Correcting on Graph: Faithful Semantic Parsing over Knowledge Graphs with Large Language Models

Ruilin Zhao, Feng Zhao[*], Hong Zhang
Natural Language Processing and Knowledge Graph Lab,
School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
{ruilinzhao,zhaof,d202381437}@hust.edu.cn

Abstract

Complex multi-hop questions often require comprehensive retrieval and reasoning. As a result, effectively parsing such questions and establishing an efficient interaction channel between large language models (LLMs) and knowledge graphs (KGs) is essential for ensuring reliable reasoning. In this paper, we present a novel semantic parsing framework Correcting on Graph (CoG), aiming to establish faithful logical queries that connect LLMs and KGs. We first propose a structured knowledge decoding that enables the LLM to generate fact-aware logical queries during inference, while leveraging its parametric knowledge to fill in the blank intermediate entities. Then, we introduce a knowledge path correction that combines the logical query with KGs to correct hallucination entities and path deficiencies in the generated content, ensuring the reliability and comprehensiveness of the retrieved knowledge. Extensive experiments demonstrate that CoG outperforms the state-of-the-art KGQA methods on two knowledge-intensive question answering benchmarks. CoG achieves a high answer hit rate and exhibits competitive F1 performance for complex multi-hop questions.

1 Introduction

Large language models (LLMs) (OpenAI et al., 2023; Dubey et al., 2024) have demonstrated remarkable capabilities in natural language processing (NLP). They demonstrate deep step-by-step reasoning capabilities, enabling them to tackle intricate questions that require multi-step analysis and nuanced understanding (Wei et al., 2022). Despite their remarkable performance across various applications, LLMs still face a significant challenge of lack of factual knowledge (Ji et al., 2023). This limitation arises from the static training process of LLMs, making it difficult to incorporate dynamically updated world knowledge. In this case, integrating LLMs with external knowledge sources, such as knowledge graphs (KGs) (Bollacker et al., 2007; Vrandecic and Krötzsch, 2014), offers a promising solution. This integration allows LLMs to generate faithful responses, reducing hallucination issues.

There are various methods for integrating LLMs with KGs (Wang et al., 2023; Luo et al., 2024; Mavromatis and Karypis, 2024; Zhao et al., 2023). Based on the frequency of interaction between LLMs and KG, these methods can be categorized into two paradigms. The first paradigm follows the iterative retrieve-then-read strategy (Nishida et al., 2018; Lewis et al., 2020), where the multi-hop question is transformed into step-by-step reasoning on the KG and the LLM is served as an agent to determine which knowledge facts should be considered at each reasoning step (Wang et al., 2023; Jiang et al., 2023a). Although this strategy is effective in knowledge selection, it typically leads to high computational costs due to the involvement of processing numerous candidate entities and relations, necessitating extensive ranking and evaluation at each step (Sun et al., 2024). Consequently, there is an increasing emphasis on the second paradigm, the semantic parsing strategy (He et al., 2021a; Luo et al., 2024), which connects LLMs and KGs through logical queries. Its goal is to convert the complex multi-hop question into formal logical queries (He et al., 2021a; Zhang et al., 2022) that allow LLMs to retrieve the necessary knowledge from KGs through a single query execution. For example, given a question “Who is the brother of Michael J. Fox?” (Bollacker et al., 2007), semantic parsing method (Luo et al., 2024) fine-tunes the LLM to generate a relation path "parent → children" and executes this query on the KG to retrieve relevant knowledge. The core of semantic parsing lies in generating valid and faithful logical queries. However, our

[*] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2025, pages 5364–5376
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

experiments reveal that 37.2% of the generated queries are invalid, where we attribute these hallucination errors to the lack of rigorous constraints during the generation process (Ji et al., 2023). This issue stems from the failure to consider the inseparable relationship between entities and relations in the KG. The entities play a crucial role in determining which relations exist between the entity and its neighbors (Lin et al., 2016). When intermediate entities are ignored, the relation path loses essential constraints, which negatively impacts the reliability of the query (Luo et al., 2024). By relying solely on relation-based reasoning and ignoring intermediate entities, LLM generates overly flexible queries that are more prone to hallucinations, leading to queries that sound plausible but fail to retrieve any information from the KG.

Similar to the structure of the chain-of-thought prompts (Wei et al., 2022), LLM+KG methods aim to retrieve a knowledge path in the KG that starts from the topic entity, traverses through multi-hop relations, and ultimately reaches the answer entity (Sun et al., 2024; Zhao et al., 2024b). Surprisingly, this knowledge path reflects the structure after incorporating intermediate entities into the logical query. In other words, to enable LLMs to generate logical queries with intermediate entity constraints, the essence is to treat the LLM as a parametric knowledge base and generate knowledge paths for question answering. This concept is highly consistent with the “Language Models as Knowledge Bases” paradigm (Petroni et al., 2019; Heinzerling and Inui, 2021; Zhao et al., 2022) and the key role of LLM is to generate the logical query while filling in the missing intermediate entities. Although LLMs are unable to store all world knowledge (Petroni et al., 2019), and the generated intermediate entities might suffer from hallucination errors (Ji et al., 2023), external KGs can effectively correct the generated paths, thus ensuring the reliability of the knowledge paths (Wang et al., 2023).

