DAMG-RAG: Document-Aware Multi-Granularity RAG for Visually Rich Document Understanding

Jiahao Gu and Ningjing Liu and Yuanzhuo Ding and Siheng Chen
Shanghai Jiao Tong University
Shanghai, China
{gu1610755435, woodserenity, ssansjhicvc, sihengc}@sjtu.edu.cn
Correspondence: sihengc@sjtu.edu.cn

Abstract

Real-world documents are visually rich and structurally organized, with text, tables, figures, and captions interleaved across pages. Although retrieval-augmented generation (RAG) is widely used for document question answering, existing pipelines often treat such documents as fixed page- or chunk-level collections. This document-agnostic construction either dilutes evidence in dense pages or fragments semantically related layout elements. Moreover, cross-page questions often require iterative evidence exploration, but existing systems typically follow fixed retrieval paths that do not adapt to the evidence already found.

We propose DAMG-RAG, a Document-Aware Multi-Granularity RAG framework for visually rich document understanding. During ingestion, DAMG-RAG constructs layout-aware atomic retrieval units and links them to page- and section-level nodes, forming a multi-granularity document tree. During inference, a state-aware agentic workflow maintains explored regions and validated evidence, dynamically selecting exploration and navigation tools over this tree. On MMLongBench, DAMG-RAG obtains 64.7% accuracy with Claude Sonnet 4.6, yielding a 3.6-point gain over the best-performing agentic baseline in our experiments while using 18.8% of its token budget.

1 Introduction

Large language models (LLMs) have made natural-language interaction with complex documents increasingly practical (Yue, 2025). Many real-world workflows, such as financial analysis (Li et al., 2023b), legal review (Chalkidis et al., 2020), scientific research (Wang et al., 2024c), and enterprise knowledge management, require question answering over visually rich documents. As shown in Figure 1, visually rich documents interleave text, tables, figures, captions, and layout structures across pages, with evidence organized differently across layout patterns. Although multimodal LLMs such as GPT-4o (Hurst et al., 2024) can directly process visual inputs, their performance degrades on long documents because relevant evidence is diluted by large amounts of irrelevant content (Sharma et al., 2024). Retrieval-augmented generation (RAG) (Lewis et al., 2020) therefore remains a practical strategy for grounding long-document question answering in targeted evidence.

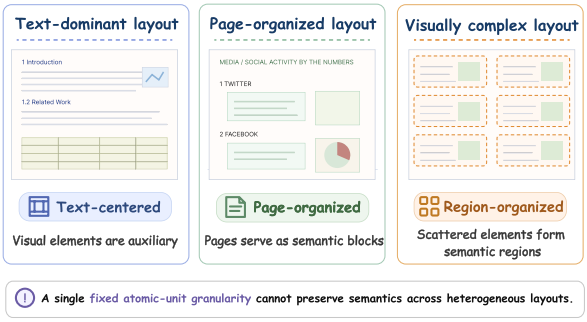


Figure 1: Examples of heterogeneous visually rich layouts with distinct text, table, and figure organization patterns.

However, applying RAG to such documents is nontrivial. In visually rich documents, relevant evidence is often embedded in the interactions among text, tables, figures, captions, and layout structures, rather than occurring in isolation (Ueda et al., 2026; Bhattacharyya et al., 2025). Effective retrieval therefore requires constructing evidence units that are semantically coherent across modalities and aligned with the document's structural and layout hierarchy (Shin et al., 2025). Furthermore, the retrieval process must be capable of leveraging diverse structural cues to guide targeted evidence collection (Sourati et al., 2025). Prior RAG systems often fall short in both aspects, giving rise to two major limitations.

1. **Document-agnostic retrieval units.** Standard RAG pipelines usually index documents at a fixed granularity, such as pages (Faysse

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1541–1560
November 6-10, 2026 ©2026 Association for Computational Linguistics

et al., 2024) or text chunks. Page-level units preserve visual context but dilute evidence in text-dense documents, while chunk-level units improve lexical specificity but may separate captions from figures, table cells from headers, or OCR (Smith, 2007) fragments from their surrounding layout. As a result, the indexed units often fail to match the semantic units needed for visually rich document QA.

