Plan-on-Graph: Self-Correcting Adaptive Planning of Large Language Model on Knowledge Graphs

Liyi Chen[1,2†], **Panrong Tong**[2], **Zhongming Jin**[2], **Ying Sun**[3], **Jieping Ye**[2*], **Hui Xiong**[3,4*]
[1] University of Science and Technology of China, [2] Alibaba Cloud Computing,
[3] Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou), [4] Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology
liyichencly@gmail.com, yings@hkust-gz.edu.cn, xionghui@ust.hk,
{panrong.tpr, zhongming.jinzm, yejieping.ye}@alibaba-inc.com

Abstract

Large Language Models (LLMs) have shown remarkable reasoning capabilities on complex tasks, but they still suffer from out-of-date knowledge, hallucinations, and opaque decision-making. In contrast, Knowledge Graphs (KGs) can provide explicit and editable knowledge for LLMs to alleviate these issues. Existing paradigm of KG-augmented LLM manually predefines the breadth of exploration space and requires flawless navigation in KGs. However, this paradigm cannot adaptively explore reasoning paths in KGs based on the question semantics and self-correct erroneous reasoning paths, resulting in a bottleneck in efficiency and effect. To address these limitations, we propose a novel self-correcting adaptive planning paradigm for KG-augmented LLM named Plan-on-Graph (PoG), which first decomposes the question into several sub-objectives and then repeats the process of adaptively exploring reasoning paths, updating memory, and reflecting on the need to self-correct erroneous reasoning paths until arriving at the answer. Specifically, three important mechanisms of *Guidance*, *Memory*, and *Reflection* are designed to work together, to guarantee the adaptive breadth of self-correcting planning for graph reasoning. Finally, extensive experiments on three real-world datasets demonstrate the effectiveness and efficiency of PoG.

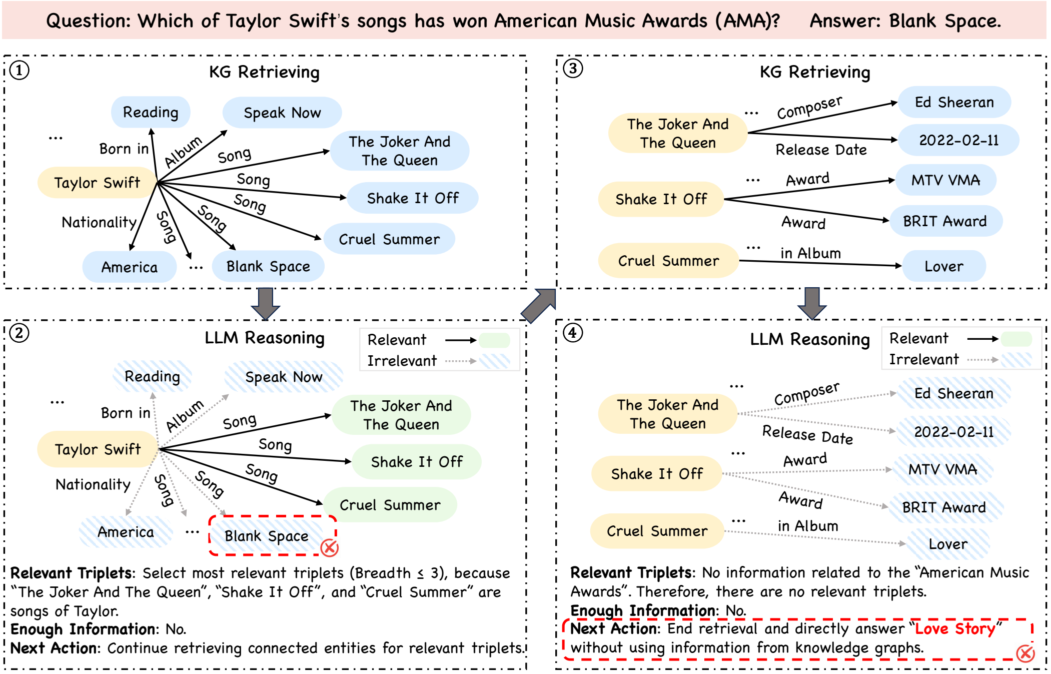
1 Introduction

Large Language Models (LLMs) have manifested outstanding performance in various natural language processing and data science tasks, such as question answering [42, 25, 62], text generation [18, 13, 8, 15], recommender systems [60, 59, 51, 44], and domain-specific applications [46, 45, 14, 50, 36]. They leverage advanced deep learning techniques and immense amounts of pre-existing text data to understand and generate human language with impressive fluency and coherence. Despite their success in numerous applications, LLMs still suffer from out-of-date knowledge, hallucinations, and opaque decision-making, highlighting the ongoing need for further investigation in this rapidly evolving field.

Intuitively, as large-scale structural knowledge bases, Knowledge Graphs (KGs) [5, 1, 12, 43] provide explicit and editable depictions of massive real-world knowledge, which have the potential to be a promising complement to the drawbacks of LLMs. Previous studies manage to integrate KGs into LLM pre-training [61, 47] or fine-tuning [53, 27] stage. However, these methods mainly compress structured knowledge in KGs into LLMs' parameters in a black-box fashion and still cannot fully

†This work was accomplished when Liyi Chen was an intern at Alibaba Cloud Computing.
*Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



Question: Which of Taylor Swift's songs has won American Music Awards (AMA)? Answer: Blank Space.

Relevant Triplets: Select most relevant triplets (Breadth ≤ 3), because "The Joker And The Queen", "Shake It Off", and "Cruel Summer" are songs of Taylor.
Enough Information: No.
Next Action: Continue retrieving connected entities for relevant triplets.

Relevant Triplets: No information related to the "American Music Awards". Therefore, there are no relevant triplets.
Enough Information: No.
Next Action: End retrieval and directly answer "**Love Story**" without using information from knowledge graphs.

Figure 1: A toy example of existing KG-augmented LLM paradigm.

enhance the flexibility, reliability, and transparency of LLMs. Therefore, several attempts [3, 2, 40] first retrieve information from KGs and then deliver explicit knowledge into LLMs. Under these circumstances, LLMs do not directly participate in the graph reasoning process, making these methods excessively dependent on the KG completeness. Recently, a KG-augmented LLM paradigm has been proposed to conduct graph reasoning, which treats the LLM as an agent to interactively explore related entities and relations on KGs and perform reasoning based on the retrieved knowledge. For instance, StructGPT [19] and ToG [35] predefine the breadth of reasoning paths explored on the KG, and leverage the LLM to iterate the process of unidirectionally extending along the reasoning paths relevant to the question and reasoning the answer using these reasoning paths. This KG-augmented LLM paradigm offers an opportunity for more comprehensively amalgamating the knowledge from both the KG and LLM by facilitating the step-by-step derivation of further insights.

However, existing paradigm may fail to plan the exploration of correct reasoning paths for many complex questions. Figure 1 illustrates an example of existing paradigm's limitations when answering the question "Which of Taylor Swift's songs has won American Music Awards? (AMA)". These limitations lie in: (1) *Predefined path breadth*: Existing paradigm requires manually setting the breadth of reasoning paths in KGs, and a fixed breadth may result in all the selected relations or entities being incorrect. When determining the relevance between paths and questions in step ②, due to the limited maximum breadth of three and the uncertainty surrounding the respective awards of songs, the LLM selected maximum numbers of songs, ignoring the correct entity "Blank Space". (2) *Irreversible exploration direction*: The path exploration in existing paradigm is unidirectional without the ability to self-correct. Even if the paths are incorrect, the LLM still continues to extend current incorrect paths and lead to the failure of reasoning on the KG. In step ③ and ④, since "The Joker And The Queen", "Shake It Off", and "Cruel Summer" were already chosen, the reasoning process continued on incorrect paths and the right answer was not found. (3) *Forgetting partial conditions*: During reasoning, the LLM may forget partial conditions in the question and cannot provide the answer that satisfies multiple conditions simultaneously. In step ④, the LLM only remembered the condition that the song was by Taylor Swift but forgot the condition about the song winning an AMA award, leading to an incorrect answer, "Love Story". Therefore, the reasoning of complex questions may heavily rely on adaptive exploration and self-correction of erroneous reasoning paths.