Motivated by this, we propose a novel semantic parsing method called **C**orrecting **o**n **G**raph (**CoG**) to retrieve reliable knowledge paths from the KG. We introduce a ***structured knowledge decoding*** that incorporates intermediate entities as constraints during the query generation process. We are inspired by the LM-as-KB paradigm (Petroni et al., 2019) and design a reciting task to fine-tune the LLM and force the LLM to memorize knowledge paths during fine-tuning. In this way, LLM can generate possible knowledge paths using its stored parametric knowledge when handling multi-hop questions. On the other hand, we aim to make the LLM generate structured output which in the form of a continuous knowledge paths. In this way, the LLM can start from the question entity for path generation. After generating the first relation, it treats the process of generating the next entity as a "fill-in-the-blank" task (Donahue et al., 2020). This approach prompts the LLM to utilize its internal knowledge to fill in the missing entity. The generated entities also implicitly carry various attributes (such as a person or a city), and these attributes help constrain the generation of the next relation, preventing the process from being overly flexible and reducing the occurrence of hallucination errors. (2) Due to the black-box nature of LLMs, The generated knowledge paths are often unreliable or incomprehensive. To address this issue, we further introduce ***knowledge path correction*** to refine the generated knowledge paths. Specifically, we correct hallucination errors in the paths, such as factual errors in intermediate entities, by retrieving reliable paths from the KG using logical queries. Additionally, we address path deficiency, as the paths generated by the LLM may not cover all possible answers. In this case, we use logical queries to match missing knowledge from the KG, thereby ensuring that the retrieved paths provide broader coverage of possible answers.

In summary, our contributions are as follows:

- We introduce a structured knowledge decoding to alleviate the excessive flexibility in the query generation process for faithful semantic parsing.
- We propose a knowledge path correcting that combines logical queries with KGs to correct the generated paths, ensuring the reliability and comprehensiveness of the knowledge paths.
- The experiments show that CoG achieves the best performance on WebQSP and CWQ. CoG achieves a high answer hit rate and exhibits competitive F1 performance for complex questions.

2 Correcting on Graph

In this section, we detailed introduce the Correcting on Graph framework. We first introduce a structured knowledge decoding to parse the multi-hop question and generate fact-aware logical queries. As shown in Figure 1, we fine-tune the LLM to memorize factual knowledge paths

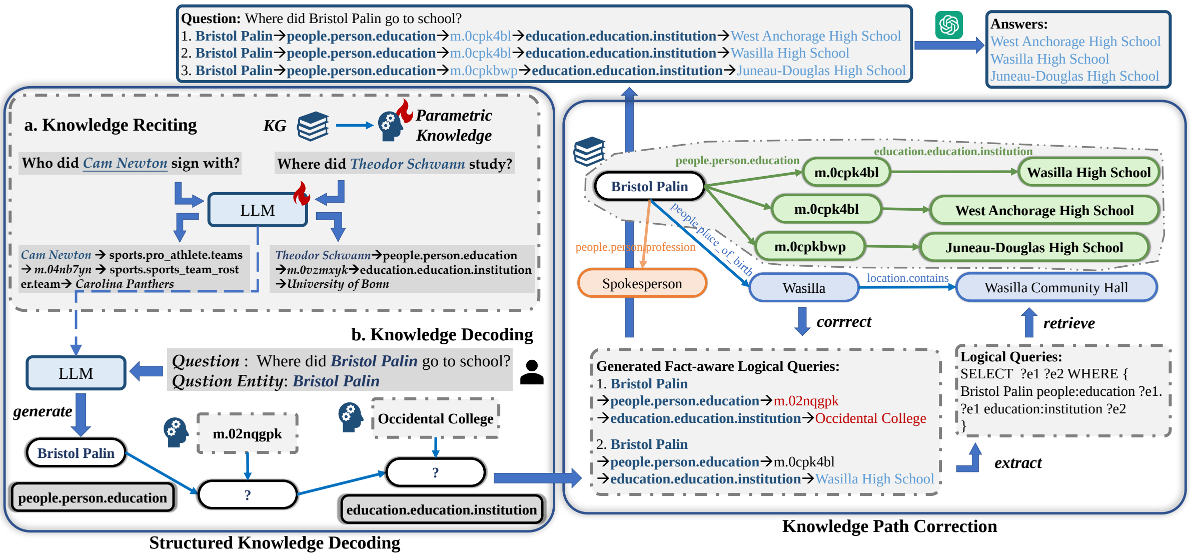
Question: Where did Bristol Palin go to school?
1. Bristol Palin→people.person.education→m.0cpk4bl→education.education.institution→West Anchorage High School
2. Bristol Palin→people.person.education→m.0cpk4bl→education.education.institution→Wasilla High School
3. Bristol Palin→people.person.education→m.0cpkbwp→education.education.institution→Juneau-Douglas High School
Answers:
West Anchorage High School
Wasilla High School
Juneau-Douglas High School
a. Knowledge Reciting
KG
Parametric Knowledge
Who did Cam Newton sign with?
Where did Theodor Schwann study?
LLM
Cam Newton → sports.pro_athlete.teams → m.04nb7yn → sports.sports_team_roster.team→ Carolina Panthers
Theodor Schwann→people.person.education →m.0vzmxyk→education.education.institution →University of Bonn
b. Knowledge Decoding
Question : Where did Bristol Palin go to school?
Qustion Entity: Bristol Palin
LLM
generate
Bristol Palin
m.02nqgpk
Occidental College
?
?
people.person.education
education.education.institution
Structured Knowledge Decoding
Bristol Palin
people.person.education
education.education.institution
m.0cpk4bl
m.0cpk4bl
m.0cpkbwp
Wasilla High School
West Anchorage High School
Juneau-Douglas High School
people.person.profession
people.place_of_birth
Spokesperson
Wasilla
location.contains
Wasilla Community Hall
correct
retrieve
Generated Fact-aware Logical Queries:
1. Bristol Palin →people.person.education→m.02nqgpk →education.education.institution→Occidental College
2. Bristol Palin →people.person.education→m.0cpk4bl →education.education.institution→Wasilla High School
Logical Queries:
SELECT ?e1 ?e2 WHERE {
Bristol Palin people:education ?e1.
?e1 education:institution ?e2
}
extract
Knowledge Path Correction

Figure 1: An overview of our proposed Correcting on Graph (CoG) framework. We first introduce a structured knowledge decoding to parse the multi-hop question and generate fact-aware logical queries. Then, we introduce a knowledge path correction to correct the fact-aware logical queries. Finally, the retrieved knowledge paths are used to assist a LLM for joint answer prediction.

through a knowledge reciting task, enable it to leverage its parametric knowledge to generate logical queries for the question while filling in the blank intermediate entities during knowledge decoding. We then introduce a knowledge path correction to correct the fact-aware logical queries. The knowledge path correction combines the extracted logical queries and the KG to ensure the reliability and comprehensiveness of the knowledge paths. Finally, the retrieved knowledge paths are used to assist a LLM for joint answer prediction.

