ReCore: Reflective Cooperative Reasoning Framework for Knowledge-Intensive Question Answering Tasks

Anonymous ACL submission

Abstract

Knowledge-intensive question-answering (QA) aim to collect dispersed knowledge from a large number of documents related to the question and derive answers through reasoning. However, traditional methods often suffer from information loss and difficulty in establishing logical connections across documents, making reasoning challenging. To address these issues, we propose Reflective Cooperative Reasoning (ReCore), a framework that combines Role-Based Multi-Perspective Collaboration with Reflective Knowledge Refinement.The former dynamically generates multiple roles based on the task and integrates their insights through a voting mechanism to identify the optimal structuring strategy. This approach facilitates the organized, concise extraction of relevant information and helps reveal relationships among fragmented content from multiple perspectives. The latter introduces a reflective construction mechanism to perform intensive reading over related documents, providing richer background knowledge for the task and enabling further refinement, segmentation, and simplification of useful content. This process mitigates information loss and enhances reasoning effectiveness. Experiments demonstrate that ReCore outperforms baseline models, delivering a 12-point boost in average answer quality (LLM Score) and a 27-percentage-point improvement in Exact Match (EM). It excels in complex tasks including spotlight locating, comparison, clustering, and chain-of-thought reasoning—all evaluated on the Loong benchmark.

1 Introduction

With the outstanding performance of large language models (LLMs) across various scenarios (Achiam et al., 2023; Yang et al., 2024a), they are widely expected to be applied to cross-document QA tasks, such as legal analysis and financial report analysis. However, due to the lack of deep domain-specific knowledge, LLMs still struggle to generate factually accurate content (i.e., "hallucinations") in tasks that require precise factual answers (Huang et al., 2025; Sui et al., 2024). To address this limitation, the Retrieval-Augmented Generation (RAG) framework (Lewis et al., 2020) has been proposed, which enhances the ability of LLMs to answer questions by incorporating external knowledge. However, traditional RAG methods process knowledge through simple text chunking (Ma et al., 2023; Li et al., 2024a), which faces the issue that document-level knowledge can be too lengthy to fit within the context window of the language model, leading to the inability of large models to consider all the knowledge, as longer documents tend to overlook key information. Additionally, the relevant information required to solve

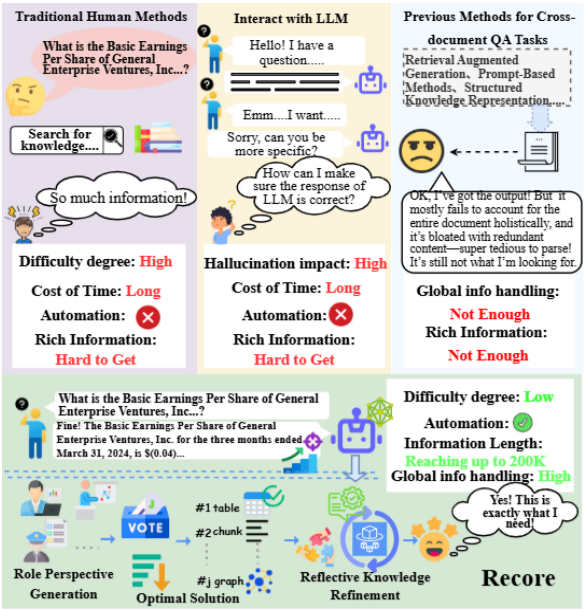

Figure 1: Illustration of core challenges in cross-document QA and the ReCore framework. ReCore combines two mechanisms to reduce info loss and enhance reasoning: Role-Based Multi-Perspective Collaboration and Reflective Knowledge Refinement.

these tasks is often dispersed across multiple documents, making it challenging for the model to efficiently integrate and utilize the scattered information for reasoning (Yang et al., 2024b; Kuratov et al., 2024; Liu et al., 2023a). For example, in legal case analysis, relevant laws, judicial opinions, and similar case situations must be extracted from a large number of samples, requiring LLMs to perform precise comparisons and comprehensive analysis. These sample documents often contain a lot of noise, making it difficult for the model to recognize the relationships between pieces of information and conduct accurate knowledge reasoning.

Many studies (Wei et al., 2022; Li et al., 2024c; Mai et al., 2023) have shown that complex tasks are better handled by treating LLMs as analogous to the human brain, drawing on human reasoning processes. Human conclusions, however, are rarely based on simple reading alone; instead, information is often summarized into structured formats to reduce complexity and improve judgment accuracy (Johnson-Laird, 1983; Paivio, 1990; Sweller, 1988; Chandler and Sweller, 1991). This process is frequently accompanied by self-reflection and verification. Similarly, LLMs possess the ability to reflect on their reasoning, identify errors, and correct them, akin to human judgment processes (Shah et al., 2025). Cognitive fit theory further suggests that humans prefer different types of structured knowledge for different tasks, such as tables for statistical analysis and graphs for long-chain reasoning (Vessey, 1991; Umanath and Vessey, 1994). Recent studies indicate that LLMs are also adept at constructing various knowledge structures (Li et al., 2023; Jain et al., 2024).However, these approaches are limited by their reliance on a single strategy, which hinders their effectiveness in handling complex, real-world information. The reliance on one method leads to difficulties in establishing logical connections and causes information loss across documents. This makes reasoning challenging, particularly when dealing with fragmented content. While LLMs can generate diverse knowledge structures and reflect on their reasoning, their inability to dynamically combine multiple strategies restricts their ability to effectively process and integrate dispersed information, which is crucial for more complex tasks.

For these reasons, we propose ReCore, which employs human-like thought processes to address the above challenges. The main contributions of our work are as follows: (1) Role-Based Multi-Perspective Collaboration: we dynamically generate roles based on the task and integrate insights from multiple perspectives through a voting mechanism to determine the optimal structuring strategy. This aids in concise information extraction and reveals relationships among fragmented content. (2) Reflective Knowledge Refinement: we introduce a reflective knowledge construction mechanism that, combined with the multi-role perspectives, enhances the diversity of results. This process enables further refinement, segmentation, and simplification of useful content, reducing information loss and improving reasoning effectiveness. (3) The experimental results show that compared to baseline methods, this method achieves substantial improvement in knowledge-intensive QA tasks such as spotlight locating, comparison, clustering, and chain of reasoning.

2 Related Work

2.1 Retrieval Augmented Generation

Due to the tendency of knowledge-intensive QA tasks to generate hallucinations in cross-domain document question answering, RAG (Lewis et al., 2020) has been proposed to address the issues of LLMs lacking domain-specific knowledge and performing poorly in answering tasks that require precise factual responses. Initially, the RAG aimed to accomplish the task in a pipeline approach by first providing the LLMs with external knowledge through a retrieval component and then combining it with its powerful generative capabilities (Karpukhin et al., 2020; Borgeaud et al., 2022). Later, considering error propagation, the REALM approach (Guu et al., 2020) optimized this process end-to-end by either jointly training the retrieval and generation processes or computing semantic similarity through word embeddings (Gao et al., 2023) to improve the model's accurate use of external knowledge. These studies have made significant progress. However, as the knowledge base expands, the application scenarios of tasks become increasingly complex. This complexity leads to challenges in managing global information. Additionally, dividing content into chunks to address the issue of excessively long documents results in segments that are too isolated from each other (Izacard and Grave, 2020; Mao et al., 2020).

