Multi-scale Granularity Alignment for Multi-hop Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Although query decomposition benefits multi-hop RAG, the entanglement of mixed granularities within un-decomposed knowledge hinders evidence alignment for attention-constrained models. To address this, we introduce MGA-RAG, a novel RAG framework that synergistically combines granularity decoupling with multi-scale granularity alignment. Specifically, MGA-RAG pioneers a multi-scale alignment strategy that reconstructs retrieved documents into representations of varying granularities and conducts scale-aware alignment with decomposed queries. This strategy shifts the focus to the alignment process prior to the generation phase, enabling the generation model to concentrate on key evidence while reducing redundancy and noise. Meanwhile, by fusing the generation results from multiple scale-specific views, MGA-RAG promotes a balanced attention to evidence across granularities. Furthermore, we introduce a Self-Correcting Decoupling Agent to audit the granularity decoupling process, mitigating error propagation caused by inaccurate granularity decomposition. Experimental results on three multi-hop QA datasets demonstrate that MGA-RAG significantly outperforms existing methods. In-depth analysis further validates the effectiveness of the proposed approach.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as the dominant paradigm for empowering Large Language Models (LLMs) with external knowledge to facilitate complex Question Answering (QA) (Gao et al., 2023; Chen et al., 2024). However, existing RAG architectures continue to face formidable challenges in multi-hop QA tasks. Particularly in scenarios constrained by limited reasoning capabilities, these systems often struggle to achieve precise alignment between query logic and retrieved knowledge (Zhao et al., 2024).

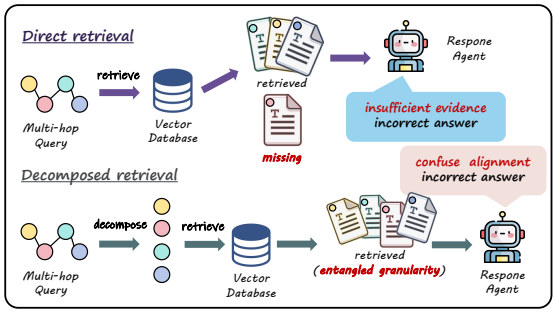

Figure 1: The schematic illustration of challenges in RAG of multi-hop questions. Direct retrieval of multi-hop queries often yields **insufficient** evidence due to missing information. Query decomposition enhances the relevance of retrieved documents, yet their entangled granularity can still **confuse** the model in selecting appropriate evidence for response generation.

To surmount this bottleneck, the research community has primarily directed its exploration along the dimensions of data structuring and workflow reconfiguration. From the data perspective, works represented by GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024) attempt to explicitly model entity relationships by transforming knowledge bases into graphs. While these approaches enhance multi-hop reasoning capabilities, their widespread adoption is hindered by prohibitive graph construction costs, complex maintenance procedures, and limited cross-domain adaptability. Conversely, at the workflow level, spurred by the emergence of Agent-based technologies, frameworks such as Iter-DRAG (Yue et al., 2025) and Iter-RetGen (Shao et al., 2023) have introduced iterative reasoning strategies that guide retrieval by decomposing complex queries into simpler subproblems. However, a critical unresolved challenge persists because the content retrieved on the "document side" remains coarse-grained and entangled even though agents effectively reduce complexity on the "query side". This "Granularity Mismatch" between fine-grained queries and coarse-grained

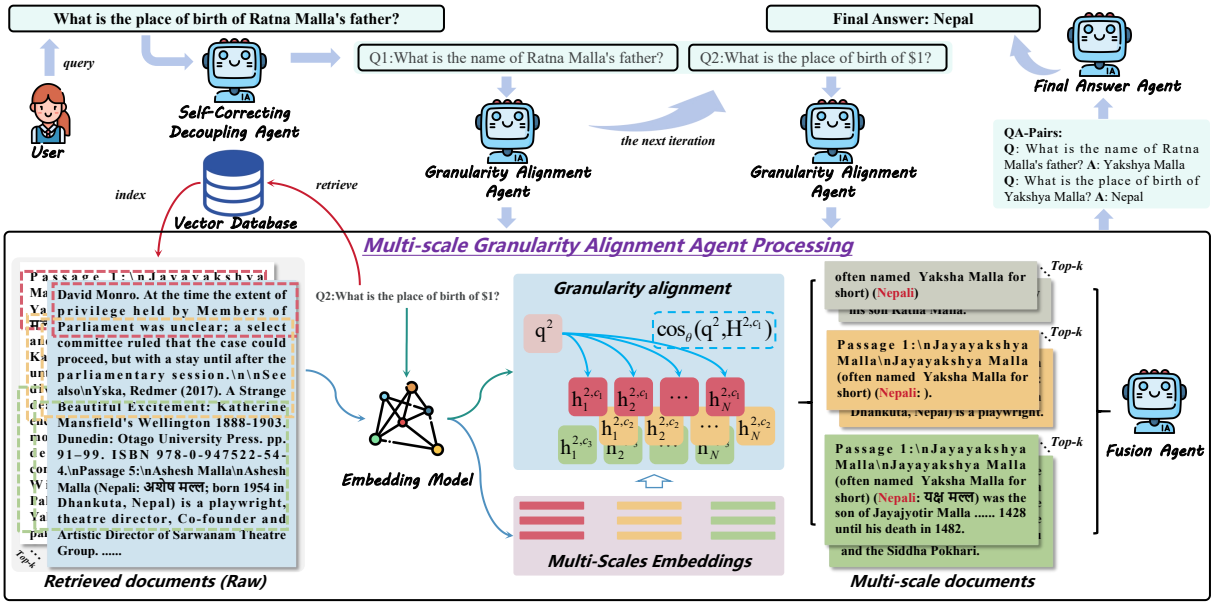

Figure 2: MGA-RAG framework illustration. MGA-RAG is a framework designed for iteratively addressing multi-hop questions in long-document question answering. It decomposes the multi-hop query into a sequence of sub-questions, which are then reviewed by a Reviser-Agent. Critically, a multi-scale granularity alignment strategy is employed to align single-granularity sub-questions with entangled granularity retrieved documents, followed by a balanced fusion of responses across multiple scales.

documents introduces substantial irrelevant noise during the reasoning process and imposes a severe cognitive burden on the model. For small-to-medium-sized models constrained by limited attention capacities, such noise interference often proves fatal by impeding their ability to extract pivotal evidence from redundant text.