2.1 Structured Knowledge Decoding

Given a multi-hop question q and a KG G, previous semantic parsing methods, such as RoG (Luo et al., 2024), aim to generate a relation path P_r:

$$P_r = r_1 \to r_2 \to \cdots \to r_n. \qquad (1)$$

The relation path consists of a series of relations that can be used to retrieve the continuous knowledge path $e_q \xrightarrow{r_1} e_1 \xrightarrow{r_2} \ldots \xrightarrow{r_n} e_a$ connecting the question entity e_q to the answer entity e_a. However, without considering the question entity e_q and the intermediate entities, the excessive flexibility in the relation path generation process leads to the fact that the generated relation paths are more prone to hallucinations, leading to queries that sound plausible but fail to retrieve any information from the KG.

Knowledge Reciting. To address this issue, we first introduce a knowledge reciting task. Given a question q and the question entity e_q, our goal is to enable the LLM to directly generate a knowledge path that is helpful for solving the question:

$$P_k = e_q \xrightarrow{r_1} e_1 \xrightarrow{r_2} \ldots \xrightarrow{r_n} e_a. \qquad (2)$$

Compared to relation paths, the first constraint of a knowledge path is the determination of which question entity as the starting entity. In complex multi-hop QA datasets, a question commonly involves multiple question entities.[1] For example: “Which movie with an actor named Goran Kostic was directed by Angelina Jolie?" This question involves two question entities, “Angelina Jolie” and “Goran Kostic”. Each entity has a different role, one as an actor and the other as a director. Without imposing a constraint on the question entity, the logical query will inevitable introduce irrelevant knowledge, such as retrieving the movies in which Angelina Jolie (*the director*) starred.

Another constraint provided by the knowledge path is the intermediate entities. Since the LLM generates responses by selecting high-probability tokens step by step, LLMs can retrieve knowledge from its parameters by continuing the generation process. For example, when the LLM generates the first relation r_1, predicting the next intermediate entity is essentially a “retrieval” process, where the LLM fills in “$e_q \xrightarrow{r_1}$?” based on its parametric knowledge. Therefore, fine-tuning the LLM to

[1] For example, in the CWQ dataset (Talmor and Berant, 2018), 48.2% of questions have multiple question entities.

generate such structured outputs can enhance its ability to utilize the parametric knowledge.

To this end, we first construct question-path pairs using the training set of two multi-hop QA datasets. Each pair consists of a question q and the corresponding knowledge path P_k, which connects the question entity e_q to the answer entity e_a through a sequence of intermediate entities and relations.

We then fine-tune the LLM by using these question-path pairs. During fine-tuning, the LLM is trained to output the knowledge path P_k given an input question q and the question entity e_q.

$$\mathcal{L} = -\sum_{i=1}^{N} \log P_\theta(P_k|q_i, e_{q_i}). \tag{3}$$

On the one hand, the reciting process is injecting the factual knowledge into the parameters of the LLM. During fine-tuning, the LLM stores these knowledge paths in its parameters and establishes connections with the semantics of the question. On the other hand, the reciting process also teaches the LLM semantic parsing. When we remove the intermediate entities involved in the knowledge path, the remaining part becomes a logical query that can be executed on the KG.

Knowledge Decoding. During inference, we leverage the fine-tuned LLM to generate the fact-aware logical query (i.e., knowledge path) for each input question.

$$P_{pred} = \text{LLM}^{ft}(q, e_q). \tag{4}$$

Since complex multi-hop questions often contain multiple answers, we leverage the beam search strategy that allows the LLM to generate multiple possible knowledge paths. Each of these paths corresponds to different possible answers, ensuring better answer coverage of the complex multi-hop question.

Once a knowledge path is generated, it can be converted into a logical query by extracting the question entity and the relations along the path. We follow RoG (Luo et al., 2024) to extract relation paths to represent logical queries, while incorporating the question entity as a constraint.

$$LQ = e_q \xrightarrow{r_1} ? \xrightarrow{r_2} \dots \xrightarrow{r_n} ?. \tag{5}$$

2.2 Knowledge Path Correction

Due to the limited parametric knowledge and decoding strategy, the generated knowledge paths may contain factual errors or fail to cover all possible answers.

Hallucination Entity. Although the LLM integrated a set of factual knowledge into its parameters during knowledge reciting, it is difficult to cover billions of knowledge facts in the KG. As a result, the LLM will inevitably encounter entities they haven't met before during inference. In this case, when the LLM attempts to generate knowledge paths using the question entities and relations, the intermediate entities in the generated path may suffer from hallucination errors, resulting in generating plausible but fabricated knowledge paths.

Formally, given a question q and the question entity e_q, the LLM generates a knowledge path:

$$P_{pred} = e_q \xrightarrow{r_1} e_1 \xrightarrow{r_2} \dots \xrightarrow{r_n} e_n. \tag{6}$$

The hallucination entity refers to an intermediate entity e_j in P_{pred} where the triple (e_i, r_j, e_j) does not exist in the KG $(e_i, r_j, e_j) \notin G$.

Path Deficiency. Due to the difficulty in controlling the decoding process of the LLM directly, it may generate multiple knowledge paths from the question entity to the same answer entity. When handling questions that have multiple answers[2], a fixed beam width decoding strategy might result in the generated knowledge paths that fail to cover all possible answers.

