BELLE: A Bi-Level Multi-Agent Reasoning Framework for Multi-Hop Question Answering

Taolin Zhang[1], **Dongyang Li**[2], **Qizhou Chen**[3,4], **Chengyu Wang**[3*], **Xiaofeng He**[4]
[1] School of Computer Science and Information Engineering, Hefei University of Technology
[2] Shanghai University of Electric Power [3] Alibaba Cloud Computing
[4] East China Normal University
tlzhang@hfut.edu.cn, chengyu.wcy@alibaba-inc.com

Abstract

Multi-hop question answering (QA) involves finding multiple relevant passages and performing step-by-step reasoning to answer complex questions. Previous works on multi-hop QA employ specific methods from different modeling perspectives based on large language models (LLMs), regardless of question types. In this paper, we first conduct an in-depth analysis of public multi-hop QA benchmarks, categorizing questions into four types and evaluating five types of cutting-edge methods: Chain-of-Thought (CoT), Single-step, Iterative-step, Sub-step, and Adaptive-step. We find that different types of multi-hop questions exhibit varying degrees of sensitivity to different types of methods. Thus, we propose a Bi-levEL muLti-agEnt reasoning (BELLE) framework to address multi-hop QA by specifically focusing on the correspondence between question types and methods, with each type of method regarded as an "operator" by prompting LLMs differently. The first level of BELLE includes multiple agents that debate to formulate an executable plan of combined "operators" to address the multi-hop QA task comprehensively. During the debate, in addition to the basic roles of affirmative debater, negative debater, and judge, at the second level, we further leverage fast and slow debaters to monitor whether changes in viewpoints are reasonable. Extensive experiments demonstrate that BELLE significantly outperforms strong baselines in various datasets. Additionally, the model consumption of BELLE is higher cost-effectiveness than that of single models in more complex multi-hop QA scenarios.

1 Introduction

Recently, large language models (LLMs) have become the fundamental infrastructure of modern NLP (Blevins et al., 2023; Zhang et al., 2024b,a; Chu et al., 2024a). Furthermore, chain-of-thought (CoT) prompting enhances the reasoning capabilities of LLMs (Wei et al., 2022; Shaikh et al., 2023; Chu et al., 2024b). Yet, the complexity of multi-hop question answering (QA) often surpasses the knowledge boundaries of LLMs, which can lead to factual errors in generated responses, also known as hallucinations (Khalifa et al., 2023; Huang et al., 2024; Chu et al., 2024a; Shi et al., 2024).

In the literature, multi-hop QA approaches with LLMs can be divided into two categories: `(1) Closed-book Reasoning:` This approach utilizes the understanding ability of LLMs for multi-hop questions, obtaining refined answers through probabilistic sampling in LLMs' response generation. CoT (Wei et al., 2022) prompts LLMs step by step for multi-hop questions to generate the reasoning process. Considering complex multi-hop reasoning paths, several works (Dua et al., 2022; Zhou et al., 2023) decompose them into sub-step questions and solve them progressively, while others (Yao et al., 2023; Chu et al., 2024a; Menon et al., 2024) model reasoning procedures as BFS or DFS search on probabilistic reasoning trees. As reported in (Borgeaud et al., 2022), the knowledge learned by LLMs is often insufficient to answer complex questions, which require external data support. `(2) Retrieval-augmented Reasoning:` Early work utilizes single-step retrieval, but often struggles to gather all necessary knowledge to answer multi-hop questions, resulting in knowledge omissions (Lazaridou et al., 2022; Borgeaud et al., 2022; Izacard et al., 2023). Several approaches leverage iterative-step retrievals by concatenating output from previous rounds with sub-step questions (Press et al., 2023; Shao et al., 2023; Jiang et al., 2024). As shown in Fig. 1, no matter what multi-hop question is given, retrieval methods directly recall external knowledge and answer the question with integrated inputs. Although the adaptive-step method leverages classifiers for dif-

* Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4184–4202
July 27 - August 1, 2025

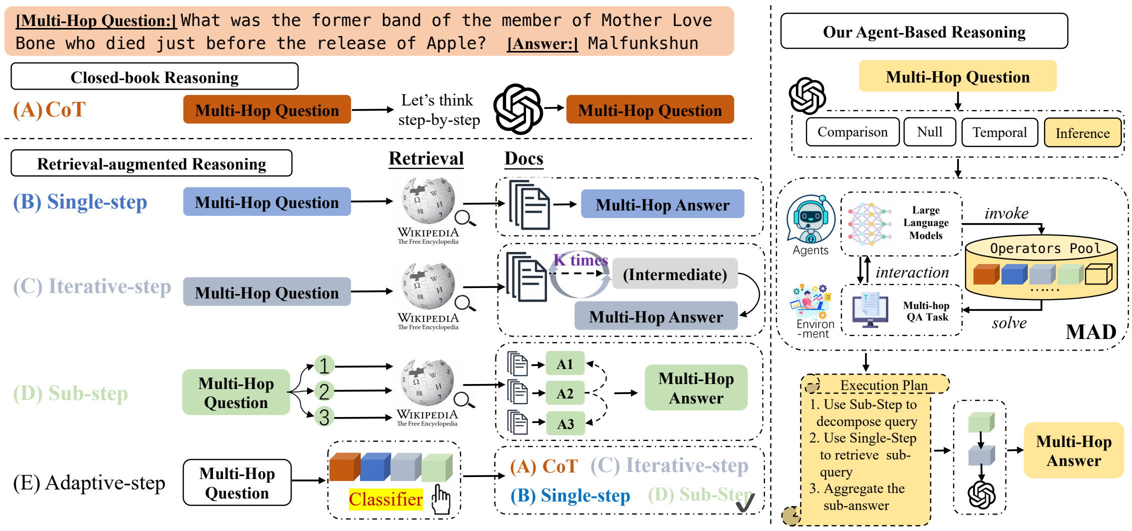

Figure 1: Comparison between our approach and existing methods for multi-hop QA. (1) `Closed-book reasoning` does not consider the requirement for external knowledge. (2) `Retrieval-augmented reasoning` leverages an end-to-end fixed solution to solve all multi-hop questions. (3) `Our agent-based reasoning` framework provides an execution plan to dynamically combine appropriate multi-hop operators with respect to multi-hop question types.

ferent questions (Jeong et al., 2024), they still use a fixed approach, regardless of question types. This also incurs an additional computational burden for relatively simple questions, which limits their usage in applications that require high inference speed (Mavi et al., 2024; Zhuang et al., 2024).

To overcome the above problems, our research focuses on the following question: *How can we dynamically combine various operators based on question types to improve the performance of multi-hop QA, while reducing the computational overhead?* Building on this motivation, we present a novel bi-level multi-agent system named BELLE, which creates and executes a plan of operators[1] for answering multi-hop questions where the plan is represented by the output summary of our multi-agent debate (MAD) system.

Specifically, we first conduct an analysis on whether different types of multi-hop questions are better answered by different operators. Following (Tang and Yang, 2024), the four question types are `Inference`, `Comparison`, `Temporal`, and `Null`. From Fig. 2, the `Temporal` and `Comparison` types are relatively simple, requiring only breaking down the question into sub-questions and using a single-step retrieval method to recall the fact. However, for the `Inference` type, due to their complexity, it is necessary to break down the question and use iterative-step retrieval to obtain more external knowledge. For other questions, we can directly use the LLM's internal knowledge to answer them.

Based on the analysis, the multi-agent pipeline consists of three modules. (i) `Question Type Classification`: We provide in-context examples formatted as new QA pairs, and inputs to LLMs are classified into the four question types. (ii) `Bi-Level Multi-agent Debate`: In addition to the basic roles in multi-agent systems (Li et al., 2024; Liang et al., 2024), we propose a bi-level architecture including a slow-debater and a fast-debater to fully utilize both the historical discussion and the current state of opposing sides to determine which multi-hop QA operators to invoke (Christakopoulou et al., 2024). Our objective is to maximize the use of information already discussed for planning operators while also preventing bias in the agent's viewpoint (Taubenfeld et al., 2024; Borah and Mihalcea, 2024). (iii) `Multi-hop QA Executor`: When the system provides a plan to invoke specific operators, we use LLMs again to generate responses according to the plan. Finally, we concatenate the results of each step to obtain sub-answers and trace back to the root node to achieve the final answer for the multi-hop question.

We evaluate BELLE on four multi-hop QA datasets, including MultiHop-RAG (Tang and Yang, 2024), 2WikiMultiHopQA (Ho et al., 2020), HotPotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022). The experiments are conducted using GPT-3.5-turbo (Brown et al.,

[1]We view specific solutions (e.g., CoT (Wei et al., 2022)) as "operators" from the perspective of prompting LLMs.

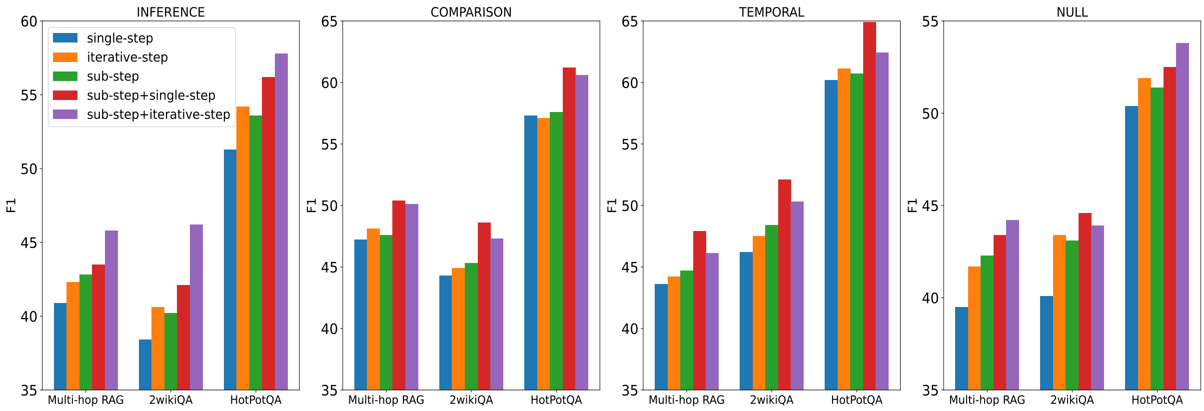

Figure 2: Comparison of single and combined operators in different multi-hop question types. The red and purple bars represent the combined operators of sub-step + single-step and sub-step + iterative-step, respectively.