To address the above challenges, we propose the Multi-scale Granularity Alignment Retrieval-Augmented Generation (MGA-RAG) framework which aims to maximize the reasoning signal-to-noise ratio under constrained model capabilities through a refined bidirectional alignment mechanism. Specifically, MGA-RAG first leverages a *Self-Correcting Decoupling Agent* to decompose complex compound queries into logically coherent atomic sub-problems. It subsequently transcends the limitations of traditional static document chunking by introducing predefined sliding windows to construct a multi-scale pool of candidate text segments. Building on this foundation, a specialized *Granularity Alignment Agent* precisely aligns specific-granularity sub-problems with low-noise text passages. Following independent reasoning across various scales, a *Fusion Agent* synthesizes the multi-source results into a final answer. This holistic granularity control ensures high consistency in semantic density between queries and evidence to effectively mitigate interference from information entanglement and maximally unleash the model's potential for semantic understanding and reasoning. Our main contributions are as follows:

- We propose MGA-RAG, a novel framework designed to address the granularity mismatch challenge in multi-hop RAG. Significantly, this framework introduces the innovative perspective of decomposing granularity at the knowledge level for the first time.
- We propose a Multi-scale Granularity Alignment strategy that aligns coherent single-granularity sub-queries generated by a Self-correcting Decoupling Agent with multi-scale evidence to effectively mitigate interference from granularity-induced noise.
- We conduct comprehensive experiments on multiple multi-hop question answering datasets. The results demonstrate that MGA-RAG significantly outperforms existing state-of-the-art methods, validating the effectiveness of our granularity control mechanism.

2 Related Work

Retrieval-Augmented Generation. With the proliferation of Large Language Models (LLMs),

Retrieval-Augmented Generation (RAG) has emerged as a standard paradigm to enhance factual accuracy and mitigate hallucinations (Su et al., 2024; Wang et al., 2024). Although modern LLMs support extended context windows, performance often degrades as input length increases—a phenomenon known as being "lost in the middle" (Liu et al., 2023). RAG addresses this by filtering redundant information and localizing task-relevant content, thus enabling models to focus on critical evidence within a manageable context (Zhou et al., 2024). However, traditional vanilla RAG typically relies on single-round retrieval based on surface-level semantic similarity (Zhuang et al., 2024), which often fails to capture the deep logical dependencies required for complex, multi-hop reasoning tasks.

Graph-based RAG. To capture higher-order structural information within documents, Graph-based RAG frameworks have been proposed. By restructuring flat text into structured graphs, methods like GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024) enable the retrieval system to traverse multi-hop reasoning paths and capture global entity relationships. To further enhance associative memory, HippoRAG (Jimenez Gutierrez et al., 2024) mimics hippocampal functions by using Personalized PageRank to integrate multi-hop evidence. Similarly, TreeRAG (Tao et al., 2025) utilizes tree-structured hierarchical indexing to organize knowledge from general concepts to specific details, ensuring semantic consistency across different abstraction levels. These approaches significantly improve the model's ability to synthesize information across disparate document segments. Nevertheless, graph-based methods often rely on pre-defined schemas or expensive graph-construction stages, which can limit their flexibility when encountering open-domain questions or rapidly evolving knowledge bases.

Iterative RAG. Recognizing the limitations of static retrieval, iterative RAG approaches introduce dynamic reasoning chains to refine information acquisition. For instance, IRCoT (Trivedi et al., 2023) interleaves retrieval with Chain-of-Thought (CoT) steps, using generated reasoning sentences to guide subsequent retrieval. While methods like Iter-RetGen (Shao et al., 2023) and IterRAG (Yue et al., 2025) focus on sub-query decomposition, DualRAG (Cheng et al., 2025) employs a dual-path framework that iteratively reconciles query-centric and context-centric perspectives to improve reasoning consistency. However, these strategies focus on query transformation while neglecting the inherent multi-scale granularity of knowledge bases, leading to alignment gaps. We address this via a bidirectional alignment paradigm that decouples query granularity and decomposes knowledge scales, ensuring precise mapping between refined queries and optimal evidence.

3 Method

3.1 Preliminary

RAG can provide effective evidence support in long document intensive knowledge question answering. Its workflow is divided into three parts: query, retrieval and generation. We define the query to be denoted using $\mathcal{Q} = \{q_1, ..., q_i\}$, the retriever for the retrieval part to be denoted by $\mathcal{R}$, and the LLM for the generation part to be denoted by $\mathcal{M}$. Therefore, the retrieval process in RAG can be expressed by the following equation (1) :

$$\mathcal{D} = \mathcal{R}(q_i), q_i \in \mathcal{Q}, \tag{1}$$

where $\mathcal{D} = \{[d_0, ..., d_k]_0, ..., [d_0, ..., d_k]_i\}$, k denotes the top-k value set during retrieval. Next, both $\mathcal{D}$ and $\mathcal{Q}$ are input into $\mathcal{M}$. The process is as follows:

$$\mathcal{O} = \mathcal{M}(\mathcal{D} \parallel \mathcal{Q}), \tag{2}$$

here, $\mathcal{O}$ represents the output of the model. When addressing multi-hop questions, a single-pass retrieval based solely on surface-level similarity between the query and the corpus typically retrieves top-k documents that are insufficient in capturing the underlying granularity required for multi-hop reasoning.

3.2 Self-Correcting Decoupling Agent

Basically, when facing multi-hop questions, $\mathcal{Q}$ will be decoupled into several sets of sub-question sequences, which effectively degrades complex multi-granularity questions (Zhou et al., 2022). To ensure the quality of decomposition, we propose a Self-Correcting Decoupling Agent that not only breaks down complex queries but also self-corrects potential logical errors. First, the agent decouples the query:

$$\mathcal{Q}' = Decoupling\text{-}Agent(I_D, \mathcal{Q}) \tag{3}$$

here, $\mathcal{Q}' = [\{q_1^1, \cdots, q_1^n\}, \cdots, \{q_i^1, \cdots, q_i^n\}]$, $\mathcal{Q}$ is decoupled into n sub-questions, I_D represents

the prompt for driving the LLM to disassemble. Compared to using the original query q_i, the content retrieved based on the decomposed sub-questions contains more targeted and relevant information that directly corresponds to each sub-question.