Formally, given a question q and the question entity e_q, the LLM generates a set of knowledge paths $\mathbf{P}_{pred}$. Path deficiency refers to the situation where the path generated by the LLM does not include all possible answers. The generated knowledge paths $\mathbf{P}_{pred}$ include path shaped like $e_q \xrightarrow{r_1} \dots \xrightarrow{r_2} \dots \xrightarrow{r_n} \dots$, but there exists an answer entity e_a such that the knowledge path from e_q to e_a is not present in $\mathbf{P}_{pred}$. This can be represented as:

$$e_q \xrightarrow{r_1} e_1 \xrightarrow{r_2} \dots \xrightarrow{r_n} e_a \notin \mathbf{P}_{pred} \tag{7}$$

Correction. Both hallucination errors and path deficiencies in the knowledge paths can be corrected through the execution of logical queries. Specifically, for hallucinated entities, the logical query retrieves reliable paths by ensuring the intermediate entities in the generated path are consistent with

[2] In the WebQSP dataset (Berant et al., 2013), 48.8% of questions have multiple answers and 12.1% of the questions have more than 10 answers

Dataset	# Train	# Test	Solvable	Multi-hop
WebQSP	2,826	1,628	95.3%	30.0%
CWQ	27,639	3,531	80.1%	42.1%

Table 1: Data statistics of the WebQSP and CWQ. Solvable refers to the proportion of questions for which there exists at least one knowledge path from the question entity to the answer entity in the KG. Multi-hop refers to the proportion of multi-hop questions.

the KG. For path deficiencies, executing the logical query ensures that all possible answer entities are retrieved, filling in the gaps left by the original knowledge path. The correction process is formalized as:

$$\mathbf{P}_{\text{correction}} = \text{exec}(G, e_q \xrightarrow{r_1} ? \xrightarrow{r_2} \dots \xrightarrow{r_n} ?). \quad (8)$$

Here, the execution of the logical query ensures that the resulting paths are faithful. The hallucination entities are removed by verifying each intermediate step, and missing knowledge paths are added through a breadth-first search (BFS) starting from e_q with the relation path $r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_n$ as constraints.

3 Experiments

3.1 Experimental Setup

Datasets. We conduct experiments on two knowledge-intensive multi-hop QA benchmarks.

- ***WebQSP*** (Berant et al., 2013) is a multi-hop question answering dataset with 4,037 questions. We use the data splits of previous work (Luo et al., 2024) for fine-tuning and evaluation.
- ***CWQ*** (Talmor and Berant, 2018) is a complex multi-hop KGQA dataset that contains 34,672 questions with intricate constraints. We use the data splits of previous work (Luo et al., 2024) for fine-tuning and evaluation.

Knowledge Graphs. We use Freebase (Bollacker et al., 2007) as the KG for both the WebQSP and CWQ datasets. WebQSP is a multi-hop question answering dataset built on Freebase, while CWQ is an extension of WebQSP, where questions are derived from the same KG. To facilitate the experiments, we follow previous work (He et al., 2021b; Luo et al., 2024) to retrieve a local knowledge subgraph for each question. As shown in Table 1, the extracted subgraphs do not provide knowledge paths for all questions. However, to ensure fairness in the experiments, we retain those unsolvable questions following previous work.

Dataset	1-hop	2-hop	> 2-hop	Unsolvable
WebQSP	65.2%	30.0%	0.0%	4.7%
CWQ	38.0%	42.1%	0.1%	19.8%

Table 2: Statistics of the length of the *shortest* path for questions in WebQSP and CWQ. The *n*-hop indicates the percentage of questions where the answer can be reached in n hops. The unsolvable representing the percentage of questions that cannot be solved by the KG.

Dataset	#Ans = 1	2 ≤ ≤ 5	6 ≤ ≤ 9	10 ≤
WebQSP	50.0%	32.5%	6.2%	11.3%
CWQ	75.8%	19.2%	2.8%	2.2%

Table 3: Data statistics on the number of answers to the questions in the WebQSP and CWQ test datasets.

Implementation Details. The dataset used for knowledge reciting to fine-tune the LLM is composed of the training sets from both WebQSP and CWQ. As shown in Table 2. Although CWQ is a complex multi-hop QA dataset, the majority of questions can be answered with knowledge paths of length 2 hops or less. Therefore, when constructing the reciting dataset for fine-tuning, we only extracted knowledge paths that are less than 3 hops to form question-path pairs. Moreover, we fine-tune the Llama 3.1-8B-instruct for structured knowledge decoding and leverage OpenAI api to call ChatGPT and GPT-4 for answer prediction and leverage Hit, Hit@1, and F1 metrics for evaluation. During decoding, we set the beam width to 2 and retain only one path for each possible answer. Further implementation details and evaluation metrics are shown in the Appendix A. Our code and data are available at https://github.com/HUSTNLP-codes/CoG.

Evaluation Metrics. We evaluate the performance on these multi-hop QA datasets using three evaluation metrics: Hit, Hit@1, and F1. Hit measures whether the model can correctly answer a given question. Specifically, it is considered a hit if the correct answer is present in the model's generated response. If the correct answer is included, the Hit value is set to 1 and otherwise 0. Hit@1 is used to evaluate whether the top-ranked candidate in the models response is correct. When the model generates multiple candidate answers, Hit@1 focuses on whether the highest-ranked candidate is correct. F1 is a comprehensive evaluation metric. As shown in Table 3, many questions contain mul-

