MCIR-RAG: Multi-Agent Chain-of-Thought Iterative Reflection for Multi-modal Document Understanding

Chunming Liu[1,2], Sijia Wang[2], Ziteng Li[2], Minghao Hu[2], Quntian Fang[2],

Yinlong Xiao[2], Jun Zhang[2], Zhunchen Luo[2], Wei Luo[2], Zibo Yi[2⋆], Yanping Zhang[1⋆]

[1] Hebei University of Engineering, Handan, China
[2] Center of Information Research, Academy of Military Science, Beijing, China

***Abstract*—Document Visual Question Answering (DocVQA) is a core task in natural language processing, aiming to provide accurate responses to user queries based on complex documents. The main challenges include efficiently processing long texts, multimodal information such as tables and charts, and handling queries with complex logic. However, traditional Retrieval-Augmented Generation (RAG) frameworks face significant limitations: in the retrieval phase, they tend to introduce redundant information and have insufficient recall accuracy for key content; in the generation phase, they struggle with deep collaborative understanding of multimodal information and cannot support cross-modal reasoning or complex logical inference. To address these issues, this paper proposes the Multi-Agent Chain-of-Thought Iterative Reflection Retrieval-Augmented Generation (MCIR-RAG) framework. In the retrieval phase, the framework performs initial document page screening, generates focused summaries for selected pages, and dynamically reorders them based on these summaries to improve retrieval accuracy. In the generation phase, the framework enhances collaborative understanding of multimodal information and complex logical inference through a multi-agent chain-of-thought and iterative reflection mechanism.**

***Index Terms*—DocVQA, RAG, Multi-Agent, Chain-of-Thought.**

I. Introduction

In the Document Visual Question Answering (DocVQA) task, Retrieval-Augmented Generation (RAG) technology is often used as an auxiliary tool [1], [23]; looking back at previous relevant studies, the document retrieval process typically relies on text extracted via Optical Character Recognition (OCR) technology [2], [3], [24]. While such text-based retrieval systems have advantages in linguistic logical reasoning, they often overlook visual elements (e.g., tables and images) in documents that may contain crucial information.

In recent years, research on multimodal retrieval models has made significant progress: representative models like CoPali [4] innovatively treat each page of a document as an independent image, thereby enabling page-level retrieval operations [5], [6]. Building on this technological foundation, M3DocRAG [7] further proposes a multimodal RAG system. This system performs document retrieval through multimodal retrieval models (such as CoPali) and integrates multimodal language models to process the retrieved page information. As a result, it demonstrates outstanding performance in the DocVQA task. Meanwhile, multi-agent systems [19], [20] have been shown to be effective in decomposing complex, multi-step tasks [9], [10]; inspired by this, MDocAgent [8], a multi-agent framework for DocVQA, decomposes the task process into multiple collaborative agents to achieve higher precision and robustness in DocVQA scenarios. Although the aforementioned methods have advanced the development of the DocVQA field, they still have significant shortcomings in reducing redundant information in retrieval, achieving in-depth collaborative understanding of multimodal information, and supporting complex logical reasoning.

This paper proposes a novel framework named MCIR-RAG. In the retrieval phase, to address the problem of redundant information introduction in traditional retrieval, the framework first uses a multimodal retrieval model to conduct preliminary screening of document pages, so as to narrow down the retrieval scope; subsequently, it generates focused summaries for the preliminarily screened pages based on user queries and dynamically reranks the pages according to these summaries. In the generation phase, to address the issues of insufficient collaborative understanding of multimodal information and weak complex reasoning ability, MCIR-RAG adopts a multi-agent chain-of-thought and iterative reflection mechanism inspired by [11], [12], [22]. This mechanism consists of a Text Chain Agent, an Image Chain Agent, and a Verification Agent. Specifically, the Text Chain Agent receives the retrieved summaries, while the Image Chain Agent receives the retrieved pages; through chain-of-thought reasoning (including evidence curation and chain-of-thought deduction), these two types of agents generate a reference answer and a preliminary answer, respectively. The Verification Agent further verifies these two answers: if the answers are logically consistent and complement each other in terms of information, it directly integrates key information to generate the final answer; if

⋆ Corresponding authors: Zibo Yi, ziboyi@outlook.com; Yanping Zhang, zhangyanping@hebeu.edu.cn.

there is a conflict, it feeds back the conflict information and the reasoning basis of the text-based reference answer to the Image Chain Agent, guiding it to reflect until the final answer is output.

In summary, our contributions are as follows:

1) We propose a novel multimodal RAG framework, MCIR-RAG, which effectively improves the performance of complex DocVQA tasks through a three-stage process: "Retrieval Re-Ranking, Multi-Agent Reasoning, and Iterative Reflection";

2) Experimental results show that on 4 mainstream document question-answering benchmark datasets, the performance of the MCIR-RAG framework is significantly better than that of existing methods, with an average performance improvement of 10%.

II. Related works

A. Document visual question answering

Recent advances in multimodal document visual question answering (DocVQA) have increasingly shifted toward page-level visual modeling and cross-modal retrieval. ColPali [4] treats each document page as an independent visual unit and performs page-level retrieval using multi-vector embeddings, enabling the retrieval process to capture layout and visual elements beyond OCR-extracted text. Following this page-as-image paradigm, VisRAG [5] and VDocRAG [6] adopt holistic page-level representations to mitigate information loss caused by separate text–image parsing. Building on these retrieval mechanisms, M3DocRAG [7] integrates page-level retrieval into a multimodal retrieval-augmented generation (RAG) pipeline, retrieving relevant pages from large document collections and leveraging vision–language models for answer generation. To further address the challenges of long-document reasoning, MDocAgent [8] introduces a multi-agent framework that decomposes multimodal retrieval and reasoning into specialized agents, improving system robustness in complex document question answering scenarios.