2.2 Structured Knowledge Representation

To streamline the document-level information in knowledge-intensive QA tasks and better assist LLMs in capturing global information, the heterogeneous retrieval framework utilizes various retrieval strategies to decouple text information into more fine-grained, compressed knowledge representations (Yang et al., 2025), enabling the utilization of knowledge according to the specific requirements of the task. Additionally, organizing knowledge into various structures can elucidate information relationships. For instance, graph structures facilitate information-to-information associations (Edge et al., 2024; Peng et al., 2024), while querying subgraphs or extracting relational triples serve as external knowledge sources. TAPAS (Herzig et al., 2020) and TAPEX (Liu et al., 2021) utilize table structures, significantly enhancing the model's reasoning capabilities. However, exclusive reliance on a single structural form may lead to information loss and hinder generalization in open-domain QA scenarios. Consequently, adaptive routers have been proposed to automatically select structured forms (Li et al., 2024b; Shuster et al., 2022). Despite this, such methods necessitate training data and can introduce complexity, potentially impacting model generation. Thus, dynamically determining the optimal knowledge representation remains a significant challenge.

2.3 Prompt-Based Methods

Language model prompting methods have been shown to exhibit good adaptability in several knowledge-intensive QA tasks as an effective way to infer with zero-shot or few-shot learning. Liu et al. (Liu et al., 2023b) showed that suitable prompts allow LLMs to be adapted to new tasks without extensive fine-tuning. However, while language model prompting approaches perform well on simple tasks, there are still challenges regarding the validity and accuracy of structured knowledge. Meng et al. (Meng et al., 2022) and Ye et al. (Ye et al., 2022) pointed out that while synthetic data generated by prompting language models improves task performance, its application to complex tasks still requires more validation. In particular, how to ensure that language models can handle tasks with a high degree of structure without relying on extensive fine-tuning remains a key issue in current research.

While significant advances have been made in research on knowledge-intensive QA tasks, gaps remain in decision question solving, knowledge representation, and utilization. The following introduces ReCore, a unified framework to address these challenges.

3 Methodology

3.1 Task Definition

This study focuses on knowledge-intensive question answering, where given a question $question$ and a set of documents $Docs$, the goal is to retrieve relevant information across $Docs$ and derive the answer A. The set Docs = $\{Doc(i)\}_{i=1}^{n}$ can contain more than 20 documents, with the total token count reaching up to 200K. A key challenge in these tasks is that useful information is dispersed across lengthy and redundant $Docs$, where each $Doc(i)$ may contain critical clues. Omitting any $Doc(i)$ could lead to an incorrect A. These tasks can be classified into four main scenarios: (1) Spotlight Locating: Identifying a single key piece of information across $Docs$, such as locating earnings data in multiple financial reports. (2) Comparison: Extracting and comparing key information, e.g., comparing legal reasoning across similar rulings. (3) Clustering: Grouping related information, such as summarizing research topics from academic papers. (4) Chain of Reasoning: Deriving answers to multi-step reasoning questions, like tracing the evolution of a legal judgment across multiple cases. These tasks require not only effective information extraction but also complex reasoning and data integration across $Docs$, a challenge that traditional methods often fail to address.

3.2 Role-Based Multi-Perspective Collaboration

Existing methods often rely on manual data annotation or unsupervised synthesis of large preference sets for training a judgment model to find the best structuring strategy for a given question (Li et al., 2024b; Zheng et al., 2023). However, these approaches have several issues. Firstly, manually annotated data are subjective and lack diversity. Secondly, poor-quality annotated data can lead to increased judgment errors in the trained model. Similarly, synthetic data can introduce noise, affecting the final results.

In contrast, we propose a solution inspired by human decision-making. This method simulates collaborative brainstorming where multiple expert

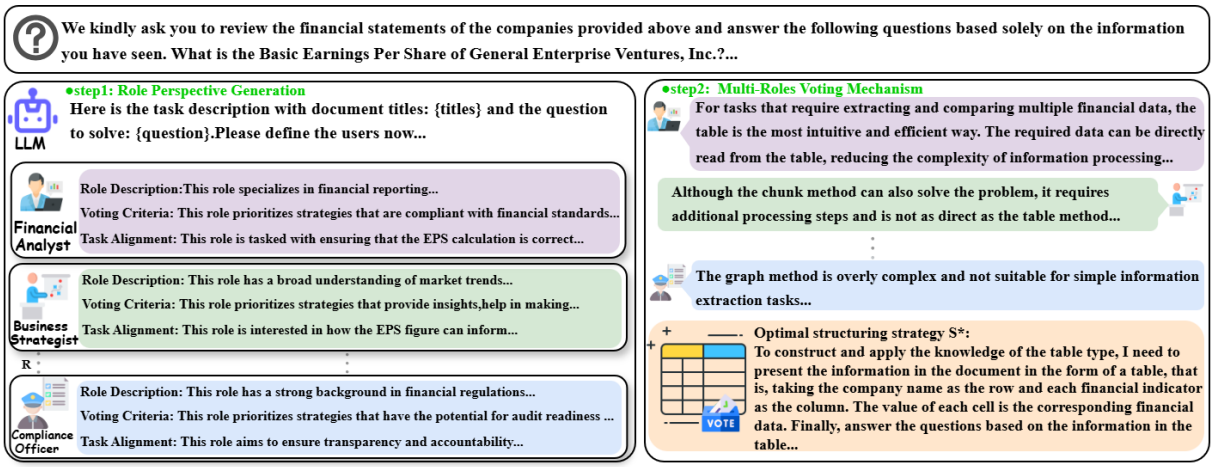

Figure 2: This describes the Role-Based Multi-Perspective Collaboration process. We first dynamically generate a set of expert roles R. These roles are domain-relevant, non-overlapping, and designed to offer diverse perspectives (step 1). Then, based on R, a candidate set of strategies S is generated. Each role evaluates and scores these strategies using role-specific voting criteria, with weights determined by task alignment. Finally, the best strategy S* is selected by aggregating the scores from R (step 2).

roles contribute different perspectives. These roles engage in discussions and vote on the most suitable strategies, providing more comprehensive and diverse inputs for decisions, as shown in Fig. 2.

3.2.1 Role Perspective Generation

We first construct a general role-definition prompt template t_R, based on the task description n, which includes the target $question$ and relevant document $titles$. t_R guides the LLM to generate multiple complementary expert roles $R \in \{r_1, r_2, ..., r_m\}$ centered around the task. R consists of three components: role description, voting criteria, and task alignment. The role description serves as a prompt, guiding the model to adopt a specific perspective when handling the task. The voting criteria define how outputs from different roles are evaluated, and task alignment assesses the weight of each role's score. To prevent overlap between R, t_R explicitly ensures minimal semantic overlap in their characteristics.