To address these limitations, we propose a novel self-correcting adaptive planning paradigm for KG-augmented LLM named **P**lan-**o**n-**G**raph (**PoG**). To the best of our knowledge, we are the first to design a reflection mechanism for self-correction and adaptive KG exploration into KG-augmented LLMs, effectively improving the ability and efficiency of LLM reasoning. Specifically, PoG first decomposes the question into several sub-objectives as guidance for planning exploration, and then repeats the process of adaptively exploring reasoning paths to access relevant KG data, updating memory to provide dynamic evidence for reflection, and reflecting on the need to self-correct reasoning paths until arriving at the answer. In PoG, three mechanisms are designed for adaptive self-correcting planning: (1) **Guidance**: To better guide adaptive exploration by harnessing conditions in the question, we employ the LLM to decompose the question into sub-objectives containing conditions, thereby benefiting the identification of relevant paths to each condition with flexible exploration breadth. (2) **Memory**: The information stored in memory offers historical retrieval and reasoning information for reflection. We record and update the *subgraph* to provide the LLM with all retrieved entities for initializing new exploration and self-correcting paths, *reasoning paths* to preserve the relationships between entities for LLM reasoning and allow for path correction, and *sub-objective status* to make the LLM recognize the known information of each condition and mitigate its forgetting in reflection stage. (3) **Reflection**: To determine whether to continue or self-correct current reasoning paths, we design a reflection mechanism to employ the LLM to reason whether to consider other entities into new exploration and decide which entities to backtrack to for self-correction based on information in memory. Finally, extensive experiments on three real-world KGQA datasets validate the effectiveness and efficiency of PoG [1]. The main contributions of this paper are listed as follows:

- We propose a novel self-correcting adaptive planning paradigm for KG-augmented LLM named PoG, which exploits the LLM to plan the adaptive breadth of reasoning paths and reflect to self-correct erroneous paths. To the best of our knowledge, we are the first to incorporate a reflection mechanism for self-correction and adaptive KG exploration into KG-augmented LLMs, effectively augmenting the LLM's reasoning ability.
- We specially design Guidance, Memory, and Reflection mechanisms for PoG. Guidance harnesses question conditions to better plan adaptive exploration by decomposing task into sub-objectives including conditions. Memory records the subgraph, reasoning paths, and sub-objective status to provide historical retrieval and reasoning information for Reflection. Based on Memory, Reflection reasons whether to self-correct reasoning paths and which entity to backtrack to for initiating new exploration.
- We conduct extensive experiments on three real-world KGQA datasets, namely CWQ, WebQSP, and GrailQA. The results demonstrate not only the effectiveness but also the efficiency of our proposed novel PoG paradigm for KG-augmented LLM.

2 Preliminary

Knowledge Graph (KG) stores massive factual knowledge in the form of a set of triplets: $G = \{(e, r, e') \,|\, e, e' \in E, r \in R\}$, where E and R denote the set of entities and relations, respectively.

Relation Paths are a sequence of relations: $z = \{r_1, r_2, ..., r_l\}$, where $r_i \in R$ denotes the i-th relation in the path and l denotes the length of the path.

Reasoning Paths are the instances of a relation path z in the KG: $p_z = e_0 \to r_1 e_1 \to r_2 e_2 \to ... \to r_l e_l$, where $e_i \in E$ denotes the i-th entity and r_i denotes the i-th relation in the relation path z.

Knowledge Graph Question Answering (KGQA) is the task of answering natural language questions based on a set of facts over the KG. Given a question q, a knowledge graph G, and topic entities T_q mentioned in q, the target of KGQA is to generate answers A_q to the question q. Following previous studies [35], we assume any entity $e_q \in T_q$ mentioned in q and answers $a_q \in A_q$ are labeled and linked to the corresponding entities in G, i.e., $T_q, A_q \subseteq E$.

3 Methodology

In this section, we introduce the technical details of the novel self-correcting adaptive planning paradigm for KG-augmented LLM named Plan-on-Graph (PoG). As illustrated in Figure 2, PoG consists of four key components: Task Decomposition, Path Exploration, Memory Updating, and Evaluation. PoG first decomposes the question into several sub-objectives as guidance of planning exploration and then repeats the process of adaptively exploring reasoning paths to access relevant KG data, updating memory to provide historical retrieval and reasoning information for reflection, and reflecting on the need to self-correct reasoning paths until arriving at the answer.

[1] `https://github.com/liyichen-cly/PoG`

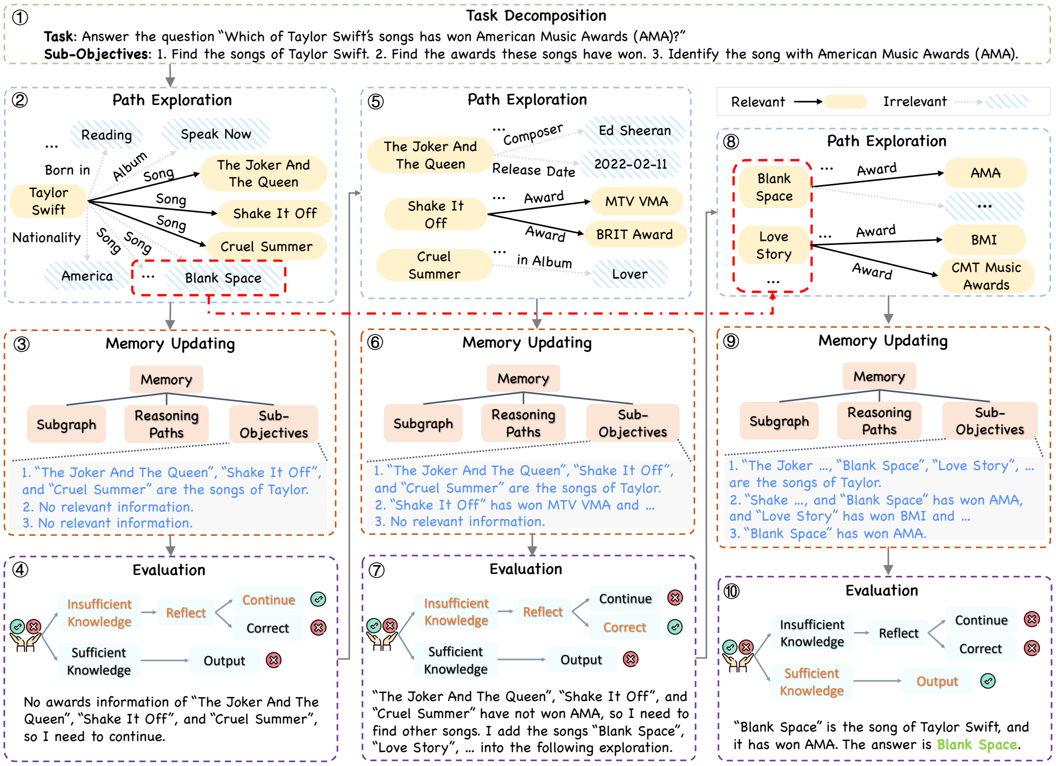

①
Task Decomposition
Task: Answer the question "Which of Taylor Swift's songs has won American Music Awards (AMA)?"
Sub-Objectives: 1. Find the songs of Taylor Swift. 2. Find the awards these songs have won. 3. Identify the song with American Music Awards (AMA).
② Path Exploration
...
Reading
Speak Now
Born in
Album
Song
The Joker And The Queen
Taylor Swift
Song
Shake It Off
Nationality
Song
Song
Song
Cruel Summer
America
...
Blank Space
③ Memory Updating
Memory
Subgraph
Reasoning Paths
Sub-Objectives
1. "The Joker And The Queen", "Shake It Off", and "Cruel Summer" are the songs of Taylor.
2. No relevant information.
3. No relevant information.
④ Evaluation
Insufficient Knowledge
Reflect
Continue
Correct
Sufficient Knowledge
Output
No awards information of "The Joker And The Queen", "Shake It Off", and "Cruel Summer", so I need to continue.
⑤ Path Exploration
...
Composer
Ed Sheeran
The Joker And The Queen
Release Date
2022-02-11
Shake It Off
...
Award
MTV VMA
Award
BRIT Award
Cruel Summer
...
in Album
Lover
⑥ Memory Updating
Memory
Subgraph
Reasoning Paths
Sub-Objectives
1. "The Joker And The Queen", "Shake It Off", and "Cruel Summer" are the songs of Taylor.
2. "Shake It Off" has won MTV VMA and ...
3. No relevant information.
⑦ Evaluation
Insufficient Knowledge
Reflect
Continue
Correct
Sufficient Knowledge
Output
"The Joker And The Queen", "Shake It Off", and "Cruel Summer" have not won AMA, so I need to find other songs. I add the songs "Blank Space", "Love Story", ... into the following exploration.
Relevant
Irrelevant
⑧ Path Exploration
Blank Space
...
Award
AMA
...
Love Story
...
Award
BMI
Award
CMT Music Awards
...
⑨ Memory Updating
Memory
Subgraph
Reasoning Paths
Sub-Objectives
1. "The Joker ..., "Blank Space", "Love Story", ... are the songs of Taylor.
2. "Shake ..., and "Blank Space" has won AMA, and "Love Story" has won BMI and ...
3. "Blank Space" has won AMA.
⑩ Evaluation
Insufficient Knowledge
Reflect
Continue
Correct
Sufficient Knowledge
Output
"Blank Space" is the song of Taylor Swift, and it has won AMA. The answer is Blank Space.

