shown in Table 8.

Table 8: Hyperparameters of exploration module on different datasets.

	WebQSP	CWQ	MetaQA-1	MetaQA-2	MetaQA-3
L	2	4	1	2	3
K	200	200	40	60	100

B.5 Baselines

We consider the following baseline methods for performance comparison:

(1) traditional KG-based methods without using LLMs:

- KV-Mem [22] utilizes a Key-Value memory network to store triples and conduct iterative read operations to deduce the answer.

[3] https://research.fb.com/downloads/babi

- GraftNet [5] first retrieves the question-relevant triplets and text sentences from the KG and corpus to build a heterogeneous subgraph. Then it adopts a graph neural network to perform multi-hop reasoning on the subgraph.
- EmbedKGQA [23] embeds entities on KG and design a scoring function to rank them based on their relevance to the question.
- NSM [7], first conducts subgraph retrieval and then employ the neural state machine with a teacher-student network for multi-hop reasoning on the KG.
- SR+NSM [25] first employs a pretrained language model to build a subgraph retriever, then use NSM for reasoning on the retrieved subgraph.
- UniKGQA [8] conducts the subgraph retrieval and reasoning process with the same model based on GNN, which achieve salient performance on KGQA task.

(2) LLM-based methods:

- KB-Binder [30] first uses Codex [41] to generates logical forms as the draft by imitating a few demonstrations. Then it bind the generated draft to an executable one with BM25 score matching.
- KAPING [15] uses a sentence embedding model to retrieve the relevant triplets to the question from KG which are then forwarded to LLMs to generate the answer.
- RoG [32] first fine-tunes the Llama2-7B to generate relation paths grounded by KGs as faithful plans. Then it uses these plans to retrieve reasoning paths from the KGs for LLMs to conduct reasoning.
- KD-CoT [35] utilizes a QA system to verify and modify reasoning traces in CoT of LLMs via interaction with external knowledge source.
- StructGPT [17] views LLMs as agents and establishes an information interaction mechanism between KG and LLMs to iteratively deduce the answer to the question.
- ToG [18] improves StructGPT by guiding the LLM to iteratively execute beam search on KG.

In Table 9, we summarize the differences between our method EtD and several representative baselines in KGQA. These methods generally involve knowledge retrieval and answer generation processes, and our exploration stage can be seen as a form of knowledge retrieval. As can be seen, our method uniquely synergizes LLM with GNN, harnessing both the precision of compositional learning and the prowess of language understanding.

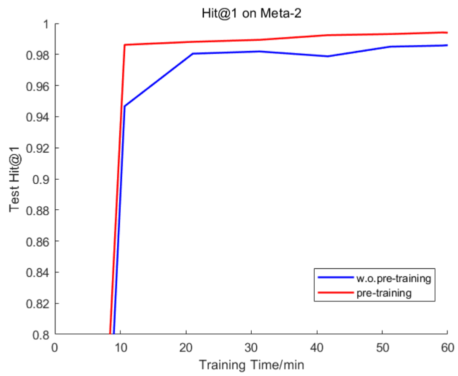
Table 9: Comparison of different methods.

Methods	Knowledge Retrieval	Answer Generation
UniKGQA	GNN	GNN
KAPING	sentence embedding model	LLM
RoG	fine-tuned LLM	LLM
StructGPT	LLM	LLM
ToG	LLM	LLM
EtD	GNN+LLM	LLM

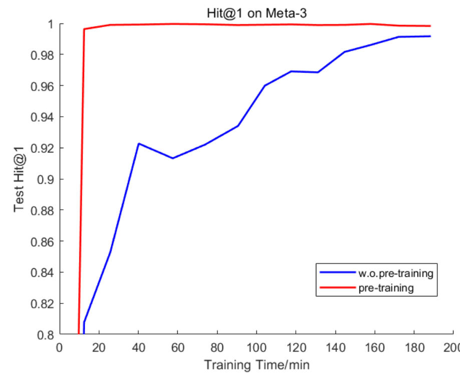
C Supplementary Experiments

C.1 Effect of Pre-training Strategy

We compared the performance of exploration module with pre-training and fine-tuning with those trained solely on individual datasets. We plot the learning curves for both training strategies. As can be seen in Figure 5, pre-training significantly accelerates the convergence speed of the model during fine-tuning on the downstream datasets, and acquires better performance. This advantage is more pronounced on more complex datasets (i.e., MetaQA-3hop).



(a) Curves on MetaQA-2.



(b) Curves on MetaQA-3.

Figure 5: Learning curves on two datasets.

To further validate the generalization capability of our method, we also conduct fine-tuning experiments on a dataset from the sports domain. We use the dataset WorldCup2014[43], which contains about 8000 questions with answers related to the 2014 World Cup, and questions are a mixture of 1-hop and 2-hop questions. As shown in table 10, our method achieve an impressive result of 100% on Hits@1, which is also superior to the existing baselines.

Table 10: Performance comparison on WorldCup2014 dataset.