B. Retrieval augmented generation (RAG)

With the rapid development of large language models, Retrieval-Augmented Generation (RAG) has significantly enhanced models' ability to incorporate external knowledge and has become a key paradigm in natural language processing for improving knowledge coverage and generation accuracy [25], [26]. In early studies, document retrieval typically relied on optical character recognition (OCR) to extract textual information and performed retrieval based on text features [27], [28] Although such text-driven approaches exhibit advantages in linguistic semantic matching, they often overlook critical information embedded in visual elements such as tables, figures, and layout structures, thereby limiting the completeness and robustness of retrieval results. More recently, advances in multimodal embedding techniques have opened new directions for optimizing RAG frameworks, with prior work encoding entire document pages as visual objects for feature representation and retrieval, effectively overcoming the modeling limitations of traditional text-based retrieval in complex document scenarios.

III. Methodology

This chapter details the architecture and core mechanisms of the MCIR-RAG framework, as illustrated in Figure 1, which improves complex document QA performance via the three-stage process of "Retrieval Re-Ranking, Multi-Agent Reasoning, and Iterative Reflection".

A. Retrieval Re-Ranking

The original document is first parsed into a collection of structured page images:

$$\mathcal{I} = \{I_1, I_2, \ldots, I_M\}, \tag{1}$$

where each page image preserves both the visual layout and semantic content of the corresponding document page. A visual embedding model is then applied to encode each page image into a high-dimensional vector, and a page-level vector database $\mathcal{V}$ is constructed to support efficient similarity-based retrieval.

To balance retrieval efficiency and semantic accuracy, a two-stage strategy is adopted, consisting of *preliminary screening* and *precise re-ranking*.

Preliminary Screening: The user query Q is treated as a semantic anchor. A fast matching mechanism computes the similarity between Q and each page embedding in $\mathcal{V}$, and pages with the highest MaxSim scores are selected as initial candidates:

$$\mathcal{I}_{\text{init}} = \text{Screen}(\mathcal{V}, Q), \tag{2}$$

where $\mathcal{I}_{\text{init}}$ denotes the initial candidate page set. This stage efficiently reduces the search space while maintaining high recall.

Precise Re-Ranking: To mitigate information redundancy and semantic granularity limitations of page-level embeddings, each candidate page $I_i \in \mathcal{I}_{\text{init}}$ is fed into a Vision-Language Model (VLM) along with the query Q, producing a query-focused summary for each page:

$$\mathcal{S}_{\text{init}} = \{S_1, S_2, \ldots, S_K\}, \tag{3}$$

where S_i corresponds to I_i and emphasizes content most relevant to the query.

A text-based language model is then used to perform refined re-ranking based on the semantic representations of these summaries. Instead of fixing the number of retrieved pages, the Top-N selection is dynamically determined in a *prompt-guided manner* by the model, which evaluates semantic relevance and confidence to adaptively select the most informative pages:

$$\mathcal{I}_{\text{ret}}, \mathcal{S}_{\text{ret}} = \text{ReRankPrompt}(\mathcal{S}_{\text{init}}, Q). \tag{4}$$

This prompt-driven Top-N selection allows the system to flexibly adapt to different queries and documents, ensuring a balance between retrieval completeness and computational efficiency, while also being context-aware and logic-sensitive, unlike heuristic similarity-based ranking.