Figure 2: The framework overview of PoG, which includes four key components: Task Decomposition, Path Exploration, Memory Updating, and Evaluation.

3.1 Task Decomposition

To harness conditions in the question to better guide the adaptive exploration process, PoG decomposes the task of answering the question into multiple sub-objectives containing conditions through semantic analysis of the LLM. Sub-objectives serve as guidance for path exploration, benefiting the identification of relevant paths to each condition outlined in the question with flexible exploration breadth. Specifically, we prompt the LLM to decompose the original question q into a list of sub-objectives for KG retrieval and reasoning. The prompt is shown in Appendix A.1. The list of sub-objectives can be denoted as $O = \{o_1, o_2, o_3, ...\}$. It is important to note that sub-objectives in O may refer to the results obtained from other sub-objectives in O, allowing for interdependencies in the reasoning process.

3.2 Path Exploration

We access relevant information from the KG by exploring reasoning paths in the KG. On the initiation of path exploration, we localize the initial entities of reasoning paths, which correspond to the topic entities mentioned in the given question. Similar to prior research [19, 35], topic entities have been pre-identified and are part of the annotated datasets. Specifically, when presented with a question q, we use topic entities to serve as the initial elements of the reasoning paths, $E^0 = T_q = \{e_1^0, e_2^0, ..., e_{N_0}^0\}$, where N_0 is the number of topic entities.

In the subsequent iterations, we continue exploring reasoning paths most relevant to the question and suspend other reasoning paths. Taking the D-th iteration as an example, before the iteration starts, each reasoning path $p_n \in P$ consists of D_{p_n} ($D_{p_n} \leq D-1$) triplets, i.e., $p_n = \{(e_{s,n}^d, r_{j,n}^d, e_{o,n}^d)\}_{d=1}^{D_{p_n}}$, where $e_{s,n}^d$ and $e_{o,n}^d$ denote subject and object entities, $r_{j,n}^d$ is a specific relation between them, $(e_{s,n}^d, r_{j,n}^d, e_{o,n}^d)$ and $(e_{s,n}^{d+1}, r_{j,n}^{d+1}, e_{o,n}^{d+1})$ are linked to each other. It is noted that the length of each reasoning path may vary, because in the D-th iteration, we only continue exploring the reasoning paths most semantically relevant to the question, which are identified in the $D-1$-th iteration. The sets of tail entities and relations to be explored are denoted as $E^{D-1} = \{e_1^{D-1}, e_2^{D-1}, \ldots, e_{N_{D-1}}^{D-1}\}$ and $R^{D-1} = \{r_1^{D-1}, r_2^{D-1}, \ldots, r_{N_{D-1}}^{D-1}\}$, respectively, where N_{D-1} is the length of E^{D-1} and

R^{D-1}. We leverage the LLM to identify the most relevant entities E^D from the neighboring entities of the current entity set E^{D-1} based on the question q and extend the reasoning paths P with E^D. In order to manage the complexity of dealing with a large number of neighboring entities using the LLM, we propose an adaptive exploration strategy that is not limited by the fixed number of relations and entities. This strategy involves a two-step process of finding relevant relations and utilizing these selected relations to explore entities.

Relation Exploration. Relation exploration is a process to retrieve the relations of all tail entities in E^{D-1} and identify the most relevant relations to the question q and the sub-objectives O. To be specific, we first conduct the search to obtain all relations linked to the tail entities in E^{D-1} as the candidate relation set $R^D_{cand} = \{r^D_{cand,1}, r^D_{cand,2}, ..., r^D_{cand,N_{D-1}}\}$. We utilize R^D_{cand} to extend the reasoning paths into candidate reasoning paths P_{cand}. Then, we employ the LLM to select a flexible number of relevant reasoning paths P ending with the tail relations in R^D from P_{cand}, based on the semantic information of the question q, tail entities E^{D-1}, candidate relations R^D_{cand}, and sub-objectives O. The prompt is shown in Appendix A.2.1, and the pre-defined query for relation search is shown in Appendix B.1.

Entity Exploration. Analogously, entity exploration is a process to retrieve neighboring entities based on R^D and E^{D-1} and detect the most relevant entities to the question q. From the previous relation exploration, we obtain extended reasoning paths P and new tail relations R^D. For each reasoning path $p_n \in P$, we can execute the queries of $(e^{D-1}_n, r^D_n, ?)$ or $(?, r^D_n, e^{D-1}_n)$ to retrieve a candidate entity set $E^D_{cand,n}$, where e^{D-1}_n and r^D_n are the tail entity and relation in p_n. When confronted with a large number of candidate entities, we use a small pre-trained DistilBERT [31] [2], to calculate the similarity between candidate entities and the question for recall. Then, we summarize all candidate entity sets into E^D_{cand} and use E^D_{cand} as the tail entities to expand P into P_{cand}. With the candidate reasoning paths P_{cand}, we exploit the LLM to choose a flexible number of relevant reasoning paths P ending with the tail entities E^D from P_{cand}, based on the semantic information of the question q and knowledge triplets composed of tail entities E^{D-1}, tail relations R^D and candidate entities E^D_{cand}. The prompt is shown in Appendix A.2.2, and the pre-defined query for entity search is shown in Appendix B.2.

3.3 Memory Updating

The information stored in memory provides historical retrieval and reasoning information for reflection. After a two-step exploration, we dynamically update the searched subgraph G_{Sub}, reasoning paths P, and sub-objective status S in memory based on the ongoing reasoning process.

Subgraph. The subgraph includes all retrieved relations and entities from the KG. We update the subgraph in memory, which can be utilized during later reflection to determine which entity to backtrack to for self-correction. In the D-th iteration, the searched subgraph G_{Sub} is updated by adding the retrieved candidate relation set R^D_{cand} and candidate entity set E^D_{cand}.

Reasoning Paths. In order to ensure that the LLM can understand relationships between entities for better reasoning and allow for path correction in reflection stage, we update reasoning paths P to preserve the semantic structure within the KG.

Sub-Objective Status. The LLM may forget partial conditions in the reasoning process. Sub-objectives obtained by decomposing the question can help the LLM remember multiple conditions in the question. The status of sub-objectives contains the current known information related to the sub-objectives, which can aid the LLM in remembering the known information of each condition and determining whether to correct the exploration direction in reflection stage. Hence, we leverage the LLM to update the currently known information relevant to sub-objectives into sub-objective status $S = \{s_1, s_2, s_3, ...\}, |S| = |O|$, based on the semantic information of the question q, sub-objectives O, historical sub-objective status, and reasoning paths P, along with the LLM's own knowledge. The prompt is shown in Appendix A.3.

3.4 Evaluation

After the path exploration and memory updating, PoG prompts the LLM to reason whether the current acquired information, including sub-objective states and reasoning paths recorded in memory, is sufficient to infer an answer. The prompt is shown in Appendix A.4.1. If the LLM determines that the

[2] `https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b`

information is sufficient, it will integrate reasoning paths, sub-objective states, and its own knowledge to provide an answer. When information is considered insufficient, there may be two situations. One is that PoG will acquire sufficient information after further extension of current paths, and the other is that current paths are incorrect. Since the reasoning capability of the LLM does not always guarantee the correctness of path exploration, there is a need to self-correct erroneous reasoning paths. Therefore, we design a reflection mechanism to determine whether and how to self-correct reasoning paths. When the LLM believes that the information is insufficient, PoG enters the stage of reflection. Specifically, PoG utilizes the LLM to reflect on whether to correct the current exploration direction based on the question q, sub-objective status S, reasoning paths P, and entities planned for the next iteration of retrieval E^D from memory. Besides, the LLM will provide the reason for the reflection result. If the LLM judges that it is necessary to incorporate additional entities beyond those in E^D for exploration, then a self-correction of reasoning paths is needed. Otherwise, PoG will continue exploring along the current reasoning paths with tail entities in E^D. For self-correction, PoG employs the LLM to decide which entities in $E_{\text{cand}} = E^1_{\text{cand}} \cup E^2_{\text{cand}} \cup ... \cup E^D_{\text{cand}}$ to backtrack to based on sub-objective states in S and the reason for additional retrieval obtained from the reflection, and adds new exploration of backtracked entities E^D_{add} into E^D for the self-correction, denoted as $E^D = E^D \cup E^D_{\text{add}}$. The prompts for reflection are shown in Appendix A.4.2.