Methods	WC-1	WC-2	WC-m
TransferNet[6]	97.9	96.5	96.8
44	97.4	98.6	96.0
EtD-Llama2-13B	100	100	100

C.2 Influence of Number of Reference Answer

Table 11 shows the impact of the number of reference answers in the determination module. As can be seen, providing three reference answers is appropriate, since more reference answers will introduce too much redundant information, interfering with the inference of the LLM.

Table 11: Comparison of different numbers of reference answers N of EtD-Llama2-13B.

N	3	4	5	6
WebQSP	77.37	77.20	77.08	76.66
CWQ	57.67	56.87	56.67	56.12

D Case Study

Table 12 and 13 present examples of the input-output of our method EtD. Compared with Figure 4, we present the complete set of prompts, including the task description. It can be observed that by combining explicit knowledge from the KG with the implicit knowledge of LLM itself, LLM ultimately selects the correct answer and provides its reasoning process, enhancing the credibility of the output.

E Variants of Prompt

Table 14 shows the variants of designed prompt introduced in Section 4.3.3, Compared with these formats, our designed knowledge-enhanced multiple-choice prompt can achieve better performance.

Table 12: Case study 1 from MetaQA-3hop.

Question–Answer	When made the dissemination of the films whose writers also wrote Birdy? - 1989.
Input Task Description	Given a question, and the reference answers with their correct probabilities and associated retrieved knowledge graph triples (entity, relation, entity) as related facts, you are asked to answer the question with this information and your own knowledge. If the reference answers contain the correct answer, please output the label and content of the answer; If not, please answer the question based on your own knowledge.
Input Question and Reference Answers	Question: when made the dissemination of the films whose writers also wrote Birdy? Reference answers: A. 1989 (correct probability: 0.996) {relevant facts: (Birdy, written_by, William Wharton), (Dad, written_by, William Wharton), (Dad, release_year, 1989)} B. 1998 (correct probability: 0.004) {relevant facts: (Birdy, has_tags, nicolas cage), (Snake Eyes, has_tags, nicolas cage), (Snake Eyes, release_year, 1998)} C. 1976 (correct probability: 0.0) {relevant facts: (Birdy, directed_by, Alan Parker), (Bugsy Malone, directed_by, Alan Parker), (Bugsy Malone, release_year, 1976)}
Output by EtD-Llama2-13B	Based on the reference answers and my own knowledge, I can answer the question. The correct answer is A. 1989. Here's why: The reference answers provide three relevant facts: 1. (Birdy, written_by, William Wharton) 2. (Dad, written_by, William Wharton) 3. (Dad, release_year, 1989) Based on these facts, we can infer that the film "Dad" was written by William Wharton and released in 1989. Therefore, the correct answer is A. 1989.

Table 13: Case study 2 from CWQ.

Question-Answer	What's the home field for the sports team owned by Mark Attanasio? - Miller Park.
Input Task Description	Given a question, and the reference answers with their correct probabilities and associated retrieved knowledge graph triples (entity, relation, entity) as related facts, you are asked to answer the question with this information and your own knowledge. If the reference answers contain the correct answer, please output the label and content of the answer; If not, please answer the question based on your own knowledge.
Input Question and Reference Answers	Question: What's the home field for the sports team owned by Mark Attanasio? Reference answers: A. Seattle Pilots (correct probability: 0.7) {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.previously_known_as, Seattle Pilots)} B. Miller Park (correct probability: 0.285) {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park)} C. Milwaukee (correct probability: 0.015) {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.location, Milwaukee)}
Output by EtD-Llama2-13B	Based on the information provided, the correct answer is B. Miller Park. Here's why: The correct reference answer with the correct probability is B. Miller Park. The relevant facts provided are: * (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio) * (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park). These facts indicate that Miller Park is the home field of the Milwaukee Brewers, which is owned by Mark Attanasio. Therefore, the answer is B. Miller Park.

Table 14: Different variants of knowledge-enhanced multiple-choice prompt.

	prompt
EtD-w.o.-mcp	Reference answers include: [Seattle Pilots, Miller Park, Milwaukee]. Their correct probabilities are [0.7, 0.285, 0.015]. Relevant facts are [(Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.previously_known_as, Seattle Pilots), (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park), (Milwaukee Brewers, sports.sports_team.location, Milwaukee)].
EtD-w.o.-cand	Relevant facts include: {(Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.previously_known_as, Seattle Pilots)}(correct probability: 0.7) {(Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park)}(correct probability: 0.285) {(Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.location, Milwaukee)}(correct probability: 0.015).
EtD-w.o.-prob	Reference answers: A. Seattle Pilots {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.previously_known_as, Seattle Pilots)} B. Miller Park {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.arena_stadium, Miller Park)} C. Milwaukee {relevant facts: (Milwaukee Brewers, sports.professional_sports_team.owner_s, Mark Attanasio), (Milwaukee Brewers, sports.sports_team.location, Milwaukee)}
EtD-w.o.-path	Reference answers: A. Seattle Pilots (correct probability: 0.7) B. Miller Park (correct probability: 0.285) C. Milwaukee (correct probability: 0.015)