However, errors in the decoupling process are often inevitable when the underlying model has limited logical reasoning capabilities. These errors may lead to the risk of error accumulation in the iterative pipeline (Yu et al., 2025). To address this, the agent employs a revision mechanism to review and revise the generated sub-question sequence. This mirrors how humans typically solve problems, by reviewing completed work for potential issues. Specifically, we design a constrained prompt (I_R) that guides model M to inspect the sub-question sequence and generate a revised version $\dot{\mathcal{Q}}'$ based on the initial output $\mathcal{Q}'$. This process can be represented as:

$$\dot{\mathcal{Q}}' = Reviser\text{-}Agent(I_R, \mathcal{Q}') \tag{4}$$

here, $\dot{\mathcal{Q}}' = [\{\dot{q}_1^1, \cdots, \dot{q}_1^n\}, \cdots, \{\dot{q}_i^1, \cdots, \dot{q}_i^n\}]$, where each set represents a revised sequence of sub-questions. These sub-question sequences are iteratively fed into the MGA-RAG framework to enable knowledge retrieval and multi-scale alignment.

3.3 Multi-Scale Granularity Alignment

To better align sub-questions with their retrieved documents, it is crucial to increase the proportion of key information within relevant content while reducing the ratio of noise within the effective information. To this end, we decompose the original retrieval documents into multiple scales, aiming to preserve key information as much as possible while filtering out noisy content. After retrieval, we employ a *Granularity Alignment Agent* to perform multi-scale alignment between single-granularity sub-queries and retrieved documents of entangled granularity. This multi-scale alignment process enables the model to better focus on key evidence during inference and answer generation. Specifically, for the retrieval documents obtained by q_i^n, the agent extracts multi-scale text segments using a multi-scale sliding window. We define multiple sets of text paragraphs of different scales as:

$$\mathcal{W}_{scale}(\mathcal{D}_i^n) = \{\mathcal{D}^{n,c_1}, \cdots, \mathcal{D}^{n,c_m}\}, \tag{5}$$

among them, $\mathcal{W}_{scale}$ is a multi-scale window for sliding and extracting text segments from $\mathcal{D}_i^n$, and $\mathcal{D}^{n,c} = \{{d_1}^{n,c}, ..., {d_N}^{n,c}\}, c \in \{c_1, c_m\}$, m is the preset number of multiple scales, and N represents the number of text segments. The rationale for employing multiple scales in subsequent alignment lies in their complementary nature: smaller scales can offer clearer, more focused evidence when the model is confused by broader contexts, while larger scales can provide more comprehensive information when finer-grained segments lack sufficient context. Next, we use the embedding model to vectorize $\mathcal{D}^{n,c}$ and a sub-question q':

$$\mathbf{H} = [Embed(\mathcal{D}^{n,c_1}), \ldots, Embed(\mathcal{D}^{n,c_m})] \in \mathbb{R}^{N \times d} \tag{6}$$

$$\mathbf{q} = Embed(q') \in \mathbb{R}^{1 \times y} \tag{7}$$

where, $\mathbf{H}$ and $\mathbf{q}$ respectively represent the embedding of multi-scale text segments and the embedding of a sub-question, and y represents the dimension size of the model output. In particular, we use the embedded $\mathbf{H}^c$ of a certain $\mathcal{D}^c$ to demonstrate the alignment process of similarity.

$${s_j}^c = \cos(\mathbf{q}, {\mathbf{h}_j}^c) = \frac{\mathbf{q} \cdot {\mathbf{h}_j}^c}{\| \mathbf{q} \| \cdot \| {\mathbf{h}_j}^c \|}, \; j = 1, \ldots, N \tag{8}$$

here, s_j denotes the similarity between the j-th passage and the sub-question q'. The complete similarity matrix can be represented as $\mathbf{s}^c = [{s_1}^c, {s_2}^c, \ldots, {s_N}^c]$. After, the top-k passages with the highest similarity scores are selected, forming a set of retrieved passages denoted as:

$$\dot{\mathcal{D}}^c = \{d_j | j \in \arg\mathrm{sort}(\mathbf{s}^c)[:k]\}. \tag{9}$$

The function $\arg\mathrm{sort}(*)$ is used to sort the similarity scores in $\mathbf{s}^c$ and extract the top-k passages. During this process, all text segments at the same scale are ranked, effectively reducing redundant information input to the model, emphasizing and aligning key content, and enabling models with limited capacity to better comprehend the provided context. To synthesize information from multiple granularities, a *Fusion Agent* is employed. Within this agent, a concurrent reasoning setting is utilized to obtain answers from various scales and fuse them. Specifically, the Fusion Agent executes parallel inference across all scales by feeding the aligned passages from each granularity level along with the corresponding prompt (I_m) into $\mathcal{M}$ to obtain scale-specific answers. This concurrent processing not only improves reasoning efficiency but

also mitigates hallucination issues by allowing the model to independently perceive aligned content at each scale. Subsequently, the Fusion Agent synthesizes these multi-granularity responses with fusion prompt (I_F) to produce the final fused answer ($\mathcal{A}_F$):

$$\mathcal{A}_F = Fusion\text{-}Agent(I_F, \{\mathcal{M}(I_m, \dot{\mathcal{D}}^c)\}_{c=c_1}^{c_m}). \tag{10}$$

By integrating reasoning results from multiple scales within the Fusion Agent, the approach effectively balances the strengths and limitations of each granularity level, mitigating the risks of over-alignment or under-alignment.