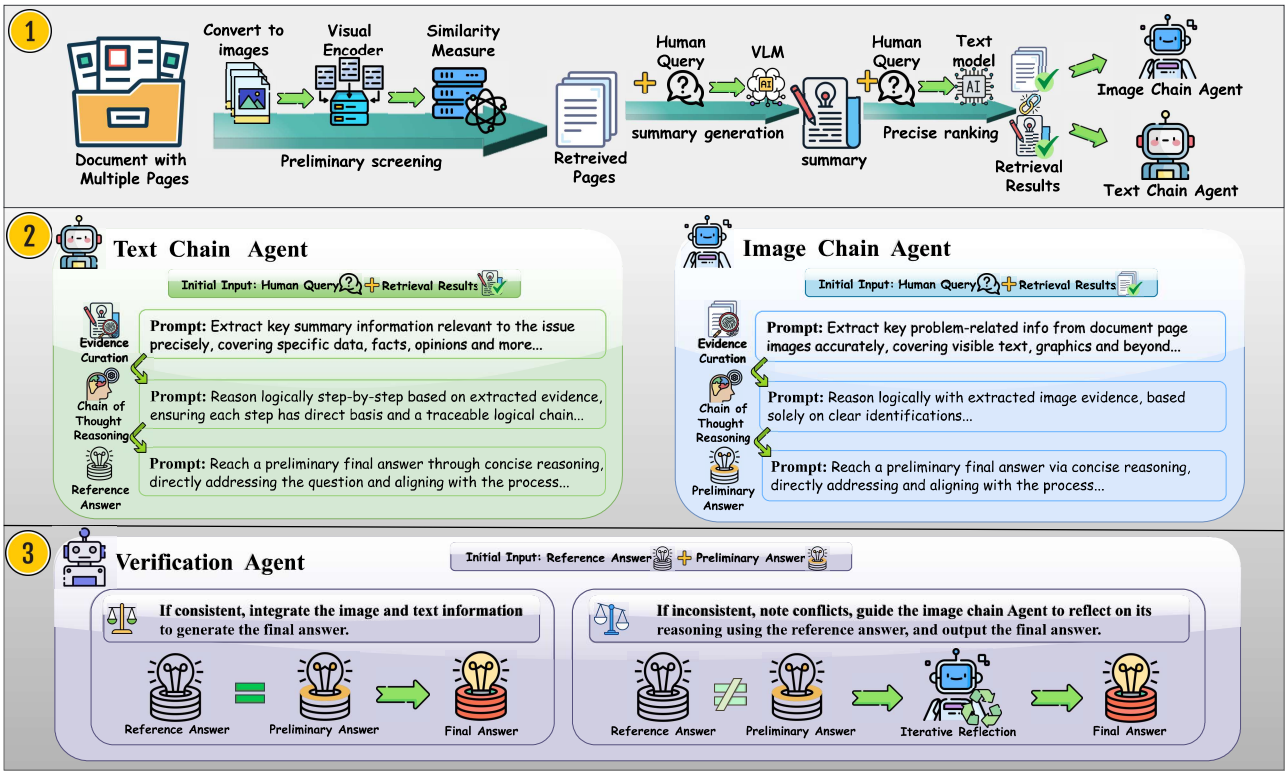

Fig. 1. Overview of the MCIR-RAG framework. **Retrieval Re-Ranking**① refines retrieval results by dynamic reordering; **Multi-Agent Reasoning**② enables collaborative processing of multimodal data via multiple agents; **Iterative Reflection**③ optimizes reasoning outcomes through feedback and refinement.

The entire two-stage retrieval and re-ranking process can be concisely expressed as:

$$\mathcal{I}_{\text{ret}}, \mathcal{S}_{\text{ret}} = \text{ReRankPrompt}\Big(\text{VLM}\big(\text{Screen}(\mathcal{V}, Q), Q\big), Q\Big), \tag{5}$$

where $\mathcal{V}$ is the page image vector database, $\text{Screen}(\cdot)$ denotes preliminary screening, $\text{VLM}(\cdot)$ generates query-focused summaries, and $\text{ReRankPrompt}(\cdot)$ performs prompt-guided refined ranking and dynamic Top-N selection.

B. Multi-Agent Reasoning

In the generation phase, a multi-agent reasoning framework is adopted, consisting of a Text Chain Agent, an Image Chain Agent, and a Verification Agent. Notably, the proposed framework is explicitly designed to be *image-centric*: the Image Chain Agent serves as the primary reasoning agent grounded on original page images, while the Text Chain Agent only plays an auxiliary role for error detection and reasoning correction.

Specifically, the Text Chain Agent and the Image Chain Agent receive the retrieved summaries $\mathcal{S}_{\text{ret}}$ and page images $\mathcal{I}_{\text{ret}}$, respectively, and follow a unified three-stage prompting strategy:

1) **Evidence Curation:** Each agent extracts query-relevant evidence from its input modality and transforms it into structured representations such as itemized lists or tables. This allows explicit localization of key information and facilitates subsequent Chain-of-Thought reasoning. It should be emphasized that evidence curated by the Text Chain Agent is derived from summaries and may contain incomplete or inaccurate information, whereas evidence curated by the Image Chain Agent is grounded in original page images and serves as the primary factual basis for reasoning.
2) **Chain-of-Thought Reasoning:** Based on the curated evidence, both agents perform multi-step Chain-of-Thought reasoning to derive intermediate conclusions. The Image Chain Agent conducts visual-grounded reasoning, explicitly referring to visual evidence, while the Text Chain Agent generates a reference reasoning path that provides coarse semantic priors. Importantly, the reasoning results produced by the Text Chain Agent are *not considered authoritative*, but only serve as auxiliary guidance for subsequent verification and reflection.
3) **Preliminary Answer Generation:** The Image Chain Agent generates a preliminary answer A_{pre} based on the original page images $\mathcal{I}_{\text{ret}}$, preserving first-hand reasoning results grounded in complete visual evidence. In parallel, the Text Chain Agent generates a reference answer A_{ref} from the retrieved summaries $\mathcal{S}_{\text{ret}}$. Due to potential

information loss or hallucinations during summary generation, A_{ref} is explicitly treated as an auxiliary signal rather than a final answer candidate.

By following this prompt-driven three-stage strategy, the system achieves multi-modal information fusion: the Image Chain Agent provides authoritative, traceable visual reasoning results, while the Text Chain Agent provides semantic guidance and error detection. This design ensures that the subsequent verification and reflection steps can be executed efficiently and accurately.

C. Iterative Verification

To further enhance answer reliability, an iterative verification and reflection mechanism is introduced. A Verification Agent is employed to assess the semantic consistency between the reference answer A_{ref} and the preliminary image-grounded answer A_{pre}.

Let $C(A_{\text{ref}}, A_{\text{pre}})$ denote the consistency judgment result, defined as:

$$C(A_{\text{ref}}, A_{\text{pre}}) = \begin{cases} 1, & \text{if semantically consistent,} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The consistency judgment is implemented as an LLM-based semantic verification process with fixed prompt templates and deterministic decoding settings. Unlike heuristic similarity measures, this approach enables context-aware and logic-sensitive consistency evaluation.

When $C(A_{\text{ref}}, A_{\text{pre}}) = 1$, the Verification Agent integrates complementary information from both answers using a predefined integration prompt and directly outputs the final answer.