4 Experiments

4.1 Experimental Setups

4.1.1 Datasets & Evaluation Metrics

To demonstrate the effectiveness of PoG on complex reasoning over knowledge graphs, we adopt three representative multi-hop KGQA datasets: CWQ [37], WebQSP [56], and GrailQA [17]. All three datasets rely on the external knowledge graph from Freebase [5]. For the large dataset GrailQA, we utilize the same testing samples as those in ToG [35] to improve computational efficiency. Following prior research [23, 19, 35], we use exact match accuracy (Hits@1) as the evaluation metric.

4.1.2 Comparison Methods

Due to variations in the performance of the method across different datasets, we select prior state-of-the-art (SOTA) approaches as baselines for each dataset. They can be categorized into two groups: (1) *LLM-only methods*, including standard prompting (IO prompt) [6], Chain-of-Thought prompting (CoT) [49], and Self-Consistency (SC) [48]. (2) *KG-augmented LLM methods*, including fine-tuned and prompting methods. For CWQ and WebQSP, we utilize UniKGQA [20], TIARA [34], RE-KBQA [7], DeCAF [57], and RoG [27] as fine-tuned baselines and KD-CoT [41], KB-BINDER [23], StructGPT [19], Interactive KBQA [52], and ToG [35] as prompting baselines. For GrailQA, we utilize RnG-KBQA [55], TIARA [34], FC-KBQA [58], Pangu [16], FlexKBQA [26], and GAIN [33] as fine-tuned baselines and KB-BINDER [23] and ToG [35] as prompting baselines. The descriptions of baselines are presented in Appendix D.

Table 1: Performance comparison of different methods on CWQ and WebQSP.

Method	**CWQ**	**WebQSP**
LLM-Only		
IO Prompt [6]	37.6	63.3
CoT [49]	38.8	62.2
SC [48]	45.4	61.1
Fine-Tuned KG-Augmented LLM		
UniKGQA [20]	51.2	79.1
TIARA [34]	-	75.2
RE-KBQA [7]	50.3	74.6
DeCAF [57]	70.4	82.1
RoG [27]	62.6	85.7
Prompting KG-Augmented LLM w/GPT-3.5 or others		
KD-CoT [41]	50.5	73.7
KB-BINDER [23]	-	74.4
StructGPT [19]	54.3	72.6
ToG [35]	57.1	76.2
PoG	**63.2**	**82.0**
Prompting KG-Augmented LLM w/GPT-4		
InteractiveKBQA [52]	59.2	72.5
ToG [35]	67.6	82.6
PoG	**75.0**	**87.3**

4.2 Performance Comparison

We compare PoG with the SOTA baselines to demonstrate its effectiveness for KG-augmented LLM. Table 1 and Table 2 present the experimental results on CWQ, WebQSP, and GrailQA datasets. Overall, PoG achieves the best performance across all three datasets. Specifically, we can make the following observations. First, compared to all prompting KG-augmented LLM baselines, PoG shows superior performance advantages. Regardless of whether GPT-3.5 or

Table 2: Performance comparison of different methods on GrailQA.

Method	GrailQA			
	Overall	**I.I.D.**	**Compositional**	**Zero-shot**
LLM-Only				
IO Prompt [6]	29.4	-	-	-
CoT [49]	28.1	-	-	-
SC [48]	29.6	-	-	-
Fine-Tuned KG-Augmented LLM				
RnG-KBQA [55]	68.8	86.2	63.8	63.0
TIARA [34]	73.0	87.8	69.2	68.0
FC-KBQA [58]	73.2	88.5	70.0	67.6
Pangu [16]	75.4	84.4	74.6	71.6
FlexKBQA [26]	62.8	71.3	59.1	60.6
GAIN [33]	76.3	88.5	73.7	71.8
Prompting KG-Augmented LLM w/GPT-3.5 or others				
KB-BINDER [23]	50.6	-	-	-
ToG [35]	68.7	70.1	56.1	72.7
PoG	**76.5**	**76.3**	**62.1**	**81.7**
Prompting KG-Augmented LLM w/GPT-4				
ToG [35]	81.4	79.4	67.3	86.5
PoG	**84.7**	**87.9**	**69.7**	**88.6**

GPT-4 is used as the underlying LLM, PoG substantially outperforms the SOTA baseline, ToG. ToG explores reasoning paths with a fixed exploration breadth and cannot detect or correct the errors, showing limitations in effect and efficiency. Meantime, we specially design self-correction and adaptive planning mechanisms, which can effectively improve both performance and efficiency. Second, although PoG is a training-free prompting method, its performance is highly competitive with fine-tuned KG-augmented LLM baselines. When using GPT-4, the performance of PoG exceeds all fine-tuned KG-augmented LLM baselines across the board. Even with GPT-3.5, the result of PoG on GrailQA surpasses all fine-tuned KG-augmented LLM methods. This suggests that our designed guidance, memory, and reflection mechanisms allow PoG's effect to surpass most of the fine-tuned methods. Third, the improvement of PoG is obvious when compared to LLM-only baselines, which do not leverage external KGs. Besides, all KG-augmented LLM methods consistently outperform LLM-only methods, indicating the value of incorporating KGs to enhance LLM performance. Moreover, PoG further improves the effectiveness of KG-augmented LLMs through its self-correctable adaptive planning. Additionally, it is worth noting that PoG with GPT-3.5 outperforms other methods on the zero-shot subset of the GrailQA dataset by a large margin, apparently outperforming all fine-tuned KG-augmented LLMs on this category. The self-correction mechanism in PoG allows it to dynamically correct errors during the reasoning process, which is crucial for zero-shot problems.

4.3 Ablation Study

In order to assess the effectiveness of each mechanism and adaptive exploration in PoG, we conduct the ablation study to remove them on three datasets, respectively. Specifically, w/o Guidance refers to the variant where entire task decomposition as guidance is removed. w/o Memory indicates the variant without the memory mechanism. w/o Reflection refers to the variant where, in the case of insufficient information, it only continues exploring along the original reasoning paths. w/o Adaptive Breadth means that the variant uses a fixed exploration space breadth instead of adapting it based on the situation. Table 3 shows the performance of all variants and the results suggest that each mechanism and adaptive breadth appears to contribute positively to the overall performance, with their removal leading to weaker results on complex question answering tasks across the evaluated datasets. These variations achieve a minimum reduction of 3.0%, 2.1%, and 3.5% on CWQ, WebQSP, and GrailQA, respectively. The performance of w/o Memory drops the

Table 3: Performance of removing each mechanism and adaptive exploration, respectively.

Method	CWQ	WebQSP	GrailQA
PoG	**63.2**	**82.0**	**76.5**
w/o Guidance	60.1	80.3	72.4
w/o Memory	58.9	77.5	69.3
w/o Reflection	59.4	78.1	70.5
w/o Adaptive Breadth	61.3	80.2	73.8

Table 4: Efficiency comparison between our proposed PoG and the baseline ToG.

Dataset	Method	LLM Call	Input Token	Output Token	Total Token	Time (s)
CWQ	ToG	22.6	8,182.9	1,486.4	9,669.4	96.5
	PoG	**13.3**	**7,803.0**	**353.2**	**8,156.2**	**23.3**
WebQSP	ToG	15.9	6,031.2	987.7	7,018.9	63.1
	PoG	**9.0**	**5,234.8**	**282.9**	**5,517.7**	**16.8**
GrailQA	ToG	11.1	4,066.0	774.6	4,840.6	50.2
	PoG	**6.5**	**3,372.8**	**202.8**	**3,575.6**	**11.5**

most, followed by w/o Reflection, because without the memory there is no information to support PoG in navigating the exploration and achieving self-correction, and PoG cannot self-correct the wrong reasoning paths without the reflection mechanism. Moreover, after setting a fixed maximum breadth for exploration, the performance deteriorates. This indicates that a fixed breadth makes the method lack flexibility and less adaptable to different questions. However, because the mechanisms of memory and reflection ensure that PoG is able to self-correct, the performance of w/o Adaptive Breadth does not decrease a lot.