3.4 Final Answer

After performing multi-granularity decoupling and revision on the multi-hop question, a multi-scale alignment strategy is applied to guide the model in leveraging evidence more effectively throughout the RAG pipeline for solving complex multi-hop reasoning. This process is conducted iteratively until the final sub-question is answered. Once all sub-questions are completed, their corresponding answers are collected. Finally, the model synthesizes and answers the original multi-hop question based on the logically connected sub-question–answer pairs. Specifically, we employ a prompt I_O to instruct the model to generate the final answer grounded on the sequence of sub-question–answer pairs.

$$\mathcal{O}_i = \mathcal{M}(I_O, [\{q_i^1, a_i^1\}, \cdots, \{q_i^n, a_i^n\}]) \tag{11}$$

Based on a clear and granite-aligned QA sequence, the model can easily understand the internal logic of multi-hop questions, even for models with limited understanding capabilities.

4 Experimental Setup

Datasets. This study focuses on addressing multi-hop questions within long-document RAG tasks. To this end, we conduct experiments on three long-text datasets curated in LongBench (Bai et al., 2023, 2024): 2WikiMQA, HotpotQA, and MuSiQue, each containing 200 complex multi-hop questions. In our experiments, the texts from each dataset are embedded into a separate vector store, simulating the application of RAG in knowledge-intensive question answering over long documents. In typical RAG scenarios applied to domain-specific contexts, documents within the same domain are stored in a unified database. **Metrics.** As this study focuses not on retrieval itself but on the optimization following retrieval, we adopt two evaluation metrics: F1 and Accuracy. The F1 score measures the exact match between the generated answer and the reference answer, based on character-level overlap. Accuracy (ACC), on the other hand, evaluates whether the generated answer conveys the same meaning as the reference answer, determined via model-based semantic judgment. The specific computation methods for these two metrics on a per-sample basis are as follows:

$$\mathrm{F1} = \begin{cases} 0, & \text{if} |P \cap G| = 0 \\ \frac{2 \cdot |P \cap G|}{|P| + |G|}, & \text{otherwise} \end{cases} \tag{12}$$

$$\mathrm{ACC} = \mathbb{1}[\mathrm{P} \approx \mathrm{G}] \tag{13}$$

here, P and G denote the final predicted answer from RAG and the ground-truth answer, respectively. The indicator function ($\mathbb{1}$) returns 1 when P and G are considered semantically equivalent. All metrics are reported as percentages.

Baselines. We select four categories of baseline methods for comparison: (1) Graph-RAG (Edge et al., 2024), LightRAG (Guo et al., 2024) and Tree-RAG (Tao et al., 2025), which enhance RAG through knowledge base structure optimization; (2) Iter-DRAG (Yue et al., 2025), which performs iterative retrieval; (3) Iter-RetGen (Shao et al., 2023), which combines CoT prompting with iterative retrieval; and (4) Naive RAG. In the comparative methods, excluding approaches such as Graph-RAG, Light-RAG, and Tree-RAG that involve additional construction of structured databases, all other methods perform retrieval using a chunk size of 512 tokens and a top-k setting of 5. For MGA-RAG, we configure three scales, with each scale corresponding to $[150, 250, 350]$ characters.

5 Main Results

As shown in Table 1, MGA-RAG consistently achieves superior performance across both F1 and Accuracy metrics. Among the evaluated datasets, MuSiQue emerges as the most challenging, while 2WikiMQA is relatively simpler. Nevertheless, MGA-RAG delivers the best results even on MuSiQue. Under the 7B setting, our method surpasses the second-best baseline in terms of Accuracy by 6.92% on 2WikiMQA, 2.83% on MuSiQue, and 3.33% on HotpotQA. Under the 14B setting, MGA-RAG achieves improvements of 11.5% on

Table 1: Performance on 2WikiMQA, MuSiQue, and HotpotQA datasets under different models.

Method	Qwen2.5-7B						Qwen2.5-14B					
	2WikiMQA		MuSiQue		HotpotQA		2WikiMQA		MuSiQue		HotpotQA	
	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC
Naive-RAG	41.81	44.00	20.94	23.67	50.45	58.33	46.99	51.50	25.20	31.17	56.51	67.00
Structured RAG												
Graph-RAG (Edge et al., 2024)	21.73	25.33	7.99	9.5	22.84	28.33	3.35	6.8	22.67	13.0	13.55	23.5
Light-RAG (Guo et al., 2024)	47.16	48.0	17.58	21.67	38.37	46.67	48.77	57.5	15.39	23.0	33.14	49.0
Tree-RAG (Tao et al., 2025)	53.66	56.83	13.81	16.83	54.40	64.5	59.58	65.5	18.51	29.0	55.04	68.5
Iterative RAG												
Iter-DRAG (Yue et al., 2025)	23.77	53.50	20.36	29.33	34.39	63.70	25.67	61.67	20.85	37.17	38.24	69.33
Iter-RetGen (Shao et al., 2023)	50.52	54.00	34.53	40.00	56.27	65.50	54.24	59.50	41.89	46.00	**60.26**	**71.50**
Ours	**58.32**	**63.17**	**39.92**	**42.83**	**57.11**	**68.83**	**60.39**	**73.17**	**43.53**	**51.83**	58.81	70.00

Table 2: Ablation of different modes on three datasets, "Nothing" refers to vanilla RAG.

Model	Mode	2WikiMQA		MuSiQue		HotpotQA	
		F1	ACC	F1	ACC	F1	ACC
Qwen2.5-7B	Nothing	41.81	44.00	20.94	23.67	50.45	58.33
	Decoupler	55.59	60.50	34.82	36.33	51.21	60.67
	MGA-RAG	**58.32**	**63.17**	**39.92**	**42.83**	**57.11**	**68.83**
Qwen2.5-14B	Nothing	46.99	51.50	25.20	31.17	56.51	67.00
	Decoupler	**65.95**	70.83	**45.33**	50.83	57.87	67.17
	MGA-RAG	60.39	**73.17**	43.53	**51.83**	**58.81**	**70.00**

Table 3: Comparison between MGA-RAG without RA and full MGA-RAG. Each cell reports F1 / ACC scores.

