BeamAggR: Beam Aggregation Reasoning over Multi-source Knowledge for Multi-hop Question Answering

Zheng Chu[1], **Jingchang Chen**[1], **Qianglong Chen**[2*], **Haotian Wang**[1]
Kun Zhu[1], **Xiyuan Du**[1], **Weijiang Yu**[3], **Ming Liu**[1,4*], **Bing Qin**[1,4]
[1]Harbin Institute of Technology, Harbin, China
[2]Zhejiang University [3]Sun Yat-sen University [4]Peng Cheng Laboratory
{zchu,jcchen,mliu}@ir.hit.edu.cn, chenqianglong.ai@gmail.com

Abstract

Large language models (LLMs) have demonstrated strong reasoning capabilities. Nevertheless, they still suffer from factual errors when tackling knowledge-intensive tasks. Retrieval-augmented reasoning represents a promising approach. However, significant challenges still persist, including inaccurate and insufficient retrieval for complex questions, as well as difficulty in integrating multi-source knowledge. To address this, we propose Beam Aggregation Reasoning (**BeamAggR**), a reasoning framework for knowledge-intensive multi-hop QA. BeamAggR explores and prioritizes promising answers at each hop of question. Concretely, we parse the complex questions into trees, which include atom and composite questions, followed by bottom-up reasoning. For atomic questions, the LLM conducts reasoning on multi-source knowledge to get answer candidates. For composite questions, the LLM combines beam candidates, explores multiple reasoning paths through probabilistic aggregation, and prioritizes the most promising trajectory. Extensive experiments on four open-domain multi-hop reasoning datasets show that our method significantly outperforms SOTA methods by 8.5%. Furthermore, our analysis reveals that BeamAggR elicits better knowledge collaboration and answer aggregation.

1 Introduction

Large language models have showcased impressive performance across various NLP tasks (OpenAI, 2023; Touvron et al., 2023). Furthermore, chain-of-thought (CoT) prompting further enhances the reasoning capabilities of LLMs (Wei et al., 2022; Kojima et al., 2022; Chu et al., 2023). However, when the question surpasses the knowledge boundaries of LLMs, it leads to factual errors, also known as hallucination. Retrieval-augmented generation (RAG) assists LLMs by retrieving supporting knowledge to alleviate factual errors, thereby drawing significant attention within the research community (Gao et al., 2023; Huang et al., 2023).

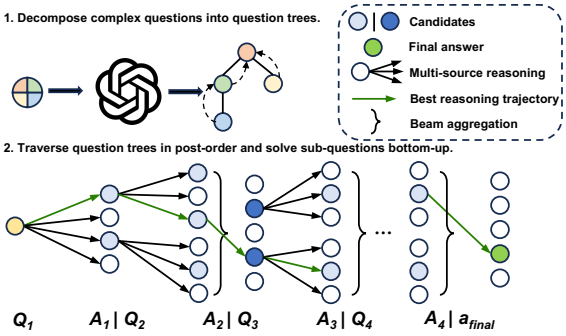

Figure 1: A brief overview of our method. Complex questions are decomposed into trees (top). Multi-source beam aggregation reasoning is conducted to find the best reasoning trajectory. (bottom)

Early attempts adopt a *retrieve-then-read* framework for one-time retrieval (Lazaridou et al., 2022; Borgeaud et al., 2022; Izacard et al., 2023; Zhang et al., 2023b). They use the original complex question as the retrieval query and achieve satisfactory performance in single-hop questions (Joshi et al., 2017; Kwiatkowski et al., 2019). Nonetheless, when confronted with complex multi-hop questions, one-time retrieval suffers from the issues of inaccurate and irrelevant retrieval, which greatly impair their performance in multi-hop reasoning.

Recent work introduces iterative multi-round retrieval (Khattab et al., 2022; Press et al., 2023; Trivedi et al., 2023; Jiang et al., 2023b; Shao et al., 2023). They use the content generated by LLMs for retrieval and, in turn, use the newly retrieved content for reasoning. Through the iterative alternation between retrieval-augmented reasoning and reasoning-augmented retrieval, the retrieval is substantially improved. Meanwhile, a portion of research decomposes complex questions into simple sub-questions, employing sub-question retrieval to

* Corresponding Authors: Ming Liu, Qianglong Chen

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1229–1248

obtain more precise information (Zhou et al., 2023; Cao et al., 2023; Park et al., 2023; Su et al., 2023).

However, there are still significant issues with these methods. Iterative retrieval is tough to achieve precise retrieval aligned with the model's reasoning. Sub-question retrieval grapples with the challenge of accurately aggregating answers, which causes cascading errors. Besides, when it comes to the open-domain setting, relying solely on knowledge from a single source proves inadequate for complex questions. Introducing multi-source knowledge may encounter knowledge conflicts, rendering efficient collaboration challenging, thus impeding its applications.

To address the aforementioned challenges, our research focuses on the following question: *How can models adaptively select and integrate knowledge from different sources during the reasoning, while reducing cascading errors in sub-questions aggregation?* Building upon this motivation, we propose Beam Aggregation Reasoning (**BeamAggR**) framework for knowledge-intensive multi-hop reasoning. Concretely, our method consists of three modules. (i) *question decomposition*: We begin by leveraging LLMs to decompose complex questions and convert them into question trees. The root node contains the original complex question, leaf nodes contain atomic sub-questions, and intermediate nodes contain composite questions that require compositional (or comparative) reasoning to obtain answers. Afterward, we traverse the question tree in post-order, employing bottom-up reasoning. (ii) *complementary multi-source reasoning*: For atomic questions, we conduct multi-source reasoning, followed by fine-grained answer aggregation, thus fostering knowledge collaboration. The aggregated answers are then normalized into a probability distribution, serving as candidates for beam aggregation. (iii) *beam aggregation*: For composite questions, we enumerate the combinations of their dependent sub-questions and conduct reasoning. Finally, the reasoning results are probabilistically aggregated, and the most promising predictions are selected.

In summary, our method explores reasoning trajectories at each hop of questions and prioritizes paths with higher likelihoods, thereby bringing reasoning insight and reducing cascading errors. Besides, the complementary multi-source reasoning helps alleviate knowledge omission and conflict.

We evaluate our method on four open-domain multi-hop reasoning datasets: HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). The experiments are conducted using GPT-3.5-turbo (Ouyang et al., 2022) and Mistral (Jiang et al., 2023a). Experimental results show that our method significantly outperforms the baselines on four datasets, with an average improvement of 8.5% compared to the previous state-of-the-art method, demonstrating its superiority. Furthermore, thorough analysis reveals the superiority of our approach to knowledge collaboration and answer aggregation.

Our contributions can be summarized as follows:

- We introduce BeamAggR, a framework for open-domain multi-hop reasoning, which outperforms the state-of-the-art methods.
- BeamAggR dynamically integrates multi-source knowledge in fine granularity during reasoning, fostering knowledge collaboration.
- BeamAggR broadens the scope of reasoning with beam combination and optimizes reasoning trajectories, mitigating cascading errors.

2 Related Work

2.1 Reasoning with Large Language Model

Wei et al. (2022) prompts LLMs to generate reasoning process before final answers, which is known as chain-of-thought (CoT) prompting. Since then, CoT prompting has been widely applied to enhance the reasoning capabilities of LLMs. Some work also designs instructions or clustering demonstrations for zero-shot reasoning (Kojima et al., 2022; Zhang et al., 2023c). Additionally, self-ensemble has been proven through extensive experiments to be an effective approach to improve performance. Wang et al. (2023) uses probabilistic sampling for multiple reasoning traces, while Qin et al. (2023) diversifies reasoning paths by using multiple languages CoTs. To address complex questions, Zhou et al. (2023); Dua et al. (2022) decompose them into sub-questions and solve them progressively, while Yao et al. (2023) models reasoning procedures as BFS or DFS search on reasoning trees.

In contrast, our method employs divide-and-conquer strategy, breaking down complex questions into trees and addressing them through bottom-up aggregation reasoning.