3.2.2 Multi-Roles Voting Mechanism

Next, the LLM, guided by the R (such as data analysts, technical experts, product managers, etc.), generates distinct structuring strategies $S = \{s_1, s_2, ..., s_j\}$ to address the $question$. Once all strategies have been proposed, each role ranks and scores S based on the effectiveness of their respective criteria. The final decision S^* is made by aggregating the scores, with the highest-ranked strategy being selected for implementation. The process is illustrated in Algorithm 1 (See Appendix A).

Next, the LLM utilizes S^* as a prompt to derive knowledge representations K, processing $Docs$ in alignment with the given $question$:

$$K = LLM(Docs, question, S^*) \quad (1)$$

3.3 Reflective Knowledge Refinement

Since K is derived from the $Docs$, it remains broad and potentially noisy. Due to the limited context window of LLMs, it is difficult for the model to process the entire $Docs$ content at once, which may result in the loss of critical information. To address this challenge, we propose a method of reflective knowledge refinement that introduces a reflective construction mechanism to optimize the reasoning process (see steps 3, 4, and 5 in Fig. 3).

3.3.1 Reflective Construction Mechanism

The core idea of this mechanism is to cache the current task memory m_i of the LLM into a fixed-size storage buffer M(See Appendix B). M records an analysis of the current task's generated results, including but not limited to error correction, information supplementation, or confirmation of the current content. The model iteratively reflects on and evaluates the generated content, dynamically adjusting the output and updating or overwriting M. Specifically, for different tasks, a corresponding prompt template t is designed. Using the t required background information I, and referencing the previously generated output b_{i-1} along with the memory m_{i-1} stored in M, the model generates the current output b_i, which is then reflected upon

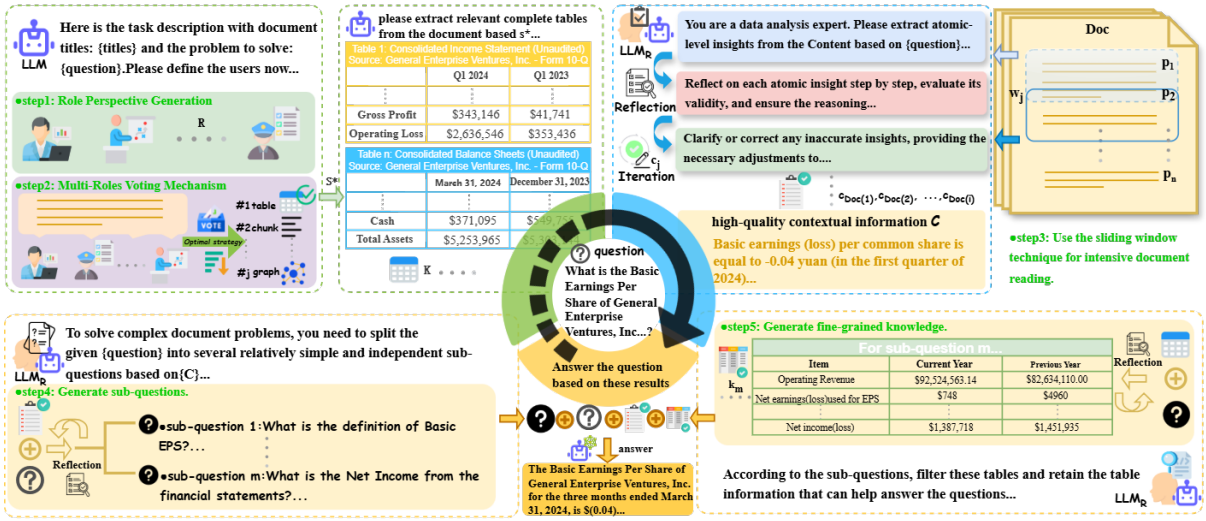

Figure 3: The overview of the ReCore framework, which simulates the human thought process in problem-solving: first, multiple expert roles R are generated (step 1), and then the optimal strategy S^* is obtained through voting from multiple expert perspectives (step 2). Then, the reflective construction mechanism is applied to conduct intensive reading of documents, decompose complex questions, and integrate information (steps 3, 4, and 5), ultimately arriving at the answer.

using the LLM's self-reflection ability to obtain the new memory m_i. If m_i deems b_i reasonable, the result is retained; otherwise, b_i is corrected based on m_i, and the updated result is stored back in the memory buffer M.

To enhance the diversity of reflection and reduce the risk of relying on a single decision-making perspective, we introduce R to collaborate within this mechanism. This collaborative reflective process is referred to as LLM_R, where all roles share the same memory buffer M. When b_i is generated, m_i includes the opinions from all the perspectives of R. The process is formulated as:

$$m_i, b_i = LLM_R(m_{i-1}, b_{i-1}, t, I), m_0, b_0 = \emptyset \tag{2}$$

3.3.2 Document Intensive Reading

To compensate for the potential loss of information in K, this phase integrates a sliding window strategy with the reflective construction mechanism, enabling the progressive extraction of atomic-level facts from $Docs$. The goal is to achieve effective information compression while preserving contextual memory throughout the reading process.

Each $Doc(i)$ is considered as a sequence of n paragraphs: $Doc(i)$=$\{P_1, P_2, \ldots, P_n\}$. We define a sliding window size w=2, meaning each local context window W_j contains two adjacent paragraphs: W_j=$\{P_j, P_{j+1}, \ldots, P_{j+w-1}\}$,$j \in [1, n-w+1]$.

Since not all content in $Docs$ is relevant to the $question$, to avoid redundancy and computational overhead, LLM selectively extracts only the most relevant information from each $Doc(i)$ in parallel. We design a document-intensive reading prompt t_{read}, I=$\{W_j, question\}$. Based on the previous summary c_{j-1} and memory m_{j-1}, the model generates the current summary c_j and memory m_j:

$$m_j, c_j = LLM_R(m_{j-1}, c_{j-1}, t_{read}, I), m_0, c_0 = \emptyset \tag{3}$$

After iterating through all windows in $Docs$ using LLM_R, each $Doc(i)$ yields a complete set of summaries $c_{Doc(i)} = \{c_1, c_2, \ldots, c_j\}$. The high-quality contextual information set is then represented as $C = \{c_{Doc(1)}, c_{Doc(2)}, \ldots, c_{Doc(i)}\}$.

3.3.3 Knowledge Integration and Reasoning

Given that the K may still contain a substantial amount of information, and that the $question$ could be complex (e.g., chain-of-reasoning problems), we further apply the reflective construction mechanism to restructure and integrate the existing content to facilitate the reasoning process.