2020) and Qwen2.5-7B (Qwen Team, 2024). The results show that our method significantly outperforms baselines. An analysis on more difficult multi-hop questions reveals the computational cost superiority of our dynamic operators combination.

2 Related Works

Multi-Hop Question Answering. Multi-hop QA is more complex than simple QA because it involves not just retrieving information, but also effectively combining related facts. Facts can be sourced from a knowledge graph (Lin et al., 2018; Cheng et al., 2023; Zhong et al., 2023), tables (Lu et al., 2016), free-form text (Yang et al., 2018; Welbl et al., 2018), or a heterogeneous combination of these sources (Chen et al., 2020; Mavi et al., 2022; Lei et al., 2023). With the development of LLMs, prompt-based methods combined with an optional retrieval module have become a popular approach for handling multi-hop QA (Press et al., 2023; Zhong et al., 2023; Zhuang et al., 2024; Chu et al., 2024a). Recently, the agent-based methods for multi-hop QA are also proposed (Shen et al., 2024; Wu et al., 2025). While all previous works focus on a specific multi-hop QA method, our approach targets a dynamic, flexible pipeline from a more fine-grained question type perspective.

Multi-Agent Debate of LLMs. Current approaches to multi-agent debate (MAD) can generally be divided into two main categories: (1) Those that adjust the model prompts and responses during the debate (Liang et al., 2024; Khan et al., 2024; Rasal, 2024; Feng et al., 2024; Yang et al., 2024). These MAD methods generate specific opinions in response to particular situations while solving a task. (2) Those that alter the structure of the debate process (Li et al., 2023; Liu et al., 2023; Chang, 2024; Hong et al., 2024). Importantly, both categories use off-the-shelf LLMs (e.g., API) and work by modifying either the inputs or outputs of these models. However, previous work did not take into account the comprehensive utilization of historical and current information in multi-agent collaboration, resulting in a waste of information.

3 Analysis of Multi-Hop Question Types

In this section, we analyze the sensitivity of different types of multi-hop questions involving single and combined operators as described previously.

We leverage four multi-hop QA datasets, namely MultiHop-RAG (Tang and Yang, 2024), 2WikiMultiHopQA (Ho et al., 2020), HotPotQA (Yang et al., 2018), and MuSiQue (Trivedi et al., 2022) as the data sources.[2] The other three datasets, except for MultiHop-RAG, do not include question type labels. Hence, we use GPT-4 (OpenAI, 2023) to annotate half of the datasets and perform cross-validation. The prompt for label annotation is shown in Appendix C.1. Considering potential annotation errors by LLMs, we refine the prompts and manually check the responses to select suitable prompts. During the manual verification of data labeling, two individuals independently test 100 samples of each type. A prompt is adopted only if both individuals agree that the labeling is consistent with the actual question type, achieving an accuracy of 95%. To maintain consistency in the label space,[3] we set it to be the same as

[2]The complete results and the analysis of the question type annotation process are shown in Appendix B.1.

[3]Due to the extensibility of our BELLE, there will be more fine-grained question type classification rules that can

that of MultiHop-RAG, which includes four types: `Inference`, `Comparison`, `Temporal`, and `Null`.

As for the combined operators, we have selected two representative methods: sub-step+single-step and sub-step+iterative-step. From Fig. 2, we can draw two conclusions:

1. Combined operators are superior to single operators in multi-hop QA tasks. Across the four question types, the method of combined operators consistently outperforms single operators. On average, the performance of combined operators is 3% higher than that of single operators across different question types and datasets.

2. Different combinations of operators have varying degrees of sensitivity to question types. For the `Inference` type, due to the increase in logical reasoning steps, it is necessary to recall more external knowledge (Mavi et al., 2022, 2024). In this case, decomposing the complex question and combining it with a multi-round retrieval scheme is more suitable for this multi-hop question type. For `Comparison` and `Temporal` types, we typically only need to identify the important subjects (e.g., entity or timestamp) for these question types and retrieve relevant content. Hence, the method based on sub-questions combined with single-step retrieval can address them effectively.

Therefore, using different combinations of operators is better for solving different types of multi-hop questions than using a specific operator alone.

4 Methodology

In this section, we provide a detailed description of BELLE, with the bi-level MAD system shown in Fig. 3. Our framework includes the following three modules: (i) `Question Type Classifier`: Multi-hop questions are classified into the corresponding question types as discussed in Sect. 3. (ii) `Bi-Level Multi-agent Debater`: In addition to conventional MAD systems, slow and fast debaters are proposed to aid opposing sides in invoking the operators with historical discussion. (iii) `Multi-hop QA Executor`: It executes the planning of operators to answer multi-hop questions.

4.1 Question Type Classifier

Compared to previous works (Cheng et al., 2023; Chu et al., 2024a; Zhuang et al., 2024) that use a specific method to coarsely solve multi-hop QA tasks, we find that the complex multi-hop reasoning task requires dynamic combinations of operators based on question types. Hence, BELLE first considers fine-grained classification of multi-hop questions as input for subsequent modules.

Specifically, this module can be directly formalized as a text classification task, denoted as $\mathcal{A}_t = \mathcal{M}_t(q)$. Here, q denotes a multi-hop question, and $\mathcal{M}_t$ is the LLM for question type classification. As for $\mathcal{A}_t$, we use the four question types analyzed in Sect. 3 as the output label space. We concatenate several QA examples as demonstrations to perform the ICL mechanism,[4] ensuring output of the correct question type labels and preventing the instruction degradation phenomenon (Brown et al., 2020; He et al., 2024a). The detailed format of templates is described in Appendix C.1.

4.2 Bi-Level Multi-agent Debate

Recently, many MAD systems have addressed specific scenarios with a setting consisting of an affirmative debater, a negative debater, and a judge (He et al., 2023; Li et al., 2024). These agents can only make a decision for task solutions based on the current state, while the historical discussion contents are not fully utilized. Consequently, the task viewpoints of both debaters may be uncontrollably altered due to the influence of one another (Taubenfeld et al., 2024; Borah and Mihalcea, 2024).

Inspired by Christakopoulou et al. (2024), we introduce a bi-level MAD system, which employs two additional memory agents named slow-debater and fast-debater to integrate the relationship between historical discussions and current viewpoints. Next, we provide a detailed description of our system, where two representative opposing debaters, two memory debaters, and a judge are involved in a debate to resolve a multi-hop question. Our framework is composed of four components divided into two levels, elaborated as follows.

4.2.1 The First Level of Debate

Meta Prompts and Operators. Considering that agents initially might not understand the task, we leverage meta prompts to introduce the question type $\mathcal{A}_t$, the number of debaters, the round limit, and other requirements, as shown in Appendix C.2. We create an atmosphere for debaters to engage in a "tit for tat" debate (see indicated contents).

For the operators pool, each element will be invoked by the following bi-level MAD system, se-

be directly used by modifying the `Meta Prompt` in the future.

[4] Other mechanisms are also analyzed in Appendix B.3.

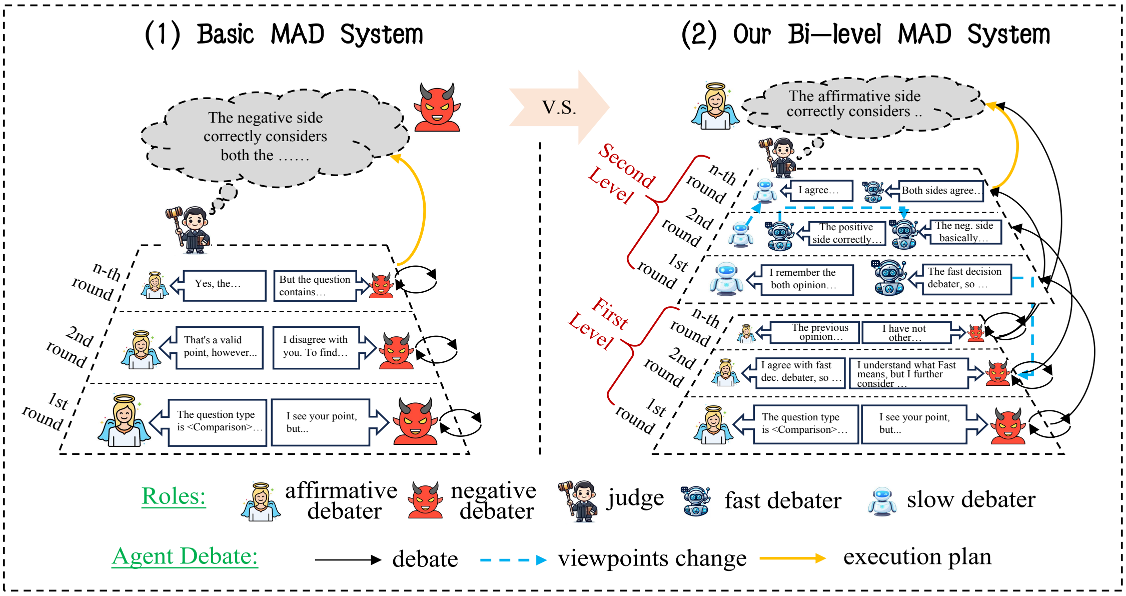

Figure 3: Model overview of BELLE. The left part is the existing MAD system containing three basic roles (i.e., an affirmative side, a negative side and a judge). The right part is the details of our bi-level MAD system including first-level and second-level debaters.

lecting from two paradigms described in Fig. 1. We choose CoT (Wei et al., 2022), single-step (Izacard et al., 2023), iterative-step (Trivedi et al., 2023), sub-step question (Press et al., 2023), and adaptive-step (Jeong et al., 2024) as representative operators.