2.2 Retrieval-augmented Reasoning

Knowledge-sensitive tasks may induce factual hallucinations in LLMs, thus necessitating external

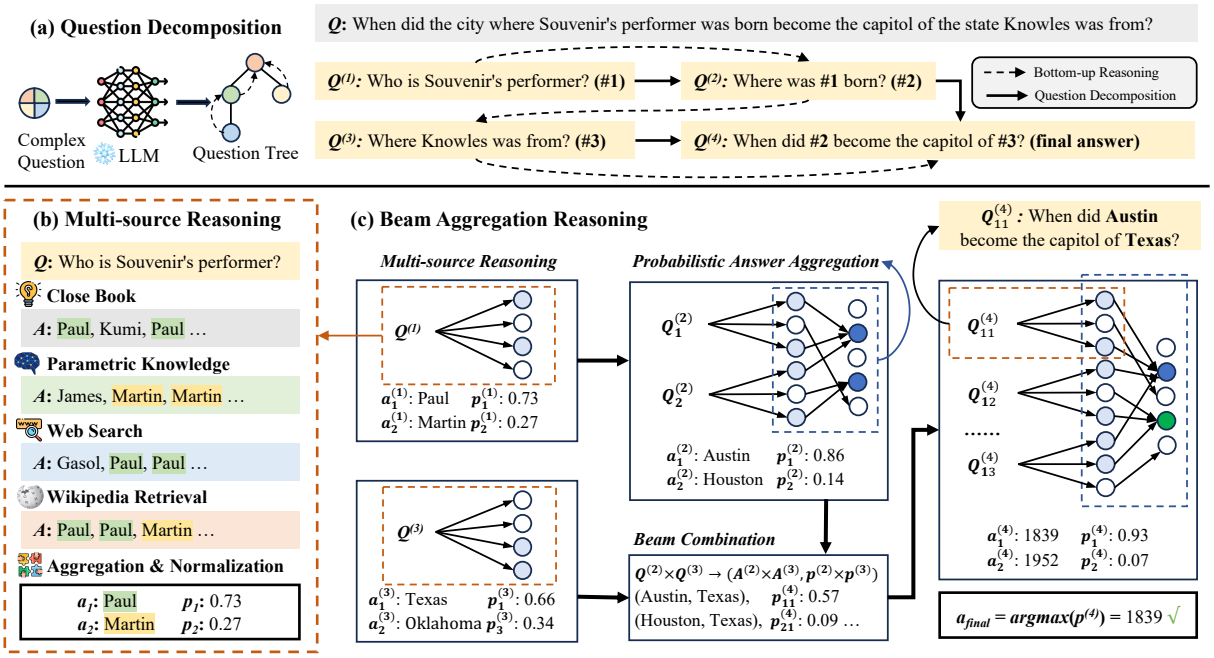

Figure 2: An overview of BeamAggR. (a) Question decomposition: decompose complex questions into trees and address them bottom-up (b) Multi-source reasoning: reason from diverse knowledge sources and normalize answers into a probability distribution (c) Beam aggregation reasoning: explore based on children's predictions, probabilistic aggregate answers and select the most promising reasoning trajectory.

retrieval for retrieval-augmented reasoning. Early work uses one-time retrieval, but they struggle to gather all the necessary knowledge to answer complex questions, resulting in knowledge omissions (Borgeaud et al., 2022; Lazaridou et al., 2022; Izacard et al., 2023). To address this, iterative retrieval is proposed. DSP (Khattab et al., 2022) engages in iterative interactions between retriever and reader through programmatically defined procedures. SelfAsk (Press et al., 2023) iteratively decomposes questions and solves them through the Google search. IRCoT (Trivedi et al., 2023) uses each reasoning step as a query for retrieval until obtaining the final answer. Similarly, ITER-RETGEN (Shao et al., 2023) conducts iterative retrievals by concatenating the output from the previous round with the original question. FLARE (Jiang et al., 2023b) introduces a lookahead mechanism, dynamically controlling the timing of retrieval based on reasoning confidence. Beam Retrieval (Zhang et al., 2023a) introduces an end-to-end framework designed to retrieve relevant paragraphs at each hop of questions through beam search. Meanwhile, some efforts achieve more precise retrieval by decomposing the problem into QDMR format (Wolfson et al., 2020). Cao et al. (2023) decomposes the problem into a tree, while Park et al. (2023) constructs a reasoning graph.

Compared to their approach, our method integrates diverse knowledge via complementary multi-source reasoning. Besides, it explores reasoning trajectories at each hop of questions and prioritizes the most promising path, thereby eliciting better answer aggregation and reducing cascading errors.

3 Beam Aggregation Reasoning

Beam Aggregation Reasoning decomposes complex questions into trees and conducts bottom-up reasoning. Throughout the bottom-up reasoning, it intricately amalgamates diverse knowledge sources, thereby mitigating the dearth of knowledge. Furthermore, beam aggregation probabilistically consolidates answers and selects reasoning pathways, consequently diminishing cascading errors.

Firstly, we decompose complex questions, then perform bottom-up multi-source knowledge reasoning in a post-order traversal manner, and employ beam aggregation at intermediate nodes until reaching the root node to obtain the final answer. The procedure is depicted in Algorithm 1, with an overview of the methodology illustrated in Figure 2. We will introduce question decomposition in § 3.1, multi-source reasoning in § 3.2 and beam aggregation in § 3.3. Detailed definitions of notations in the algorithm and formulas can be found in Table 7. Task definition is given in Appendix A.1.

Algorithm 1 Beam Aggregation Reasoning

Require: Complex multi-hop questions, Q
Require: Multi-source knowledge retriever, R
Require: Large language model, LLM
Require: Candidate size in aggregation, k

$Q_{decomp} = LLM(Q)$
for $N^{(i)}$ in PostOrderTraverse(Q_{decomp}) **do**
  if $N^{(i)}$ is leaf-node **then**
    $\hat{\boldsymbol{a}} = LLM(q^{(i)},\ R)$
    $\boldsymbol{a}^{(i)},\ \boldsymbol{p}^{(i)} = \text{Vote}(\hat{\boldsymbol{a}})[1:k]$
  else
    initialize $\hat{\boldsymbol{a}},\ \hat{\boldsymbol{p}} = list,\ list$
    for $c',\ p'$ in CartProd(sons($N^{(i)}$)) **do**
      $q' = \text{MaskFill}(q_i,\ c')$
      $\boldsymbol{a}^t, \boldsymbol{p}^t = \text{Vote}(LLM(q',\ c',\ R))$
      $\hat{\boldsymbol{a}} \leftarrow \text{Concat}(\hat{\boldsymbol{a}},\ \boldsymbol{a}^t)$
      $\hat{\boldsymbol{p}} \leftarrow \text{Concat}(\hat{\boldsymbol{p}},\ \boldsymbol{p}^t \cdot p')$
    end for
    $\boldsymbol{a}^{(i)},\ \boldsymbol{p}^{(i)} = \text{Aggr}(\hat{\boldsymbol{a}},\ \hat{\boldsymbol{p}})[1:k]$
  end if
end for
return $\boldsymbol{a}^{root}[1]$

3.1 Question Decomposition

Multi-hop questions entail complex structures, such as bridge, comparison, composition and their integration. To address this, we parse complex questions into trees to express the reasoning structure. As shown in Figure 2 (a), the complex question Q is decomposed into a tree comprising four simpler sub-questions, $Q^{(1)}$ to $Q^{(4)}$, with (compositional) dependencies among them. The root node represents the original complex question, the leaf nodes represent atomic sub-questions, and the intermediate nodes require compositional (comparative) reasoning to obtain answers. Following Cao et al. (2023); Su et al. (2023), we adopt #i as the placeholder for intermediate questions, enabling us to replace incomplete questions with solved sub-questions as we traverse to that node. Specifically, the decomposed question is represented in QDMR format (Wolfson et al., 2020). Afterward, we tackle the complex questions in a bottom-up fashion, following a post-order traversal sequence. The tree is represented as a post-order traversal node sequence, $Q_{decomp} = \{N^{(1)}, N^{(2)} \dots\}$, with each node containing a question, candidate answers, and the associated probabilities, $N^{(i)} = \{q^{(i)},\ \boldsymbol{a}^{(i)},\ \boldsymbol{p}^{(i)}\}$. It is noteworthy that the model may produce structural incorrect decomposition, which needs post-filtering to ensure the validity of decomposition. We present the formal definitions in Table 7(b).