Humans typically approach complex questions through a gradual, step-by-step process that progresses from shallow to deep understanding. Inspired by this, we design a sub-question decomposition prompt t_{sub}, which takes as input I=$\{C, question\}$ and generates a set of sub-questions $Sub\text{-}question_i$ that are relatively simple and mutually independent. If the $question$ is already sufficiently atomic or cannot be further decomposed, this step is skipped. Subsequently, with

$Sub\text{-}question_i$ and K as input I, we employ a knowledge extraction prompt t_{task} to further extract the relevant information k_i from K for $Sub\text{-}question_i$:

$$m_i, Sub\text{-}question_i = LLM_R\left(m_{i-1}, Sub\text{-}question_{i-1}, t_{sub}, I\right), m_0, Sub\text{-}question_0 = \emptyset \tag{4}$$

$$m_i, k_i = LLM_R(m_{i-1}, k_{i-1}, t_{task}, I), m_0, k_0 = \emptyset \tag{5}$$

This process continues until both $Sub\text{-}question_i$ and k_i are confirmed—based on the model's reflective memory—as being in their finalized forms. The finalized sub-questions and corresponding knowledge elements are denoted as $Sub\text{-}question_m$ and k_m, respectively.

Finally, LLM performs reasoning over the original $question$ by jointly utilizing $Sub\text{-}question_m$, k_m, and C, to generate the final answer A:

$$A = LLM_{Infer}(question, C, Sub\text{-}question_m, k_m) \tag{6}$$

where LLM_{Infer} refers to the process where the LLM is guided by prompts to infer A.

4 Experiments

4.1 Datasets

Our work uses the Loong benchmark (Wang et al., 2024) to evaluate performance on knowledge-intensive QA tasks. The benchmark includes four tasks (Spotlight Locating, Comparison, Clustering, and Chain of Reasoning) and four document length settings (Set1, Set2, Set3, Set4), with the longest exceeding 200k tokens. As document length increases, the relevant information for solving the task becomes more dispersed. The benchmark covers English and Chinese, two of the most widely used languages, effectively simulating real-world open-domain QA tasks. For evaluation metrics, we follow the original Loong benchmark setup, assessing performance using LLM scores (0–100) and exact match (EM) rate (0–1).

Additionally, we reference Podcast Transcripts, a method that generates test questions based on content scraped from Microsoft's podcast site, using LLMs as the evaluation standard. By comparing the results from two different methods via prompt-based querying, we can quantify relative performance (Zheng et al., 2023). Specifically, we scraped and selected 1,027 recent blog posts from Microsoft's site, following the GraphRAG (Edge et al., 2024) setup to diversify the evaluation of the generated outputs (usefulness, richness, insightfulness, and user-friendliness).

4.2 Experimental Settings

We use Qwen2-72B-Instruct as the base model framework (Yang et al., 2024a) and deploy the model using vllm3 via API, maintaining the same settings as Loong (Wang et al., 2024). In our method, the hyperparameters are set to $m = 3$ (number of generated roles) and $w = 2$ (to read content between two adjacent document segments).

4.3 Baseline Methods

To comprehensively evaluate the performance of the proposed ReCore framework in intensive reasoning tasks, we compare it against the following baseline methods: (1) RAG (Lewis et al., 2020), which chunks the document and uses a retriever to keep the most relevant blocks for augmentation; (2) Long-context, which extends the LLM input window to 128K tokens using extrapolation, allowing large documents to be directly input to the model (Yang et al., 2024a); (3) RQ-RAG (Chan et al., 2024), which refines queries to return more relevant text blocks for reasoning; (4) GraphRAG (Edge et al., 2024), which processes document information into a graph structure and retrieves structured, concise information to help generate model answers; (5) StructRAG, which dynamically selects the document structure by training a router with synthetic data, significantly improving RAG system performance (Li et al., 2024b). For RAG and Long-context, we follow the experimental setup reported in Loong (Wang et al., 2024). For RQ-RAG, GraphRAG, and StructRAG, we evaluate performance based on their official codebases. Similarly, Qwen2-72B-Instruct serves as the base model for baseline methods to ensure fair comparisons.

4.4 Experimental Results

4.4.1 Main experiment

We compared our proposed method with baseline methods on the Loong benchmark and Podcast Transcripts task, and the experimental results are shown in Table 1 and Table 2.

The results in Table 1 clearly show that, for shorter documents (Set 1), the Long-context method, which inputs global information into the model, outperforms the traditional RAG method. In simpler tasks (e.g., Spotlight Locating), complex

Table 1: Experimental results on the Loong benchmark. Our method achieves an average increase of 12 percentage points in LLM scores and a 27 percentage point improvement in EM rate, demonstrating superior performance on complex knowledge-intensive QA tasks. Detailed results are provided in Table 5 of Appendix C.

Method	Set 1		Set 2		Set 3		Set 4	
	LLM	EM	LLM	EM	LLM	EM	LLM	EM
RAG	46.11	0.23	45.42	0.19	42.60	0.18	29.29	0.07
Long-context	60.11	0.29	45.71	0.17	35.94	0.09	28.92	0.06
RQ-RAG	53.51	0.17	47.09	0.10	40.93	0.05	31.91	0.01
GraphRAG	40.82	0.18	33.06	0.03	33.28	0.04	23.47	0.05
Structrag	69.43	0.35	60.95	0.24	57.92	0.21	51.42	0.10
Our	**80.98**	**0.60**	**74.96**	**0.55**	**70.05**	**0.48**	**61.96**	**0.35**

Table 2: Win rates of Recore against baselines on the Podcast Transcripts task. Prompt is provided in Fig. 12. Abbreviations: Long-ctx = Long-context, U = Usefulness, R = Richness, I = Insightfulness, UF = User-Friendliness, Avg = Average. Our method outperforms all baselines, demonstrating superior effectiveness.

Method Pair	U	R	I	UF	Avg
Our vs. RAG	0.94	0.96	0.97	0.92	0.95
Our vs. Long-ctx	0.86	0.99	0.98	0.67	0.88
Our vs. RQ-RAG	0.86	0.97	0.97	0.57	0.84
Our vs. GraphRAG	0.87	0.74	0.81	0.76	0.80
Our vs. StructRAG	0.67	0.56	0.57	0.39	0.55

Table 3: Experimental results of 122 randomly selected datasets tested via APIs of other models (considering cost constraints), demonstrating the generalizability of our proposed method.

Model	LLMscore	EM
DeepSeek-V3.2	64.32	0.22
Our	**72.23**	**0.43**
GPT-4o	53.47	0.20
Our	**56.75**	**0.22**

structures like GraphRAG lead to poorer LLM performance. Furthermore, methods like StructRAG, which dynamically selects optimal structures, and RQ-RAG, which allows the model to reason step by step by breaking down subproblems, show significant performance gains. Our method effectively combines the strengths of these approaches, demonstrating its excellence.