Opposing Debaters. There are two debaters that play the roles of the affirmative and the negative, respectively. In each debate round, the debaters take turns presenting arguments based on their own previous debate history. For the affirmative debater, denote the debate history from all `t-1` rounds as H_{ad}^{t-1}. The result of the t-th round discussion for the affirmative debater is defined as follows:

$$f_{ad}^{t} = \mathcal{M}(H_{ad}^{t-1}, f_{fast}^{t-1}, f_{slow}^{t-1}) \tag{1}$$

where $\mathcal{M}$ is the same LLM as $\mathcal{M}_t$. f_{fast}^{t-1} and f_{slow}^{t-1} represent the discussion results of the fast and slow debaters in the `(t-1)`-th round, respectively. The definitions for the debate history and discussion results of the negative debater, denoted as f_{nd}^{t}, are similarly defined.

4.2.2 The Second Level of Debate

The first level of discussion focuses on each side's positions without evaluating the rationality of operator selection. Therefore, in our proposed bi-level debate mechanism, the second level comprehensively evaluates the operator selection in the current t-th round (fast debater) and summarizes historical debates (slow debater).

Fast Debater. In the discussion process of the fast debater, the main goal is to assess whether the operators selected in the current discussion between both sides are reasonable. This involves the participation of three roles: the affirmative and negative sides in the t-th round, as well as the previous discussion results of the fast debater. We denote the debate history of the fast debater from all previous `t-1` rounds as H_{fast}^{t-1}. Hence, the current t-th debate result of the fast debater is as follows:

$$f_{fast}^{t} = \mathcal{M}(f_{ad}^{t}, f_{nd}^{t}, H_{fast}^{t-1}) \tag{2}$$

Note that the fast debater only considers the situation in the current t-th debate, making it susceptible to the viewpoints of both sides, as illustrated by the blue dashed line in Fig. 3.

Slow Debater. Compared to the fast debater, the slow debater integrates all historical information to judge the rationality of operator selection. The more important goal is to prevent debaters from losing confidence in correct viewpoints, which may lead to oscillation (Zhang et al., 2023). The slow debater process involves the affirmative, negative, fast, and historical roles of the slow debater. Similar to the fast debater, the debate history from all previous `t-1` rounds is H_{slow}^{t-1}. The current viewpoint of the slow debater is as follows:

$$f_{slow}^{t} = \mathcal{M}(f_{ad}^{t}, f_{nd}^{t}, f_{fast}^{t}, H_{slow}^{t-1}) \tag{3}$$

Judge. Finally, we design a judge J to oversee the debate process, providing an execution plan of combined operators. The judge operates in two modes: (a) Hard Mode, where judge J decides if a correct combination of operators can be determined after all debaters present their viewpoints. If possible, the debate concludes; otherwise, it continues. (b) Soft Mode, where judge J extracts useful operator suggestions based on the slow debater's history, H^t_{slow}, since no correct solution is found within the debate's round limit. The judge's template is in Appendix C.2, which produces a summarized plan for invoking operators step by step.

4.3 Multi-hop QA Executor

Through the discussion of our bi-level MAD system, we have obtained the specific plan for solving the multi-hop question. Then, we progressively invoke the corresponding multi-hop operators to obtain the final answer. To ensure consistency in the LLM's understanding, we use the same LLM $\mathcal{M}$ to execute the sub-steps of the operator planning process. An example is shown in Appendix C.3.

5 Experiments

Due to space limitation, we describe datasets, baselines and implementation details in Appendix A.

5.1 Experimental Results

5.1.1 Results of Multi-hop QA Tasks

Main Results. Table 1 shows the general performance of BELLE across the four multi-hop QA datasets. We observe that: (1) Generally, due to the requirement for external knowledge in complex multi-hop questions (Mavi et al., 2024; Minaee et al., 2024), retrieval-augmented reasoning methods show more significant improvement compared to closed-book methods. However, a comparable improvement can still be achieved by reasoning step by step using CoT (Wei et al., 2022). (2) Among retrieval-augmented methods, the simple retrieval method does not significantly improve the effectiveness of multi-hop QA. Other methods with additional enhancement operations, such as ProbTree (Cao et al., 2023) and BeamAggR (Chu et al., 2024a), achieve significant improvements. (3) Since the agent-based methods are designed with special modules, the collaborative semantic understanding of multi-hop questions by these methods has not been fully utilized compared to our unified operators' framework. Therefore, an agent-based approach is still insufficient in solving multi-hop QA tasks. (4) BELLE consistently achieves the best results. Through careful debate for choosing combined operators, our model achieves the greatest improvement on the extremely difficult MuSiQue dataset under 2, 3, and 4 hops settings.

Results of Question Types. We present the results for the four types in Fig. 4, using two strong baselines: CoT (Wei et al., 2022) and BeamAggR (Chu et al., 2024a). Specifically, we observe that (1) The retrieval-based method that introduces external knowledge performs much better on various types of multi-hop questions than simply using an LLM to answer. Meanwhile, our combined operators method also consistently performs better than the strongest multi-source knowledge-enhanced method. (2) Our model shows no significant improvement for `Comparison` and `Temporal` due to the simple answer patterns. For `Comparison` questions, the model only needs to decompose the question into two parts that require comparison, and the answers are concise (e.g., "Yes" and "True"). For `Temporal` questions, it is usually necessary to find the important timestamp for answering. However, for the remaining two types, `Inference` and `Null`, which are much more difficult, our BELLE model achieves significant improvements. `Inference` type questions require reasoning across multiple documents.[5] Due to the lack of a unified pattern for `Null` questions, it requires invoking different operators for adaptive combination.

5.1.2 Ablation Study

In Table 2, we select three crucial components for our ablation study. Specifically, when we remove the question type classifier, `<Question Type>` will not be inserted into the meta prompts for the subsequent bi-level MAD system. The first-level debate is replaced with an LLM without a debating environment, and the viewpoints are directly optimized by the second-level debate. When we remove the second-level debate, the overall system degrades to a basic MAD system associated with question types. The results show that removing the second-level debaters has the greatest impact regardless of the LLMs used. It indicates that this level leverages the history of debating to make reasonable operator selection opinions, compared to the basic first-level system alone. We also find that introducing question types as prior knowledge into the MAD system is crucial for the selection of combined operators.

[5]For example, there are two gold paragraphs and eight distractors in HotpotQA (Yang et al., 2018) for each question.

Dataset→	Multi-hop RAG			HotpotQA			2WikiQA			MuSiQue		
Models↓	EM	F1	Acc	EM	F1	Acc	EM	F1	Acc	2hop	3hop	4hop
Closed-book Reasoning												
SP	39.4	47.5	44.3	32.1	38.9	37.4	27.8	33.9	31.6	16.4	16.2	12.6
CoT	43.6	50.5	49.7	40.5	46.5	47.3	36.2	42.3	43.7	30.2	22.5	13.2
Retrieval-augmented Reasoning												
Single-step	47.2	52.3	51.3	48.7	55.3	54.6	38.1	42.9	41.3	22.1	10.6	10.4
Self-Ask	49.8	54.6	52.6	44.5	49.4	50.2	40.5	46.9	48.5	24.4	8.8	7.5
IRCoT	55.1	59.2	58.4	51.2	56.2	55.4	50.7	56.8	52.3	31.4	19.2	16.4
FLARE	54.9	58.7	59.2	50.8	56.1	58.3	58.2	60.1	63.7	40.9	27.1	15.0
ProbTree	56.5	62.5	60.1	56.3	60.4	60.6	64.3	67.9	65.4	41.2	30.9	14.4
EffiRAG	49.2	55.3	54.7	52.9	57.9	55.4	47.7	51.6	53.8	32.7	23.6	12.5
BeamAggR	61.9	<u>67.2</u>	66.8	55.6	<u>62.9</u>	59.2	66.1	<u>71.6</u>	69.2	<u>45.9</u>	<u>36.8</u>	<u>21.6</u>
Agent-based Reasoning												
LONGA.	53.6	56.8	57.4	52.4	59.3	58.1	60.1	65.6	62.8	40.5	25.8	16.4
GEAR	50.7	52.5	51.9	50.4	54.6	54.8	47.4	52.3	51.6	35.1	20.9	15.3
RopMura	52.6	53.7	58.2	49.2	53.1	55.7	58.8	63.2	64.0	41.1	24.6	16.2
BELLE	**64.7**	**70.4** (↑ 3.2)	**68.5**	**59.2**	**66.5** (↑ 3.6)	**63.7**	**69.7**	**75.7** (↑ 4.1)	**72.8**	**50.5** (↑ 4.6)	**42.1** (↑ 5.3)	**29.2** (↑ 7.6)

Table 1: The general results of BELLE. The best and second results are highlighted by **bold** and <u>underline</u>. We show the F1 for 2,3,4-hops of MusiQue. T-tests show the improvements are statistically significant with $p < 0.05$ (%).