3.2 Complementary Multi-source Reasoning

To avoid the hallucination caused by lack of knowledge, we use four reasoning strategies combined with answer normalization to fuse information from diverse knowledge sources, as shown in Figure 2(b). The knowledge sources include implicit internal knowledge, explicit internal knowledge, Wikipedia knowledge, and web search knowledge.

$$\boldsymbol{a}_s = \text{LLM}(q, K_s) \tag{1}$$

where $s \in \{closebook, parametric, wiki, serp\}$ is reasoning strategy, K is retrieved knowledge.

Internal Knowledge Reasoning For implicit knowledge reasoning, we prompt the model with chain-of-thought demonstrations to perform closed-book reasoning. There are also studies suggesting that the model's parametric knowledge can serve as a retrieval source (Yu et al., 2023; Zhang et al., 2023b). We first prompt the model to generate parametric knowledge relevant to the question, and then employ it for explicit knowledge reasoning.

External Knowledge Reasoning We utilize Wikipedia and search engines as external retrieval for external knowledge reasoning. Regarding Wikipedia, we employ BM25 to conduct sparse retrieval over the full Wikipedia dumps. For search engines, we call the Google Search API and use the organic results as the retrieval content. After retrieval, we employ few-shot prompts to conduct reasoning on the retrieved documents.

Answer Normalization After completing four sets of independent knowledge reasoning, we merge them through voting to achieve knowledge fusion. Following that, we normalize the answer-frequency pairs to probability distributions for subsequent beam aggregation, as shown in Eq. (2).

$$p_i = \frac{\exp(f_i/\tau)}{\sum_{j=1}^{k} \exp(f_j/\tau)} \tag{2}$$

where f is frequency and τ is temperature.

3.3 Beam Aggregation

In beam aggregation, we conduct reasoning over combinations of candidate answers inherited from sub-questions to expand reasoning breadth and exploration space, as shown in Figure 2(c). We then select reasoning paths by maximizing the marginal probability distribution of predictions.

Beam Combination In intermediate nodes, we need to conduct reasoning based on the answers derived from previous sub-questions, and each sub-question is associated with a set of answers and probabilities, termed candidates (or beams). The intermediate question may depend on multiple sub-questions, so we enumerate combinations among the candidates. Specifically, we compute the Cartesian product among the candidates, as shown in Eq. (3). To prevent combinatorial explosion, we restrict the exploration space with beam size. Afterward, for each combination, we substitute the placeholders in the question with candidate answers and conduct multi-source reasoning. We take a composition node with two sub-questions as an example.

$$\{\langle a_i^{(x)}, a_j^{(y)} \rangle, p_i^{(x)} p_j^{(y)} \mid i, j = 1, 2, \ldots, k\} \quad (3)$$

where $^{(x),(y)}$ denote sub-questions, k is candidate size, a is answer and p is associated probability.

As illustrated in Figure 2(c), $Q^{(4)}$ relies on $Q^{(2)}, Q^{(3)}$. We calculate the Cartesian product of candidates of $Q^{(2)}$ and $Q^{(3)}$, and perform substitution to derive new questions. For example, $Q_{11}^{(4)}$ is obtained by substitute the #1 with $a_1^{(2)}$ and #2 with $a_1^{(3)}$, "*When did **Austin** become the capital of **Texas**?*", with $P(Q_{11}^{(4)}) = p_1^{(2)} p_1^{(3)}$. Subsequently, we conduct multi-source reasoning on all substituted sub-questions.

Probabilistic Answer Aggregation We have explored numerous combinations of sub-questions in beam combination, yielding multiple sets of answers and probabilities. Next, we will aggregate the above answers according to the probabilities to determine the optimal reasoning path. This can be formalized as argmax for maximizing the marginal probabilities of predictions, as described below.

$$P(y) = \sum_{q_i \in Q} P(y|x = q_i) \cdot P(q_i) \quad (4)$$

where Q is the original intermediate question, q_i is a substituted question, $P(y|x = q_i)$ is the answers distribution of q_i, and $P(q_i)$ is the weight of q_i.

After probabilistic answer aggregation, we keep the top-k answers $\boldsymbol{a}$ and their probabilities $\boldsymbol{p}$ as candidates. The aggregated candidates will continue propagating bottom-up, participating in subsequent beam aggregations until reaching the root node. Upon reaching the root node, we regard the answer with the highest probability as the final answer.

4 Experimental Setup

4.1 Datasets

We evaluate BeamAggR on four open-domain multi-hop reasoning datasets. HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), and Bamboogle (Press et al., 2023) consist of two-hop questions, and MuSiQue (Trivedi et al., 2022) contains questions with 2 to 4 hops. For HotpotQA, 2WikiMQA, and MuSiQue, we use the same development and test set provided by IRCoT (Trivedi et al., 2023). The test set and development set are both extracted from the original development set, consisting of 500 and 100 instances, respectively. For Bamboogle, we use all 125 instances as the test set, without a separate development set, and use the same hyperparameters with 2WikiMQA. The evaluation metric is token-level F1.

On all four datasets, we conduct experiments in the open-domain setting, employing the entire Wikipedia dumps and web search for retrieval.

4.2 Implementation Details

In the main experiment, we use GPT-3.5-turbo as the backbone. Since OpenAI has deprecated GPT-3.5 (text-davinci series), we reproduce the baselines with GPT-3.5-turbo for a fair comparison[1]. Please refer to Appendix A.2 for more details.

4.3 Baselines

Standard Prompting (Brown et al., 2020) directly generates the final answer with few-shot demonstrations. We use a 20-shot demonstration.

Chain-of-Thought Prompting (Wei et al., 2022) generates reasoning steps before the final answer. We use a 20-shot demonstration.

One-time Retrieval involves using the original question as the query, concatenating sparse retrieval and Google search results into the prompt to guide the model to perform CoT reasoning.

Self-Ask (Press et al., 2023) adopts an iterative approach to decompose complex questions. It generates sub-questions iteratively based on existing reasoning, and then retrievals and answers them until reaching the final answer.

IRCoT (Trivedi et al., 2023) interleaves retrieval-augmented reasoning and reasoning-augmented retrieval until the retrieval information is sufficient to answer the question.

[1] https://platform.openai.com/docs/deprecations

Methods	HotpotQA			MuSiQue				2WikiMQA					Bamboogle
	Overall	*Bridge*	*Comp.*	**Overall**	*2hop*	*3hop*	*4hop*	**Overall**	*Bridge*	*Infer.*	*Comp.*	*B.C.*	**Overall**
Close-book Reasoning													
SP	38.9	37.5	45.3	15.6	16.4	16.2	12.6	33.9	13.9	23.9	53.3	57.0	27.8
CoT	46.5	44.6	55.5	24.7	30.2	22.5	13.2	42.3	25.7	25.1	58.0	68.5	53.6
Retrieval-augmented Reasoning													
OneR ♢	55.3	52.9	66.5	16.4	22.1	10.6	10.4	42.9	24.3	28.7	75.7	51.2	46.8
Self-Ask ♠	49.4	45.3	68.6	16.2	24.4	8.8	7.5	46.9	31.6	40.5	71.5	52.6	51.9
IRCoT ♠	56.2	53.4	69.6	24.9	31.4	19.2	16.4	56.8	44.2	22.7	89.0	69.4	55.0
FLARE ♠	56.1	54.2	64.4	31.9	40.9	27.1	15.0	60.1	46.2	54.5	81.4	66.3	58.1
ProbTree ♣	60.4	59.2	65.9	32.9	41.2	30.9	14.4	67.9	49.8	**66.4**	81.7	**91.1**	66.6
Ours	**62.9**(↑2.5)	**60.5**	**74.2**	**36.9**(↑4.0)	**43.3**	**36.1**	**20.5**	**71.6**(↑3.7)	**55.1**	64.9	**89.9**	90.4	**74.8**(↑8.2)

Table 1: Experimental results on four open-domain multi-hop reasoning datasets: HotpotQA, MuSiQue, 2WikiMQA and Bamboogle. Best and second results are highlighted by **bold** and underline. The evaluation metric is F1. All experiments are done with *gpt-3.5-turbo-instruct* through in-context learning. All baselines are instantiated with both sparse retriever and search engine. ♢ : One-time retrieval. ♠ : Iterative retrieval. ♣ : Sub-question retrieval.