Moreover, it is unsurprising that for shorter, simpler documents, the LLM performs well on the Spotlight Locating task. However, we found that our method, despite having moderately sized documents (Set 2 and Set 3), yields Spotlight Locating task scores similar to those for longer documents (Set 4). Upon analysis, we found that the poorer performance is primarily in financial tasks, where all Set 4 documents are in English, while Set 2 and Set 3 include a mix of English and Chinese. This mix leads to semantic ambiguity, such as confusion caused by multiple translations of a single word, which increases the difficulty of information retrieval and can result in lower scores due to imprecise wording. Nevertheless, our method still shows an overall improvement in the EM rate, indicating a higher number of correct answers. As document length increases and task difficulty rises, scores and EM rates decrease, reflecting more severe information loss and greater memory burden on the model.

4.4.2 Ablation Study

To validate the contribution of each method in Recore, we conducted an ablation study. As shown in Table 4, "w/o Reflective Construction Mechanism" refers to directly constructing C, $Sub\text{-}question_m$, and k_m, "w/o Document Intensive Reading" refers to the removal of C, and "w/o Knowledge Integration" refers to the removal of the decomposed $Sub\text{-}question_m$, with the corresponding k_m also being unconstructable, leaving only C and K.

The results demonstrate that each method makes a positive contribution to the overall framework. Removing any method leads to a decrease in LLM scores and EM rates. Long documents may cause information loss, but integrating high-quality information helps maintain the relationship between global information, allowing the LLM to better understand the context and provide more comprehensive answers, especially in chain-of-reasoning tasks. Like humans, LLMs have limitations. The reflection mechanism and structured information make the data more refined and accurate. The ablation comparison of the Role-Based Multi-Perspective Collaboration method is more complex, and we will analyze it in detail in the next section.

Table 4: Ablation study results for our framework. We abbreviate Reflective Construction Mechanism as RCM, Document Intensive Reading as DIR, and Knowledge Integration as KI.

Method	Set1		Set2		Set3		Set4		Overall	
	LLMscore	EM	LLMscore	EM	LLMscore	EM	LLMscore	EM	LLMscore	EM
Our	**80.98**	**0.60**	**74.96**	**0.55**	**70.05**	**0.48**	**61.96**	**0.35**	**72.83**	**0.51**
- w/o RCM	76.88	0.54	69.41	0.45	67.93	0.42	60.31	0.34	69.16	0.44
- w/o DIR	69.43	0.35	60.95	0.24	57.92	0.21	51.42	0.10	60.38	0.23
- w/o KI	67.21	0.31	63.61	0.26	57.16	0.17	52.71	0.12	60.82	0.22

4.5 Analysis

To better analyze the advantages and limitations of ReCore, this section will examine the task-adaptive method for dynamically determining role generation in our framework, as well as analyze the low-score results.

To ensure fairness in the experiments, we also chose Qwen2-72B-Instruct as the base LLM and compared the performance of the random structure (randomly selecting a structure to process knowledge), single structure (using a prompt to instruct the LLM to choose a structure), predefined roles (Data Analyst: efficiency priority; Accuracy Expert: accuracy priority; User Experience Expert: comprehensive analysis), and our method. As shown in Fig. 4, our method outperforms the others.

To further validate the accuracy of our multi-role decision-making strategy, we invited 10 graduate students to label and cross-check 1600 questions from the Loong benchmark. They used human-preferred structures as the gold standard for processing document information (see Appendix D).

The impact of role variability on the reproducibility of experimental results is discussed in detail in Appendix E. As shown in Table 6, our method aligns closely with human preferences. Due to cost limitations, we only tested the impact of 1-5 dynamically generated expert roles on decision-making. The results indicate that too many experts lead to poor performance, as diverse roles may introduce irrelevant factors that interfere with judgment.

We also analyzed the causes of low and zero-score results. Low scores are often attributed to answers that include excessive details and background information, which can dilute the response's focus and make it appear too verbose. Zero scores typically occur in questions requiring direct numeric answers, such as "What is the value of the right-of-use asset for company X?" in financial reports.

A detailed analysis revealed several causes for zero scores: First, the lack of clear explanations for specialized terms like "right-of-use asset" in the question; second, the use of varied expressions (e.g., "asset under lease" or "leasehold asset"), which may confuse the model's understanding; and third, the presence of multiple time or unit conversions in the question, increasing the chance of errors.

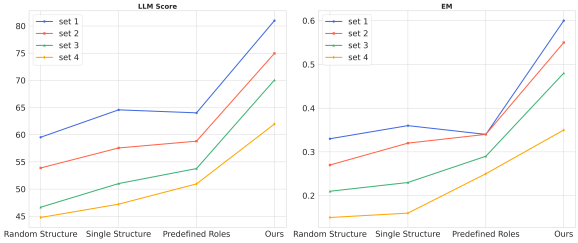

Figure 4: Impact of different methods on strategy s^*.

Overall, our method demonstrates strong robustness, providing effective answers in most cases.

5 Conclusion

In this paper, we propose a new framework, ReCore, for assisting LLMs to obtain more refined external knowledge and utilize it, for the challenges faced in existing knowledge-intensive QA tasks, concerning the steps humans take to deal with difficult questions. The framework dynamically generates multi-roles perspective based on the task, sets decisions through a voting mechanism, and then introduces a reflective knowledge-refinement approach to integrate the refined information and finally conclude the answer. Experiments show that the reasoning accuracy is significantly improved in knowledge-intensive tasks such as spotlight locating, comparison, clustering, and chain of reasoning. Therefore, this paper offers a novel idea for developing more powerful RAG systems in the future, focused on how to allow the model to acquire and utilize quality knowledge.

Limitations

In this section, we will discuss our work's limitations and focus on future directions for improvement.

Firstly, due to the application of multiple agents, this research is inevitably influenced by cost and equipment constraints. Future research could explore ways to optimize the framework to alleviate these cost-related challenges. Furthermore, due to the instability of the generation process, the benchmarks used to evaluate the results of knowledge-intensive reasoning tasks may not be perfect. Currently, the most reliable standard remains manual annotation, highlighting the need for the design of a more comprehensive evaluation benchmark.

Additionally, the breadth of languages is a crucial factor for the generalization of this study. At present, only the two most widely spoken languages, Chinese and English, are considered. Future studies could extend the research to include other languages.

Finally, our approach still faces limitations when dealing with very long documents. This is because excessively long documents place a significant burden on the model. Regardless of how much the representation is streamlined, truncation occurs when the document exceeds the model's maximum input length. Even humans struggle to retain comprehensive and extensive knowledge, and we believe that, with further optimization of large models and advancements in self-learning capabilities, the performance will improve, particularly in terms of memory retention.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, and 1 others. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610*.

Paul Chandler and John Sweller. 1991. Cognitive load theory and the format of instruction. *Cognition and instruction*, 8(4):293–332.

Darren Edge, Ha Trinh, N Cheng, J Bradley, A Chao, A Mody, S Truitt, and J Larson. 2024. From local to global: A graph rag approach to query-focused summarization. arxiv 2024. *arXiv preprint arXiv:2404.16130*.

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).