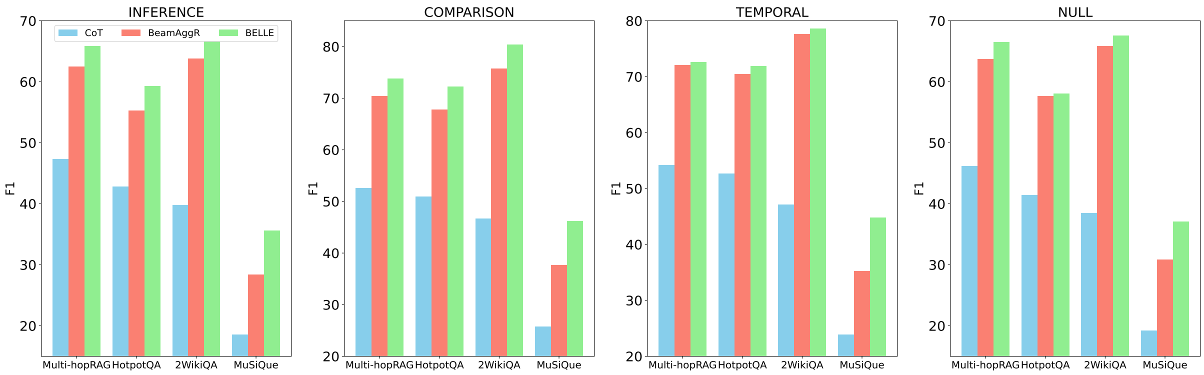

Figure 4: Results of different question types in terms of F1 (%).

In the ablation experiment involving each debater, we further explore the influence of specific debaters. For affirmative and negative debaters, since removing a debater would disrupt the "tit for tat" atmosphere, we maintain the number of agents unchanged by using corresponding prompts. When removing the fast debater, the modeling methods of the other debaters are also synchronously removed. To remove the slow debater, we use the last round result of the fast debater as the summary result. We observe the following: (1) Compared to designs that completely remove the first level, using several agents of the same type at the first level to obtain operator plans is beneficial for multi-hop QA tasks. (2) Removing either the fast or slow agent adversely affects task performance to some degree, with the removal of the slow summarizer having a more significant impact.

5.2 Detailed Analysis

Due to space constraints, we present other detailed statistical results of our bi-level MAD system in Appendix B.5.

5.2.1 Changes in Operator Selection

From Fig. 5, we investigate the impact of the debating contents between the first-level and second-level debaters using HotpotQA questions with the `Inference` type. Specifically, for the four important debaters in two levels, there are two situations to be considered: (1) In the same round of debating, the impact of the first-level (i.e., affirmative and negative debaters) on the second layer (i.e., slow and fast debaters) and (2) In different rounds of de-

Model ↓ Dataset →	D1	D2	D3	D4	Avg.
Qwen2.5-7B					
BELLE	64.1	59.4	68.5	32.8	56.2
BeamAggR	55.8	51.8	62.4	23.2	48.3
w/o Type Classifier	59.6	54.1	63.5	25.9	50.8
w/o First Level Debate	61.2	55.4	64.6	28.9	52.5
w/o Second Level Debate	58.8	53.5	62.1	25.4	50.0
GPT-3.5-turbo					
BELLE	70.4	66.5	75.7	40.6	63.3
BeamAggR	67.2	62.9	71.6	34.8	59.1
w/o Type Classifier	67.9	63.4	73.2	37.6	60.5
w/o First Level Debate	68.2	63.7	73.5	38.1	60.9
w/o Second Level Debate	66.8	62.8	72.3	36.5	59.6
w/o affir.&neg. Debater	68.4	64.1	73.9	38.5	61.2
w/o Fast Debater	67.3	63.2	72.9	37.4	60.2
w/o Slow Debater	67.0	63.1	72.7	36.9	59.9

Table 2: Ablation study of BELLE in terms of F1 (%). Due to space limitation, we use the abbreviations "D1", "D2", "D3", and "D4" to represent Multi-hop RAG, HotpotQA, 2WikiQA, and MuSiQue, respectively.

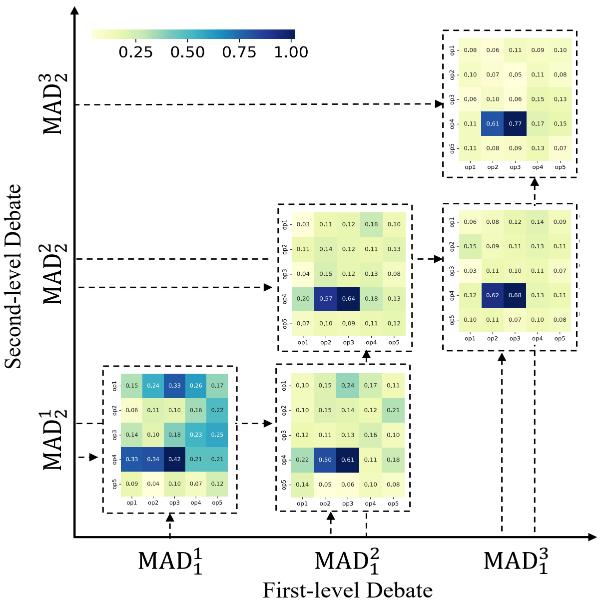

Figure 5: Changes in the selection of combined operators. MAD^j_i denotes the debate stage at i-level and j-th debate round. (Best viewed in color.)

bating, the impact of the previous second-level on the current first-layer debating. Hence, we define the following formula to quantitatively measure the attitude change of the bi-level system:

$$F_{f^t \to s^t} = \alpha(F^t_{ad} + F^t_{nd}) + (1-\alpha)(F^t_{fast} + F^t_{slow})$$

and

$$F_{s^{t-1} \to f^t} = \beta(F^t_{ad} + F^t_{nd}) + (1-\beta)(F^{t-1}_{fast} + F^{t-1}_{slow})$$

where $F_{f^t \to s^t}$ denotes the score for situation (1) and $F_{s^{t-1} \to f^t}$ for situation (2). Each score is a $\mathbb{R}^{5\times5}$ matrix, representing the combined score between 5 operators. F^t_{ad}, F^t_{nd}, F^t_{fast}, and F^t_{slow} represent the t-th round score of the four debaters, respectively. Considering that the content discussed by the first-layer debaters in situation (1) provides information for subsequent discussion, its importance is higher. Thus, we have assigned a value of 0.8 to α and 0.8 to β. The specific score for each debater (e.g., F^t_{ad}) is based on the viewpoint similarity between the two operators. We use GPT-4 (OpenAI, 2023) to score the output content of debaters and the template content composed of two operators.[6] As shown in Fig. 5, we observe that: (1) The bi-level MAD system becomes increasingly focused on which combined operators to use. The scores in the subgraph may fluctuate slightly, but the scoring trend of the combined operators is stable. (2) In our bi-level MAD system, the number of debate rounds is relatively small, reducing the cost of computational resources. It typically requires only 2 rounds to determine operators.

Model ↓ Dataset →	D1	D2	D3	D4	Avg.
Agent-based Methods					
BELLE	18,324	19,520	21,402	23,723	20,742
LONGA.	38,943	74,216	44,283	36,529	48,493
GEAR	32,077	58,541	41,976	35,128	41,931
RopMura	32,885	113,183	46,821	34,547	56,859
Debate Levels					
L0	21,376	26,801	27,542	26,634	25,588
L1	20,988	24,572	23,894	27,149	24,151
L2	18,324	19,520	21,402	23,723	20,742
L3	23,729	25,863	31,154	27,269	27,004
Num. of Debaters					
$\mathbf{N_{f2} \to N_{s2}}$	18,324	19,520	21,402	23,723	20,742
$N_{f3} \to N_{s3}$	26,465	32,841	28,072	35,917	30,824
$N_{f4} \to N_{s4}$	32,053	38,716	34,579	41,839	36,797
$N_{f5} \to N_{s5}$	39,236	45,170	42,585	47,736	43,682

Table 3: Consumption of prompt token quantity under different agent settings. $N_{fi} \to N_{sj}$ refers to i debaters in the first layer and j debaters in the second layer. L_i indicates different settings of the meta prompt.

5.2.2 Analysis of Computational Overhead

Comparison with Retrieval Methods. Retrieval methods often involve frequent invocation of LLMs, resulting in significant computational overhead. We specifically select more challenging examples of prediction errors by plain LLMs to evaluate the models. In Fig. 6, previous methods exacerbate reasoning overhead while improving performance. In contrast, our method not only surpasses the SOTA (e.g., BeamAggR (Chu et al., 2024a)) in

[6] We define the similarity level with a corresponding score between them, such as "very similar" → 0.7.

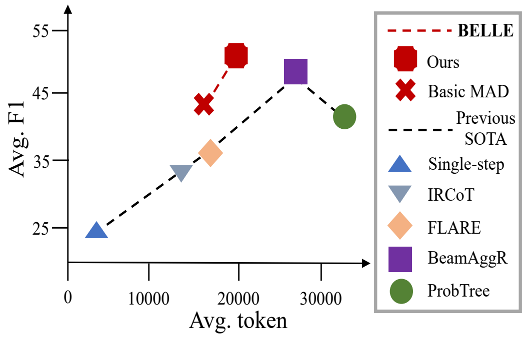

Figure 6: Analysis of the relation between performance and number of retrieval tokens (Best viewed in color).

performance but also reduces reasoning overhead in terms of required tokens. The main advantage of our model is in fully utilizing the current state and historical information, making the execution planning of the combined operators for the multi-hop question more reasonable. Hence, it reduces the number of rounds of combined operator retrieval and lowers the cost of prompt inference length. Detailed statistics are in Appendix B.4.

Comparison with Agent-based Settings. We compare the different debater settings, including agent-based methods, debater levels, and the number of debaters for each level. The debate levels indicate the atmosphere of the debate prompts, as shown in Table 11. As shown in Table 3, (1) due to the establishment of a second-level reflection and judgment mechanism (de Winter et al., 2024; Zeng et al., 2024), our BELLE framework effectively determines the current state of the task to reduce token consumption. (2) Setting different debate levels and adjusting the number of agents for competition can improve BELLE models. By controlling the debate level of token consumption, it is unnecessary to mandate a confrontational discussion atmosphere. A relatively relaxed discussion mechanism, coupled with clear MAD system objectives, yields better results for the BELLE framework while reducing token usage. Meanwhile, excessive focus on increasing the number of agents may not necessarily enhance performance, and token consumption could increase sharply.