4.4 Efficiency Study

We study the efficiency of PoG and the SOTA prompting KG-augmented LLM baseline, ToG. Table 4 presents the average LLM call, token consumption, and time required by both methods to answer a question across three datasets. In all datasets, PoG demonstrates clear advantages over ToG in terms of all metrics. For average number of LLM calls, PoG consistently requires fewer calls to the LLM, and reduces it by at least 40.8%. This highlights PoG's ability to reason more efficiently with fewer LLM interactions. Regarding token consumption, PoG exhibits a notable advantage in both input and output token usage. On CWQ, compared to ToG's input tokens, PoG shows a reduction of approximately 4.6% in input token consumption. As for output tokens, PoG produces just 353.159 output tokens, representing a substantial decrease of roughly 76.2%. This indicates the effectiveness of PoG in reducing the overall token consumption during the reasoning process. Most importantly, PoG achieves superior time efficiency compared to ToG. On CWQ and GrailQA, PoG presents a speedup of over 4 times. ToG predefines the breadth of exploration, leading to the exploration of many irrelevant paths. Additionally, ToG lacks a self-correction mechanism, and when there is insufficient information to answer a question, it can only extend the current reasoning paths, sacrificing a lot of efficiency on irrelevant explorations. By contrast, the efficiency advantages of PoG can be attributed to its adaptive exploration and self-correction of reasoning paths based on the semantics of the question. The adaptive breadth reduces unnecessary exploration, and effective correction avoids the extending of wrong current paths.

4.5 Case Study

Figure 3 shows a typical case from the testing results on CWQ dataset. We compare the results of PoG, ToG, and CoT in answering the question "Who is in control of the place where the movie 'The Naked and the Dead' takes place?". The underlying LLMs they used are all based on GPT-3.5. PoG initially identifies a flexible number of relations related to the topic entities. Specifically, for "The Naked and the Dead", PoG successfully discovers that the movie takes place in Panama, while for "President of Panama", the LLM thinks that only the relation "government.government_office_or_title.jurisdiction" is relevant. Upon retrieval, no information is found regarding the person in control of Panama. This triggers reflection as PoG realizes that it lacks sufficient information. With the memory, PoG refers to the sub-objective status and recognizes that it already knows the movie location (Panama) for sub-objective #1 but is unaware of the person in control of Panama for sub-objective #2. Based on the current reasoning paths, PoG makes a decision to execute self-correction and returns to exploring the relation not previously explored for "President of Panama". Due to the task decomposition, during the self-correction process, it becomes easier to identify the correct relation "government.government_office_or_title.office_holders" according to the sub-objectives. Through the guidance, memory, and reflection mechanisms, PoG successfully finds the correct answer, "Juan Carlos Varela". In contrast, ToG fails to identify the most relevant relation concerning "President of Panama" and continues exploring incorrect paths. This consumes a significant amount of time and ultimately leads to an erroneous answer due to the hallucination. CoT refuses to answer directly since the LLM realizes its lack of knowledge regarding the answer and requires additional information to be provided. From this analysis, it is evident that PoG outperforms ToG and CoT. PoG successfully

Question: Who is in control of the place where the movie "The Naked and the Dead" takes place?	
CoT	**Answer**: [Refused to answer.] I'm sorry, but I need to clarify the question a bit further in order to provide an accurate answer. Could you please specify whether you are referring to the character or the entity in control of the place where the movie "The Naked and the Dead" takes place? Thank you for your understanding.
ToG	**Answer**: [KG exploration failed, and the LLM answered the question according to its own knowledge.] First, the movie "The Naked and the Dead" takes place during World War II. Second, the United States Army is in control of the place where the movie takes place. **Paths**: The Naked and the Dead → *film.film.country* → United States of America ↘ *film.film.featured_film_locations* → Panama ↖ President of Panama → *government.government_office_or_title.jurisdiction* ↘ ↘ *government.governmental_jurisdiction.government_positions* ⋯ *government.government_position_held.office_position_or_title* ⋯
PoG	**Answer**: [Through reflection to self-correct paths, the answer was found on the revised paths.] The movie "The Naked and the Dead" takes place in Panama. The President of Panama, Juan Carlos Varela, is in control of Panama based on the knowledge triplets. **Sub-Objectives**: 1. Identify the place where the movie "The Naked and the Dead" takes place; 2. Determine the authority or governing body in control of that place. **Sub-Objective Status**: 1. The movie "The Naked and the Dead" takes place in Panama; 2. The President of Panama, Juan Carlos Varela is in control of Panama. **Paths**: The Naked and the Dead → *film.film.country* → United States of America ↘ *film.film.featured_film_locations* → Panama ↖ President of Panama → *government.government_office_or_title.jurisdiction* ↘ *government.government_office_or_title.office_holders* → m.010gg02t Juan Carlos Varela ← *government.government_position_held.office_holder* ↙

Figure 3: A typical case to compare different methods to answer the complex question. For the convenience of display, we only provide the sub-objective status and partial reasoning paths stored in memory. Topic entities, wrong answers, and correct answers are highlighted in blue, red, and green. The revised path is highlighted with a yellow background.

leverages sub-objective status to self-correct the exploration path in the reflection stage and finally provides the correct answer.

5 Related Work

LLM Reasoning. To encourage LLMs to engage in reasoning rather than simply providing answers directly, many researchers instruct LLMs to generate the process of thinking in their outputs [49, 22, 63]. In the early stages, Chain of Thought (CoT) [49] was designed to provide a few examples of intermediate natural language reasoning steps as the prompt. After that, several variants of CoT reasoning with different forms like Tree-of-Thought [54], Graph-of-Thought [4], Memory of Thought [24], and Skeleton-of-Thought [30] were proposed to enhance the thinking process. However, LLMs may make mistakes during the reasoning process. Hence, many works [32, 21, 28, 29] designed self-correction mechanisms based on feedback to rectify flawed reasoning and ensure accuracy. Additionally, large efforts were dedicated to guiding LLMs in understanding complex graph structures [39, 38] and improving their graph reasoning across different graph tasks [11, 10, 9]. However, it is still an open issue to address the outdated knowledge, hallucinations, and opaque decision-making for LLM reasoning.

KG-Augmented LLM. Despite the pre-training of LLMs on massive corpora, they still suffer from limitations such as outdated knowledge, hallucinations, and opaque decision-making. An effective approach to address these limitations is to leverage KGs for explicit and editable knowledge provision to LLMs. Previous studies integrated KGs into LLM pre-training [61, 47] or fine-tuning [53, 27] stage, but they merely inject structured knowledge into LLMs' parameters and still leave these limitations unexplored. Therefore, several works [3, 2, 40] first retrieved information from KGs and then directly fed explicit knowledge into LLMs. In this way, LLMs do not involve the graph reasoning process and cannot provide potential insights. Then, a novel KG-augmented LLM paradigm [19, 35] was proposed to treat the LLM as an agent to interactively explore related entities and relations on KGs and perform reasoning based on the retrieved knowledge. Although this KG-augmented LLM paradigm has achieved impressive performance, it still faces the challenges of adaptively exploring the KG based on question semantics and self-correcting erroneous reasoning paths. To the best of our knowledge, our work stands out as a pioneering effort in successfully integrating a reflection mechanism for self-correction and adaptive KG exploration into KG-augmented LLMs, effectively enhancing the LLM's reasoning ability.

6 Conclusion

In this paper, we proposed a novel self-correcting adaptive planning paradigm for KG-augmented LLM named Plan-on-Graph (PoG). To the best of our knowledge, we were the first to incorporate a reflection mechanism for self-correction and adaptive KG exploration into KG-augmented LLMs, effectively augmenting LLM's reasoning ability and efficiency. PoG first decomposed the question into several sub-objectives, and then repeated the process of exploring reasoning paths, updating memory, and reflecting on the need to self-correct reasoning paths until arriving at the answer. To be specific, three important mechanisms were designed to work together to guarantee the adaptive breadth of self-correcting planning for graph reasoning, i.e., Guidance, Memory, and Reflection. Finally, extensive experiments on three real-world KGQA datasets validated not only the effectiveness but also the efficiency of the proposed PoG.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Key Research and Development Program of China (Grant No. 2023YFF0725001), the National Natural Science Foundation of China (Grant No. 92370204, 62306255), the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023B1515120057, 2024A1515011839), the Guangzhou-HKUST (GZ) Joint Funding Program (Grant No. 2023A03J0008), and the Alibaba Research Intern Program.

References

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *international semantic web conference*, pages 722–735. Springer, 2007.

[2] Agnes Axelsson and Gabriel Skantze. Using large language models for zero-shot natural language generation from knowledge graphs. In *Proceedings of the Workshop on Multimodal, Multilingual Natural Language Generation and Multilingual WebNLG Challenge (MM-NLG 2023)*, pages 39–54, 2023.

[3] 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.

[4] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[5] 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.

[6] 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.

[7] Yong Cao, Xianzhi Li, Huiwen Liu, Wen Dai, Shuai Chen, Bin Wang, Min Chen, and Daniel Hershcovich. Pay more attention to relation exploration for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2119–2136, 2023.

[8] Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8297–8309, 2023.