Dataset	Model	w/o RA	MGA-RAG
2WikiMQA	Qwen2.5-7B	54.60 / 59.00	**58.32 / 63.17**
	Qwen2.5-14B	66.03 / 71.67	60.39 / **73.17**
MuSiQue	Qwen2.5-7B	36.17 / 39.17	**39.92 / 42.83**
	Qwen2.5-14B	45.75 / 53.00	43.53 / 51.83
HotpotQA	Qwen2.5-7B	54.58 / 64.00	**57.11 / 68.83**
	Qwen2.5-14B	54.46 / 67.67	**58.81 / 70.00**

Table 4: Performance under different Chunking mode.

Model	Mode	2WikiMQA		MuSiQue		HotpotQA	
		F1	Acc	F1	Acc	F1	Acc
Qwen2.5-7B	(512, 5)	58.3	61.2	39.6	40.0	57.1	61.3
	(512, 10)	**60.0**	**63.3**	**40.2**	**41.2**	**59.9**	64.7
	(1024, 10)	57.2	62.5	35.2	39.3	59.8	**68.3**
	(1024, 5)	58.0	63.2	35.2	38.3	58.3	67.0
Qwen2.5-14B	(512, 5)	60.4	72.2	44.4	47.2	57.5	63.7
	(512, 10)	**68.8**	74.0	**46.1**	48.7	61.1	66.2
	(1024, 10)	67.6	**75.0**	41.9	**50.5**	**61.5**	73.0
	(1024, 5)	66.7	74.0	38.7	45.7	60.1	**73.2**

Table 5: Accuracy with different model sizes, using Qwen2.5-7B or Qwen2.5-14B as the decoupler.

Dataset	Decouple Model	1.5B	7B	14B
2WikiMQA	Qwen2.5-7B	47.3	63.2	66.0
	Qwen2.5-14B	**51.2**	**64.3**	**73.2**
MuSiQue	Qwen2.5-7B	32.0	42.8	51.8
	Qwen2.5-14B	**34.5**	**44.2**	**51.8**
HotpotQA	Qwen2.5-7B	53.8	**68.8**	68.8
	Qwen2.5-14B	**57.3**	66.0	**70.0**

2WikiMQA and 5.83% on MuSiQue. However, on HotpotQA, Iter-RetGen slightly outperforms MGA-RAG with the larger model, likely due to its enhanced CoT prompting, which becomes more effective with increased model capacity. Although LightRAG leverages a graph-structured knowledge base to facilitate multi-hop reasoning, it requires the model to jointly interpret complex queries and traverse logically entangled graph structures, thereby complicating fine-grained alignment. Moreover, Tree-RAG tends to retrieve and input all potentially relevant documents into the model, which becomes highly inefficient when dealing with longer documents. This results in suboptimal performance, as the model's attention capacity is inherently limited and requires more precise localization of relevant passages. In contrast, MGA-RAG aligns retrieved documents at multiple scales guided by sub-question decomposition, enabling reasoning at suitable granularities while reducing redundancy. Moreover, F1 metric indicate that MGA-RAG better captures query intent, leading to more relevant and faithful responses.

Overall, MGA-RAG demonstrates strong performance and competitiveness across all evaluated methods. This is attributed to its effective granularity decoupling for complex multi-hop questions, coupled with explicit intervention to ensure the correctness and coherence of the decomposition. Additionally, the multi-scale alignment based on the sub-question sequence allows the model to concentrate more precisely on relevant evidence, avoiding the noise that often leads to persistent hallucina-

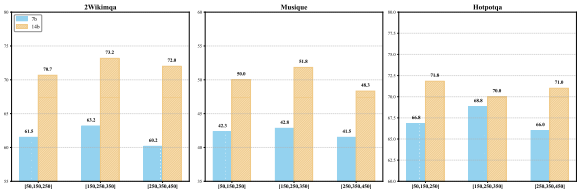

Figure 3: Scale study of scale combinations. The scale combinations include $[50, 150, 250]$, $[150, 250, 350]$, and $[250, 350, 450]$.

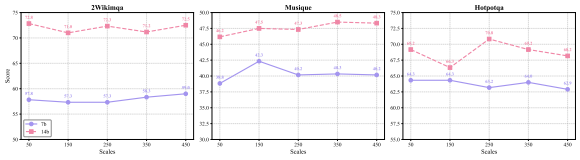

Figure 4: The single-scale granularity study spans from 50 to 450 characters.

tions.

6 Discussions

We further conduct a series of analytical experiments, including ablation study, retrieval content study, scales study, customization study, and case study. The ablation study primarily investigates the effectiveness of the proposed method, while the other analyses explore the impact of different parameter settings on the overall reasoning pipeline.

6.1 Ablation Study

As shown in Table 2, the performance of MGA-RAG consistently improves with the addition of each functional component, particularly for smaller models. This improvement is largely attributed to the multi-scale granularity alignment, which increases the proportion of key information across different scales, thereby facilitating more precise evidence localization and alignment by the model. For the F1 metric, however, a slight decline is observed in the 14B model on the 2WikiMQA and MuSiQue datasets. This is likely because more capable models tend to generate partially explanatory outputs, resulting in additional irrelevant characters and thus affecting the F1 score. Moreover, as shown in Table 3, the Reviser-Agent (RA) proves more effective in assisting the decomposition of complex queries when applied to the 7B model. This highlights the necessity of RA, even though a performance drop is observed with the 14B model on the MuSiQue dataset. Therefore, for models with weaker capabilities, the use of a RA is essential to ensure the proper decoupling of multi-granularity queries. In contrast, the multi-scale granularity alignment process proves effective across both strong and weak models, enhancing performance regardless of model capacity.

6.2 Retrieval Content Study

From Table 4, which reports the ChunkSize and TopK can observe that increasing TopK from 5 to 10 consistently enhances F1 and Accuracy in most settings. For instance, with a 14B model on 2WikiMQA, F1 rises from 60.39 to 68.81 as TopK increases from 5 to 10 (ChunkSize = 512). Smaller chunks generally outperform larger ones, with higher TopK, likely due to more precise retrieval and greater coverage of relevant information. Conversely, large chunks with low TopK yield suboptimal results due to reduced evidence coverage. These findings suggest that smaller chunks with higher TopK optimize precision-recall balance in multi-hop QA, aligning with the proposed multi-scale granularity alignment, where finer initial granularity ensures a more stable alignment process.