6 Conclusion and Future Work

In this paper, we introduce BELLE to effectively address the challenges of multi-hop QA by aligning specific question types with appropriate reasoning methods. By incorporating diverse operators and a bi-level debate mechanism, it achieves significant improvements over existing baselines. In the future, we aim to investigate the integration of BELLE with real-world applications to assess its efficacy in dynamic and evolving environments.

Limitations

While our proposed BELLE framework demonstrates significant improvements over existing methods, several limitations still persist. One major issue is its reliance on multiple agents interacting iteratively, especially during the debate process. Refining the debate rules and strategies could potentially reduce overhead while maintaining or even enhancing performance. Additionally, although BELLE exhibits robustness against known question types, it may struggle with novel or previously unseen question formats. To address this, adaptation to accommodate new question types will be crucial for further improvements in various applications.

References

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *ACL*, pages 6649–6663.

Angana Borah and Rada Mihalcea. 2024. Towards implicit bias detection and mitigation in multi-agent LLM interactions. In *EMNLP*, pages 9306–9326.

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 *ICML*, pages 2206–2240.

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 *NeurIPS*.

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 *EMNLP*, pages 12541–12560.

Edward Y. Chang. 2024. Socrasynth: Multi-llm reasoning with conditional statistics. *CoRR*, abs/2402.06634.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *EMNLP*, pages 1026–1036.

Zhen Cheng, Jianwei Niu, Shasha Mo, and Jia Chen. 2023. Genboost: Generative modeling and boosted learning for multi-hop question answering over incomplete knowledge graphs. In *ICPADS*, pages 1131–1138.

Konstantina Christakopoulou, Shibl Mourad, and Maja J. Mataric. 2024. Agents thinking fast and slow: A talker-reasoner architecture. *CoRR*, abs/2410.08328.

Zheng Chu, Jingchang Chen, Qianglong Chen, Haotian Wang, Kun Zhu, Xiyuan Du, Weijiang Yu, Ming Liu, and Bing Qin. 2024a. Beamaggr: Beam aggregation reasoning over multi-source knowledge for multi-hop question answering. In *ACL*, pages 1229–1248.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2024b. Navigate through enigmatic labyrinth A survey of chain of thought reasoning: Advances, frontiers and future. In *ACL*, pages 1173–1203.

Joost C. F. de Winter, Dimitra Dodou, and Yke Bauke Eisma. 2024. System 2 thinking in openai's o1-preview model: Near-perfect performance on a mathematics exam. *CoRR*, abs/2410.07114.

Dheeru Dua, Shivanshu Gupta, Sameer Singh, and Matt Gardner. 2022. Successive prompting for decomposing complex questions. In *EMNLP*, pages 1251–1265.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. 2024. Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-llm collaboration. In *ACL*, pages 14664–14690.

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. 2024a. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *CoRR*, abs/2410.15553.

Zeyu He, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Shaurya Rohatgi, and Ting-Hao Kenneth Huang. 2024b. If in a crowdsourced data annotation pipeline, a GPT-4. In *CHI*, pages 1040:1–1040:25.

Zhitao He, Pengfei Cao, Yubo Chen, Kang Liu, Ruopeng Li, Mengshu Sun, and Jun Zhao. 2023. LEGO: A multi-agent collaborative framework with role-playing and iterative feedback for causality explanation generation. In *EMNLP*, pages 9142–9163.

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 *COLING*, pages 6609–6625.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for A multi-agent collaborative framework. In *ICLR*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*

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.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *NAACL*, pages 7036–7050.

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 *EMNLP*, pages 7969–7992.

Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2024. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. *CoRR*, abs/2407.13101.

Yimin Jing, Deyi Xiong, and Yan Zhen. 2019. Bipar: A bilingual parallel dataset for multilingual and cross-lingual reading comprehension on novels. In *EMNLP*, pages 2452–2462.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Few-shot reranking for multi-hop QA via language model prompting. In *ACL*, pages 15882–15897.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. In *ICML*.

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.

Fangyu Lei, Xiang Li, Yifan Wei, Shizhu He, Yiming Huang, Jun Zhao, and Kang Liu. 2023. S3HQA: A three-stage approach for multi-hop text-table hybrid question answering. In *EMNLP*, pages 1731–1740.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: communicative agents for "mind" exploration of large scale language model society. *CoRR*, abs/2303.17760.

Renhao Li, Minghuan Tan, Derek F. Wong, and Min Yang. 2024. Coevol: Constructing better responses for instruction finetuning through multi-agent cooperation. In *EMNLP*, pages 4703–4721.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, pages 17889–17904.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2018. Multi-hop knowledge graph reasoning with reward shaping. In *EMNLP*, pages 3243–3253.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. 2023. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *CoRR*, abs/2310.02170.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Zhengdong Lu, Hang Li, and Ben Kao. 2016. Neural enquirer: learning to query tables in natural language. *IEEE Data Eng. Bull.*, 39(3):63–73.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *CoRR*, abs/2204.09140.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. Multi-hop question answering. *Found. Trends Inf. Retr.*, 17(5):457–586.

Sachit Menon, Richard S. Zemel, and Carl Vondrick. 2024. Whiteboard-of-thought: Thinking step-by-step across modalities. In *EMNLP*, pages 20016–20031.

Shervin Minaee, Tomás Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *CoRR*, abs/2402.06196.

Yee Man Ng and Ilia Markov. 2024. Leveraging open-source large language models for native language identification. *CoRR*, abs/2409.09659.

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

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 *EMNLP*, pages 5687–5711.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392.

Sumedh Rasal. 2024. LLM harmony: Multi-agent communication for problem solving. *CoRR*, abs/2401.01312.

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

Omar Shaikh, Hongxin Zhang, William Held, Michael S. Bernstein, and Diyi Yang. 2023. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In *ACL*, pages 4454–4470.

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 *EMNLP*, pages 9248–9274.

Zhili Shen, Chenxin Diao, Pavlos Vougiouklis, Pascual Merita, Shriram Piramanayagam, Damien Graux, Dandan Tu, Zeren Jiang, Ruofei Lai, Yang Ren, and Jeff Z. Pan. 2024. Gear: Graph-enhanced agent for retrieval-augmented generation. *CoRR*, abs/2412.18431.

Zhengliang Shi, Shuo Zhang, Weiwei Sun, Shen Gao, Pengjie Ren, Zhumin Chen, and Zhaochun Ren. 2024. Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *ACL*, pages 7339–7353.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *CoRR*, abs/2401.15391.

Amir Taubenfeld, Yaniv Dover, Roi Reichart, and Ariel Goldstein. 2024. Systematic biases in LLM simulations of debates. In *EMNLP*, pages 251–267.

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 *ACL*, pages 10014–10037.

Thomas Walshe, Sae Young Moon, Chunyang Xiao, Yawwani Gunawardana, and Fran Silavong. 2025. Automatic labelling with open-source llms using dynamic label schema integration. *Preprint*, arXiv:2501.12332.

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 *NeurIPS*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Trans. Assoc. Comput. Linguistics*, 6:287–302.

Feijie Wu, Zitao Li, Fei Wei, Yaliang Li, Bolin Ding, and Jing Gao. 2025. Talk to right specialists: Routing and planning in multi-agent system for question answering. *Preprint*, arXiv:2501.07813.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. Watch every step! LLM agent learning via iterative step-level process refinement. In *EMNLP*, pages 1556–1572.

Ruixin Yang, Dheeraj Rajagopal, Shirley Anugrah Hayati, Bin Hu, and Dongyeop Kang. 2024. Confidence calibration and rationalization for llms via multi-agent deliberation. *CoRR*, abs/2404.09127.

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 *EMNLP*, pages 2369–2380.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*.

Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, Linling Shen, Jianqiao Lu, Haochen Tan, Yukang Chen, Hao Zhang, Zhan Shi, Bailin Wang, Zhijiang Guo, and Jiaya Jia. 2024. Mr-ben: A meta-reasoning benchmark for evaluating system-2 thinking in llms. In *NeurIPS*.

Bin Zhang, Hangyu Mao, Jingqing Ruan, Ying Wen, Yang Li, Shao Zhang, Zhiwei Xu, Dapeng Li, Ziyue Li, Rui Zhao, Lijuan Li, and Guoliang Fan. 2023. Controlling large language model-based agents for large-scale decision-making: An actor-critic approach. *CoRR*, abs/2311.13884.

Taolin Zhang, Qizhou Chen, Dongyang Li, Chengyu Wang, Xiaofeng He, Longtao Huang, Hui Xue', and Jun Huang. 2024a. Dafnet: Dynamic auxiliary fusion for sequential model editing in large language models. In *ACL*, pages 1588–1602.

Taolin Zhang, Dongyang Li, Qizhou Chen, Chengyu Wang, Longtao Huang, Hui Xue, Xiaofeng He, and Jun Huang. 2024b. R^4: Reinforced retriever-reorder-responder for retrieval-augmented large language models. In *ECAI*, pages 2314–2321.

Taolin Zhang, Chengyu Wang, Minghui Qiu, Bite Yang, Zerui Cai, Xiaofeng He, and Jun Huang. 2021. Knowledge-empowered representation learning for chinese medical reading comprehension: Task, model and resources. In *ACL*, pages 2237–2249.

Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Longagent: Scaling language models to 128k context through multi-agent collaboration. *CoRR*, abs/2402.11550.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *EMNLP*, pages 15686–15702.

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 *ICLR*.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. 2024. Efficientrag: Efficient retriever for multi-hop question answering. In *EMNLP*, pages 3392–3411.

A Implementation Details of BELLE

A.1 Model Details

Retrieval Setup. To retrieve external knowledge for retrieval-augmented reasoning operators, we use the October 2017 Wikipedia dumps[7] as the candidate document pool. Considering the computational cost of retrievers, we use the sparse model BM25 (Robertson and Zaragoza, 2009) to replace the complex models. [8] We set a range of 3 to 10 candidate documents in each dataset for the multi-hop questions corresponding to these methods.