2. **State-agnostic evidence exploration.** Many questions require following structural cues across pages, such as section hierarchy, table or figure references, and page-level visual signals (Wang et al., 2025; Sun et al., 2025). Existing RAG systems often use predefined retrieval paths and do not condition subsequent searches on the evidence collected so far (Du et al., 2026). This makes it difficult to decide when to expand local context, inspect a related section, follow a reference, or stop searching.

To address these limitations, we propose DAMG-RAG, a document-aware multi-granularity RAG framework for long visually rich document QA. It constructs document-adaptive retrieval units and uses LLM agents to perform state-aware evidence exploration across multiple document granularities.

Our design is motivated by the complementary strengths of retrieval tools and human document reading. Retrieval tools can efficiently localize candidate regions. Human readers, by contrast, are effective at making state-dependent reading decisions. DAMG-RAG combines these strengths by using retrieval tools to quickly locate candidate regions, while allowing LLM agents to decide how to further explore evidence within a document-aware structure.

This idea is implemented in two stages. During ingestion, the framework determines suitable atomic retrieval units based on the document's layout characteristics. It then extracts a hierarchical outline, aligns atomic units with pages and sections, and enriches them with concise contextual semantics. The resulting document tree preserves semantically coherent atomic units while connecting them to higher-level document structure, providing the basis for multi-granularity retrieval and navigation.

At inference time, DAMG-RAG conducts state-aware evidence exploration over the document tree through a multi-agent workflow. It maintains a persistent memory of explored regions and validated evidence as a global search state (Park et al., 2023). Conditioned on this state, the workflow determines the appropriate retrieval granularity and invokes the corresponding tools (Yao et al., 2023) to explore the document tree, and refines retrieved candidates through staged filtering and verification. As a result, retrieval is formulated as an adaptive, multi-granularity evidence exploration process rather than a fixed-path matching procedure.

Our contributions can be summarized as follows:

- We propose a document-aware ingestion strategy that constructs layout-aware atomic units and links them to page- and section-level nodes, yielding a multi-granularity document tree for visually rich documents.
- We introduce a state-aware evidence exploration workflow that maintains explored regions and validated evidence, enabling the system to choose among retrieval, context expansion, and structure-guided navigation tools during inference.
- We evaluate DAMG-RAG on MMLongBench and show that it outperforms strong agentic baselines by 3.5 and 3.6 accuracy points with GPT-4o and Claude Sonnet 4.6 (Anthropic, 2026), respectively, while using only 18.8% of the token budget.

2 Related Work

2.1 RAG for Visually Rich Documents

Traditional RAG methods (Lewis et al., 2020; Karpukhin et al., 2020) for document question answering primarily operate over textual chunks, which limits their ability to exploit visual and layout information in visually rich documents. To incorporate visual evidence, recent methods (Faysse et al., 2024) have introduced page-level visual retrieval models, where document pages are encoded as images and retrieved according to their visual relevance to the query. These methods improve access to visual information, but they often treat an entire page as the retrieval unit, which can be too coarse for text-dense or structurally complex documents.

Another line of work (Wan and Yu, 2025; Du et al., 2026) relies on OCR or document parsing tools (PyMuPDF Developers, 2026) to decompose pages into fine-grained textual or visual elements,