FLARE (Jiang et al., 2023b) dynamically adjusts the retrieval timing based on the confidence of reasoning and conducts retrieval based on upcoming reasoning sentences.

ProbTree (Cao et al., 2023) parses the question into a tree, employing logprobs-based sub-question aggregation to obtain the final answer.

5 Experimental Results

5.1 Main Results

The experimental results on four multi-hop reasoning datasets are presented in Table 1. We observe that one-time retrieval can improve performance in HotpotQA, 2WikiMQA and Bamboogle. However, when facing the more challenging MuSiQue, one-time retrieval even impairs performance (24.7 → 16.4), which indicates that inaccurate retrieval and knowledge conflicts may exacerbate hallucination.

The iterative retrieval approach has made notable progress compared to one-time retrieval by offering more comprehensive knowledge. Additionally, methods based on sub-question retrieval have more explicit queries, which leads to a further improvement in retrieval accuracy, serving as the best-performing baseline method.

As shown in Table 1, our method demonstrates superiority over all baselines. It outperforms the previous state-of-the-art, ProbTree, on all four datasets: HotpotQA (+2.5), MuSiQue (+4.0), 2WikiMQA (+3.7), and Bamboogle (+8.2). We attribute the improvement to three aspects: (i) Question decomposition leads to more accurate retrieval. Compared to iterative retrieval based on generation content, sub-questions serve as clearer retrieval queries. (ii) Complementary reasoning facilitates collaboration between diverse knowledge sources. Our method dynamically selects and leverages knowledge from multiple sources with fine granularity, thereby mitigating knowledge conflicts and omissions. (iii) Probability-based beam aggregation mitigates cascading errors and optimizes the reasoning trajectories through consistency.

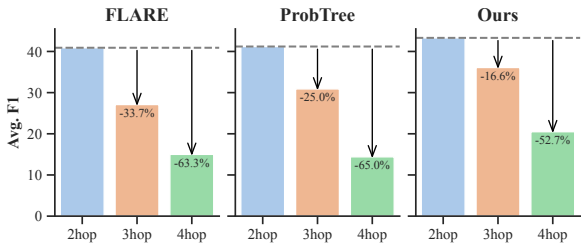

Figure 3: Performance gap with different reasoning steps. We adopt the original split in MuSiQue and report the average f1 score in each subset. As the number of reasoning steps escalates, the model's performance declines. Our method exhibits a slower performance decline as reasoning steps increase, indicating its ability to effectively alleviate cascading errors.

We observe significant improvements, particularly for *Comp* questions. Comparison questions involve interactions among sub-questions. With the aid of beam aggregation, the model can enumerate various combinations during reasoning, thereby encompassing a broader spectrum of possibilities (65.9 → 74.2, 81.7 → 89.9). Additionally, our method is proficient at addressing complex questions (3/4-hop), as shown in Figure 3. Our method exhibits a slower performance decline as reason-

Methods	HotpotQA	MuSiQue	2WikiMQA	Bamboogle
Close-book Reasoning				
SP	23.1	4.3	8.5	11.0
CoT	31.3	16.1	34.2	33.0
Retrieval-augmented Reasoning				
OneR	43.6	11.4	36.3	28.1
FLARE	45.7	20.1	39.6	41.3
ProbTree	50.4	27.0	59.9	61.1
Ours	**55.2**	**32.3**	**63.2**	**74.0**
Ours (GPT-3.5)	62.9	36.9	71.6	74.8

Table 2: Experimental results on *Mistral-7B*. Our method still outperforms all baselines and is compatible with *GPT-3.5-turbo*, suggesting its generalizability.

ing depth increases (8.4% in 3-hop and 12.3% in 4-hop), indicating its ability to precisely aggregate answers and effectively alleviate cascading errors.

5.2 Results on Open-source Model

To prove the generalizability of our method to various models, we also conduct experiments on open-source models. We select Mistral-7B (Jiang et al., 2023a), the state-of-the-art LLM among similar scales. As shown in Table 2, Beam Aggregation significantly outperforms previous SOTA on all four datasets, demonstrating its model-agnostic nature and effectiveness. Furthermore, it is comparable to GPT-3.5-turbo on the Bamboogle dataset.

5.3 Ablation Study

Effect of Multi-source Reasoning We conduct two sets of ablations: removing a single knowledge source and removing a type of knowledge, as shown in Table 3(a). Removing any knowledge source results in a certain performance decrease, suggesting that each type of knowledge contributes to reasoning (a.1-a.4). Furthermore, the declining trends (4.1% internal knowledge and 9.7% external knowledge) indicate that external knowledge makes a greater contribution compared to internal knowledge. Using only internal knowledge for reasoning results in a severe performance drop (a.5). Nevertheless, our method still outperforms chain-of-thought reasoning, which reflects the effectiveness of aggregation reasoning. Completely removing internal knowledge leads to a substantial decline, as shown in (a.6). This suggests that internal knowledge can complement external retrieval, proving the effectiveness of our method in knowledge collaboration.

Effect of Beam Aggregation To validate the effectiveness of beam aggregation, we first employ log-prob-based aggregation, which can only select a single knowledge source. As shown in Table 3(b), this leads to a notable decline, suggesting that coarse-grained polarized aggregation struggles to effectively integrate knowledge, thus underscoring the superiority of our beam aggregation. Probabilistic aggregation enables the distinction of the importance of answers. To investigate its effect, we conduct an ablation where the aggregated answers are treated with equal weights, as shown in (b.2). The performance drops by 5% and 4% on two datasets, suggesting that prioritizing reasoning trajectories can elicit better reasoning. Finally, we investigate the effect of candidate size. Larger candidate sizes result in broader reasoning breadth, but they also increase reasoning overhead. When it is set to 1, greedy aggregation strategy is employed. It is a cost-efficient variant of beam aggregation reasoning. As shown in (b.3), it causes a slight performance decrease, indicating that some erroneous reasoning may be corrected as the reasoning process progresses with the help of a broader scope of reasoning. We conduct a detailed analysis of reasoning performance and overheads in section 6.3.

Setting	2WikiMQA	MuSiQue
Beam Aggregation	**71.6**	**36.9**
(a) Multi-Source Knowledge		
1. w/o closebook	69.2	35.1
2. w/o parameter	68.3	35.6
3. w/o wikipedia	65.8	33.1
4. w/o web search	63.4	32.7
5. internal only	52.0	27.3
6. external only	68.5	33.4
(b) Beam Aggregation		
1. polarization aggregation	65.9	30.9
2. w/o probability	67.8	35.4
3. greedy aggregation	70.1	36.2

Table 3: Ablation results on multi-source knowledge reasoning and beam aggregation. Internal only: conduct internal reasoning only. External only: conduct external reasoning only. W/o probability: do not distinguish the weights of aggregated answers. Polarization aggregation: aggregate answers with logprobs.

6 Analysis

6.1 BeamAggR Facilitates Knowledge Collaboration

To investigate the impact of Beam Aggregation on knowledge collaboration, we conduct a prelimi-

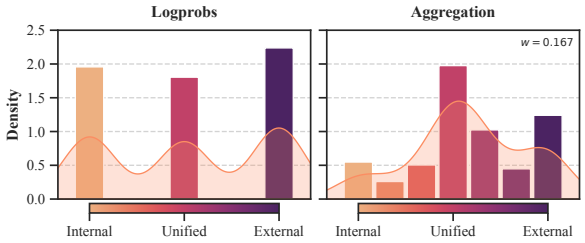

Figure 4: Distribution of knowledge integration in reasoning. **Unified** represents the integration of multi-source knowledge in reasoning, while the two ends represent reliance on single-source knowledge. The bars represent original discrete distributions, and the curve is the kernel density estimate (KDE).