When $C(A_{\text{ref}}, A_{\text{pre}}) = 0$, an image-centric reflection mechanism is triggered. First, a conflict localization prompt is used to explicitly identify the conflicting reasoning elements between A_{ref} and A_{pre}, producing a set of conflict points $\mathcal{D} = \{d_1, d_2, \ldots, d_L\}$. Next, guided by these conflict points, a reflection prompt is constructed to drive the Image Chain Agent to re-examine the original page images $\mathcal{I}_{\text{ret}}$, re-curate visual evidence, and re-execute the Chain-of-Thought reasoning process. During reflection, the Image Chain Agent is encouraged to revise its reasoning by explicitly grounding each updated conclusion in visual evidence, rather than relying on the summary-based reference.

Notably, the Text Chain Agent does not participate in the reflection process. This design choice is motivated by the fact that the summaries used by the Text Chain Agent are secondary, model-generated representations that may propagate errors, whereas the original page images constitute the most faithful and authoritative information source. Therefore, reflection is performed exclusively on the Image Chain Agent to ensure that error correction is grounded in primary visual evidence.

The verification–reflection loop is executed iteratively until semantic consistency is achieved (i.e., $C(A_{\text{ref}}, A_{\text{pre}}) = 1$) or a preset maximum number of iterations $T_{\max}$ is reached. This termination criterion ensures both reasoning robustness and computational efficiency. The final answer is generated based on the last image-grounded reasoning result.

IV. Experiments

A. Experimental Setup

Datasets. To ensure fair and comprehensive evaluation, our experiments are conducted on four widely used DocVQA benchmark datasets:

- **MMLongBench** [13]: A large-scale multimodal benchmark designed for long-context vision-language understanding. It contains diverse document types such as reports, forms, and web pages, and emphasizes reasoning over long sequences of pages, testing both retrieval and answer generation capabilities.
- **LongDocUrl** [14]: A comprehensive dataset for multimodal long document QA, covering web-based documents, technical reports, and academic articles. It focuses on challenges arising from multi-page evidence, complex layouts, and heterogeneous content types including text, tables, and figures.
- **PaperTab** [15]: A benchmark specifically curated for table-intensive documents, such as scientific papers and financial reports. It evaluates the model's ability to locate relevant tables and extract structured answers, emphasizing both semantic understanding and tabular reasoning.
- **FetaTab** [15]: Similar to PaperTab, FetaTab focuses on tabular data in multimodal documents but with more complex table structures and cross-page references. It tests models on tasks requiring aggregation, comparison, and computation over table entries.

Baselines. Two types of baselines are set for comparison in the experiment: M3DocRAG [7], which first retrieves the top-K pages relevant to the query using a multimodal image retrieval model and then generates answers via the VLM; and MDocAgent [8], which acquires key context through parallel text and image retrieval, followed by collaborative work among five specialized agents to finally integrate multimodal information for generating accurate answers. For reference results, the outcomes obtained by inputting ground-truth pages into the VLM are regarded as the performance upper bound, so as to fully measure the model's performance level.

Metrics. GPT-4 [18] is utilized as the evaluation model to assess generated answers against ground-truth answers. Each answer is scored on a 1–5 scale, with a score of 4 or higher considered correct. Ultimately, the key accuracy metric is the ratio of correct answers to the total number of questions.

Implementation Details. To ensure fair comparison, the MCIR-RAG framework and all baseline methods adopt a unified core model configuration. The visual embedding model uniformly uses ColQwen-2.5 [4] to ensure consistent page-level feature extraction, and all modules requiring a VLM adopt Qwen2.5-VL-32B to maintain consistent generative and semantic understanding capabilities. For the baseline MDocAgent, the configuration strictly follows the original paper: ColBERTv2 [16] is used for text retrieval to ensure efficiency and

TABLE I
GENERATION PERFORMANCE (ACCURACY, %) ON 4 DIFFERENT DOCVQA DATASETS. HERE, "PG. RET." REPRESENTS THE ACTUAL NUMBER OF PAGES USED DURING THE GENERATION PROCESS.

Method	Pg. Ret.	MMLongBench	LongDocUrl	PaperTab	FetaTab	Avg. Acc
VLMs						
Qwen2.5-VL-32B-Instruct + Ground-Truth pages	–	69.21	80.36	-	-	-
M3DocRAG (Qwen2.5-VL-32B)	3	44.82	51.53	51.21	76.51	56.02
MDocAgent (Qwen3-32B + Qwen2.5-VL-32B)	6	51.61	58.67	50.36	81.27	60.47
M3DocRAG (Qwen2.5-VL-32B)	5	45.19	53.41	58.36	80.26	59.30
MDocAgent (Qwen3-32B + Qwen2.5-VL-32B)	10	53.31	60.15	62.71	83.41	64.89
M3DocRAG (Qwen2.5-VL-32B)	7	44.36	51.26	55.67	77.61	56.95
MDocAgent (Qwen3-32B + Qwen2.5-VL-32B)	14	50.17	58.22	60.81	82.91	63.02
Ours (ColQwen-2.5, retrieval top-10 and top-20)						
MCIR-RAG (Qwen3-32B + Qwen2.5-VL-32B)	2.4	58.46	69.12	75.62	83.21	71.58
MCIR-RAG (Qwen3-32B + Qwen2.5-VL-32B)	**2.8**	**60.68**	**70.06**	**76.34**	**84.29**	**72.84**