Model	LLM	WebQSP			CWQ		
		Hit	Hit@1	F1	Hit	Hit@1	F1
Traditional KGQA Methods							
GRAFT-Net (Sun et al., 2018)	/	–	66.7	62.4	–	36.8	32.7
NSM (He et al., 2021a)	/	–	68.7	62.8	–	47.6	42.4
SR+NSM(+E2E) (Zhang et al., 2022)	/	–	69.5	64.1	–	50.2	47.1
NSM+h (He et al., 2021a)	/	–	74.3	67.4	–	48.8	44.0
UniKGQA (Jiang et al., 2023b)	/	–	77.2	72.2	–	51.2	49.1
LLM-based KGQA Methods							
KD-CoT (Wang et al., 2023)	ChatGPT	68.6	–	52.5	55.7	–	–
StructGPT (Jiang et al., 2023a)	ChatGPT	72.6	–	–	–	–	–
KB-BINDER (Li et al., 2023)	Codex	74.4	–	–	–	–	–
ToG+ChatGPT (Sun et al., 2024)	ChatGPT	76.2	–	–	58.9	–	–
ToG+GPT-4 (Sun et al., 2024)	GPT-4	82.6	–	–	69.5	–	–
KG-CoT (Zhao et al., 2024b)	ChatGPT	82.1	–	–	51.6	–	–
GoG (Xu et al., 2024)	ChatGPT	–	78.7	–	–	55.7	–
Interactive-KBQA (Xiong et al., 2024)	GPT-4	72.4	–	71.2	59.1	–	49.0
EffiQA+GPT-4 (Dong et al., 2025)	GPT-4	82.9	–	–	69.5	–	–
RoG (Luo et al., 2024)	Fine-tuned Llama2-7B-chat	85.7	80.0	70.8	62.6	57.8	56.2
GNN-RAG (Mavromatis and Karypis, 2024)	Fine-tuned Llama2-7B-chat	85.7	80.6	71.3	66.8	61.7	59.4
CoG+ChatGPT (Ours)	Fine-tuned Llama3.1-8B-Instruct	**88.5**	**81.4**	**76.9**	**68.9**	**62.5**	**60.4**
CoG+GPT-4 (Ours)	Fine-tuned Llama3.1-8B-Instruct	**90.5**	**83.3**	**78.0**	**70.3**	**63.9**	**60.8**

Table 4: The experiment results of GNN-based methods and LLM-based maethods on WebQSP and CWQ. The LLM column indicates the LLM used by each method. For example, ToG+ChatGPT uses ChatGPT as the LLM agent to traverse on the KG. RoG uses a fine-tuned Llama2-7B-chat model to perform both semantic parsing and question answering. CoG uses a fine-tuned Llama3.1-8B-Instruct model for semantic parsing, and uses ChatGPT and GPT-4 to perform answer prediction based on the corrected knowledge paths.

tiple answers in WebQSP and CWQ. The F1 score evaluates both the model's ability to recall the correct answers and its capacity to handle noisy or distracting information effectively.

Baselines. We compare with traditional KGQA methods and recent LLM-based KGQA methods.

For traditional KGQA methods, GRAFT-NET (Sun et al., 2018) leverages graph neural network to handle graph information for question answering. NSM (He et al., 2021a), SR+NSM (Zhang et al., 2022), and NSM+h (He et al., 2021a) leverage neural symbolic machine for semantic parsing over KGs. UniKGQA (Jiang et al., 2023b) introduces a pre-training task to perform retrieval and reasoning within a unified framework.

For LLM-based KGQA methods, KD-CoT (Wang et al., 2023) corrects the reasoning chain of LLMs by accessing the KG dynamically. FIT (Ye et al., 2023) and GRT (Zhao et al., 2024a) leverage cross-modal pre-training to enhance LLMs using KGs. StructGPT (Jiang et al., 2023a) and ToG (Sun et al., 2024) leverage LLMs to perform structured reasoning on the KG directly. KB-BINDER (Li et al., 2023) binds the relation and entity to constrain the path selection during multi-hop reasoning. KG-CoT (Zhao et al., 2024b) integrates Chain-of-Thought prompting with KG retrieval to guide LLMs in step-by-step reasoning using relevant KG facts. GoG (Xu et al., 2024) treats the LLM as an agent, allowing it to generate missing entities or relations on incomplete KGs for multi-hop reasoning. Interactive-KBQA (Xiong et al., 2024) enables interactions between LLMs and the KG, allowing the model to iteratively decompose and resolve complex queries. EffiQA (Dong et al., 2025) combines LLMs with a KG exploration model in a collaborative framework for question answering. RoG (Luo et al., 2024) fine-tunes the LLM to generate logical queries and perform joint reasoning. GNN-RAG (Mavromatis and Karypis, 2024) leverages GNNs to locate candidate answers and retrieve knowledge paths toward these entities.

3.2 Main Results

We show the performance comparison between our proposed Coin and several baseline models in Table 1. Our proposed Coin achieves the best results across all metrics on the two datasets.

Effectiveness of CoG. Compared to the LLM agent baselines, CoG outperforms ToG (Sun et al., 2024), with a 9.5% improvement in Hit on WebQSP and a 1.4% improvement on CWQ. These results highlight the capability of CoG to generate answer correctly across both datasets. Un-

Method	WebQSP			CWQ		
	P	R	F1	P	R	F1
CoG	79.3	**81.4**	**76.9**	**55.5**	**66.2**	**60.4**
CoG w/o decoding	64.2	51.9	61.0	38.6	38.2	38.8
CoG w/o correction	**80.4**	61.6	65.1	50.8	63.8	56.5

Table 5: Ablation studies on CoG. ’w/o decoding’ refers to directly retrieving 2-hop knowledge paths for answer prediction. ’w/o correction’ refers to using the generated knowledge paths for answer prediction.

Method	WebQSP	CWQ
ChatGPT	51.8	39.9
+ ToG (Sun et al., 2024)	76.2	58.9
+ RoG (Luo et al., 2024)	81.5	52.7
+ GNN-RAG (Mavromatis and Karypis, 2024)	85.3	64.1
+ CoG (Ours)	**88.5**	**68.9**

Table 6: Performance of different methods as knowledge retrievers (Hit). The retrieved knowledge paths are used for answer prediction under the same LLM.

like LLM agent methods (Wang et al., 2023; Jiang et al., 2023a; Sun et al., 2024) that require to iteratively interact with KGs, CoG leverages semantic parsing strategy and only requires one single interaction with KGs to retrieve knowledge paths during knowledge path correction. Moreover, CoG only requires two interactions with LLMs, which significantly reduces the inference latency.