[9] Liyi Chen, Zhi Li, Yijun Wang, Tong Xu, Zhefeng Wang, and Enhong Chen. Mmea: Entity alignment for multi-modal knowledge graph. In *International Conference on Knowledge Science, Engineering and Management*, pages 134–147. Springer, 2020.

[10] Liyi Chen, Zhi Li, Tong Xu, Han Wu, Zhefeng Wang, Nicholas Jing Yuan, and Enhong Chen. Multi-modal siamese network for entity alignment. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 118–126, 2022.

[11] Liyi Chen, Chuan Qin, Ying Sun, Xin Song, Tong Xu, Hengshu Zhu, and Hui Xiong. Collaboration-aware hybrid learning for knowledge development prediction. In *Proceedings of the ACM on Web Conference 2024*, pages 3976–3985, 2024.

[12] Liyi Chen, Ying Sun, Shengzhe Zhang, Yuyang Ye, Wei Wu, and Hui Xiong. Tackling uncertain correspondences for multi-modal entity alignment. In *Proceedings of the 38th Conference on Neural Information Processing Systems*, 2024.

[13] Xi Chen, Xinjiang Lu, Haoran Xin, Wenjun Peng, Haoyang Duan, Feihu Jiang, Jingbo Zhou, and Hui Xiong. A table-to-text framework with heterogeneous multidominance attention and self-evaluated multi-pass deliberation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 607–620, 2023.

[14] Xi Chen, Chuan Qin, Zhigaoyuan Wang, Yihang Cheng, Chao Wang, Hengshu Zhu, and Hui Xiong. Pre-dygae: Pre-training enhanced dynamic graph autoencoder for occupational skill demand forecasting. In *Proceedings of the 33th International Joint Conference on Artificial Intelligence*, 2024.

[15] Zheng Gong and Ying Sun. Graph reasoning enhanced language models for text-to-sql. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2447–2451, 2024.

[16] Yu Gu, Xiang Deng, and Yu Su. Don't generate, discriminate: A proposal for grounding language models to real-world environments. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4928–4949, 2023.

[17] Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488, 2021.

[18] Bin Ji, Huijun Liu, Mingzhe Du, and See-Kiong Ng. Chain-of-thought improves text generation with citations in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18345–18353, 2024.

[19] 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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, 2023.

[20] 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 International Conference on Learning Representations*, 2023.

[21] Byoungjip Kim, Youngsoo Jang, Lajanugen Logeswaran, Geon-Hyeong Kim, Yu Jin Kim, Honglak Lee, and Moontae Lee. Prospector: Improving llm agents with self-asking and trajectory ranking. 2023.

[22] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[23] 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.

[24] Xiaonan Li and Xipeng Qiu. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374, 2023.

[25] Xiaoxi Li, Yujia Zhou, and Zhicheng Dou. Unigen: A unified generative framework for retrieval and question answering with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8688–8696, 2024.

[26] Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18608–18616, 2024.

[27] 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*, 2024.

[28] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Large language models can do parallel decoding. In *The Twelfth International Conference on Learning Representations*, 2024.

[31] V Sanh. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[32] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Yiheng Shu and Zhiwei Yu. Distribution shifts are bottlenecks: Extensive evaluation for grounding language models to knowledge bases. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 71–88, 2024.

[34] Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. Tiara: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8108–8121, 2022.

[35] 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*, 2024.

[36] Ying Sun, Hengshu Zhu, Lu Wang, Le Zhang, and Hui Xiong. Large-scale online job search behaviors reveal labor market shifts amid covid-19. *Nature Cities*, 1(2):150–163, 2024.

[37] 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.

[38] Yanchao Tan, Hang Lv, Xinyi Huang, Jiawei Zhang, Shiping Wang, and Carl Yang. Musegraph: Graph-oriented instruction tuning of large language models for generic graph mining. *arXiv preprint arXiv:2403.04780*, 2024.

[39] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2023.

[40] Jianing Wang, Qiushi Sun, Nuo Chen, Xiang Li, and Ming Gao. Boosting language models reasoning with chain-of-knowledge prompting. *arXiv preprint arXiv:2306.06427*, 2023.

[41] 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.

[42] Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19162–19170, 2024.

[43] Shuyao Wang, Yongduo Sui, Chao Wang, and Hui Xiong. Unleashing the power of knowledge graph for recommendation via invariant learning. In *Proceedings of the ACM on Web Conference 2024*, pages 3745–3755, 2024.

[44] Shuyao Wang, Yongduo Sui, Jiancan Wu, Zhi Zheng, and Hui Xiong. Dynamic sparse learning: A novel paradigm for efficient recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 740–749, 2024.

[45] Tianfu Wang, Qilin Fan, Chao Wang, Leilei Ding, Nicholas Jing Yuan, and Hui Xiong. Flagvne: A flexible and generalizable rl framework for network resource allocation. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, 2024.

[46] Tianfu Wang, Li Shen, Qilin Fan, Tong Xu, Tongliang Liu, and Hui Xiong. Joint admission control and resource allocation of virtual network embedding via hierarchical deep reinforcement learning. *IEEE Transactions on Services Computing*, 17(03):1001–1015, 2024.

[47] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

[48] 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 *International Conference on Learning Representations*, 2023.

[49] 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.

[50] Shiwei Wu, Joya Chen, Tong Xu, Liyi Chen, Lingfei Wu, Yao Hu, and Enhong Chen. Linking the characters: Video-oriented social graph generation via hierarchical-cumulative gcn. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4716–4724, 2021.

[51] Wei Wu, Chao Wang, Dazhong Shen, Chuan Qin, Liyi Chen, and Hui Xiong. Afdgcf: Adaptive feature de-correlation graph collaborative filtering for recommendations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1242–1252, 2024.

[52] Guanming Xiong, Junwei Bao, and Wen Zhao. 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 (Volume 1: Long Papers*, pages 10561–10582, 2024.

[53] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[54] 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.

[55] Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 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)*, pages 6032–6043, 2022.

[56] 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.

[57] 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 International Conference on Learning Representations*, 2023.

[58] Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. Fc-kbqa: A fine-to-coarse composition framework for knowledge base question answering. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[59] Shengzhe Zhang, Liyi Chen, Chao Wang, Shuangli Li, and Hui Xiong. Temporal graph contrastive learning for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9359–9367, 2024.

[60] Yuting Zhang, Ying Sun, Fuzhen Zhuang, Yongchun Zhu, Zhulin An, and Yongjun Xu. Triple dual learning for opinion-based explainable recommendation. *ACM Transactions on Information Systems*, 42(3):1–27, 2023.

[61] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, 2019.

[62] Lili Zhao, Qi Liu, Linan Yue, Wei Chen, Liyi Chen, Ruijun Sun, and Chao Song. Comi: Correct and mitigate shortcut learning behavior in deep neural networks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 218–228, 2024.

[63] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *The International Conference on Learning Representations*, 2023.

Appendix

A Prompts

Here, we provide all the prompts used in PoG. To facilitate the LLM output parsing, we require the LLM to provide answers using specific data structures, such as lists and JSON. Besides, we require the LLM not to output any other irrelevant information to the results. The specific in-context few-shot is shown in code files.

A.1 Task Decomposition

```
Please break down the process of answering the question into as few sub-objectives
as possible based on semantic analysis.

In-Context Few-Shot

Now you need to directly output sub-objectives of the following question in list
format without other information or notes.
Q: {}
```

A.2 Path Exploration

A.2.1 Relation Exploration

```
Please provide as few highly relevant relations as possible to the question and its
sub-objectives from the following relations (separated by semicolons).

In-Context Few-Shot

Now you need to directly output relations highly related to the following question
and its sub-objectives in list format without other information or notes.
Q: {}
Sub-Objectives: {}
Topic Entity: {}
Relations: {}
```

A.2.2 Entity Exploration

```
Which entities in the following list ([] in Triples) can be used to answer the
question? Please provide the minimum possible number of entities, and strictly
adhering to the constraints mentioned in the question.

In-Context Few-Shot

Now you need to directly output the entities from [] in Triplets for the following
question in list format without other information or notes.
Q: {}
Triplets: {}
```

A.3 Memory Updating

```
Based on the provided information (which may have missing parts and require further
retrieval) and your own knowledge, output the currently known information required
to achieve the sub-objectives.

In-Context Few-Shot

Now you need to directly output the results of the following question in JSON format
 without other information or notes.
Q: {}
```

```
Sub-Objectives: {}
Memory: {}
Knowledge Triplets: {}
```

A.4 Evaluation

A.4.1 Answer Question

```
Please answer the question based on the memory, related knowledge triplets and your
knowledge.

In-Context Few-Shot

Now you need to directly output the results of the following question in JSON format
 (must include "A" and "R") without other information or notes. If the triplets
explicitly contain the answer to the question, prioritize the fact of the triplet
over memory.
Q: {}
Memory: {}
Knowledge Triplets: {}
```

A.4.2 Reflection

```
Based on the current set of entities to be retrieved and the known information
including memory and triplets, is it necessary to add additional entities for
answering the question?

In-Context Few-Shot

Now you need to directly output the results of the following question in the JSON
format (must include "Add" and "Reason") without other information or notes.
Q: {}
Entities set to be retrieved: {}
Memory: {}
Knowledge Triplets: {}
```

```
Please select the fewest necessary entities to be retrieved for answering the Q from
 Candidate Entities, based on the current known information (Memory), the reason for
 additional retrieval, and your own knowledge.

In-Context Few-Shot

Now you need to directly output the results for the following Q in the list format
without other information or notes.
Q: {}
Reason: {}
Candidate Entities: {}
Memory: {}
```

B Search SPARQL

To automatically process the KG data in PoG, we pre-define the SPARQL for Freebase queries, which can be executed by filling in the entity's mid and relation.