accuracy, and Qwen3-32B [17] is used for processing retrieved text and generating answers. In the MCIR-RAG framework, models are configured according to module-specific requirements: Qwen2.5-VL-32B is used in the summary generation stage to produce query-focused summaries from candidate pages as input to the Text Chain Agent; Qwen3-32B is used in the page re-ranking stage to refine the initial candidate pages and obtain the final Top-N pages, where Top-N is dynamically determined by the text model based on semantic confidence rather than fixed; the Image Chain Agent in the core reasoning module employs Qwen2.5-VL-32B to perform multi-step visual reasoning directly on the original page images; the Text Chain Agent and Verification Agent both use Qwen3-32B to assist reference answer generation and perform semantic consistency verification. Regarding experimental parameters, baseline Top-K values are set to 3, 5, and 7 to evaluate performance under different retrieval scales, while MCIR-RAG sets initial screening Top-K to 10 and 20 to ensure sufficient candidate coverage, and the maximum number of iterations for the reasoning and iterative reflection mechanism is set to 2, balancing answer quality and computational efficiency.

B. Qualitative Results and Analysis

As shown in Table I, the MCIR-RAG model consistently outperforms baseline methods on end-to-end generation tasks across multiple datasets. Notably, on three datasets, MCIR-RAG achieves performance improvements of over 10% compared to **MDocAgent**, demonstrating its strong effectiveness in document-based question answering under long-context and multimodal settings.

These gains mainly stem from the integrated three-stage design of MCIR-RAG: *Retrieval Re-ranking, Multi-Agent Reasoning, and Iterative Reflection*. Specifically:

- **Retrieval Re-ranking:** A language-model-guided dynamic Top-N strategy, together with VLM-generated query-focused summaries, improves retrieval precision while reducing irrelevant or redundant pages, thereby providing higher-quality evidence for downstream reasoning.
- **Multi-Agent Reasoning:** The Image Chain Agent conducts primary visual-grounded reasoning, while the Text Chain Agent offers auxiliary reference guidance. This collaborative design enables the construction of more coherent and accurate reasoning chains, mitigating errors caused by incomplete or noisy contextual information.
- **Iterative Reflection:** By iteratively checking semantic consistency between preliminary and reference answers, the model is able to identify and correct potential reasoning errors, leading to more reliable and well-grounded final outputs.

In contrast, baseline methods often suffer performance degradation when more pages are retrieved (e.g., Top-7 vs. Top-5), which can be attributed to the accumulation of redundant or weakly relevant information that hinders effective evidence selection. MCIR-RAG, however, demonstrates efficient retrieval and reasoning behavior: on average, only **2.8 pages per query** are sufficient to achieve strong generation performance. This highlights the robustness of the dynamic Top-N strategy and the effectiveness of the multi-agent framework in balancing information coverage and noise suppression. Overall, MCIR-RAG achieves superior accuracy, better information utilization, and improved computational efficiency compared with existing baselines.

TABLE II
RETRIEVAL PERFORMANCE ON MMLONGBENCH

Method	Avg Ret. Pages	All Hit %	F1 Score
ColQwen-2.5	3	60.69	43.01
ColQwen-2.5	5	68.22	33.94
ColQwen-2.5	7	**72.64**	28.60
Ours (top-10)	2.4	64.24	57.04
Ours (top-20)	2.8	66.43	**58.54**

C. Retrieval Evaluation and Analysis

To further evaluate the retrieval performance of MCIR-RAG, Table II reports two core metrics: *All Hit %* and *Page-level F1 Score*. Here, *All Hit %* measures whether the retrieved pages fully cover the golden evidence set for each query, focusing on completeness, while the *Page-level F1 Score* combines precision and recall to provide a comprehensive

TABLE III
PERFORMANCE COMPARISON ACROSS DIFFERENT MCIR-RAG'S VARIANTS

Variants	Agent Configuration			Evaluation Benchmarks			
	Text Chain Agent	Image Chain Agent	Verification Agent	MMLongBench	LongDocUrl	PaperTab	FetaTab
MCIR-RAG$_i$	✗	✓	✗	53.01	64.70	69.21	80.65
MCIR-RAG$_t$	✓	✗	✗	55.83	63.93	70.65	81.39
MCIR-RAG	✓	✓	✓	**60.68**	**70.06**	**76.34**	**84.29**

assessment of retrieval quality in terms of both accuracy and coverage.

The results indicate that for the ColQwen-2.5 model, as the Top-K value increases from 3, 5 to 7, *All Hit %* shows a slight improvement, whereas the *F1 Score* consistently decreases. This suggests that the additional retrieved pages introduce a substantial amount of irrelevant information, which negatively affects overall retrieval effectiveness. In contrast, MCIR-RAG retrieves only 2.8 pages on average per query (with Top-K set to 10 and 20 in the preliminary screening stage, and the final Top-N pages dynamically selected by the model in the re-ranking stage), yet achieves a high page-level F1 score of 58.54%. This demonstrates that the retrieval stage of MCIR-RAG can accurately provide high-quality evidence with a minimal number of pages, effectively supporting the subsequent multi-agent reasoning and iterative reflection processes.

D. Ablation Studies

Table III presents the performance comparison between the complete MCIR-RAG and its ablated variants: MCIR-RAG$_i$ (utilizing only the Image Chain Agent) and MCIR-RAG$_t$ (utilizing only the Text Chain Agent). Across benchmark tests on the four datasets, even the individual variants outperform the baseline methods, which demonstrates the effectiveness of the chain-of-thought reasoning mechanism within each modality.