6.3 Scales Study

To investigate the impact of different scale configurations in MGA-RAG, we examine the relationship between scale combinations and model performance. Accuracy is reported across three multi-hop QA datasets using three scale settings: [50,150,250], [150,250,350], and [250,350,450]. As shown in Figure 3, the [150,250,350] configuration consistently yields the most stable performance across datasets. This is because it balances fine- and coarse-grained textual segments, allowing the fusion answer to benefit from complementary information. Moreover, as shown in Figure 4, the performance of MGA-RAG exhibits significant fluctuations across datasets as single scale increases. This indicates that a single scale alone struggles to provide a comprehensive view and fails to match the performance of combined scales. These results demonstrate that integrating multiple scales leads to more balanced reasoning performance, highlighting the necessity of multi-scale fusion.

6.4 Customization Study

This study evaluates the performance of RAG systems using distinct models for query decomposition and multi-scale granularity alignment. Table 5 shows that a 1.5B parameter model for

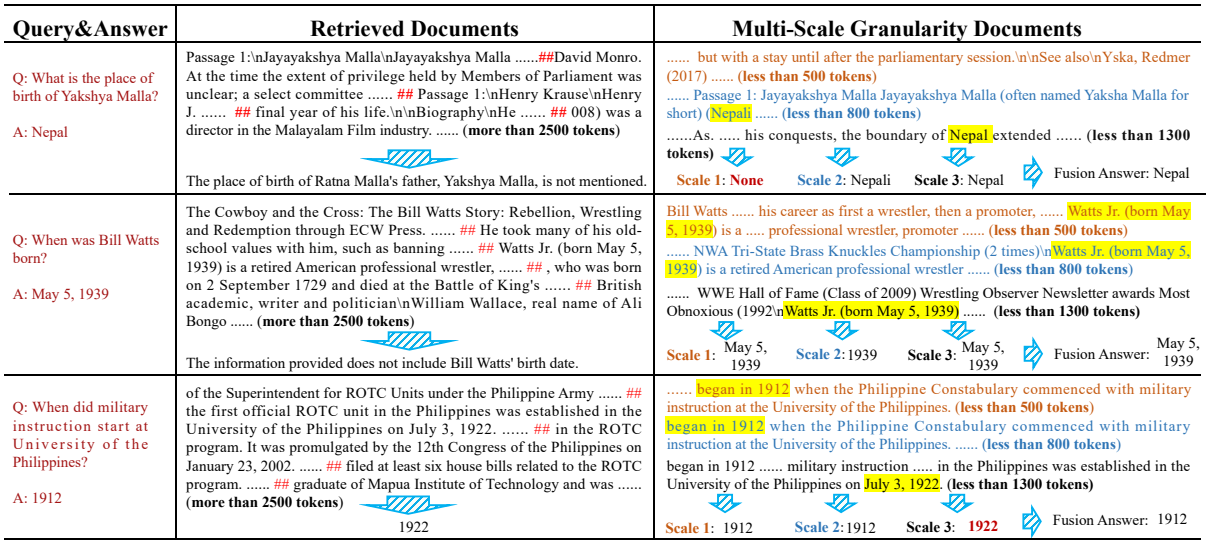

Figure 5: Case study. The figure presents inference examples of a sub-question from the 2WikiMQA, HotpotQA, and MuSiQue datasets (top to bottom) using Qwen2.5-7B. While directly retrieved documents are entangled and long, the multi-scale granularity passages contain fewer tokens while preserving key evidence, enabling the model to better locate relevant information.

alignment on decomposed sub-queries significantly outperforms Naive-RAG. Notably, Qwen2.5-1.5B achieves an Accuracy of 51.17%, which is already comparable to that of Qwen2.5-14B (Accuracy: 51.50%). The results highlight the synergy of combining smaller and larger models to balance computational efficiency and performance. Additionally, increasing model scale reveals a clear scaling law effect (Fang et al., 2024), demonstrating the framework's adaptability for customized RAG deployments across varied computational constraints.

6.5 Case Study

As shown in Figure 5, directly reasoning over retrieved documents often introduces interference from redundant knowledge, leading to hallucinations or low-confidence answers (e.g., when the model deems no evidence available). In contrast, aligning sub-questions with multi-scale granularity passages effectively reduces noise while preserving key evidence. Even when one scale produces a hallucinated response, such as the largest scale in the third example, the complementary information from other scales enables the model to generate a correct fused answer. Moreover, when the finest granularity lacks sufficient context, larger scales still provide reliable support.

7 Conclusion

In this paper, we propose MGA-RAG, a novel RAG framework for handling multi-hop queries in long-document question answering. MGA-RAG decouples complex queries into single-granularity sub-questions and aligns them with retrieved content using a multi-scale strategy. This approach reduces the influence of irrelevant information and enables sub-questions to more effectively match key evidence. Additionally, we introduce a Reviser-Agent to audit the decoupling process, mitigating error accumulation in the RAG pipeline, particularly for models with limited capacity. Comprehensive experiments demonstrate the superiority and effectiveness of MGA-RAG across multiple datasets.

Limitations

Despite the significant performance superiority demonstrated by MGA-RAG over competing methods, it remains subject to certain limitations. The first concerns computational cost, which is an inherent challenge within the iterative RAG paradigm arising from the necessity of multiple LLM interactions. MGA-RAG is not exempt from this overhead. The second limitation relates to semantic continuity within the chunking strategy. Since MGA-RAG employs fixed-size text segmentation, there is a potential risk that this rigid approach may lead to semantic incoherence. Nevertheless, these limitations delineate clear avenues for future research and optimization.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, and 1 others. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. *arXiv preprint arXiv:2504.18243*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling laws for dense retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1339–1349.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

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. *arXiv preprint arXiv:2305.15294*.

Weihang Su, Qingyao Ai, Xiangsheng Li, Jia Chen, Yiqun Liu, Xiaolong Wu, and Shengluan Hou. 2024. Wikiformer: Pre-training with structured information of wikipedia for ad-hoc retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19026–19034.