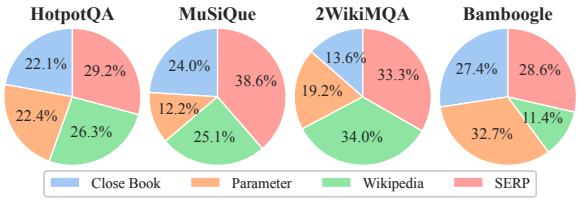

Figure 5: The contribution of each reasoning strategy to the final answer in percentage. HotpotQA is balanced, while MuSiQue and 2WikiMQA leans to external knowledge, and Bamboogle favors internal knowledge.

nary experiment. We track the model's utilization of knowledge from different sources during the reasoning process, as depicted in Figure 4. The polarity aggregation method based on log-probs excessively relies on reasoning on a single source of knowledge. In more than 2/3 of cases, it can only utilize knowledge from a single source. In contrast, our approach facilitates better knowledge collaboration in reasoning. In summary, our beam aggregation can effectively employ knowledge from multiple sources at a finer granularity.

Furthermore, we carry out a more detailed examination of the proportions of each type of knowledge in reasoning. We track the contribution of different knowledge sources to the final answer, as illustrated in Figure 5. Disparities in reasoning contributions across various datasets are observed. 2WikiMQA and MuSiQue tend to favor external reasoning, Bamboogle leans towards internal reasoning, and HotpotQA strikes a balance among different types of knowledge. It is noteworthy that, without manual adjustment of the weight of knowledge, Beam Aggregation can adaptively adjust its proportions during the reasoning process. This indicates that our method can effectively integrate multi-source knowledge.

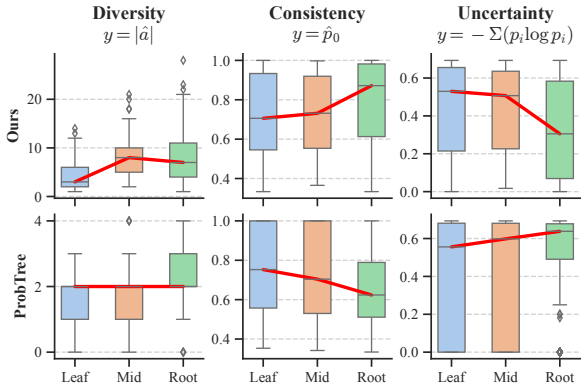

Figure 6: The tendency of candidate answers distribution from leaf node to root. We define three metrics to measure the distribution. Diversity: The number of distinct answers. Consistency: The proportion of the majority answer. Uncertainty: The information entropy of answer distribution.

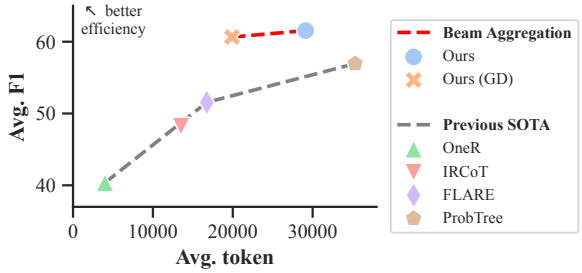

Figure 7: Pareto chart for token consumption and performance (F1). The upper-left quadrant indicates higher efficiency. Results are averaged over four datasets.

6.2 Trends of Answer Aggregation in Bottom-up Reasoning

We conduct a detailed analysis of the trends in answer aggregation throughout the reasoning process. To assess the characteristics of the answer distribution during the bottom-up reasoning process, we define three metrics: diversity, consistency, and uncertainty. The definition and tendency are shown in Figure 6. It can be observed that through the reasoning process, the diversity of answers improves, indicating its exploration of a wider range of possibilities. In contrast, consistency and uncertainty continue to decrease as the reasoning depth increases. This suggests that our method is capable of filtering out reasoning from erroneous during the bottom-up aggregation, dynamically choosing appropriate knowledge sources and reasoning trajectories. Conversely, methods based on log-probs aggregation are affected by inaccurate aggregation and cascading errors, thus unable to achieve these, highlighting the superiority of our method.

6.3 Analysis of Reasoning Cost

Retrieval-augmented generation often involves frequent invocation of LLMs, resulting in significant computational overhead. We compare the performance and overhead of five RAG methods. As illustrated in Figure 7, previous methods exacerbate reasoning overhead while improving performance. In contrast, our method not only surpasses the previous SOTA in performance but also incurs lower overhead. Moreover, our method can further reduce reasoning overhead through greedy aggregation. In summary, Beam Aggregation is Pareto efficient in balancing performance and reasoning overhead. Detail statistics can be found in Table 5.

7 Conclusion

This paper introduces BeamAggR for knowledge-intensive multi-hop reasoning. BeamAggR utilizes a divide-and-conquer strategy, breaking down complex questions into a tree structure and conducting reasoning in a bottom-up fashion. At each step of reasoning, BeamAggR builds upon previous candidates to identify the most likely reasoning path. Furthermore, it incorporates multi-source reasoning to enhance knowledge collaboration. Overall, BeamAggR facilitates multi-hop reasoning through precise sub-question retrieval, efficient knowledge collaboration, accurate answer aggregation, and broad exploration of reasoning trajectories. Extensive experiments on four open-domain multi-hop reasoning datasets demonstrate its effectiveness.

Limitations

Our method involves beam combination across multiple candidates and self-consistency in each reasoning step, thereby increasing reasoning overhead. This overhead can be mitigated through greedy aggregation reasoning. The effectiveness of our framework is contingent upon the accurate decomposition of questions, which may pose significant challenges for large language models. Currently, we only use internal knowledge and unstructured external knowledge for reasoning. In future work, structured external knowledge, such as knowledge bases, can be integrated into reasoning to provide more comprehensive knowledge repositories (Li et al., 2024).

Acknowledgements

The research in this article is supported by the National Key Research and Development Project (2021YFF0901602), the National Science Foundation of China (U22B2059, 62276083), and Shenzhen Foundational Research Funding (JCYJ20200109113441941), Major Key Project of PCL (PCL2021A06). Ming Liu and Qianglong Chen are the corresponding authors.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shulin Cao, Jiajie Zhang, Jiaxin Shi, Xin Lv, Zijun Yao, Qi Tian, Lei Hou, and Juanzi Li. 2023. Probabilistic tree-of-thought reasoning for answering knowledge-intensive complex questions. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12541–12560. Association for Computational Linguistics.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *CoRR*, abs/2309.15402.

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. Successive prompting for decomposing complex questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1251–1265. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *CoRR*, abs/2311.05232.

Gautier Izacard, Patrick S. H. Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *CoRR*, abs/2310.06825.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7969–7992. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *CoRR*, abs/2212.14024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *CoRR*, abs/2203.05115.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6097–6109. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jinyoung Park, Ameen Patel, Omar Zia Khan, Hyunwoo J. Kim, and Joo-Kyung Kim. 2023. Graph-guided reasoning for multi-hop question answering in large language models. *CoRR*, abs/2311.09762.

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

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2695–2709. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.

Xin Su, Tiep Le, Steven Bethard, and Phillip Howard. 2023. Semi-structured chain-of-thought: Integrating multiple sources of knowledge for improved language model reasoning. *CoRR*, abs/2311.08505.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

Tomer Wolfson, Mor Geva, Ankit Gupta, Yoav Goldberg, Matt Gardner, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Trans. Assoc. Comput. Linguistics*, 8:183–198.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *CoRR*, abs/2305.10601.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2023a. Beam retrieval: General end-to-end retrieval for multi-hop question answering. *CoRR*, abs/2308.08973.