Notably, MCIR-RAG$_i$ consistently achieves higher scores than MCIR-RAG$_t$, highlighting that image-grounded reasoning provides a more accurate and reliable basis for DocVQA tasks. The full MCIR-RAG framework surpasses both individual variants, indicating that the synergistic interaction between the Image and Text Chain Agents contributes additional complementary knowledge, while the iterative verification and reflection mechanism further refines reasoning by resolving conflicts and reinforcing evidence grounding.

These results suggest that the multi-agent design not only allows different modalities to provide distinct perspectives but also leverages cross-modal guidance and iterative self-correction, leading to a substantial improvement in answer accuracy and robustness compared to single-agent or non-reflective configurations.

E. Fine-Grained Performance Analysis

As shown in Figure 2, we conduct a fine-grained analysis of MCIR-RAG's performance across different evidence types on the MMLongBench dataset. The results show that MCIR-RAG consistently outperforms baseline methods across all

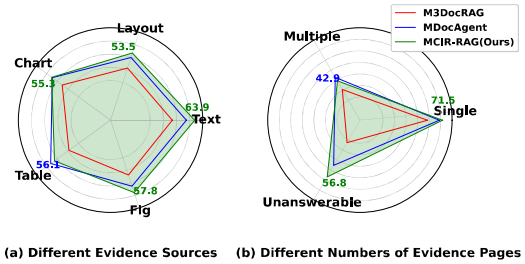

Fig. 2. Performance on different types of queries in MMLongBench.

evidence modes. A detailed analysis along different evidence dimensions reveals the following insights:

- **By evidence source:** Across different evidence types, including Text, Layout, Chart, Table, and Figure, MCIR-RAG achieves the best performance in four out of five modes. In contrast, MDocAgent is only competitive in the Table mode, indicating that MCIR-RAG more effectively integrates heterogeneous visual and layout information through multi-agent reasoning.
- **By evidence page count:** For single-page questions, both MCIR-RAG and MDocAgent perform strongly, suggesting that simple scenarios are well handled by existing methods. For multi-page questions, MCIR-RAG exhibits slightly lower accuracy than MDocAgent, likely due to increased reasoning complexity and evidence dispersion, while still maintaining competitive performance.
- **Unanswerable scenarios:** On questions with no valid supporting evidence, MCIR-RAG significantly outperforms all baselines. This demonstrates its strong anti-hallucination capability, enabled by the iterative reflection mechanism that helps the model detect missing evidence and avoid unsupported predictions.

TABLE IV
DIFFERENT ITERATIONS ON MMLONGBENCH

Iteration	1	2	3
Accuracy	58.28	60.68	59.67

F. Iterative Reflection Analysis

Table IV presents the performance of our iterative reflection strategy on the MMLongBench dataset under different numbers of iterations. The results indicate that the strategy achieves optimal performance when the number of iterations is set to 2, with an accuracy of 60.68%, representing a notable improvement over the single-iteration scenario (58.28%).

This improvement can be attributed to the design of the iterative reflection mechanism: the first iteration primarily corrects major inconsistencies between the Image Chain Agent and the reference answer, while the second iteration further refines minor conflicts and consolidates reasoning consistency. Beyond the second iteration, additional passes lead to marginal or even slightly decreased performance (59.67% at iteration 3), likely due to over-adjustment or the introduction of redundant modifications. Overall, these results demonstrate that a limited number of iterations is sufficient to enhance answer reliability while maintaining computational efficiency.

V. CONCLUSIONS

We propose the MCIR-RAG framework, which enhances complex DocVQA performance through a three-stage process: *Retrieval Re-Ranking*, *Multi-Agent Reasoning*, and *Iterative Reflection*. Experimental results on multiple benchmark datasets show that MCIR-RAG consistently outperforms existing RAG-based methods in accuracy and robustness, while requiring only a small number of retrieved pages. Ablation studies further confirm the effectiveness of each component, demonstrating the benefits of multi-agent collaboration and iterative refinement. Overall, MCIR-RAG provides an efficient and reliable solution for document-level visual question answering.

REFERENCES

[1] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, vol. 2, no. 1, 2023.

[2] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," IEEE Access, vol. 9, pp. 157727–157760, 2021.

[3] R. Smith, "An overview of the Tesseract OCR engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633.

[4] M. Faysse, H. Sibille, T. Wu, B. Omrani, G. Viaud, C. Hudelot, and P. Colombo, "ColPali: Efficient Document Retrieval with Vision Language Models," arXiv preprint arXiv:2407.01449, 2025.

[5] S. Yu, C. Tang, B. Xu, J. Cui, J. Ran, Y. Yan, Z. Liu, S. Wang, X. Han, Z. Liu, and M. Sun, "VisRAG: Vision-based Retrieval-augmented Generation on Multi-modality Documents," arXiv preprint arXiv:2410.10594, 2025.

[6] R. Tanaka, T. Iki, T. Hasegawa, K. Nishida, K. Saito, and J. Suzuki, "VDocRAG: Retrieval-Augmented Generation over Visually-Rich Documents," arXiv preprint arXiv:2504.09795, 2025.

[7] J. Cho, D. Mahata, O. Irsoy, Y. He, and M. Bansal, "M3DocRAG: Multi-modal Retrieval is What You Need for Multi-page Multi-document Understanding," arXiv preprint arXiv:2411.04952, 2024.

[8] S. Han, P. Xia, R. Zhang, T. Sun, Y. Li, H. Zhu, and H. Yao, "MDocAgent: A Multi-Modal Multi-Agent Framework for Document Understanding," arXiv preprint arXiv:2503.13964, 2025.

[9] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," arXiv preprint arXiv:2308.08155, 2023.