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Parag Jain, Andreea Marzoca, and Francesco Piccinno. 2024. Structsum generation for faster text comprehension. *arXiv preprint arXiv:2401.06837*.

Philip Nicholas Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. 6. Harvard University Press.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Yury Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Tong Li, Zhihao Wang, Liangying Shao, Xuling Zheng, Xiaoli Wang, and Jinsong Su. 2023. A sequence-to-sequence&set model for text-to-table generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5358–5370, Toronto, Canada. Association for Computational Linguistics.

Xiaoxi Li, Zhicheng Dou, Yujia Zhou, and Fangchao Liu. 2024a. Corpuslm: Towards a unified language model on corpus for knowledge-intensive tasks. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 26–37.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2024b. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *arXiv preprint arXiv:2410.08815*.

Zhuoqun Li, Hongyu Lin, Yaojie Lu, Hao Xiang, Xianpei Han, and Le Sun. 2024c. Meta-cognitive analysis: Evaluating declarative and procedural knowledge in datasets and large language models. *arXiv preprint arXiv:2403.09750*.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2021. Tapex: Table pre-training via learning a neural sql executor. *arXiv preprint arXiv:2107.07653*.

Kaixin Ma, Hao Cheng, Yu Zhang, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2023. Chain-of-skills: A configurable model for open-domain question answering. *arXiv preprint arXiv:2305.03130*.

Jinjie Mai, Jun Chen, Guocheng Qian, Mohamed Elhoseiny, Bernard Ghanem, and 1 others. 2023. Llm as a robotic brain: Unifying egocentric memory and control.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Allan Paivio. 1990. *Mental representations: A dual coding approach*. Oxford university press.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, and 1 others. 2025. Rethinking reflection in pre-training. *arXiv preprint arXiv:2504.04022*.

Kurt Shuster, Mojtaba Komeili, Leonard Adolphs, Stephen Roller, Arthur Szlam, and Jason Weston. 2022. Language models that seek for knowledge: Modular search & generation for dialogue and prompt completion. *arXiv preprint arXiv:2203.13224*.

Peiqi Sui, Eamon Duede, Sophie Wu, and Richard Jean So. 2024. Confabulation: The surprising value of large language model hallucinations. *arXiv preprint arXiv:2406.04175*.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.

Narayan S Umanath and Iris Vessey. 1994. Multiattribute data presentation and human judgment: A cognitive fit perspective. *Decision Sciences*, 25(5-6):795–824.

Iris Vessey. 1991. Cognitive fit: A theory-based analysis of the graphs versus tables literature. *Decision sciences*, 22(2):219–240.

Minzheng Wang, Longze Chen, Cheng Fu, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, and 1 others. 2024. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. *arXiv preprint arXiv:2406.17419*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024a. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Peiru Yang, Xintian Li, Zhiyang Hu, Jiapeng Wang, Jinhua Yin, Huili Wang, Lizhi He, Shuai Yang, Shangguang Wang, Yongfeng Huang, and 1 others. 2025. Heterag: A heterogeneous retrieval-augmented generation framework with decoupled knowledge representations. *arXiv preprint arXiv:2504.10529*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Gui, Ziran Jiang, Ziyu Jiang, and 1 others. 2024b. Crag-comprehensive rag benchmark. *Advances in Neural Information Processing Systems*, 37:10470–10490.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Role-Based Multi-Perspective Collaboration Details

This appendix provides specific implementation details of the Role-Based Multi-Perspective Collaboration mechanism in ReCore. It elaborates on the criteria for dynamic role generation, the specific process of multi-perspective insight collection, and the voting mechanism for optimal structuring strategy selection.

For the given task, there are multiple solutions, each leveraging a different type of knowledge. You need to rank these solutions based on their effectiveness and determine the most appropriate solution.

You are required to define one distinct user personas, each with unique skills and perspectives. These personas will be used to analyze the task and vote on the best solution.
Each persona should be clearly differentiated from one another, ensuring minimal overlap in their characteristics. The task description provides insights that will help you shape these personas.

When crafting each persona, focus on the following:

1. **Role Description:** What is the role's area of expertise or focus? What strengths and perspectives does the role bring to the task?
2. **Voting Criteria:** How does the role prioritize factors when evaluating solutions? What criteria (e.g., effectiveness, efficiency, cost, impact, ethical considerations) does the role use to make a decision?
3. **Task Alignment:** How does the task align with the role's goals and interests? What challenges or needs are the role trying to address with their analysis?

Prevent inluding specific entities or names in the persona descriptions to maintain a general and abstract representation. Each persona will evaluate the task and provide their ranking of the proposed solutions. Your final decision will be based on the collective insights and votes from all personas." ""
Here is the task description with document titles: {titles} and the problem to solve: {question}.
Examples of user personas:

1. Data Analyst
- **Role Description:** The Data Analyst specializes in analyzing datasets, identifying patterns, and providing statistical insights.
- **Voting Criteria:** This role prioritizes solutions that are backed by data, use robust analytical methods, and offer actionable insights based on data patterns.
- **Task Alignment:** The Data Analyst is interested in extracting meaningful insights from the data and recommending data-driven solutions.

\nPlease define the users now.

Figure 5: Prompts for R Generation.

Instruction:
You are a professional with the skills of {role}. Faced with the current task, you must fully leverage your expertise to analyze all proposed solutions and assess which one is best suited to address the problem at hand.
Focus on organizing relevant information and evaluating the strengths and weaknesses of each approach, rather than providing a direct final answer. The goal is to build a solid foundation for identifying the optimal solution through careful analysis.

Document Info:
{titles}

Question:
{question}

Solutions:
{solutions}

Prior Discussion:
{discussion}

Requirements:
Analyze the strengths and weaknesses of each solution in detail, including the specific evaluation criteria you use during the analysis.

Rank the solutions explicitly using the format ">", "<", "=" to show how each compares to the others.

Your final evaluation should focus on how effectively each solution addresses the current problem, especially in terms of the core task (e.g., sorting, content extraction, information linking, etc.).