Effectiveness of structured knowledge decoding. Compared to semantic parsing baselines (Li et al., 2023; Luo et al., 2024), CoG outperforms the state-of-the-art semantic parsing method RoG (Luo et al., 2024) in both Hit and Hit@1 metrics. This improvement is attributed to the knowledge reciting of the structured knowledge decoding, which enables the LLM to incorporate factual knowledge during fine-tuning while considering constraints on intermediate entities when generating knowledge paths.

Effectiveness of knowledge path correction. Compared with the state-of-the-art LLM-based methods, CoG also outperforms RoG (Luo et al., 2024) and GNN-RAG (Mavromatis and Karypis, 2024) on the F1 metric. CoG achieves an F1 score of 78.0 on WebQSP and 60.8 on CWQ, while GNN-RAG achieves 71.3 and 59.4, respectively. These improvements in F1 score indicate that the knowledge path correction method not only results in better answer recall but also effectively reduces the introduction of irrelevant answers. This demonstrates the advantage of CoG in generating faithful logical queries and knowledge paths.

Method	$\text{Hit}_{\text{question}}$	$\text{Hit}_{\text{answer}}$
CoG	**94.0%**	**47.2%**
CoG w/o correction	91.2% (↓ 2.8%)	21.2% (↓ 26.0%)
RoG (Luo et al., 2024)	88.6%	27.5%

Table 7: Answer hit rate of knowledge paths retrieved by CoG, CoG w/o correction, and RoG on WebQSP. ↓ denotes the performance drop compared to CoG.

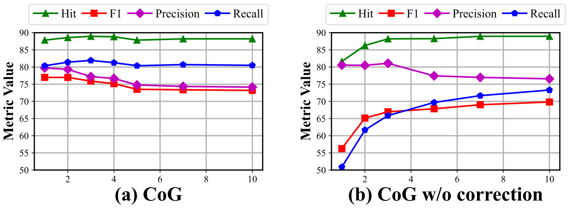

Figure 2: Performance of CoG and CoG w/o correction under different beam sizes on WebQSP.

3.3 Further Analysis

Ablation Study. In Table 4, we evaluate the effectiveness of each component in CoG. **1)** w/o decoding refers to directly retrieving 2-hop knowledge paths from the KG, without using fine-tuned LLM to generate knowledge paths and logical queries. **2)** w/o correction refers to directly using the knowledge paths generated by the fine-tuned LLM for answer prediction, without correcting the hallucination entity and path deficiency.

The experimental results demonstrate the effectiveness of structured knowledge decoding. Directly retrieving 2-hop knowledge paths introduces a large amount of irrelevant knowledge paths, which prevents the LLM from answering the question correctly. Furthermore, the recall drops significantly without knowledge path correction, as the uncorrected knowledge paths may contain hallucinated entities or path deficiencies. This demonstrates that knowledge path correction can significantly enhance the capability of the LLM to recall correct answers.

Quality of Knowledge Paths. In Table 5, we treat ToG (Sun et al., 2024), RoG (Luo et al., 2024), GNN-RAG (Mavromatis and Karypis, 2024), and our proposed CoG as knowledge path retrievers, and use ChatGPT for answer prediction based on the retrieved knowledge paths. The experimental results show that the knowledge paths retrieved by CoG significantly improve accuracy of question answering. Since both RoG (Luo et al., 2024) and GNN-RAG (Mavromatis and Karypis, 2024) use the fine-tuned LLM both knowledge

Case 1: Hallucination Entity
Question: What is the nationality of Khaosai Galaxy? [Thailand]
Decoding: Khaosai Galaxy →*people.person.nationality*→United States of America. **Logical Query:** Khaosai Galaxy →*people.person.nationality*→? **Correction:** Khaosai Galaxy →*people.person.nationality*→Thailand. (*corrected*)
Case 2: Path Deficiency
Question: Who played in Barbara Gordon? [Melinda McGraw, Hannah Gunn, Ilyssa Fradin]
Decoding: Barbara Gordon →*film.character.portrayed_in_films*→m.0c04kpn→*film.performance.actor* →Melinda McGraw Barbara Gordon→*film.character.portrayed_in_films*→m.0y54_x4→*film.performance.actor*→Hannah Gunn **Logical Query:** Barbara Gordon →*film.character.portrayed_in_films*→?→*film.performance.actor*→? **Correction:** Barbara Gordon →*film.character.portrayed_in_films*→m.0c04kpn→*film.performance.actor*→Melinda McGraw, Barbara Gordon →*film.character.portrayed_in_films*→m.0y54_x4→*film.performance.actor*→Hannah Gunn Barbara Gordon →*film.character.portrayed_in_films*→m.041w0vy→*film.performance.actor* →Ilyssa Fradin (*corrected*)

Table 8: The case studies demonstrate the hallucinated entity and path deficiency in the generated knowledge paths during knowledge decoding, and how knowledge path correction addresses these issues using logical queries.

retrieval and answer prediction, this experiment also demonstrates that the superior performance of CoG is attributed to the high quality of the retrieved knowledge paths, rather than only relying on powerful LLMs for answer prediction.

Effectiveness of Path Correction. To further demonstrate the effectiveness of knowledge path correction, we evaluate the answer hit rate of the retrieved knowledge paths in Table 7. $\text{Hit}_{\text{question}}$ indicates that if the retrieved path hits at least one correct answer of the question, $\text{Hit}_{\text{question}}$ is 1, otherwise 0. $\text{Hit}_{\text{answer}}$ refers to the hit rate for all answer of a question. For example, if a question has three answers, we consider the hit rate for each answer in the retrieved knowledge paths.

The experiment shows that the corrected knowledge paths allow more questions to contain knowledge paths leading to the correct answer, which is the reason why CoG achieves good performance in Hit and Hit@1 in Table 4. On the other hand, the results demonstrate that for questions with multiple answers, the corrected knowledge paths can cover more possible answers, revealing why CoG can significantly improve recall and F1 performance in Table 5.