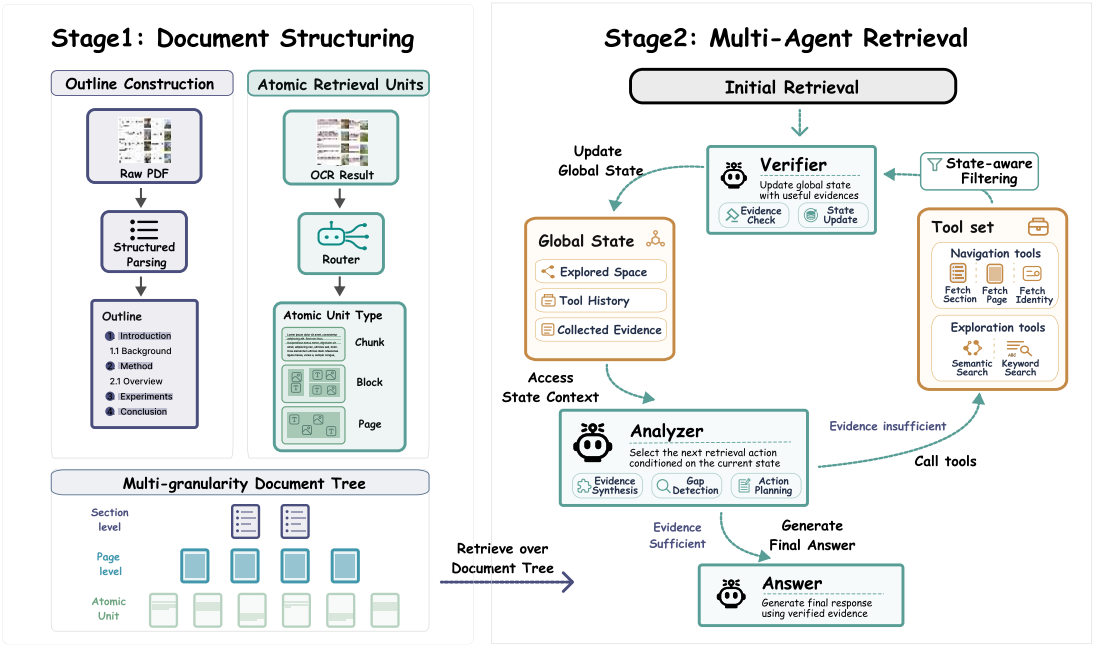


Figure 2: Overview of DAMG-RAG. Document structuring builds a multi-granularity document tree from outline and atomic retrieval units, while multi-agent retrieval uses a global search state to guide tool-based evidence exploration.

which are then indexed as retrieval chunks. These methods offer localized access, but their predefined units still provide only a single structural view, making it difficult to capture evidence spanning local semantic blocks and higher-level document structure.

2.2 Multi-Agent Systems for Document Understanding

Multi-agent systems have shown promise in complex reasoning tasks by assigning specialized roles to different agents and coordinating their interactions (Hong et al., 2024; Li et al., 2023a; Wu et al., 2024). In document understanding area, recent methods typically instantiate agents according to either modality or functional role. Some systems (Han et al., 2025) use modality-specific agents to process textual and visual evidence separately before answer aggregation, while others adopt iterative seeker-reviewer workflows (Sun et al., 2025; Wang et al., 2025), where a seeker agent retrieves candidate evidence and a reviewer agent verifies answer-evidence consistency and determines whether additional retrieval is needed.

Although these designs improve multimodal processing and introduce deliberation before retrieval, they often rely on limited tools and operate within a single document granularity. They therefore struggle to explore different document regions in a targeted way and lack an explicit global search state over explored regions and accumulated evidence.

3 Method

Given a visually rich document and a question, DAMG-RAG operates in two stages, as illustrated in Figure 2. First, Document Structuring constructs atomic units according to layout and content characteristics, aligns them with page and section nodes, and forms a multi-granularity document tree (Section 3.1). Second, Multi-Agent Retrieval maintains a global search state and iteratively selects exploration or navigation tools to collect, filter, and verify evidence before answer generation (Section 3.2).

3.1 Document Structuring

Document-Aware Atomic Unit Construction. DAMG-RAG first routes each document to an appropriate atomic-unit construction strategy. This routing step is motivated by the observation that OCR chunks can directly serve as reliable units in well-structured documents, but may become fragmented or semantically misaligned in visually complex layouts.

Given the initial OCR-derived chunks, we compute a lightweight page-level visual-complexity score based on image-chunk ratio and chunk density. The top-K pages are then converted into lay-

out visualizations, where chunk bounding boxes are rendered with distinct colors. A VLM-based router assesses whether these pages are best represented by parser chunks, whole pages, or visually grouped blocks, and assigns the document to the corresponding construction mode:

$$\tau = \mathrm{Router}(\mathcal{P}_K^{\mathrm{layout}}), \quad \tau \in \{\tau_{\mathrm{chunk}}, \tau_{\mathrm{page}}, \tau_{\mathrm{group}}\}.$$

Here, $\mathcal{P}_K^{\mathrm{layout}}$ denotes the selected top-K layout pages, and τ selects one of three construction strategies:

(1) Chunk-based strategy (τ_{chunk}): Atomic units are constructed from the original chunks when textual content is dominant and the initial layout is already sufficiently regular. Following the standard RAG setting, text is segmented into fixed-size overlapping chunks, while images are preserved as independent visual chunks.