B.1 Relation Search

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?relation
WHERE {
  ns:mid ?relation ?x .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?relation
WHERE {
  ?x ?relation ns:mid .
}
```

B.2 Entity Search

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
  ns:mid ns:relation ?tailEntity .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
  ?tailEntity ns:relation ns:mid .
}
```

B.3 Entity Name Search

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?tailEntity
WHERE {
  {
    ?entity ns:type.object.name ?tailEntity .
    FILTER(?entity = ns:mid)
  }
  UNION
  {
    ?entity <http://www.w3.org/2002/07/owl#sameAs> ?tailEntity .
    FILTER(?entity = ns:mid)
  }
}
```

C Datasets

In this paper, we use three complex multi-hop KGQA datasets: ComplexWebQuestions [37], WebQSP [56], and GrailQA [17]. The statistics of datasets are shown in Table 5. WebQSP contains questions from WebQuestions that are answerable by Freebase. It tests I.I.D. generalization on questions. ComplexWebQuestions (CWQ) extends WebQSP and encompasses four types of complex questions: conjunction, composition, comparative, and superlative. GrailQA is a diverse KGQA dataset built on Freebase, and is designed to test three levels of generalization of models: I.I.D., compositional, and zero-shot.

Table 5: Statistics of KGQA datasets.

Dataset	**Answer Format**	**Train**	**Test**	**Licence**
ComplexWebQuestions	Entity	27,734	3,531	-
WebQSP	Entity/Number	3,098	1,639	CC Licence
GrailQA	Entity/Number	44,337	1,000	-

D Baseline Descriptions

The baselines we compare can be categorized into two groups: (1) LLM-only methods; (2) KG-augmented LLM methods, including fine-tuned and prompting methods.

LLM-Only Methods

- Standard prompting (IO prompt) [6] verifies the ability of LLMs to achieve better performance in task-agnostic, few-shot problems than traditional LMs.
- Chain-of-Thought prompting (CoT) [49] generates a series of intermediate reasoning steps in prompts to help LLMs perform better in several NLP tasks.
- Self-Consistency (SC) [48] samples multiple, diverse reasoning paths through few-shot CoT, and uses the generations to select the most consistent answer.

Finetuned KG-Augmented LLM Methods

- UniKGQA [20] unifies the graph retrieval and reasoning process into a single model with LLMs.
- TIARA [34] first uses BERT to retrieve a set of schema items, which are further used as the input, together with the question, to T5 for plan generation. They also apply constrained decoding but only for grammaticality.
- RE-KBQA [7] capitalizes relations in KGs to enhance entity representations and introduce additional supervision to improve the selection of reasoning paths.
- DeCAF [57] combines semantic parsing and LLMs reasoning to jointly generate answers, which also reach salient performance on KGQA tasks.
- RoG [27] collaborates LLMs with KGs to achieve trustworthy reasoning to leverage structural information.
- RnG-KBQA [55] first uses BERT to rank a set of enumerated candidate programs (up to a limited complexity), and then uses T5 to edit the top programs into more complex programs.
- FC-KBQA [58] proposes a fine-to-coarse composition framework to avoid knowledge entanglement and guarantee both generalization ability and logical interpretability.
- Pangu [16] considers leveraging the discriminative ability of LLMs. It consists of a symbolic agent with a cooperative neural LLM.
- FlexKBQA [26] is a flexible KGQA framework with LLMs. It can utilize a limited set of annotated data to build KGQA for different KGs and query languages.
- GAIN [33] pays attention to the robustness of KGQA models. It proposes a data augmentation method to alleviate this problem and further evaluates the distribution shifts including from different aspects.

Prompting KG-Augmented LLM Methods

- KB-BINDER [23] is developed to challenge the heterogeneity of items from different KGs. It enables few-shot in-context learning over KGQA tasks.
- KD-CoT [41] retrieves relevant knowledge from KGs to generate faithful reasoning plans for LLMs.
- StructGPT [19] defines the interface of KG data to implement knowledge access and filtering with finite quantity, and leverage the LLM to infer the answer or subsequent planning repeatedly.
- Interactive KBQA [52] interacts with KGs directly and then generates logical forms. The interactions are under three designed universal APIs for KGs.
- ToG [35] iteratively retrieves relevant triplets from KGs and employs the LLM to assess whether the reasoning paths in beam search are sufficient for answering the question and if further retrieval of the next hop is necessary.

E Implementation Details

In our experiments, we use GPT-3.5 and GPT-4 to serve as the underlying LLMs. We call them by the OpenAI official API [3]. We set the temperature parameter to 0.3, frequency penalty to 0, and presence penalty to 0. The maximum token length for generation is 1024. In all experiments, the depth of exploration is set to 4 to avoid endless exploration. The experiments are conducted on a server with two Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz and 256 GB RAM memory.

F Depth Sensitivity

Since LLMs are not entirely certain about when to stop, we need to manually set the depth of KG exploration to avoid endless exploration. To investigate the impact of exploration depth on PoG performance, we conduct experiments with depth settings ranging from 1 to 5 on CWQ dataset. As shown in Figure 4, increasing the depth will improve the performance of PoG. Beyond a depth of 4, the improvement becomes less noticeable. The increase in depth leads to exponential growth in resource and time consumption. Considering the balance between efficiency and effectiveness, we set the depth to 4.

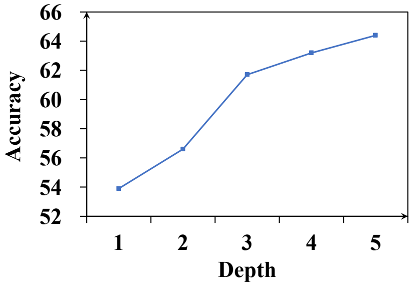


Figure 4: The impact of exploration depth on the performance of PoG.

G Case Analysis

In PoG, we design a reflection mechanism to provide the opportunity for self-correction for exploring reasoning paths. In Figure 5, we calculate the proportion of cases with reverse occurrences among all questions in CWQ, and the results show that 24% of cases involve reversing during the exploration process to achieve self-correction. This demonstrates that LLMs are indeed not always capable of making correct judgments in KG exploration and that self-correction is necessary for KG-augmented LLMs. Figure 6 presents the proportion of correct answers obtained by PoG after self-correction on three datasets. Overall, the self-correction in PoG appears to have positively impacted the accuracy of KGQA, particularly for the WebQSP and CWQ datasets, where the proportion of correct answers reached 64% and 48% after the self-correction process. This analysis suggests that the reflection mechanism in PoG has the potential to enhance the reasoning capabilities of KG-augmented LLM and improve the performance across various datasets by allowing for self-correction and exploration of alternative reasoning paths.

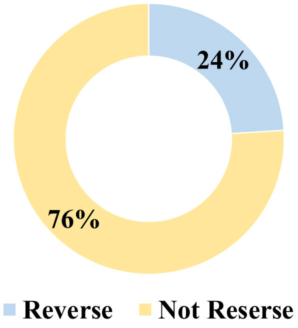


Figure 5: The proportion of cases with reverse occurrences among all data.

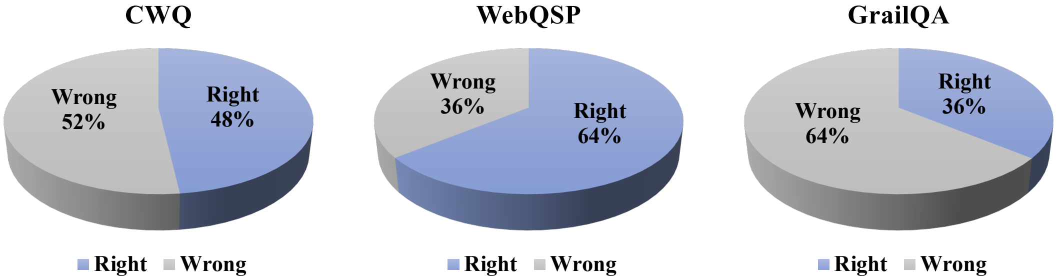
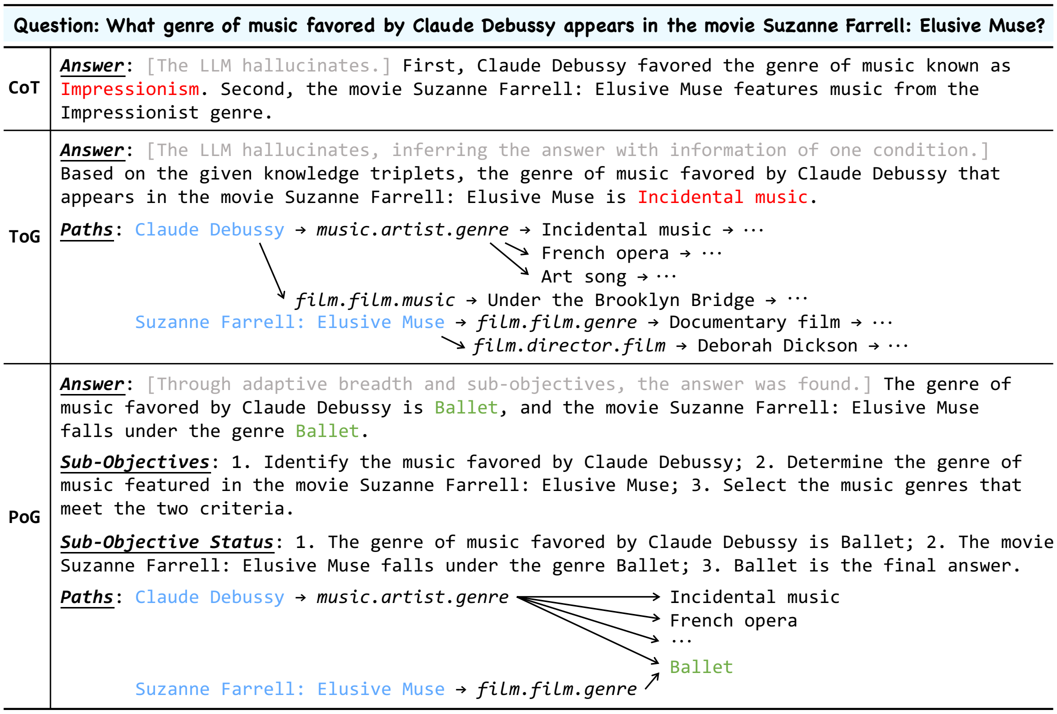