Metrics. The evaluation metrics are token-level EM (Exact Match), F1 and Acc (Accuracy). The difference between EM and Acc is that EM must be strictly included in the ground-truth string, while Acc uses the LLM to perform semantic consistency checks on prediction and ground-truth.

Baselines. (1) **SP** denotes the standard prompting for obtaining the response. (2) **Chain-of-Thought (CoT)** generates logic reasoning steps before the final answer (Wei et al., 2022). We use 4-shot for each question, providing an example for each type of question respectively. (3) **Single-step Retrieval** involves using the multi-hop question as the query to retrieve the candidate documents one time and then concatenating the search results into the prompt to perform prompt reasoning (Lazaridou et al., 2022). (4) **Self-Ask** uses an iterative method to break down complex questions, progressively generating and addressing sub-questions until the final answer is reached (Press et al., 2023). (5) **IRCoT** alternates among the retrieval-augmented reasoning methods until the retrieved information is adequate to answer the question (Trivedi et al., 2023). (6) **FLARE** dynamically adjusts the retrieval timing according to the confidence in reasoning and performs retrieval based on the subsequent reasoning sentences (Jiang et al., 2023b). (7) **ProbTree** breaks down the question into a tree structure, using logprobs-based aggregation of sub-questions to derive the final answer (Cao et al., 2023). (8) **BeamAggR** also breaks down complex questions into tree structures, which consist of atomic and composite questions, and then applies bottom-up reasoning (Chu et al., 2024a). (9) **EfficientRAG** iteratively generates new questions without requiring LLM calls in each round and filters out irrelevant information (Zhuang et al., 2024). (10) **GEAR** (Shen et al., 2024) presents a new graph-based retriever called SyncGE, which uses an LLM to identify initial nodes for graph exploration. (11) **RopMura** (Wu et al., 2025) is a multi-agent system that integrates both a planner and a router to support QA across various knowledge domains. (12) **LONGAGENT** (Zhao et al., 2024) scales LLMs (e.g., LLaMA (Touvron et al., 2023)) to a context of 128K based on MAD system and demonstrates potential superiority in long-text processing. [9]

Experimental Settings. Our main experiments are conducted using GPT-3.5-turbo(Brown et al., 2020) as the backbone, provided by the Azure OpenAI 2024-01-25 version. In addition, we perform experiments using GPT-4 (OpenAI, 2023), with the Azure OpenAI 2024-06-13 version, to ensure the accuracy of classification in Sect. 3, despite a higher response cost.[10] To verify the effectiveness of our LLM-agnostic multi-hop QA framework, we replace the backbone of all baselines with Qwen2.5-7B (Qwen Team, 2024) and Mistral-7B (Jiang et al., 2023a).

For the SFT experiment in Appendix B.3, we use Qwen2.5-7B-instruct, training on 8 $\times$ Nvidia A100 GPUs for about 15 hours. We use the full tuning paradigm to perform the SFT process. The hyperparameters are as follows: batch size is 1, learning rate is 1e-5, with the AdamW optimizer (Loshchilov and Hutter, 2019), and the number of epochs is 1.

A.2 Dataset Details

Datasets. We evaluate BELLE on four open-domain multi-hop QA datasets: MultiHop-RAG (Tang and Yang, 2024), 2WikiMultiHopQA (Ho et al., 2020), HotPotQA (Yang et al., 2018), and

[7] https://hotpotqa.github.io/wiki-readme.html

[8] The retriever can be replaced by other high-precision neural models (Karpukhin et al., 2020; Izacard et al., 2022) as long as the candidate documents are prepared in advance.

[9] Due to the space limitation, we abbreviate the model name "EfficientRAG" to "EffiRAG" and "LONGAGENT" to "LONGA" in Table 1, respectively.

[10] https://learn.microsoft.com/en-us/azure/ai-services/openai/

Type ↓ Data →	D1	D2	D3	D4
Inference	816	2158	4758	938
Comparison	856	2495	3819	856
Temporal	583	1033	2691	414
Null	301	1719	1308	251
Total	2556	7405	12576	2459

Table 4: The number of multi-hop question types included in each dataset. "D1", "D2", "D3", and "D4" represent Multi-hop RAG, HotpotQA, 2WikiQA, and MuSiQue respectively.

MuSiQue (Trivedi et al., 2022). These datasets contain questions with 2 to 4 hops. For HotPotQA, 2WikiMultiHopQA, and MuSiQue, we use the same development and test sets extracted from the original dataset similar to IRCoT (Trivedi et al., 2023). In Table 4, we present the data distribution of different multi-hop question types in four datasets. Here, we refer to the Multi-hop RAG (Tang and Yang, 2024), providing the description of different multi-hop question types as follows: (1) **Inference:** This type requires identifying the internal logical semantics of multi-hop questions and connecting them through intermediate entities for answering. The final answer is an entity string. (2) **Comparison:** This is usually achieved by comparing the similarities and differences related to the entities or topics in the multi-hop questions. The answer is typically a definitive word such as "Yes", "No" or "Consistently". (3) **Temporal:** These questions are mainly answered based on the sequence of events occurring at different time points. The answer is also typically words such as "Yes", "No", or a temporal indicator word like "before". (4) **Null:** These are questions whose answer cannot be obtained from the retrieved documents or are other free-form questions. The answer is generally a noun with an indefinite form. Particularly, we choose the distractor setting dataset of HotpotQA (Yang et al., 2018), and all hops (i.e., 2, 3, and 4-hop) in MuSiQue (Trivedi et al., 2022) are used.

SFT QA Dataset: We collect the SFT QA pair data for the experiment of question classifier analysis in Appendix B.3. The training prompt is shown in Fig. 7. We use the training datasets of HotpotQA-hard, and 2WikiQA-hard to form the SFT data. The number of training data points is 15,661 and 12,576, respectively.

Reasoning Cost Dataset: To demonstrate the effectiveness and computational resource cost of our BELLE model, we design an inference consumption in Sect. 5.2.2. We choose various retrieval-augmented reasoning methods as our strong baselines. The metrics are the retrieved tokens required and the average F1 results. We particularly select the difficult multi-hop questions as the dataset for this experiment, randomly selecting 5,000 samples with various types from the prediction errors of LLMs.

B Additional Experimental Discussion

B.1 Annotation Process of Question Types

The Complete Results: Considering that there are too many combinations between operators, we limit the experiment to the two most typical combinations. In Fig. 8, we present the overall results for data analysis (see Sect. 3). Due to the relatively small range of MuSiQue results compared to others, we have considered space limitations and placed its results in Appedix. The conclusions in Sect. 3 are consistently effective.

Analysis of Question Type Annotation: For the question type annotation process, to ensure the accuracy of data labeling, we use the GPT-4 model rather than GPT-3.5-turbo. It has been widely adopted in many works for data labeling (Ng and Markov, 2024; He et al., 2024b; Walshe et al., 2025). The process of cross validation involves two NLP experts conducting separate labeling and discussing results with inconsistent cases until the error is controlled within 5%. This mechanism of labeling from coarse-grained to fine-grained manual review is widely used in many works (Rajpurkar et al., 2016; Jing et al., 2019; Zhang et al., 2021). Therefore, after selecting reliable models and experts, the labeling results of data analysis can be trusted. Due to the flexibility of our framework, we can directly add type descriptions in `Meta prompt` to expand fine-grained multi-hop question types. For example, we have added two new types of fine-grained inference "Bridge-comparison" and "Compositional" (Ho et al., 2020). Specifically, we add two examples and twp multi-hop QA question type descriptions in Fig. 9.

- The `Meta Prompt` is transformed to: "As an assistant, ... 'Inference', 'Comparison', 'Temporal' , 'Bridge-comparison', 'Compositional' and 'Null' "
- The demonstration examples are added: "Ex-

SFT for Question Classifier

- **SFT Prompt**

Instruction: You are now an experienced question type annotator who can identify the specific type of input questions. You will receive an input text, please determine which data label the text belongs to. The data labels you can choose are as follows:
-------- Data labels --------
[Inference, Comparison, Temporal, Null]
-------Examples-------
There are some examples for you to refer to:
Example 1:
Input: "Which football club, recently discussed in articles by The New York Times, The Guardian, and Sky Sports, experienced a home defeat in the Premier League, has a player named Reece James who may undergo a late fitness check, and is expanding its U.S. presence under Todd Boehly's co-ownership?"
Output: Inference
Example 2:
Input: "Does the Yahoo News article praise the Biden administration's effectiveness on climate issues, while the Engadget article suggests that society is being positively impacted by the interconnectedness of climate change, the modern internet, and authoritarianism?"
Output: Comparison
Example 3:
Input: "Between the report by The Verge on Apple's defense of its Google Search deal published on September 26, 2023, and the report by TechCrunch on the class action antitrust suit against Google published on December 15, 2023, was there a change in the portrayal of Google's business practices with other companies?"
Output: Temporal
Example 4:
Input: "Considering the information from a BBC News article and a Rolling Stone piece on Dua Lipa: At Your Service, which single character would be common when referencing the title of the podcast mentioned by both sources?"
Output: Null
-------Starts Working------
Input: "What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?"
Output: Inference [Example from HotpotQA]

Figure 7: The SFT template for our experiment in Appendix B.3.