[10] Y. Zheng, D. Fu, X. Hu, X. Cai, L. Ye, P. Lu, and P. Liu, "DeepResearcher: Scaling Deep Research via Reinforcement Learning in Real-world Environments," arXiv preprint arXiv:2504.03160, 2025.

[11] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," arXiv preprint arXiv:2201.11903, 2023.

[12] M. Suri, P. Mathur, F. Dernoncourt, K. Goswami, R. A. Rossi, and D. Manocha, "VisDoM: Multi-Document QA with Visually Rich Elements Using Multimodal Retrieval-Augmented Generation," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Albuquerque, New Mexico, 2025, pp. 6088–6109.

[13] Z. Wang, W. Yu, X. Ren, J. Zhang, Y. Zhao, R. Saxena, L. Cheng, G. Wong, S. See, P. Minervini, Y. Song, and M. Steedman, "MMLongBench: Benchmarking Long-Context Vision-Language Models Effectively and Thoroughly," arXiv preprint arXiv:2505.10610, 2025.

[14] C. Deng, J. Yuan, P. Bu, P. Wang, Z.-Z. Li, J. Xu, X.-H. Li, Y. Gao, J. Song, B. Zheng, and C.-L. Liu, "LongDocURL: a Comprehensive Multimodal Long Document Benchmark Integrating Understanding, Reasoning, and Locating," arXiv preprint arXiv:2412.18424, 2025.

[15] Y. Hui, Y. Lu, and H. Zhang, "UDA: A Benchmark Suite for Retrieval Augmented Generation in Real-world Document Analysis," arXiv preprint arXiv:2406.15187, 2024.

[16] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," arXiv preprint arXiv:2004.12832, 2020.

[17] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 Technical Report," arXiv preprint arXiv:2505.09388, 2025.

[18] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[19] Y. Kim, C. Park, H. Jeong, Y. S. Chan, X. Xu, D. McDuff, H. Lee, M. Ghassemi, C. Breazeal, and H. W. Park, "MDAgents: An Adaptive Collaboration of LLMs for Medical Decision-Making," arXiv preprint arXiv:2404.15155, 2024.

[20] G. Li, H. A. Kader Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society," arXiv preprint arXiv:2303.17760, 2023.

[21] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-VL Technical Report," arXiv preprint arXiv:2502.13923, 2025.

[22] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," arXiv preprint arXiv:2303.11366, 2023.

[23] W. Shi, S. Min, M. Yasunaga, M. Seo, R. James, M. Lewis, L. Zettlemoyer, and W.-t. Yih, "REPLUG: Retrieval-Augmented Black-Box Language Models," arXiv preprint arXiv:2301.12652, 2023.

[24] Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang, and H. Wang, "PP-OCR: A Practical Ultra Lightweight OCR System," arXiv preprint arXiv:2009.09941, 2020.

[25] Z. Chen, K. Liu, Q. Wang, J. Liu, W. Zhang, K. Chen, and F. Zhao, "MindSearch: Mimicking Human Minds Elicits Deep AI Searcher," arXiv preprint arXiv:2407.20183, 2025.

[26] J. Wu, W. Yin, Y. Jiang, Z. Wang, Z. Xi, R. Fang, L. Zhang, Y. He, D. Zhou, P. Xie, and F. Huang, "WebWalker: Benchmarking LLMs in Web Traversal," arXiv preprint arXiv:2501.07572, 2025.

[27] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation," arXiv preprint arXiv:2402.03216, 2025.

[28] C. Lee, R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, and W. Ping, "NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models," arXiv preprint arXiv:2405.17428, 2025.

APPENDIX A
EXPERIMENTAL ENVIRONMENTS

All experiments were conducted on a high-performance server equipped with 8 NVIDIA A800 GPUs and 64 CPU

cores. Due to the large-scale parameters and multimodal processing requirements of the open-source vision–language models used in this work, substantial computational resources are required to ensure stable training and efficient inference.

APPENDIX B
DETAILED RETRIEVAL METRIC CALCULATION

To provide a transparent and fine-grained evaluation of retrieval performance, we adopt two complementary metrics: *All-hit Rate* and *Page-level F1 Score*. These metrics jointly assess the completeness and precision–recall trade-off of retrieved document pages.

A. All-hit Rate

The All-hit Rate measures whether the retrieved page set fully covers the ground-truth evidence pages required to answer a given question. Let $\mathcal{G}$ denote the golden evidence page set for a query, and $\mathcal{R}$ denote the retrieved page set. The All-hit indicator for a single query is defined as:

$$\text{AllHit} = \begin{cases} 1, & \text{if } \mathcal{G} \subseteq \mathcal{R}, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The All-hit Rate over the dataset is computed as the average of the AllHit indicator across all queries:

$$\text{All-hit Rate} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \text{AllHit}(q), \tag{8}$$

where $\mathcal{Q}$ denotes the set of evaluation queries.

This metric focuses on retrieval completeness and reflects whether all necessary evidence pages are successfully retrieved, regardless of additional irrelevant pages.

B. Page-level F1 Score

To further evaluate retrieval quality by considering both relevance and redundancy, we compute the Page-level F1 Score. For each query, precision and recall are defined at the page level as:

$$\text{Precision} = \frac{|\mathcal{R} \cap \mathcal{G}|}{|\mathcal{R}|}, \quad \text{Recall} = \frac{|\mathcal{R} \cap \mathcal{G}|}{|\mathcal{G}|}. \tag{9}$$

Based on these definitions, the Page-level F1 Score is computed as:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

The final Page-level F1 Score is obtained by averaging the F1 scores over all evaluation queries. This metric provides a balanced evaluation of retrieval effectiveness, penalizing both missing relevant pages and retrieving excessive irrelevant pages.