Zhebin Zhang, Xinyu Zhang, Yuanhang Ren, Saijiang Shi, Meng Han, Yongkang Wu, Ruofei Lai, and Zhao Cao. 2023b. IAG: induction-augmented generation framework for answering reasoning questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 1–14. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023c. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations,*

ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A Appendix

A.1 Task Defination

This paper focuses on open-domain multi-hop question-answering tasks. The task's input is a multi-hop question Q that requires multi-step reasoning to solve, with each step of reasoning necessitating specific knowledge. Given Q as the query, the retriever $\mathcal{R}$ retrieves relative documents $\mathcal{D} = \{d_i\}_{i=1}^{|\mathcal{D}|}$. The retrieved documents $\mathcal{D}$ are then fed as context into the model for knowledge reasoning. $y = \mathrm{LM}(Q, \mathcal{D})$. Our method employs LLMs with few-shot prompts and external retrieval to address the tasks.

A.2 Implementation Details of BeamAggR

Retrieval Setup We use the October 2017 Wikipedia dumps[2] as the retrieval corpus, employing BM25 (Robertson and Zaragoza, 2009) implemented by Elasticseach as the sparse retriever. For the web search, we use the Google SERP provided by Serper[3]. For SERP results, we consider the top-3 organic results as well as the answer box (if available), resulting in 3 or 4 retrieval documents. We use wikipedia[4] package to access raw Wikipedia content, and perform fuzzy matching to extract snippets in the document with fuzzywuzzy[5] package. To ensure experimental fairness, we incorporate Google search results as additional knowledge for the baseline methods.

Hyperparameters Our main experiments are conducted using *gpt-3.5-turbo-instruct* provided by Azure OpenAI 12-01-preview version. The experiments on open-source models are conducted using Mistral-7B (Jiang et al., 2023a). Table 4 shows detailed hyperparameters.

[2] https://hotpotqa.github.io/wiki-readme.html
[3] https://serper.dev/
[4] https://pypi.org/project/wikipedia/
[5] https://pypi.org/project/fuzzywuzzy/

Evaluation We use token-level F1 as evaluation metric. Besides, if entity aliases are available, we calculate the maximum F1 score between the prediction and each ground truth alias.

Prompts For datasets except for Bamboogle, we adopt the question decomposition results provided by Cao et al. (2023). We convert their decomposition into our format and filter out those with invalid decomposition formats, regenerating them as necessary. For open-source models, limited by their capabilities, they conduct reasoning based on the question decomposition provided by GPT-3.5.

Hyperparameters	Values
# Retrieval doc	5
# Retieval serp	3 or 4
# Parametric knowledge	1
# Closebook prompt	24
# Parametric prompt	5
# Wikipedia prompt	3
# SERP prompt	5
# Question Decomp. prompt	24
Beam size	2
Temperature	3
Self-consistency ($\tau = 0.7$)	5

Table 4: Hyperparameters of BeamAggR.

A.3 Details of Dataset

We provide statistics of the datasets, along with examples of each question type, as shown in Table 6.

HotpotQA consists of 2-hop Bridge and Comparison questions. Bridge: Inferring through the bridge entity to complete the 2-hop question. Comparison: Comparing two entities from the same category, including yes/no questions.

MuSiQue composites one-hop questions through bridge and composition into multi-hop complex questions, covering six categories of 2 to 4-hop questions. Bridge: Composite two single-hop questions into a 2-hop question through a bridge entity. Composition: A question is connected to two subquestions via two bridge entities, constituting a 3-hop question. Bridge and Composition: Combining 2-hop bridge questions and 3-hop composition questions in different order yields 4hop composition/bridge questions.

2WikiMQA includes four types of questions: bridge, inference, comparison, and bridge comparison. Bridge and comparison questions are similar to those in HotpotQA. Inference questions are similar to bridge questions, except they use inference relationships instead of bridge entities, such as *grandfather of*. Bridge-comparison: Comparing entities from two bridge sub-questions.

Bamboogle consists of 2-hop bridge questions, which are similar to 2hop (bridge) questions in MuSiQue. All questions in Bamboogle cannot be directly answered through search engines.

A.4 Details of Analysis

Distribution of knowledge integration In order to facilitate a fair comparison between log-probs and probabilistic aggregation in the preliminary study, we limit our statistical analysis to the proportion of internal and external knowledge contained within the top-1 answers in atomic questions. This ensures that the outcome of the statistics is not affected by the types of decomposition (iterative or sub-question retrieval) or different reasoning paths. The experimental results are averaged from 500 entries each from the HotpotQA and Musique datasets. We plot the discrete distributions (Histplot) and the kernel density estimate (KDEplot) in Figure 4. The y-axis is probabilistic density, for discrete distributions, the density is the percentage divided by the width of the bar. In this figure, the bar width w is set to $1/6 \approx 0.167$.

Contribution of each knowledge source Beginning with the final answer (the top-1 answer from the root node), we trace the reasoning path utilized from top to bottom, counting the knowledge sources that participated in the voting for sub-question answers along this path. Finally, we compute the percentage of knowledge sources within the entire path. To ascertain that this combination of knowledge aggregation led to the correct answer, we omit erroneous samples from our analysis.

Tendency of candidate answers distribution In the reasoning process, errors can be accumulated, leading to cascading errors. Through multiple reasoning paths and probabilistic aggregation, Beam Aggregation can gradually reduce the uncertainty. To illustrate this, we classify reasoning hop into three types: leaf, mid, and root. We statistically analyze the candidate answer distribution in each hop and use diversity, consistency, and uncertainty to measure this distribution. Figure 6 shows a boxplot figure of these statistics. We also line the median value of each hop to show the trend more clearly. The analysis is conducted on the MuSiQue dataset, which has a clearer reasoning hop structure.

Methods	HotpotQA		MuSiQue		2WikiMQA		Bamboogle	
	#token	*f1*	*#token*	*f1*	*#token*	*f1*	*#token*	*f1*
One-Time Retrieval								
OneR	4053	55.3	3941	16.4	3892	42.9	4082	46.8
Iterative Retrieval								
IRCoT	13550	56.2	16877	24.9	13273	56.8	10347	55.0
FLARE	17793	56.1	19180	31.9	16651	60.1	13358	58.1
Sub-question Retrieval								
ProbTree	25607	60.4	46431	32.9	39249	67.9	30004	66.6
Ours	23720	62.9	36336	36.9	30178	71.6	26276	74.8
Ours (GA)	17887	62.3	22522	36.2	21703	70.1	17507	74.0

Table 5: Detailed token cost per instance and performance in four datasets. GA: greedy aggregation. (§6.3)

Token consumption We compare the efficiency between one-time retrieval (OneR), iterative retrieval (IRCoT, FLARE), and sub-question retrieval (ProbTree, Ours). We use a Pareto chart to visualize the token consumption of each method, as shown in Figure 7. The token consumption per instance and performance are averaged in four datasets. The upper left indicates better efficiency (less token consumption and higher performance). Detailed token consumption in each dataset is shown in Table 5. We will detail the method for calculating token consumption in the following section.

We measure the computational cost by evaluating the average token consumption per question. Specifically, the cost of each instance includes prompt tokens (demonstrations, question, retrieved documents) and completion tokens (reasoning traces, answer). For retrieval-augmented reasoning, the cost of a single API call is approximately 4000 tokens, whereas for non-retrieval reasoning, the cost is less than 1000 tokens.

A.5 Prompts

We provide manually annotated demonstrations on the 2WikiMQA dataset for reference. Our prompts are derived from IRCoT[6] (Trivedi et al., 2023) and ProbTree[7] (Cao et al., 2023), to which we have made some modifications. The question decomposition is carried out using 24-shot demonstrations (Figure 10). For implicit internal knowledge, closed-book reasoning is conducted using 20-shot chain-of-thought demonstrations (Figure 11). For

[6] https://github.com/StonyBrookNLP/ircot

[7] https://github.com/THU-KEG/ProbTree

explicit internal knowledge, we first have the LLM generate parametric knowledge (Figure 12), followed by knowledge reasoning using 5-shot demonstrations (Figure 13). For external knowledge reasoning based on web search and Wikipedia, we use prompts of 5-shot and 3-shot (Figure 15, 14).

A.6 Formal Definition of Notations

We describe the formal definition of the notations used in the algorithm pseudocode and formulas in this paper, as shown in Table 7.