Figure 6: Prompts for Multi-Roles Voting Mechanism

Algorithm 1 Role-Based Multi-Perspective Collaboration

Input: n, t_R and Number of roles m
Output: Optimal structuring strategy S^*

$R \leftarrow \text{GenerateRoles}(m, n, t_R)$
$S \leftarrow \text{GenerateAllStrategies}(R, n)$
$\text{TaskWeight}(R) \leftarrow \text{Sim}(\text{TaskAlignment}_R, n)$
$scores \leftarrow \emptyset$
for each role $r_i \in R$ **do**
  for each strategy $s_j \in S \setminus \text{own}(r_i)$ **do**
    $Score_{r_i}(s_j) \leftarrow \text{Evaluate}(s_j \mid \text{VotingCriteria}_{r_i})$
    $scores[s_j] \leftarrow scores[s_j] + \text{TaskWeight}(r_i) \cdot Score_{r_i}(s_j)$
  end for
end for
$S^* \leftarrow \arg\max_{s_j \in S} scores[s_j]$
return S^*

B Reflective Knowledge Refinement Details

This appendix presents the complete prompt templates underpinning the Reflective Knowledge Refinement mechanism.

```
You are {R}.  Please evaluate whether the following atomic insights for my question are correct.


## Task Description
- If the atomic insights are correct, please answer the question directly.
- If there are any incorrect parts, please provide the corrected version.

## Question
{question}

## Answer Format
- If you can answer the question directly, provide your answer as follows:
{
"think": "Step-by-step reasoning to solve the question",
"answer": True
}

- If the insights are incorrect, provide the corrected information:
{
"think": "Evaluate the atomic insights one by one, and solve the question step by step",
"feedback": "Clarify the necessary information for each insight",
"answer": False
}
Please provide the answer in JSON format:
```

Figure 7: Prompts for reflection

```
You are a {R}. Based on the question below, extract atomic-level insights from the content, and express them using the Subject–Verb–Object (SVO) format.

Extraction Requirements:
Use the original numerical value from the content (no modifications).

Apply banker's rounding and truncate to 4 decimal places if conversion is needed.

Use appropriate units: CNY, ten thousand CNY, or million CNY.

Do not infer or assume missing information.

Incorporate reflective opinions from previous content or characters to provide context, ensuring the insights align with earlier generated information.

Ensure that the insights are coherent with previous conclusions or data points, drawing from past reflections where applicable.

Memory:
{M}

Prior Content:
{prior content}

Output Format:
Subject ‖ Relation ‖ Object

If irrelevant:
If no logical relation exists between content and query → Output: "none"

If numeric data is present but no conversion is needed → keep the original number.

Question:
{question}

Content:
{content}

Output:
```

Figure 8: Prompts for Document Intensive Reading

```
Instruction:
To effectively address complex document-related questions, you are required to break down the given question into multiple simpler and logically independent sub-questions.

Requirements:
1.Doc Info provides a description of the document and may help guide your decomposition—use it as a reference where applicable.

2.  If the given question is already sufficiently simple, or if no meaningful decomposition is possible, do not attempt to split it.

3.  Consider multiple character opinions, reflective thoughts, and insights from previously generated content.  Integrate any relevant information from past discussions, reflecting on the conclusions made in earlier steps to ensure consistency and continuity.

4.  Ensure that the decomposition aligns with the broader context of previous information, incorporating insights and memories from prior interactions.

Memory:
{M}

Prior Content:
{prior content}

Do not include any additional explanation or commentary in your output.

Description:
{C}

Query:
{query}

Output:
```

Figure 9: Prompts for question decomposition

```
Instruction:
The following structured data segments are independently constructed from multiple documents.  Based on the question, filter and retain only the segments that are relevant and helpful for answering the question.  Carefully analyze the entities and attributes mentioned in the question, and use them as the basis for filtering.
The information required to answer the question is guaranteed to exist in one or more of the provided data segments.  You must review each segment individually to determine its relevance.  In addition, consider the perspectives of multiple characters, their reflective opinions, and memories from prior generated content to form a cohesive and well-rounded answer.

Memory:
{M}

Prior Content:
{prior content}

Data Segments:
{struct_content}

Question:
{subquestion}

Output:
(List only the segments that are useful for answering the question.  Do not include any additional explanation or information outside of the filtered relevant segments.  Ensure that past reflections or memories are integrated to align with the answer provided.)
```

Figure 10: Prompts for further extract the relevant information

```
1.Answer the Question based on retrieval results.
2. Find the relevant information from given retrieval results and output as detailed, specific, and lengthy as possible.
3. The output must be a coherent and smooth piece of text.
```

Figure 11: Prompts for infer the final answer

C Experimental Settings and Comparative Results

This appendix details the evaluation prompts for the Podcast Transcripts task, and presents the full main experimental data (see Table 5).

```
You are tasked with comparing two summaries provided in response to a given context.
Follow these guidelines:

1. Careful Reading: Begin by carefully reading the original context and the two summaries.
2. Scoring Criteria: Evaluate which summary does better in terms of the following
dimensions:
- Usefulness / Correctness: How accurate and reliable is the summary?  Does it capture the
key points and provide useful information?
- Richness / Diversity: How varied and detailed is the summary?  Does it cover different
aspects of the original content or provide in-depth explanations?
- Insightfulness / Deep Understanding: Does the summary demonstrate a deep
comprehension of the content?  Does it offer thoughtful or original insights into the subject?
- User-Friendliness: How easy is the summary to read and understand?  Is it well-
structured, concise, and accessible to the reader?
3. Explanation: After evaluating the summaries based on the criteria, provide a detailed
explanation for your rankings.  Justify your assessment with specific reasons.

Output Format: (Assume the summaries are labeled A and B)

Usefulness / Correctness: A
Reason: ...
Richness / Diversity: A
Reason: ...
Insightfulness / Deep Understanding: A
Reason: ...
User-Friendliness: A
Reason: ...

Ensure that your evaluation and explanations are well-supported by your analysis of the
content. Your evaluation should be thorough, fair, and based on the quality of the summaries
provided.  A good summary is not always longer, but more accurate and logical.

Here is the context information:
-- start of context --
{context}
-- end of context --

Here are two summaries:
-- start of summary A --
{answers}
-- end of summary B --

Respond according to the guidelines provided, do not include any other information in the
response. Strictly follow the format provided above.
```

Figure 12: Prompts for Podcast Transcripts task

D Annotation Protocols for Human Preference Alignment

The annotation process followed three core steps: (1) Document type classification: annotators first categorized each question's corresponding document into four types (Financial Report, Legal Document, Technical Paper, News Article); (2) Structure definition: predefined processing structures were assigned to each document type (Financial Report → Table structure, Legal Document → Chunk structure, Technical Paper → Hierarchical structure, News Article → Keyword structure); (3) Cross-check: 10 annotators were divided into 5 groups to conduct double-blind labeling, and inconsistent labels were resolved through group discussion to ensure annotation accuracy.