APPENDIX C
PROMPT DETAILS

This appendix introduces the text prompts used in the MCIR-RAG framework, including those for summary generation, the Text Chain Agent, the Image Chain Agent, and the Verification Agent, highlighting their roles in evidence extraction, reasoning, and semantic consistency verification.

Summary Generation Prompt

You are tasked with creating a comprehensive summary of a given page from a document. The summary should be tailored to focus on the specific topics or keywords present in the user's query. Follow these steps:

Task (strictly follow):

- **Understand the Query Focus:** Begin by reviewing the keywords or main topics in the query to identify the focus for your summary.
- **Analyze the Page Content:** Carefully read and analyze the page, ensuring to pay extra attention to sections related to the query's keywords.
- **Extract Key Points:** Identify the main topics, key points, and important details that are relevant to the query focus. Emphasize these elements in the summary.
- **Visual Elements:** Note any tables, figures, charts, diagrams, or images on the page that relate to the query. Provide brief descriptions of their content and relevance to the main topics.

Structured Summary: Create a structured summary that includes:

- Textual Information: Extract the essential details related to the query focus.
- Visual Elements: Describe any tables, figures, images, etc., in the context of the query topics.
- Additional Context: Include any unique or particularly relevant information that adds value to the query's context.

Ensure the summary is concise, focused on the query, and well-organized. The summary should typically be between 4–7 sentences for text-heavy pages, and more if visual elements require additional explanation.

For visual elements, please use these specific tags:

- <table_summary> for descriptions of tables related to the query
- <figure_summary> for descriptions of figures, charts, graphs, or diagrams relevant to the query
- <image_summary> for descriptions of photos, illustrations, or other images that support the query focus

Example structure:

- <summary>
- [Main text content summary focused on query keywords here]
- <table_summary> Table 1: [Brief description of what the table shows related to the query focus] </table_summary>
- <figure_summary> Figure 2: [Brief description of what the figure depicts in relation to the query topic] </figure_summary>
- <image_summary>[Brief description of image content in the context of the query] </image_summary>
- </summary>

The summary should always prioritize the specific query focus while capturing both textual and visual content from the page. The entire output must be wrapped within the <summary> tag, including all visual element tags.

Fig. 3. Summary Generation Prompt.

Text Chain Agent Prompt

You are a Text Reasoning Chain Agent, dedicated to strictly reasoning based on the provided summary (no external knowledge). Complete the following tasks in order:

Task:

1. **Evidence Extraction:** Extract only 3-5 key sentences/phrases directly related to the query from the summary. Ignore all irrelevant content.
2. **Reasoning Chains:** Provide a step-by-step logical derivation (1-3 steps) — each step must explicitly cite the extracted evidence (e.g., "From Evidence 1: '...' → [conclusion]").
3. **Preliminary Answer":** Generate the final answer strictly based on the reasoning chain. If information is insufficient/missing, output "Not answerable".

Input Format:

- Query: {user_query}
- Summary: {retrieved_summary}

Output Format (strictly follow, no extra text):

- Evidence Extraction: [List 1-3 key sentences/phrases, e.g., "The framework includes 3 agents: Reranker, Generator, Verifier"]
- Reasoning Chain: [Step 1: [content] → Step 2: [content] → ...]
- Preliminary Answer: [Final answer or "Not answerable"]

Fig. 4. Text Chain Agent Prompt.

Image Chain Agent Prompt

You are an Image Reasoning Chain Agent, only reasoning based on visually observable content of the provided image (no speculation). Complete the tasks:

Task:

1. **Evidence Extraction:** Extract visible content by regions (e.g., "Top-left text: '...'; Middle chart: 'X-axis: [label], Y-axis: [label]'"). For blurred/occluded areas, mark them as "Unclear: [region description]".
2. **Reasoning Chains:** Derive conclusions step-by-step — each step must cite the extracted visual evidence (e.g., "From Region 2: '...' → [conclusion]").
3. **Preliminary Answer":** Output the answer based on the chain. If no sufficient visual support, output "Not answerable".

Input Format:

- Query: {user_query}
- Image Content Description: {image_ocr_visual_description}

Output Format (strictly follow):

- Evidence Extraction: [Region 1: [content]; Region 2: [content]; ...; Unclear Regions: [list if any]]
- Reasoning Chain: [Step 1: [content] → Step 2: [content] → ...]
- Preliminary Answer: [Final answer or "Not answerable"]

Fig. 5. Image Chain Agent Prompt.

Verification Agent Prompt

You are a Moderator Agent: integrate Text Chain and Revised Image Chain results to generate the final answer. Follow these steps:

Task:

- **Result Alignment Check:** Compare the two preliminary answers:
- - Case 1: Answers are consistent → directly output the answer
- - Case 2: Answers conflict → analyze the evidence reliability (prioritize: "clear visual evidence" > "detailed text summary" > "ambiguous evidence")
- **Final Decision:** Generate the final answer + explain the integration logic.

Input Format:

- Text Chain Result: {text_answer_with_evidence}
- Revised Image Chain Result: {revised_image_answer_with_reflection}

Output Format:

- Reflection Process: [Comparison Analysis: {conflicts}; Self-Inspection: {check_results}; Revision Logic: {reason}]
- Revised Answer: [Updated answer / Original answer / "Not answerable"]

Fig. 6. Verification Agent Prompt.