Question Type	#Examples	Question Example
(a) HotpotQA		
Bridge	412	Which team does the player named 2015 Diamond Head Classic's MVP play for?
Comparison	88	Did LostAlone and Guster have the same number of members?
(b) MuSiQue		
2hop (Bridge)	254	Who succeeded the first President of Namibia?
3hop1 (Bridge)	122	What currency is used where Billy Giles died?
3hop2 (Composition)	32	When was the first establishment that McDonaldization is named after, open in the country Horndean is located?
4hop1 (Bridge)	51	When did Napoleon occupy the city where the mother of the woman who brought Louis XVI style to the court died?
4hop2 (Bridge + Composition)	16	How many Germans live in the colonial holding in Aruba's continent that was governed by Prazeres's country?
4hop3 (Composition + Bridge)	25	When did the people who captured Malakoff come to the region where Philipsburg is located?
(c) 2WikiMQA		
Comparison	119	Who was born first, Albert Einstein or Abraham Lincoln?
Inference	79	Who is the maternal grandfather of Abraham Lincoln?
Bridge	197	Who is the founder of the company that distributed La La Land film?
Bridge-comparison	105	Which movie has the director born first, La La Land or Tenet?
(d) Bamboogle		
2hop (Bridge)	125	When did the last king from Britain's House of Hanover die?

Table 6: Statistics and examples of each question type in HotpotQA, MuSiQue, 2WikiMQA and Bamboogle. It should be noted that some instances in these datasets trigger the content filter of Azure OpenAI API, where the model refuses to respond. And thus we have omitted these filtered samples when calculating metrics.

Symbol	Dim / Contains	Description
(a) Inputs		
Q	-	Complex knowledge-intensive multi-hop question.
$R(\dots)$	-	Multi-source knowledge retriever.
$LLM(\dots)$	-	Large language model prompted with few-shot demonstration. $LLM(\dots, R)$ indicates that the reasoning is augmented by the retriever.
k	-	Hyper-parameter: Candidate size in beam aggregation.
(b) Question Tree		
Q_{decomp}	$\{N^{(1)}, N^{(2)} \dots\}$	Decomposed questions tree, contains tree nodes. The root node N^{root} represents the original question, and the children of a node $\text{sons}(N^{(i)})$ are sub-questions.
$N^{(i)}$	$\{q^{(i)},\ \boldsymbol{a}^{(i)},\ \boldsymbol{p}^{(i)}\}$	Tree node i, contains the sub-question, answers and probabilities.
$q^{(i)}$	$\mathbb{R}$	Sub-question for node i, obtained in decompose step. The question may contain placeholder tokens (e.g. #1) which need to be replaced by the answer in sub-questions (sons) during reasoning.
$\boldsymbol{a}^{(i)},\ \boldsymbol{p}^{(i)}$	$\mathbb{R}^k,\ \mathbb{R}^k$	Top-k candidate answers and corresponding probabilities in node i, obtained in bottom-up reasoning step.
(c) Variables in Reasoning Step		
$\hat{\boldsymbol{a}},\ \hat{\boldsymbol{p}}$	$\mathbb{R}^?,\ \mathbb{R}^?$	Intermediate results for answers and corresponding probabilities, need to be further voted or aggregated.
$\boldsymbol{a}^t,\ \boldsymbol{p}^t$	$\mathbb{R}^?,\ \mathbb{R}^?$	Same as above.
$\boldsymbol{c}',\ p'$	$\mathbb{R}^{\lvert\text{sons}\rvert}, \mathbb{R}$	One combination of sub-questions answer and corresponding probabilities. A combination is one element in the cartesian product of the children's answers $\boldsymbol{c}' \in \bigtimes_{i \in \text{sons}} \boldsymbol{a}^{(i)} = \boldsymbol{a}^{\text{sons}_1} \times \boldsymbol{a}^{\text{sons}_2} \dots$ and p' is the production of children's probabilities for this combination.
(d) Functions in Pseudocode		
PostOrderTraverse(Q)	$[N] \to [N]$	Return a new node sequence in post-order (child0 < child1 < ... < root).
CartProd$(sons)$	$[N] \to [[a],\ p]$	Cartesian product of the answers and the associated probabilities of the children. For instance, if a node has two children x, y, this function will return a set $\{\langle a_i^{(x)}, a_j^{(y)} \rangle, p_i^{(x)} p_j^{(y)} \mid i, j = 1, 2, \dots, k\}$.
MaskFill(q, c)	$q, [a] \to q$	Replaces the placeholder token $(\#i)$ in the sub-question q with candidate answer c_i.
Vote$(\boldsymbol{a})$	$[a] \to [a], [p]$	Deduplicate answers and obtain a probabilities distribution based on their frequency. Returns are arranged in descending order of probabilities.
Aggr$(\boldsymbol{a}, \boldsymbol{p})$	$[a], [p] \to [a], [p]$	Return the unique answers and their normalized probabilities, arranged in descending order of probabilities.

Table 7: The formal definition of notations used in the algorithms and formulas within this paper.

Reasoning Example on **Bamboogle**

Question: The fourth largest city in Germany was originally called what?
Decomposition:
Q1. What is the fourth largest city in Germany?
Q2. What was #1 originally called?

Q1: What is the fourth largest city in Germany?
closebook answer: [Frankfurt, 3], [Cologne, 2]
parametric answer: [Cologne, 5]
document answer: [Regensburg, 3]
serp answer: [Darmstadt, 5]
> aggregated answer: [Cologne, 0.6607], [Darmstadt, 0.3392]

Q2-1: What was [Cologne] originally called?
Q2-2: What was [Darmstadt] originally called?
> aggregated answer: [Colonia Claudia Ara Agrippinensium, 0.5229], [Colonia Agrippina, 0.1378], [Darmundestat, 0.2988], [the Grand Duchy of Hesse, 0.0404]

Q: The fourth largest city in Germany was originally called what?
aggregated answer: [Colonia Claudia Ara Agrippinensium, 0.6363], [Darmundestat, 0.3636]
> final answer: Colonia Claudia Ara Agrippinensium ✓
ground truth: Colonia Claudia Ara Agrippinensium

Figure 8: An example of reasoning on Bamboogle dataset

Question Decomposition Examples

(a) MuSiQue
Question: Who played who sang is she really going out with him in the who influenced Beyonce movie?
Decomposition:
Q1. Which movie was influenced by the Who?
Q2. Who sang 'Is She Really Going Out With Him'?
Q3. Who played #2 in #1?

(b) HotpotQA
Question: What German state was Karl Julius Perleb born in?
Decomposition:
Q1. Where was Karl Julius Perleb born?
Q2. What German state is #1 in?

(c) 2WikiMQA
Question: Which film has the director who was born later, Money On The Street or She-Devils On Wheels?
Decomposition:
Q1. Who is the director of film Money On The Street?
Q2. When was #1 born?
Q3. Who is the director of film She-Devils On Wheels?
Q4. When was #3 born?
Q5. Which film has the director who was born later, Money On The Street or She-Devils On Wheels? (#2, #4)

(d) Bamboogle
Question: Who built the fastest air-breathing manned aircraft?
Decomposition:
Q1. What is the name of the fastest air-breathing manned aircraft?
Q2. Who built #1?

Figure 9: Examples of question decomposition

Demonstrations of **Question Decomposition**

Question: Who is the performer of Live at this studio that employs the person who coined the term theatre of the absurd?
Decompose: "Who is the performer of Live at this studio that employs the person who coined the term theatre of the absurd?": ["Where did the person who coined the term the theatre of the absurd work?", "Who is the performer at the Live at the #1 event?"], "Where did the person who coined the term the theatre of the absurd work?": ["Who coined the term the theatre of the absurd", "Where is #1 worked?"]

Question: Who is the general treasurer of the state where Israel Arnold House is located?
Decompose: "Who is the general treasurer of the state where Israel Arnold House is located?": ["What state is Israel Arnold House located?", "Who is the general treasurer of #1?"]