Wenyu Tao, Xiaofen Xing, Yirong Chen, Linyi Huang, and Xiangmin Xu. 2025. Treerag: Unleashing the power of hierarchical storage for enhanced knowledge retrieval in long documents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 356–371.

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)*, pages 10014–10037.

Shiqi Wang, Yeqin Zhang, and Cam-Tu Nguyen. 2024. Mitigating the impact of false negative in dense retrieval with contrastive confidence regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19171–19179.

Xiaohan Yu, Pu Jian, and Chong Chen. 2025. Tablerag: A retrieval augmented generation framework for heterogeneous document reasoning. *arXiv preprint arXiv:2506.10380*.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. 2024. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Jianbing Shen, Guodong Long, Can Xu, and Daxin Jiang. 2024. Fine-grained distillation for long document retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19732–19740.

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. *arXiv preprint arXiv:2408.04259*.

Algorithm 1 MGA-RAG Framework

Input: Multi-hop query $\mathcal{Q}$
Output: Final answer $\mathcal{O}$

Step 1: Query Decoupling
$\mathcal{Q}' \leftarrow Decoupler(I_D, \mathcal{Q})$ // Decompose $\mathcal{Q}$ into sub-questions
$\dot{\mathcal{Q}}' \leftarrow Reviser - Agent(I_R, \mathcal{Q}')$ // Revise $\mathcal{Q}'$ into better sub-questions
// $\dot{\mathcal{Q}}' = [\{\dot{q}_1^1, \cdots, \dot{q}_1^n\}, \cdots, \{\dot{q}_i^1, \cdots, \dot{q}_i^n\}]$
Step 2: Iterative Multi-hop Processing
$\text{QAPairs} \leftarrow \emptyset$ // Initialize question-answer pairs
for each sub-question sequence $\{\dot{q}_i^1, \cdots, \dot{q}_i^n\} \in \dot{\mathcal{Q}}'$ **do**
  for each sub-question $\dot{q}_i^j$ in the sequence **do**
    $\mathcal{D}_i^j \leftarrow \mathcal{R}(\dot{q}_i^j)$ // Retrieve documents for $\dot{q}_i^j$
    Step 2.1: Multi-scale Granularity Alignment
    $\text{ScaleAnswers} \leftarrow \emptyset$
    for each granularity scale $s \in \text{Scales}$ **do**
      $\text{Segments} \leftarrow \mathcal{W}_{scale}(\mathcal{D}_i^j, s)$ // Segment documents using scale s
      $\text{AlignedPassages} \leftarrow \text{AlignAndRank}(\text{Segments}, \dot{q}_i^j, \text{TopK})$
      $a_s \leftarrow \mathcal{M}(I_m, \text{AlignedPassages}, \dot{q}_i^j)$ // Generate answer for scale s
      $\text{ScaleAnswers} \leftarrow \text{ScaleAnswers} \cup \{a_s\}$
    end for
    Step 2.2: Multi-scale Answer Fusion
    $a_i^j \leftarrow \mathcal{M}(I_F, \text{ScaleAnswers}, \dot{q}_i^j)$ // Fuse answers across scales
    $\text{QAPairs} \leftarrow \text{QAPairs} \cup \{(\dot{q}_i^j, a_i^j)\}$
  end for
end for
Step 3: Final Answer Generation
$\mathcal{O} \leftarrow \mathcal{M}(I_O, \text{QAPairs})$ // Generate final answer from QA pairs
Return $\mathcal{O}$

Appendix

A Appendix: Pseudocode

According to the methodology section, we construct the corresponding pseudocode to provide a clearer understanding of the simple and effective pipeline of MGA-RAG. Specifically, the pseudocode is presented in Algorithm 1.

B Experimental Environment and Infrastructure

To better simulate RAG applications under privatized deployment scenarios, which typically involve querying user-specific knowledge bases, we locally deployed Qwen2.5-7B and Qwen2.5-14B. Both models are deployment-friendly and support long-context inference of up to 32K tokens. Specifically, we adopted the latest official release of the vLLM framework for local deployment [1]. All experiments were conducted on a single server equipped with four NVIDIA RTX 4090 GPUs. The PyTorch version used was aligned with the vLLM version to ensure compatibility. During inference, all models were configured with a temperature of 0 to ensure deterministic outputs. The maximum output length was set to 512 tokens, and all other decoding hyperparameters were left at their default values. In addition, the embedding model was also deployed locally using vLLM, with a maximum input length of 8192 tokens [2]. For vector storage, we employed the Chroma vector database from the LangChain framework to manage and index the external knowledge corpus [3].

B.1 Implementation Details

To ensure a fair comparison, we implemented all methods using the same framework and infrastructure. Specifically, we utilized the vLLM framework for all methods, including MGA-RAG,

[1] https://github.com/vllm-project/vllm

[2] https://huggingface.co/BAAI/bge-m3

[3] https://www.langchain.com/

Graph-RAG, Light-RAG, Tree-RAG, Iter-DRAG, Iter-RetGen, and Naive-RAG. For Graph-RAG and Light-RAG, we use the unified Light-RAG repository, with Graph-RAG operating in global mode and Light-RAG in hybrid mode. Parameters such as chunk size and top-k are set to their default values [4]. For Tree-RAG, we set top-k to 3, as Tree-RAG indexes all relevant nodes, which imposes a substantial context burden. For Iter-DRAG and Iter-RetGen, we conduct three iterations and use the same vector store, chunk size, and top-k settings as MGA-RAG.

C Prompt Design

RAG is an LLM-based methodology that heavily relies on prompt engineering, where the design and quality of prompts directly impact the performance of the RAG pipeline.

C.1 Prompt of MGA-RAG

Our method incorporates the following key prompts: I_D for the granularity decomposition of multi-hop questions, I_m for reasoning on each scale after multi-scale alignment, I_F for fusing the results from different scales, and I_O for generating the final output. The detailed prompt designs are illustrated in Figures 1-5.

C.2 Prompt of other methods

For the other methods, we adopt their original prompt formats for implementing the RAG pipeline. Specifically, for Graph-RAG, Light-RAG, and Tree-RAG, we incorporate an additional prompt for final answer refinement, as these methods tend to generate overly lengthy responses, which can adversely affect the F1 score. Accordingly, we define the final answer refinement prompt (Prompt I_{Syn}) as illustrated in Figure 11.