E Impact of Role Variability on Result Reproducibility

Table 6: Impact of the number of expert roles m on decision results: This table presents the performance of different methods in the Loong benchmark and analyzes the impact of the number of expert roles on both performance and alignment with human preferences.

m	Accuracy	F1 Score	Overall	
			LLMscore	EM
1	0.74	0.85	68.42	0.48
2	0.81	0.89	70.63	0.495
3	**0.93**	**0.96**	**72.83**	**0.51**
4	0.78	0.88	67.45	0.505
5	0.70	0.83	62.07	0.50

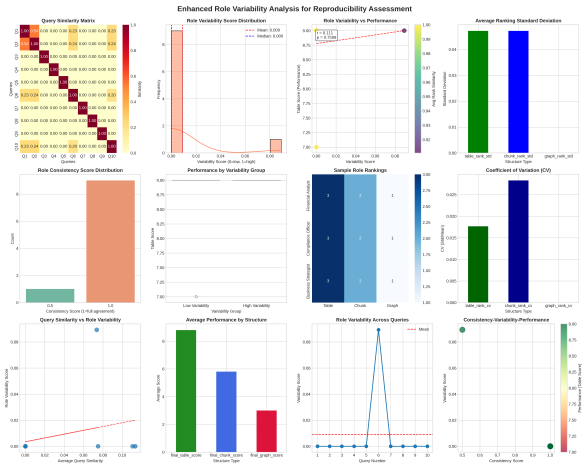

Figure 13: Role Variability Analysis for Reproducibility (Controlled Financial Domain, Similar Query Types)

This panel set evaluates variability in a financially focused, Chinese-English dataset with manually screened similar query types (to isolate role variability from task variability): (1) Role Variability Score Distribution shows low variability (majority near 0.00) even within consistent query types; (2) Performance by Variability Group (Low/High Variability) shows no significant performance divergence, ruling out query-type confounds; (3) Average Performance by Structure maintains stable rankings across role variations for parallel financial tasks. The similar query type design ensures our finding—role variability does not threaten reproducibility—is specific to (and valid for) aligned financial domain scenarios, addressing potential concerns about task heterogeneity.

Table 5: Experimental results on the Loong benchmark. As shown, our method achieves an average increase of 12 percentage points in LLM scores and a 27 percentage point improvement in EM rate. The increase in EM rate indicates that the number of perfect answers has grown, demonstrating that the model performs better on complex, knowledge-intensive QA tasks.

Method	Spot.		Comp.		Clus.		Chain.		Overall	
	LLM Score	EM	LLM Score	EM	LLM Score	EM	LLM Score	EM	LLM Score	EM
Set1										
RAG	51.08	0.35	44.53	0.27	37.96	0.05	53.95	0.35	46.11	0.23
Long-context	68.49	0.55	60.60	0.37	47.08	0.08	70.39	0.36	60.11	0.29
RQ-RAG	72.31	0.54	48.16	0.05	47.44	0.07	58.96	0.25	53.51	0.17
GraphRAG	31.67	0.00	27.60	0.00	40.71	0.14	54.29	0.43	40.82	0.18
Structrag	74.53	0.47	75.58	0.47	65.13	0.23	67.84	0.34	69.43	0.35
Recore	**86.32**	**0.75**	**82.83**	**0.65**	**83.14**	**0.50**	**74.38**	**0.60**	**80.98**	**0.60**
Set2										
RAG	66.27	0.46	46.28	0.31	38.95	0.05	46.15	0.22	45.42	0.19
Long-context	64.53	0.43	42.60	0.21	38.52	0.05	51.18	0.20	45.71	0.17
RQ-RAG	57.35	0.35	50.83	0.16	42.85	0.03	47.60	0.10	47.09	0.10
GraphRAG	24.80	0.00	14.29	0.00	37.86	0.00	46.25	0.12	33.06	0.03
Structrag	68.00	0.41	63.71	0.36	61.40	0.17	54.70	0.19	60.95	0.24
Recore	66.93	**0.59**	**74.04**	**0.59**	**81.91**	**0.55**	**67.57**	**0.50**	**74.96**	**0.55**
Set3										
RAG	73.69	0.55	42.20	0.27	32.78	0.02	37.65	0.13	42.60	0.18
Long-context	46.99	0.27	37.06	0.13	31.50	0.02	35.01	0.07	35.94	0.09
RQ-RAG	50.50	0.13	44.62	0.00	36.98	0.00	36.79	0.07	40.93	0.05
GraphRAG	15.83	0.00	27.40	0.00	42.50	0.00	43.33	0.17	33.28	0.04
Structrag	68.62	0.44	57.74	0.35	58.27	0.10	49.73	0.13	57.92	0.21
Recore	62.12	**0.51**	**69.58**	**0.49**	**78.22**	**0.48**	**61.96**	**0.43**	**70.05**	**0.48**
Set4										
RAG	52.17	0.24	24.60	0.10	26.78	0.00	17.79	0.00	29.29	0.07
Long-context	33.18	0.16	26.59	0.08	29.84	0.01	25.81	0.04	28.92	0.06
RQ-RAG	29.17	0.08	40.36	0.00	26.92	0.00	34.69	0.00	31.91	0.01
GraphRAG	17.50	0.00	26.67	0.00	20.91	0.00	33.67	0.33	23.47	0.05
Structrag	56.87	0.19	55.62	0.25	56.59	0.00	35.17	0.05	51.42	0.10
Recore	**63.51**	**0.47**	**64.74**	**0.38**	**70.92**	**0.33**	**45.09**	**0.26**	**61.96**	**0.35**

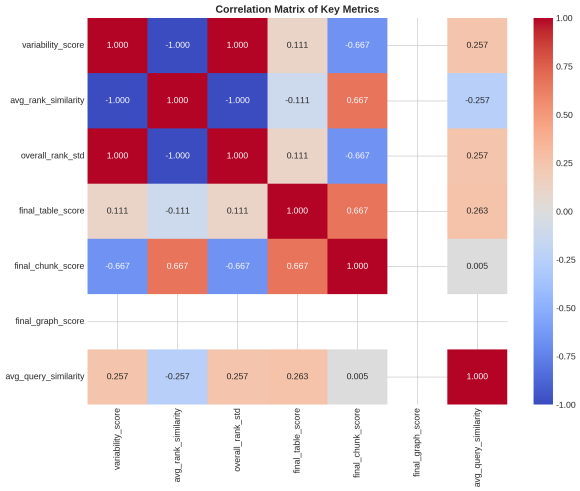

Figure 14: Correlation Matrix of Key Metrics for Role Variability and Reproducibility

This correlation matrix quantifies pairwise relationships between role variability (e.g., $variability_score$, $overall_rank_std$), consistency (e.g., $avg_rank_similarity$), and performance metrics (e.g.,$final_table_score$). Notable observations include: (1) A perfect negative correlation (-1.00) between $variability_score$ and $avg_rank_similarity$ reflects the expected inverse relationship between variability and consistency; (2) Weak correlations ($\leq$ 0.263) between variability metrics (e.g.,$variability_score$, $overall_rank_std$) and performance scores (e.g.,$final_table_score$, $final_chunk_score$); (3) A near-zero correlation (0.005) between $final_chunk_score$ and $avg_query_similarity$. These results indicate that role variability has minimal impact on core performance metrics, supporting that our method's results remain stable (i.e., reproducible).

F Computational Overhead

On average, processing a document collection to derive answers for each question requires 1–2 hours. Details can be referred to the Data provided in the submitted paper.

G Artifact Citations, Licenses and Usage Compliance

All third-party artifacts used in this work, including the base LLM model and dataset, are properly cited in the References section. The base model is licensed under Apache 2.0 for unrestricted academic use, and the dataset follows CC BY-NC-SA 4.0 with non-commercial research as its intended use. Our utilization of these artifacts strictly complies with their license terms, with no derivatives of the dataset deployed outside of the research context. The self-developed ReCore framework code is available upon request for academic validation.