Question: What weekly publication in the place of death of George Townsend is issued by the employer of the Yale labor historian who advised younger historians?
Decompose: "What weekly publication in the place of death of George Townsend is issued by the employer of the Yale labor historian who advised younger historians?": ["Where the Yale labor historian who advised younger labor historians works?", "Where did George Townsend die?", "What weekly publication in #2 is issued by #1?"], "Where the Yale labor historian who advised younger labor historians works?": ["Which Yale labor historian advised other younger labor historians?", "Where #1 works?"]

Question: When was the SEC championship game between the winner of the most national titles in NCAA football and Georgia?
Decompose: "When was the SEC championship game between the winner of the most national titles in NCAA football and Georgia?": ["Who has the most national titles in NCAA football?", "When was the SEC championship game between #1 and georgia?"]

Question: Who sings Never Say Never with the performer of As Long as You Love Me?
Decompose: "Who sings Never Say Never with the performer of As Long as You Love Me?": ["Who is the performer of As Long as You Love Me?", "Who sings Never Say Never with #1?"]
.....

Figure 10: Demonstrations of question decomposition. (24 shot)

Demonstrations of **Close-book Reasoning**

Question: When did the director of film Hypocrite (Film) die?
Answer: The film Hypocrite was directed by Miguel Morayta. Miguel Morayta died on 19 June 2013. So the answer is **19 June 2013**.

Question: Do director of film Coolie No. 1 (1995 Film) and director of film The Sensational Trial have the same nationality?
Answer: Coolie No. 1 (1995 film) was directed by David Dhawan. The Sensational Trial was directed by Karl Freund. David Dhawan's nationality is India. Karl Freund's nationality is Germany. Thus, they do not have the same nationality. So the answer is **no**.

Question: Are both Kurram Garhi and Trojkrsti located in the same country?
Answer: Kurram Garhi is located in the country of Pakistan. Trojkrsti is located in the country of Republic of Macedonia. Thus, they are not in the same country. So the answer is **no**.

Question: Who was born first out of Martin Hodge and Ivania Martinich?
Answer: Martin Hodge was born on 4 February 1959. Ivania Martinich was born on 25 July 1995. Thus, Martin Hodge was born first. So the answer is **Martin Hodge**.

Question: Which album was released more recently, If I Have to Stand Alone or Answering Machine Music?
Answer: If I Have to Stand Alone was published in the year 1991. Answering Machine Music was released in the year 1999. Thus, of the two, the album to release more recently is Answering Machine Music. So the answer is **Answering Machine Music**.
.....

Figure 11: Demonstrations of close-book reasoning on 2WikiMQA dataset. (20 shot)

Instruction of Parametric Knowledge Generation

Provide the necessary background knowledge to answer the given question.
Question: {}
Knowledge:

Figure 12: Instruction of parametric knowledge generation. (zeroshot)

Demonstrations of Explicit Parametric Reasoning

Given a question and the relevant documents, answer the question and explain why. If you are unsure, answer Unknown.

#1 Document:
Kurram Garhi Kurram Garhi is a town located in the Kurram District of Khyber Pakhtunkhwa, Pakistan. It is situated on the bank of Kurram River and is approximately 12 kilometers away from the city of Parachinar, the district's headquarters. The word "Kurram" is derived from the Sanskrit word "Kramar," which means "a place to live." It is believed that Kurram Garhi was named by the Hindu king, Raja Karanpal, who ruled the area in the 10th century. The town has a rich history and has been a significant strategic location throughout the centuries. It has been a part of various empires and has seen many battles between different rulers. In the 18th century, Kurram Garhi was under the control of the Mughal Empire, and later it became a part of the Durrani Empire.
Question: Which country is Kurram Garhi located in?
Answer: Kurram Garhi is located in the country of Pakistan. So the answer is **Pakistan**.

#1 Document:
Monte Galbiga is a mountain located in the province of Como, Lombardy, Italy. It has an elevation of 1,690 meters (5,545 feet) above sea level. The mountain is also known as the "Balcone d'Italia" (Balcony of Italy) due to its panoramic views of Lake Como and the surrounding mountains. The name "Galbiga" is derived from the Lombard word "galb" which means "height". The mountain is a popular destination for hikers and offers various trails and viewpoints. It is also a popular spot for paragliding and hang gliding. In addition to its natural beauty, Monte Galbiga also has historical significance. During World War II, it was used as a strategic observation point by the Italian army. Remains of fortifications and bunkers can still be found on the mountain.
Question: In which country is the mountain Monte Galbiga located?
Answer: The mountain Monte Galbiga is located in Italy. So the answer is **Italy**.

.....

Figure 13: Demonstrations of explicit parametric reasoning on 2WikiMQA dataset. (5 shot)

Demonstrations of Open-book Reasoning (Wikipedia)

Given a question and the relevant Wikipedia text, answer the question and explain why. If you are unsure, answer Unknown.

#1 Wikipedia Title: Hypocrite (film)
Text: Hypocrite (Spanish: Hipócrita..!) is a 1949 Mexican thriller film directed by Miguel Morayta ...
#2 Wikipedia Title: When the Legends Die
Text: When The Legends Die is a 1963 novel, by Hal Borland, and a DeLuxe Color film released ...
#3 Wikipedia Title: Who Is the Guilty?
Text: Who is the Guilty? (sometimes" Who is to Blame?") is a 1925 Georgian silent film ...
#4 Wikipedia Title: Miguel Morayta
Text: Miguel Morayta(15 August 1907 – 19 June 2013) was a Spanish film director and screenwriter ...
#5 Wikipedia Title: Joselito vagabundo
Text: Joselito vagabundo(" Joselito Vagabond") is a 1966 Mexican film. It stars Sara García and is directed by ...
Question: When did the director of film Hypocrite (Film) die?
Answer: The film Hypocrite was directed by Miguel Morayta. Miguel Morayta died on 19 June 2013. So the answer is **19 June 2013**.

.....

Figure 14: Demonstrations of open-book reasoning (wikipedia) on 2WikiMQA dataset. (3 shot)

Demonstrations of **Open-book Reasoning (SERP)**

Please answer the question based on the snippet from Google search and provide an explanation. If you are unsure, answer Unknown.

#1 Wikipedia Title: Shah dynasty
Snippet: Dravya Shah was the youngest son of Yasho Brahma Shah, Raja (King) of Lamjung and grandson of Kulamandan Shah Khad, Raja (King) of Kaski ...
#2 Wikipedia Title: List of monarchs of Nepal
Snippet: The monarchs of Nepal were members of the Shah dynasty who ruled over the Kingdom of Nepal from 1743 to its dissolution in 2008 ...
#3 Wikipedia Title: Krishna Shah
Snippet:Krishna Shah (10 May 1938 – 13 October 2013) was an Indian-American/Gujarati film and theatre director, screenwriter, playwright, producer, and production/distribution executive ...
Question: Who is the child of Krishna Shah (Nepalese Royal)?
Answer: Krishna Shah was the father of Rudra Shah. So the answer is **Rudra Shah**.

#1 Wikipedia Title: Kurram Garhi Hydropower Plant
Snippet: Kurram Garhi Hydropower Plant (KGHPP) is a small, low-head, run-of-the-river hydroelectric power generation station of 4.0 megawatt generation capacity ...
#2 Wikipedia Title: Kurram District
Snippet: Kurram District is a district in the Kohat Division of the Khyber Pakhtunkhwa province of Pakistan.The name Kurram comes from the river Kwarma ...
#3 Wikipedia Title: Kurrama River
Snippet: The Kurrama River, or Kurram River, originates from the watershed of Spin Ghar region in the Paktia province of Afghanistan and the Kurram District of Pakistan. ...
#4 Answerbox Title: Kurram Garhi
Snippet: Kurram Garhi is a small village located near the city of Bannu, which is the part of Khyber Pakhtunkhwa province of Pakistan.
Question: When did the director of film Hypocrite (Film) die?
Answer: The film Hypocrite was directed by Miguel Morayta. Miguel Morayta died on 19 June 2013. So the answer is **19 June 2013**.

.....

Figure 15: Demonstrations of open-book reasoning (SERP) on 2WikiMQA dataset. (5 shot)