C.2.1 Prompt of Accuracy metric

Our primary focus is on optimizing the retrieved documents to enable more effective generation by LLMs. To evaluate the performance of different methods, we employ both F1 and Accuracy metrics. In particular, Accuracy is assessed using an LLM-based evaluation, which captures cases where the generated answer is semantically correct but does not exactly match the reference at the character level. This metric directly reflects whether the output fulfills the user's intent. Accordingly, we design a unified prompt (Prompt I_{Sim}), as shown in Figure 12, to guide the LLM in evaluating Accuracy across all methods.

[4] https://github.com/HKUDS/LightRAG

Prompt I_D: Sub-question Decomposition Instruction

You are a helpful assistant that generates up to 5 sub-questions to break down an input question into coherent, independently answerable parts.

- Sub-questions must lead to the final answer.
- Use `$1` for intermediate answers, `$ANSWER` for the final answer.
- Output valid JSON with `question` and `subquestions` fields.
- Each sub-question is formatted as: `["question text", "placeholder"]`.
- Please make sure to break down the relationships in the question completely.

Example 1:

```
{"question": "Who is the mother of the director of film X?", "subquestions": [["Who is the director of film X?", "$1"], ["Who is the mother of $1?", "$ANSWER"]]}
```

Example 2:

```
{"question": "Do directors of films A and B have the same nationality?", "subquestions": [["Who is the director of film A?", "$1"], ["Who is the director of film B?", "$2"], ["Do $1 and $2 have the same nationality?", "$ANSWER"]]}
```

Output Format:

```
{"question": "<input question>",
"subquestions": [["<sub-question>", "<$1 or $ANSWER>"], ...]
}
Input question:
```

Figure 6: Prompt I_D used for decomposing a complex input question into structured sub-questions.

Prompt I_m: Alignment and Response

You are provided with the following retrieved knowledge and a question. Please answer the question based on the retrieved knowledge.

- **Only output the answer, no other content**.

Retrieved Knowledge:

```
{context}
```

Question:

```
{query}
```

Instruction (repeated):

```
Only output the answer, no other content.
Answer:
```

Figure 7: Prompt I_m drives the alignment and response of the granularity at each scale.

Prompt I_F: Fusion Across All Scales

You are given a question and answers generated from different scale documents. Please synthesize a final accurate and complete answer.

- **Only output the answer, no other content**.

Question:
`{query}`
Original Documents:
`{"\n\n".join(documents)}`
Answers from different scale documents:
```
1. {answers[0]}
2. {answers[1]}
3. {answers[2]}
```
Final Instruction (repeated):
```
Only output the answer, no other content.
Final Answer (be concise and correct):
```

Figure 8: Prompt I_F used to integrate the alignment results at each scale.

Prompt I_O: Final Answer Generation

You are given a main query along with several sub-question and answer pairs that were generated by logically decomposing the main query. Your task is to synthesize an accurate and complete answer to the main query based on the information provided in the sub-questions and their answers.

- Pay special attention to the intent and semantic structure of the main query — determine whether it concerns time, location, tasks, or events.
- Carefully distinguish temporal relationships — do not confuse sequences or time references across sub-answers.
- Ensure your final answer aligns directly with the main query's intent.
- Avoid including irrelevant or misleading details.
- **Only provide the answer — no extra explanations or justifications**.

[Sub-question and answer pairs]:
`{sub_qa_pairs}`
[Main query]:
`{query}`
Final Instruction:
```
Only output the answer, no other content.
Final Answer (to the original question):
```

Figure 9: Prompt I_O for synthesizing an accurate final answer based on a series of sub-question and answer pairs aligned with a decomposed main query.

Prompt I_R: Sub-question Revision Instruction

You are a subquestion reviser. You're given a JSON object with:

- a `"question"`: the original question.
- a `"subquestions"` list: a step-by-step decomposition used to answer the question.

Your job:

1. Inspect the first subquestion.
2. If it misses key entities or fails to preserve the core meaning, mark it as invalid.
3. If invalid, regenerate all subquestions to:
    - Retain key content from the original question in the first subquestion.
    - Ensure each subquestion is answerable via knowledge base.
    - Use logical placeholders like `$1`, `$2`, etc.
4. Ensure the question is decomposed — not rephrased.
5. Output the final JSON in the original format.

You must revise subquestions when:

- Key named entities are missing in the first subquestion.
- The subquestion is too generic or not semantically aligned.
- The subquestion is a restatement, not a decomposition.

Example (Invalid → Revised):

```
Input: {"question": "Where does the director of film Wine Of Morning work at?", "subquestions":
[["What is the title of the film?", "$1"], ["Who is the director of $1?", "$2"], ["Where does
$2 work at?", "$ANSWER"]]}
Output: {"question": "Where does the director of film Wine Of Morning work at?", "subquestions":
[["Who is the director of the film Wine Of Morning?", "$1"], ["Where does $1 work?", "$ANSWER"]]}
```

Now revise this JSON (preserving full structure):

Figure 10: Prompt I_R for detecting and revising subquestions that fail to preserve the semantics or key entities of the original question.

Prompt I_{Syn}: Answer Synthesis

Please organize the following provisional answer according to the question to form the final answer.

- The answer may be entities such as names of people, organizations, places, relationships, times, dates, etc.
- The answer may also be nothing, if not found.
- **Only output the final answer, no other content.**

Provisional Answer:

`{provisional_answer}`

Question:

`{question}`

Final Answer (only the final answer, no other content):

Figure 11: Prompt I_{Syn} for refining the provisional answer into a final concise response.

Prompt I_{Sim}: Answer Similarity Judgment

Please determine whether Answer A and Answer B involve similar elements, such as:

- the same event,
- the same person,
- the same time,
- the same number, etc.

If they are similar, return `"Yes"`; otherwise, return `"No"`.

Answer A:

`{A}`

Answer B:

`{B}`

Output (Only "Yes" or "No"):

Figure 12: Prompt I_{Sim} for binary similarity judgment between two answers.