Impact of the Beam Width during Decoding. During structured knowledge decoding, we increase the beam width during decoding to allow the LLM to generate more possible knowledge paths. However, as shown in Figure 2 a, as the beam width increases, the performance first improves and then decreases. This is because introducing too many knowledge paths may mislead the LLM with irrelevant knowledge. As shown in Figure 2 b, correction plays a crucial role in improving the quality of the knowledge paths. Directly using the knowledge paths generated by the LLM cannot cover all the answers, leading to low recall and F1. As a result, it requires a very large beam width to ensure the performance.

3.4 Case Study

Case 1: Hallucination Entity. Case 1 shows that when the LLM generates a knowledge path from a question entity, it may suffer from hallucination when completing the next-hop neighboring entity. This is because the LLM may encounter entities or knowledge they haven't met before during the fine-tuning stage. However, the attribute information for the generated intermediate entity, such as whether it is a person or a city, can constrain the generation process of the next relation. In such cases, using the extracted logical queries, CoG can easily identify hallucination errors in the knowledge paths and correct them.

Case 2: Path Deficiency. Case 2 shows that the LLM may overlook knowledge paths leading to some possible answers. This is because there may be possible path toward these answers, and for the LLM, "retrieving" parametric knowledge through generation is difficult to control directly. In this case, knowledge path correction improves the coverage of answers by executing logical queries, ensuring that the retrieved knowledge path encompass more possible answers and enhancing the completeness of the generated paths.

4 Related Work

Knowledge Graph Question Answering (KGQA) aims to answer natural language questions by reasoning over structured KGs.

Semantic Parsing Methods. Semantic parsing methods transform input questions into logical queries executable over KGs. Neural Symbolic Machines (NSM) (He et al., 2021a), SR+NSM (Zhang et al., 2022), and NSM+h (He et al., 2021a) adopt neural-symbolic models to map questions into programs. QGG (Lan and Jiang, 2020) enhances logical form generation by incorporating constraints and extending relational paths. RNG-KBQA (Ye et al., 2022) introduces a ranker-generator framework for producing structured queries. ArcaneQA (Gu and Su, 2022) dynamically restricts the token space to control query generation and reduce errors. Besides, Embed-KGQA (Saxena et al., 2020) and TransferNet (Shi et al., 2021) formulate multi-hop QA as a link prediction problem. GRAFT-NET (Sun et al., 2018) applies graph neural networks to aggregate evidence from the KG. UniKGQA (Jiang et al., 2023b) unifies retrieval and reasoning in a single pre-trained framework.

LLM-based Methods Recent methods leverage LLMs for KGQA, either by integrating retrieved KG facts into reasoning steps or by guiding the entire reasoning process. Self-Ask (Press et al., 2023) decomposes complex questions into subquestions. IR-CoT (Trivedi et al., 2023) interleaves retrieval and reasoning via Chain-of-Thought prompting. KD-CoT (Wang et al., 2023) grounds each reasoning step in KG facts to reduce hallucinations. DoG (Li et al., 2024) iteratively retrieves and reasons over relevant knowledge. Interactive-KBQA (Xiong et al., 2024) frames QA as a multi-turn interaction with the KG. GoG (Xu et al., 2024) treats the LLM as both a reasoner and a generator of missing KG links. Moreover, StructGPT (Jiang et al., 2023a) and ToG (Sun et al., 2024) leverage LLMs as agents to simulate structured reasoning over KGs. EffiQA (Dong et al., 2025) combines LLMs with a KG exploration module to improve efficiency and faithfulness. RoG (Luo et al., 2024) fine-tunes LLMs to jointly generate queries and reason over retrieved knowledge paths. KG-CoT (Zhao et al., 2024b) incorporates KG evidence into each step of CoT reasoning. GNN-RAG (Mavromatis and Karypis, 2024) locates answer candidates using GNNs and retrieves paths supporting the answers.

5 Conclusion

In this paper, we propose a novel Correcting on Graph Framework to establish faithful logical queries that connect LLMs and KGs directly. We first fine-tune the LLM to recite factual knowledge paths, enabling it to generate fact-aware logical queries for the question. Then, we combine logical queries with KGs to correct the hallucination entity and path deficiency, ensuring the reliability and comprehensiveness of the retrieved knowledge. Extensive experiments demonstrate that CoG achieves a high answer hit rate and exhibits competitive F1 performance for complex multi-hop questions.

Limitations

Although generating logical queries to connect LLMs with KGs can effectively reduce frequent interactions between them, compared to the retrieve-then-read paradigm, generating logical queries often requires additional fine-tuning of the LLM to ensure it can accurately output the relation name correctly. On the other hand, fine-tuning is also necessary for LLMs to learn the specific schema of the KG. For example, retrieving a person's spouse might seem like a single-hop query, but it actually is a 2-hop query in Freebase. For instance, retrieving Richard Nixon's wife involves a 2-hop knowledge path: Richard Nixon → people.person.spouse_s → m.02h98gq → people.marriage.spouse → Pat Nixon, where "m.02h98gq" is an intermediate entity. Therefore, fine-tuning is essential for the LLM to learn the unique schema of a specific KG.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2023YFF0905503, National Natural Science Foundation of China under Grants No. 62472188.

References

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA*, pages 1533–1544. ACL.

Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. 2007. Freebase: A shared database of structured general human knowledge. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1962–1963.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2492–2501.

Zixuan Dong, Baoyun Peng, Yufei Wang, Jia Fu, Xiaodong Wang, Xin Zhou, Yongxue Shan, Kangchen Zhu, and Weiguo Chen. 2025. Effiqa: Efficient question-answering with strategic multi-model collaboration on knowledge graphs. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 7180–7194.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, and et al. 2024. The llama 3 herd of models. *arXiv preprint*, arxiv:2407.21783.

Yu Gu and Yu Su. 2022. ArcaneQA: Dynamic program induction and contextualized encoding for knowledge base question answering. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1718–1731.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021a. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 553–561.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021b. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *The Fourteenth ACM International Conference on Web Search and Data Mining, WSDM 2021, Virtual Event, Israel, March 8-12, 2021*, pages 553–561.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1772–1791.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248:1–248:38.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023a. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9237–9251.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 969–974.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020*, pages 9459–9474.