(2) Page-based strategy (τ_{page}): Atomic units are constructed from entire pages when each page naturally forms a coherent semantic boundary. This strategy is suitable for slide-like documents, where the page layout typically organizes content into complete topical units.

(3) VLM-grouping strategy (τ_{group}): This strategy is used when the original chunks are fragmented or poorly aligned with semantic boundaries. Instead of treating each chunk independently, we ask a VLM to identify groups of chunks that should form coherent atomic units. Given the layout visualization $\mathcal{P}^{\mathrm{layout}}$ and the original chunk set $\mathcal{X} = \{x_i\}_{i=1}^N$, the VLM predicts a set of chunk groups:

$$\mathcal{G} = \{g_j\}_{j=1}^M = \mathrm{VLM}_{\mathrm{group}}(\mathcal{X}, \mathcal{P}^{\mathrm{layout}}), \quad g_j \subseteq \mathcal{X}.$$

Each group g_j contains chunks that are spatially adjacent and semantically related. We then merge the bounding boxes of chunks in g_j and crop the corresponding page region to construct one multimodal atomic unit.

Atomic Unit Representation. After construction, each atomic unit is represented as a retrieval node. To support both semantic and lexical retrieval, DAMG-RAG annotates each node with a compact semantic description and a set of salient keywords generated by the VLM. The semantic description is used for dense retrieval, while the keywords support explicit lexical lookup in the multi-agent retrieval stage.

Multi-Granularity Document Tree Construction. After constructing and annotating atomic units, DAMG-RAG organizes them into a document tree that connects atomic units with their page and section-level context. In our implementation, we obtain the document outline from the structured output of Adobe PDF Extract API (Adobe), and align each outline node with its corresponding pages. Atomic units are then attached to the page nodes from which they are constructed. The outline source is replaceable: any OCR-derived text hierarchy can also be used as the outline.

Formally, the resulting document tree is defined as

$$\mathcal{T} = (\mathcal{V}, \mathcal{L}), \qquad \mathcal{V} = \mathcal{V}^{\mathrm{sec}} \cup \mathcal{V}^{\mathrm{page}} \cup \mathcal{V}^{\mathrm{atom}}.$$

Here, $\mathcal{V}^{\mathrm{sec}}$, $\mathcal{V}^{\mathrm{page}}$, and $\mathcal{V}^{\mathrm{atom}}$ denote section units, page units, and atomic units, respectively. The link set $\mathcal{L}$ is induced by natural containment relations. This tree provides a unified multi-granularity structure for evidence retrieval and navigation.

3.2 Multi-Agent Retrieval

Global Search State. During inference, DAMG-RAG maintains an explicit global search state over the document tree $\mathcal{T} = (\mathcal{V}, \mathcal{L})$, tracking explored nodes, accumulated evidence and previous actions. Before step t, the state is defined as

$$S_t = \left(\mathcal{V}_t^{\mathrm{exp}}, \mathcal{E}_t, \{\mathcal{A}_i\}_{i=1}^{t-1}\right).$$

Here, $\mathcal{V}_t^{\mathrm{exp}} \subseteq \mathcal{V}$ denotes the set of nodes explored before step t, which is used to avoid redundant retrieval. $\mathcal{E}_t$ stores the accumulated evidence extracted from verified useful nodes, while $\{\mathcal{A}_i\}_{i=1}^{t-1}$ records previous retrieval actions. This state provides the global context for subsequent agent planning.

State-Guided Search Planning. To enable structured planning over the document tree, we define a compact toolset $\mathcal{U}$ as the agent action space:

$$\mathcal{U} = \mathcal{U}^{\mathrm{exp}} \cup \mathcal{U}^{\mathrm{nav}}.$$

Exploration tools $\mathcal{U}^{\mathrm{exp}}$ introduce new evidence candidates by searching over atomic units outside the current retrieval trajectory. Specifically, SEMANTIC SEARCH retrieves units using dense similarity over semantic descriptions, while KEYWORD SEARCH performs lexical lookup over salient keywords.

Navigation tools $\mathcal{U}^{\text{nav}}$ instead use accumulated evidence or structural cues to move within the document tree. FETCH IDENTITY fetches referenced figures, tables, or other identified elements; FETCH PAGE retrieves page-level context; and FETCH SECTION expands retrieval to section-level context. Detailed tool implementations are provided in Appendix A.2.