Figure 6: The proportion of correct answers obtained by PoG after self-correction.

Besides, Figure 7 shows another typical case from the testing results on CWQ dataset. We compare the results of PoG, ToG, and CoT in answering the question "What genre of music favored by Claude Debussy appears in the movie Suzanne Farrell: Elusive Muse?". PoG first adaptively identifies the

[3] https://platform.openai.com/docs/api-reference.



Question: What genre of music favored by Claude Debussy appears in the movie Suzanne Farrell: Elusive Muse?

CoT

Answer: [The LLM hallucinates.] First, Claude Debussy favored the genre of music known as Impressionism. Second, the movie Suzanne Farrell: Elusive Muse features music from the Impressionist genre.

ToG

Answer: [The LLM hallucinates, inferring the answer with information of one condition.] Based on the given knowledge triplets, the genre of music favored by Claude Debussy that appears in the movie Suzanne Farrell: Elusive Muse is Incidental music.

Paths: Claude Debussy → *music.artist.genre* → Incidental music → ⋯

French opera → ⋯
Art song → ⋯
film.film.music → Under the Brooklyn Bridge → ⋯
Suzanne Farrell: Elusive Muse → *film.film.genre* → Documentary film → ⋯
film.director.film → Deborah Dickson → ⋯

PoG

Answer: [Through adaptive breadth and sub-objectives, the answer was found.] The genre of music favored by Claude Debussy is Ballet, and the movie Suzanne Farrell: Elusive Muse falls under the genre Ballet.

Sub-Objectives: 1. Identify the music favored by Claude Debussy; 2. Determine the genre of music featured in the movie Suzanne Farrell: Elusive Muse; 3. Select the music genres that meet the two criteria.

Sub-Objective Status: 1. The genre of music favored by Claude Debussy is Ballet; 2. The movie Suzanne Farrell: Elusive Muse falls under the genre Ballet; 3. Ballet is the final answer.

Paths: Claude Debussy → *music.artist.genre* → Incidental music

Figure 7: A typical case to compare different methods to answer the complex question. For the convenience of display, we only provide the sub-objective status and partial reasoning paths stored in memory. Topic entities, wrong answers, and correct answers are highlighted in blue, red, and green.

most relevant relation "music.artist.genre" to the topic entity "Claude Debussy". Without constraining the breadth of reasoning paths, PoG considers multiple candidate entities as potentially relevant to the question. In the subsequent exploration of the topic entity "Suzanne Farrell: Elusive Muse", PoG adaptively chooses only "Ballet" as the relevant entity, as it records the known information of sub-objective #1 in memory. Through adaptive breadth and memorization of sub-objective status, PoG successfully and efficiently provides the correct answer. In contrast, ToG randomly explores paths when faced with multiple candidate entities, only finding one condition from sub-objective #1. Finally, ToG only remembers the genre of music favored by "Claude Debussy" but forgets the condition from sub-objective #2, answering "Incidental music". CoT directly hallucinates an irrelevant answer, "Impressionism". This case indicates the effectiveness of adaptive breadth and memorization of sub-objective status.

H Broader Impact & Limitation

In the current research landscape, PoG carries a significant broader impact, primarily reflected in its enhancement of complex reasoning capabilities for KG-augmented LLM. By innovatively integrating guidance, memory, and reflection mechanisms, PoG not only strengthens the model's flexibility and accuracy when facing complex queries but also enhances its ability to self-correct erroneous reasoning paths. This self-correcting adaptive planning paradigm enables the model to backtrack and adjust reasoning directions when faced with invalid initial assumptions or impasses, resulting in an optimal solution search. Additionally, the broader impact of PoG is manifested in several other aspects: (1) Improving Efficiency and Effectiveness in Problem-Solving: By dynamically adjusting exploration breadth and employing self-correction mechanisms, PoG can more efficiently handle complex questions and provide more accurate answers, significantly enhancing the overall performance of KGQA systems. (2) Enhancing the Robustness and Adaptability of LLMs: Through its memory mechanism, which records and tracks the completion status and reasoning paths of each sub-objective, PoG enables the LLM to more robustly deal with the uncertainty and complexity of questions, making it more precise and reliable across a wide range of applications. (3) Fostering Innovation in the Field of Artificial Intelligence: PoG's integration of meta-cognitive capabilities into reasoning and planning processes represents an innovative attempt that could further propel research and innovation in broader AI technologies within the artificial intelligence field. (4) Improving User Experience and Expanding Application Domains: With PoG's reasoning capabilities, user experience is greatly improved due to more accurate and quicker responses. Meanwhile, the domains where it can be applied will also expand, particularly in environments that require handling complex queries and responses involving large volumes of data.

There are still limitations in using PoG for addressing more complex problems. Some of the key limitations include: (1) Low Self-Confidence: LLMs are still not entirely certain about what information is needed, how many steps are required to extract the information, when to perform dynamic updates, or if the current information is sufficient. In future work, we will focus on the evaluation of LLM's self-confidence. For example, this can be alleviated by training a small model specifically for this evaluation task to improve accuracy. (2) Efficiency: Answering complex questions requires multiple steps. In future work, we aim to design strategies to reduce steps and improve task execution efficiency in situations of high self-confidence. (3) Non-Standardized Query: For less standardized queries, semantic understanding might be insufficient due to limitations in the capabilities of the LLM itself, leading to decreased effectiveness. In future work, we will address this issue by employing SOTA query rewriting methods or interacting with the user to refine the query.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please see the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the Appendix H.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please see Section 3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see Section 4.1 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets were released in public GitHub.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see Appendix E and Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Please see Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conduct the research with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please see Appendix H.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets are from `https://github.com/IDEA-FinAI/ToG/tree/main/data`, and we present the licenses in Appendix C.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We will provide the documentation with assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.