Kun Li, Tianhua Zhang, Xixin Wu, Hongyin Luo, James R. Glass, and Helen Meng. 2024. Decoding on graphs: Faithful and sound reasoning on knowledge graphs through generation of well-formed chains. *arxiv preprint*, arxiv:2410.18415.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6966–6980.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2016. Knowledge representation learning with entities, attributes and relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2866–2872.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Costas Mavromatis and George Karypis. 2024. GNN-RAG: graph neural retrieval for large language model reasoning. *arxiv preprint*, arxiv:2405.20139.

Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 647–656.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, and et al. 2023. GPT-4 technical report. *arxiv preprint*, arxiv:2303.08774.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711.

Apoorv Saxena, Aditay Tripathi, and Partha P. Talukdar. 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, ACL 2020, Online, July 5-10, 2020*, pages 4498–4507.

Jiaxin Shi, Shulin Cao, Lei Hou, Juanzi Li, and Hanwang Zhang. 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, EMNLP 2021, Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4149–4158.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W. Cohen. 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 EMNLP 2018, Brussels, Belgium, October 31 - November 4, 2018*, pages 4231–4242.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Alon Talmor and Jonathan Berant. 2018. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018*, pages 641–651.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037.

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

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Guanming Xiong, Junwei Bao, and Wen Zhao. 2024. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10561–10582.

Yao Xu, Shizhu He, Jiabei Chen, and et al. 2024. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 18410–18430.

Qichen Ye, Bowen Cao, Nuo Chen, Weiyuan Xu, and Yuexian Zou. 2023. Fits: Fine-grained two-stage training for knowledge-aware question answering. *arXiv preprint*, arXiv:2302.11799.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. RNG-KBQA: generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6032–6043.

Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. 2022. Subgraph retrieval enhanced model for multi-hop knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5773–5784.

Feng Zhao, Hongzhi Zou, and Cheng Yan. 2023. Structure-aware knowledge graph-to-text generation with planning selection and similarity distinction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8693–8703.

Ruilin Zhao, Feng Zhao, Liang Hu, and Guandong Xu. 2024a. Graph reasoning transformers for knowledge-aware question answering. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 19652–19660.

Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. 2024b. KG-CoT: chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 6642–6650.

Ruilin Zhao, Feng Zhao, Guandong Xu, Sixiao Zhang, and Hai Jin. 2022. Can language models serve as temporal knowledge bases? In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 2024–2037.

A Detailed Implementation

The dataset used for knowledge reciting to fine-tune the LLM is composed of the training sets from both WebQSP and CWQ. We perform a breadth-first search starting from each question entity to find the shortest path to the answer entity, and then use these shortest paths as the ground truth paths of the question for fine-tuning. We fine-tune the Llama 3.1-8B-instruct for structured knowledge decoding. During fine-tuning, the batch size is set to 4, the learning rate to 2e-5, the warmup ratio to 0.03, the learning rate scheduler type to cosine, and the number of epochs to 3. We use two NVIDIA A100 (80GB) GPUs for fine-tuning. For inference, we utilize the OpenAI API to call ChatGPT and GPT-4 for answer prediction.

In Table 7, we introduce $\text{Hit}_{\text{question}}$ and $\text{Hit}_{\text{answer}}$ to further evaluate the answer hit rate of the knowledge paths.$\text{Hit}_{\text{question}}$ is used to measure whether a knowledge path contains at least one correct answer. If the path contains a correct answer, $\text{Hit}_{\text{question}}$ is 1, otherwise it is 0. $\text{Hit}_{\text{answer}}$ measures how many correct answers are contained in the knowledge path. For each correct answer, if it is in the path, $\text{Hit}_{\text{answer}}$ is 1, otherwise it is 0. For example: if a question has three correct answers and the generated path only contains two of them, $\text{Hit}_{\text{question}}$ would be 1 because the path contains at least one correct answer. $\text{Hit}_{\text{answer}}$ would be 1 for the two correct answers in the path, and 0 for the missing one. In this way, $\text{Hit}_{\text{question}}$ and $\text{Hit}_{\text{answer}}$ reflect the effectiveness of the knowledge path from different perspectives.

Instruction:
Please generate a knowledge path starting from the question entity to possible answers for the following question.

Input:
Question:
# Qusetion
Question Entity:
# Question Entity 1, # Question Entity 2, ..., # Question Entity k

Output:
# Knowledge Path

Figure 3: The prompt template for structured knowledge decoding.

B Prompts

Figures 3 and 4 show the prompt templates we used in CoG.

During fine-tuning, we construct the training dataset using the prompt template shown in Figure 3. On one hand, we treat the LLM as a parametric knowledge base and force the LLM to memorize these knowledge facts during fine-tuning. In this way, LLM can generate possible knowledge paths using its stored parametric knowledge when handling with knowledge-intensive questions. On the other hand, we aim to make the LLM generate structured output which in the form of a continuous knowledge paths. In this way, the LLM can

Instruction:
Please answer the following question based on the knowledge paths. Please only return the answers, with one answer per line.

Input:
Knowledge Paths:
# Knowledge Path 1 …
# Knowledge Path 2 …
…
# Knowledge Path n

Question:
# Question

Output:
# Answer 1
# Answer 2
…
# Answer k

Figure 4: The prompt template for joint reasoning with knowledge paths.

start from the question entity for path generation. After generating the first relation, it treats the process of generating the next entity as a "fill in the blank" task. This approach prompts the LLM to utilize its parametric knowledge. thereby prompting the LLM to utilize its internal knowledge for question answering. The generated entities also implicitly carry various attributes (such as a person or a city), and these attributes help constrain the generation of the next relation, preventing the process from being overly flexible and reducing the occurrence of hallucination errors.

After correcting the knowledge paths, we combine the input question and corrected knowledge paths to allow the LLM to perform joint reasoning. By having the LLM output one answer per line during answer prediction, we make it easier to extract the answers generated by the LLM for calculating the F1 scores.