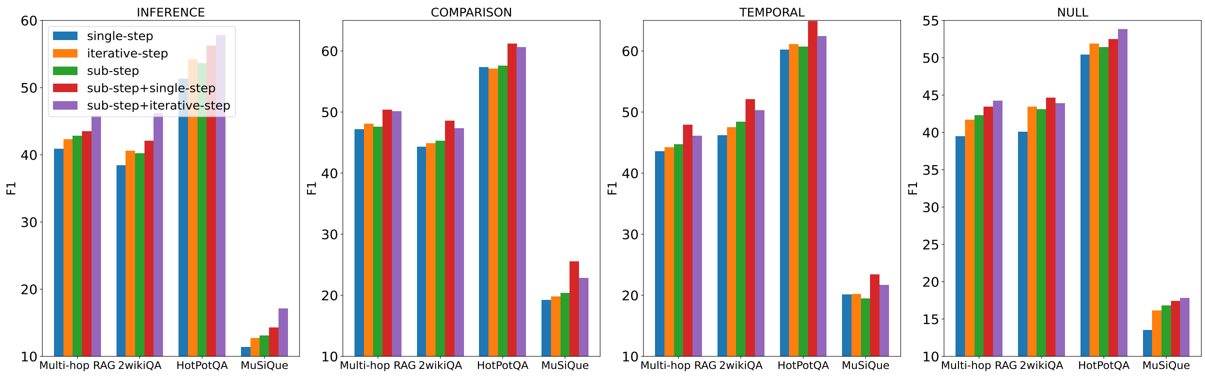

Figure 8: Overall Comparison of single and combined operators in different multi-hop questions.

ample 5: Why did the founder of Versus die? (Output: Compositional)" and "Example 6: Are both director of film FAQ: Frequently Asked Questions and director of film The Big Money from the same country? (Output: Bridge-comparison) "

Then we perform the experiments on two new types, our BELLE framework further improves the performances over the four datasets to "65.1 (+0.4) / 71.2 (+0.8)", "59.9 (+0.7) / 67.8 (+1.3)", "71.4 (+1.7) / 79.3 (+3.6)", "30.4 (+0.2) / 41.8 (+1.2) " in terms of EM and F1 (%) respectively. These results indicate that by incorporating meaningful question types for multi-hop QA tasks, our framework continues to achieve performance improvements under the bi-layer reflection mechanism guided by question types. This experiment roughly verifies the effectiveness of our BELLE framework for multi-hop QA tasks with simple extensions.

B.2 Results on Different Backbones

To demonstrate the generalization ability of our method to various backbones, we also conduct experiments on open-source models and those with larger parameters. We choose Qwen2.5-7B (Qwen Team, 2024) and Mistral-7B (Jiang et al., 2023a) as our open-source backbones and GPT-4 (OpenAI, 2023) as the larger closed-book model. We report the F1 metric for these datasets and the average results over 2, 3, and 4 hops in MuSiQue.

As shown in Table 6, we observe that our BELLE model with respect to 7B open-source backbones can achieve SOTA results on all four multi-hop QA datasets compared to previous strong baselines, demonstrating its model-agnostic nature and effectiveness. On datasets Multi-hop RAG and HotpotQA, Mistral-7B performs better than Qwen2.5-7B due to the specialized training in long context dialogue ability. When we replace them in BELLE

Dataset→	Multi-hop RAG		HotpotQA		2WikiQA		MuSiQue		Avg.	
Models↓	# token	F1	#token	F1	#token	F1	#token	F1	#token	F1
Single-step	4109	30.5	3876	29.4	3652	21.5	4356	18.3	3998	24.9
IRCoT	15368	39.2	14677	45.8	13924	31.6	14229	22.7	14550	34.8
FLARE	17212	41.4	19516	44.8	16592	33.4	17285	24.1	17651	35.9
ProbTree	30975	45.7	28360	47.3	37241	37.2	40032	28.1	34152	39.6
BeamAggR	26940	52.3	25463	54.9	31943	43.6	34260	30.1	29651	45.2
Basic MAD	16439	49.3	13530	53.2	21402	42.5	22593	28.3	18491	43.3
BELLE	18324	56.4	19520	62.8	22394	47.2	23723	33.5	20742	50.0

Table 5: Token consumption per multi-hop questions and performance in four datasets.

Model ↓ Dataset →	D1	D2	D3	D4	Avg.
Qwen2.5-7B					
CoT	24.9	22.5	19.9	11.8	19.8
ProbTree	50.7	47.1	55.6	17.3	42.7
BeamAggR	55.8	51.8	62.4	23.2	48.3
BELLE	**64.1**	**59.4**	**68.5**	**32.8**	**56.2**
Mistral-7B					
CoT	26.3	25.1	19.2	10.6	20.3
ProbTree	51.4	48.7	53.8	16.9	42.7
BeamAggR	56.6	54.3	59.9	22.7	48.4
BELLE	**65.8**	**61.3**	**64.4**	**29.7**	**55.3**
GPT-4					
CoT	51.8	47.2	44.9	24.6	42.1
ProbTree	62.8	61.5	68.3	30.5	55.8
BeamAggR	67.6	63.4	72.7	36.2	60.0
BELLE	**71.3**	**66.9**	75.3	**41.3**	**63.7**
BELLE (GPT-3.5-turbo)	70.4	66.5	**75.7**	40.6	63.3

Table 6: Results of different LLMs in terms of F1 (%).

Type Strategy	D1	D2	D3	D4	Avg.
Qwen2.5-7B					
ICL	64.1	**59.4**	68.5	**32.8**	**56.2**
SFT	**64.5**	58.9	**69.1**	31.2	55.9
Zero-shot	61.5	57.2	66.3	29.7	53.7
GPT-3.5-turbo					
ICL	70.4	**66.5**	75.7	**40.6**	**63.3**
SFT	**70.6**	65.8	**75.9**	38.2	62.6
Zero-shot	68.1	63.5	71.3	36.7	59.9

Table 7: Performance of multi-hop QA tasks with different question type strategies in terms of F1 (%).

with a larger backbone, the performance further improves on average (+0.4%). Since the GPT-4 needs higher price to obtain response, we use GPT-3.5-turbo to perform the main experiments.

B.3 Impact of Type Classifier

From the results of the ablation study in Table 2, we can find that incorporating question types is crucial for guiding our MAD system to provide reasonable planning of combined operators. Hence, we further analyze the methods used to obtain question types: in-context learning (ICL), SFT, and zero-shot prompting. For the ICL mechanism, we provide a sample for each type of multi-hop question combined with instructions to form the input prompt of the LLMs. In addition, we use the existing question types and QA pairs to test the SFT mechanism and the training datasets are described in Appendix A.2. In zero-shot prompting, we only use the instruction and label space to prompt the LLMs. From the results in Table 7, although ICL may fluctuate on some datasets compared to SFT, it can achieve the best average performance regardless of the parameter size of the LLMs. However, zero-shot prompting results in a rapid decrease in effectiveness due to the complex reasoning required for multi-hop questions.

B.4 Detailed Reasoning Cost Results

In Table 5, we provide the comprehensive token consumption per instance, where performance is averaged across four datasets. We assess the computational cost by measuring the average token usage per question. Specifically, it includes calculating the cost of prompt tokens, such as demonstrations, questions, and retrieved documents. For iterative-step methods such as IRCoT (Trivedi et al., 2023), we have summed the number of document tokens recalled by all steps. In our BELLE model, we count the number of recalled document tokens for the combined operators.

The main advantage of our model lies in fully utilizing the current state and historical information, making the execution planning of the combined operators for the multi-hop question more reason-

# of Debaters	D1	D2	D3	D4	Avg.
`Qwen2.5-7B`					
2 (Default)	64.1	**59.4**	68.5	**32.8**	56.2
$N_{f3} \rightarrow N_{s3}$	63.9	58.7	68.1	32.3	55.8
$N_{f4} \rightarrow N_{s4}$	**64.5**	59.2	**68.6**	32.7	**56.3**
$N_{f5} \rightarrow N_{s5}$	63.2	58.4	67.7	31.8	55.3
`GPT-3.5-turbo`					
2 (Default)	70.4	66.5	**75.7**	40.6	63.3
$N_{f3} \rightarrow N_{s3}$	69.8	66.9	75.2	39.9	63.0
$N_{f4} \rightarrow N_{s4}$	**71.2**	**67.4**	75.5	**41.3**	**63.9**
$N_{f5} \rightarrow N_{s5}$	69.4	65.8	74.9	39.7	62.5

Table 8: Results of multi-hop QA tasks with more debaters in terms of F1 (%). $N_{fi} \rightarrow N_{sj}$ means i debaters in the first layer and j debaters in the second layer.

Debate Level	D1	D2	D3	D4	Avg.
`Qwen2.5-7B`					
L2 (Default)	**64.1**	**59.4**	68.5	**32.8**	**56.2**
L0	63.8	59.1	**68.6**	31.5	55.8
L1	62.6	57.3	67.8	29.2	54.2
L3	61.5	55.8	67.4	27.4	53.0
`GPT-3.5-turbo`					
L2 (Default)	**70.4**	**66.5**	**75.7**	**40.6**	**63.3**
L0	69.6	65.7	73.8	39.4	62.1
L1	68.2	63.5	72.4	38.8	60.7
L3	67.3	63.1	71.5	37.5	59.9

Table 9: Performance of multi-hop QA tasks with different debate levels in terms of F1 (%).

able. Hence, it can reduce the number of rounds of combined operator retrieval and lowering the cost of prompt inference length.

B.5 Analysis of Debaters

Impact of Debater Number. In this experiment, we increase the number of debaters in each layer for a more comprehensive discussion. Specifically, we increase the number of debaters to three, four, and five for each layer, and then analyze the results of the bi-layer debate. For the three debaters, we allocate two to the affirmative side and one to the negative side in the first level. The same settings apply to the second level. We evenly allocate the number of roles within four debaters. For the five debaters, the allocation mechanism is similar to that of three debaters. In Table 8, we can observe that (1) As the number of debaters increases, the performance of the model decreases (63.3 $\rightarrow$ 63.8 using GPT-3.5-turbo). Considering the performance and cost of debating (see Sect. 5.2.2), we choose 2 debaters to report the main results. (2) The debate effect steadily improves when the number of debaters is balanced (e.g., 2 debaters and 4 debaters).

Impact of Debate Level. We then study whether the atmosphere of the debate prompt has an impact on the results. Hence, we design different instructions (see Appendix C.4) to initialize the debaters' meta prompt. In Table 9, asking debaters to "tit for tat" is necessary for our bi-level MAD system to achieve good performance. However, we find that "must disagree with each other on every point" does not lead to the best performance and may even result in a certain decrease (e.g., $\downarrow$ 3.4 in L3). We speculate that both levels can basically reach a mutually agreed viewpoint in the early rounds of debate round friendly (see Fig. 5).

B.6 Discussion of Framework Dependence

As for the dependence of predefined heuristics and manual annotations of our BELLE framework, the previous MAD system (Feng et al., 2024; Xiong et al., 2024; Liang et al., 2024) for solving NLP tasks utilizes task characteristics for prompt settings and the multi-agent collaboration design. For the edge cases or evolving domains, the fast-debater of the second-layer judges the current discussion of the first-layer based on specific tasks without large-scale heuristic prompt debugging using `Meta Prompt`, while the slow debater comprehensively outputs a response based on historical information. For some special task examples of edge cases or evolving domains, our second-layer MAD mechanism can perform reflective collaboration to further alleviate the possible operator viewpoint bias in high-difficulty examples at parameter scales such as GPT-3.5-turbo (e.g. 1st round to 2nd round in. Fig. 10).

C The Templates of BELLE

C.1 Question Type Annotation

Our question type annotation prompt is shown in Fig. 9. We choose an example from the HotpotQA dataset (Yang et al., 2018) and use GPT-4 (OpenAI, 2023) to annotate the type of answer as "{"type": "Inference"}". This template is also used for the question type classifier (see Sect. 4.1), replaced with GPT-3.5-turbo (Brown et al., 2020) due to the high cost of responses.

Question Type Annotation

- **Question Type Annotation Prompt**

As an assistant, your task is to answer the question type after. Your answer should be after in JSON format with key ”type” and its value should be string. There are four types you can choose from: ‘Inference’, ‘Comparison’, ‘Temporal’ and ‘Null’.

-------Examples-------
There are some examples for you to refer to:
Example 1:
Input: “Which football club, recently discussed in articles by The New York Times, The Guardian, and Sky Sports, experienced a home defeat in the Premier League, has a player named Reece James who may undergo a late fitness check, and is expanding its U.S. presence under Todd Boehly's co-ownership?”
Output: {‘type’: ‘inference’}
Example 2:
Input: “Does the Yahoo News article praise the Biden administration's effectiveness on climate issues, while the Engadget article suggests that society is being positively impacted by the interconnectedness of climate change, the modern internet, and authoritarianism?”
Output: {‘type’: ‘comparison’}
Example 3:
Input: “Between the report by The Verge on Apple's defense of its Google Search deal published on September 26, 2023, and the report by TechCrunch on the class action antitrust suit against Google published on December 15, 2023, was there a change in the portrayal of Google's business practices with other companies?”
Output: {‘type’: ‘Temporal’}
Example 4:
Input: “Considering the information from a BBC News article and a Rolling Stone piece on Dua Lipa: At Your Service, which single character would be common when referencing the title of the podcast mentioned by both sources?”
Output: {‘type’: ‘Null’}
-------Starts Working------
Input: “What is the name of this American musician, singer, actor, comedian, and songwriter, who worked with Modern Records and born in December 5, 1932?”
Output: {‘type’: ‘Inference’}[Example from HotpotQA]

Figure 9: Prompt examples of question type annotation.

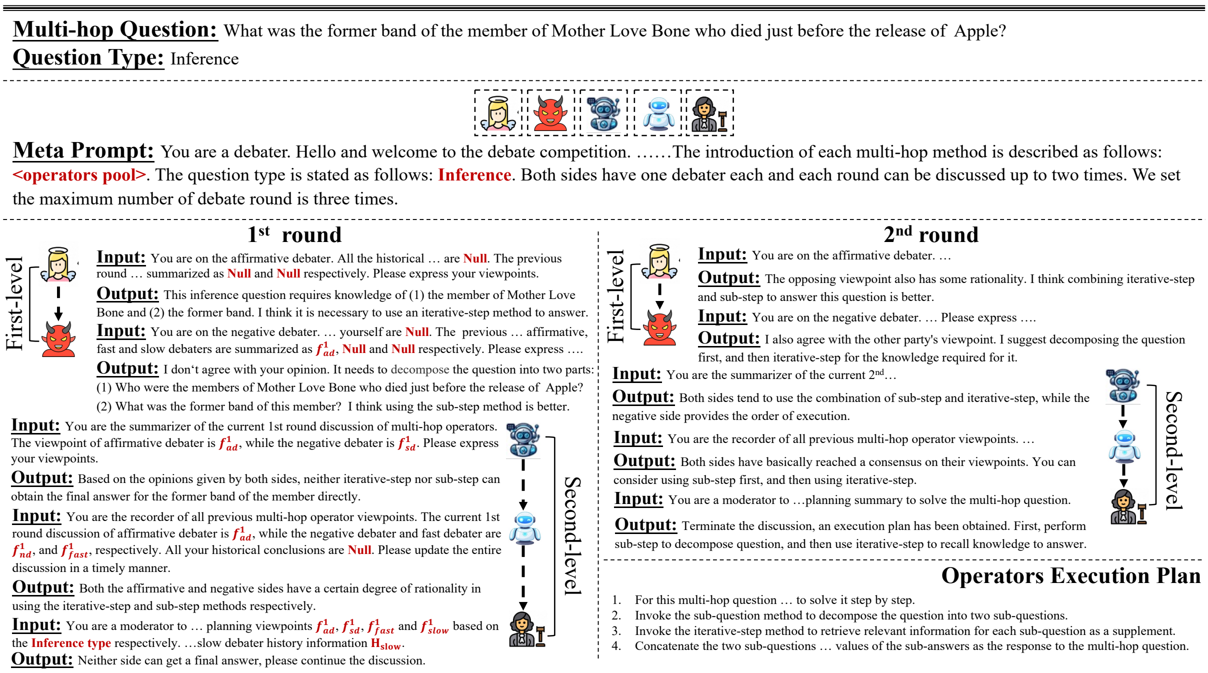

Figure 10: An example of our bi-level MAD process. Due to the excessive output content of the debater, we have replaced it with the corresponding mathematical symbols described in Sect. 4. In first round, we represent it using “Null” as some placeholder information has not been obtained yet.

C.2 Meta Prompts

Table 10 illustrates our meta prompt used to initialize the debaters. The speaking order of the debaters is as follows: affirmative debater and negative debater in the first level, followed by fast debater and slow debater in the second level, and finally the judge in each round.

C.3 An Example of Operator Planning

To facilitate the readers’ understanding of the operation process of our bi-level debate system, we provide an example from the HotpotQA dataset (Yang et al., 2018) in Fig. 10, detailing how to obtain the combined operators through a step-by-step planning process.

Meta Prompt	You are a debater. Hello and welcome to the debate competition. It's not necessary to fully agree with each other's perspectives, as our objective is to find the correct execution plan of operators to answer the multi-hop question based on its type. You can freely combine the methods from the operator pool to solve the task. The introduction of each multi-hop method is described as follows: <operators pool>. The question type is stated as follows: <question type>. Both sides have one debater each and each round can be discussed up to two times. We set the maximum number of debate round is three times.
Affirmative Debater	You are on the affirmative debater. All the historical round discussion results of yourself are <H_{ad}^{t-1}>. The previous round state of fast and slow debaters are summarized as <f_{fast}^{t-1}> and <f_{slow}^{t-1}> respectively. Please express your viewpoints.
Negative Debater	You are on the negative debater. You disagree with the affirmative debater's points. All the historical round discussion results of yourself are <H_{nd}^{t-1}>. The previous round state of affirmative, fast and slow debaters are summarized as <f_{ad}^{t}>, <f_{fast}^{t-1}> and <f_{slow}^{t-1}> respectively. Please express your viewpoints.
Fast Debater	You are the summarizer of the current t-th round discussion of multi-hop operators. The viewpoint of affirmative debater is <f_{ad}^{t}>, while the negative debater is <f_{nd}^{t}>. Please express your viewpoints.
Slow Debater	You are the recorder of all previous multi-hop operator viewpoints. The current t-th round discussion of affirmative debater is <f_{ad}^{t}>, while the negative debater and fast debater are <f_{nd}^{t}> and <f_{fast}^{t}> respectively. All your historical conclusions are <H_{slow}^{t-1}>. Please update the entire discussion in a timely manner.
Judge	You are a moderator to give a operator planning summary to solve the multi-hop question. There is a bi-level opposing debaters involved in a debate competition at the of last round. They have already presented their operator planning viewpoints <f_{ad}>, <f_{nd}>, <f_{fast}> and <f_{slow}> based on the <question type> respectively. If you can get a clear summary, you can end the discussion process of the multi-hop question after outputting. If you determine that you cannot output a summary, you can extract the solution from the slow debater history information <H_{slow}>.

Table 10: The debating prompts for all debaters in our bi-level MAD system of BELLE. Each debater needs to fill content into the symbol "<>" before performing the discussion process.

Level	Prompt
0	Both sides must reach a full consensus on every point of the debate. Each multi-hop operator selection must be agreed upon by both sides.
1	Most of the debate should be characterized by disagreements, but there may still be a small amount of consensus on less important operators selection based on question types.
2 (Default)	It's not necessary to fully agree with each other's perspectives, as our objective is to find the correct execution plan of operators to answer the multi-hop question based on its type.
3	Both sides must disagree with each other on every point of the multi-hop QA operators debate. There should be no consensus whatsoever.

Table 11: The different debate levels for bi-level MAD process.

C.4 Different Debate Levels

In Table 11, we set four debate-level prompts to evaluate the influence of our bi-level MAD process.