Within the multi-agent workflow, the analyzer performs state-guided search planning over the toolset $\mathcal{U}$. At step t, it takes the question q, the global search state S_t, and the toolset $\mathcal{U}$ as input:

$$d_t, \mathcal{A}_t = \text{Analyzer}(q, S_t, \mathcal{U}).$$

The analyzer assesses the current evidence coverage, identifies missing information, and outputs a planning decision d_t together with a set of executable tool actions $\mathcal{A}_t$. Each action specifies a selected tool $u_t \in \mathcal{U}$ and its input arguments. If the accumulated evidence is sufficient, d_t indicates termination and $\mathcal{E}_t$ is passed to the answer generation stage.

Candidate Filtering and Evidence Verification. Given the planned tool actions $\mathcal{A}_t$, DAMG-RAG retrieves a candidate set $\mathcal{C}_t$ from the document tree. The candidates first undergo state-aware filtering, which removes units already recorded in the explored-node set $\mathcal{V}_t^{\text{exp}}$. The remaining candidates are then passed to an evidence verifier, which identifies units that directly support the current question or retrieval goal and extracts grounded evidence snippets from them.

After verification, all candidates retrieved in the current step are marked as explored, and the extracted evidence is merged into the evidence memory:

$$\mathcal{V}_{t+1}^{\text{exp}} = \mathcal{V}_t^{\text{exp}} \cup \mathcal{C}_t, \qquad \mathcal{E}_{t+1} = \mathcal{E}_t \cup \Delta\mathcal{E}_t.$$

The updated state S_{t+1} is then returned to the analyzer for subsequent search planning.

Answer Generation. When the analyzer terminates or the maximum round budget is reached, the answer generator receives the question and the accumulated evidence memory $\mathcal{E}_t$. It produces the final answer using only the verified evidence snippets.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on MMLongBench (Ma et al., 2024) and LongDocURL (Deng et al., 2025), two long-document QA benchmarks requiring evidence localization over lengthy PDFs. Their predominantly short factual answers further support reliable automatic evaluation. Table 1 summarizes the dataset statistics used in our experiments.

Dataset	MMLongBench	LongDocURL
Questions	1082	2325
Documents	135	396
Avg. Pages	47.5	85.6
Avg. Tokens	21214	43623

Table 1: Dataset statistics for MMLongBench and LongDocURL.

Baselines. We compare DAMG-RAG against two categories of baselines. (i) *Evidence-page oracle*, where the ground-truth evidence pages are directly provided to a large vision-language model (LVLM) for answer generation. (ii) *Agentic multimodal RAG baselines*, including ViDoRAG (Wang et al., 2025), DocAgent (Sun et al., 2025), and MDocAgent (Han et al., 2025). ViDoRAG adopts an iterative seek-and-review framework over candidate images. DocAgent conducts iterative document exploration through keyword- and section-based operations. MDocAgent retrieves both textual and visual candidates, delegates them to modality-specific agents, and aggregates the resulting answers. For fair comparison, all baselines are implemented with the same LVLM backbone and evaluated under the same evaluation protocol. Their token consumption is further analyzed in Section 4.4. Further implementation details of all baselines are provided in Appendix A.3.

Evaluation Metrics. We report end-to-end accuracy based on an LLM-as-a-judge evaluation protocol, following recent work that uses strong LLMs as scalable evaluators (Zheng et al., 2023; Liu et al., 2023). We use GPT-4o as the evaluator to judge whether each model prediction is semantically equivalent to the corresponding ground-truth answer. Details of the evaluator prompt and its agreement with human evaluation are provided in Appendix B.1.

Implementation Details. Key implementation settings are as follows. MinerU (Wang et al., 2024a) is employed for OCR and multimodal document parsing during the document pre-building stage. Dense retrieval is implemented with `text-embedding-3-small` (OpenAI, 2026). In

Method	MMLongBench							LongDocURL
	Academic	Brochure	Guidebook	Industry	Report	Tutorial	Average	Average
GPT-4o								
Oracle	45.0	57.0	56.4	66.2	54.0	76.1	57.1	64.5
ViDoRAG	28.4	35.0	52.9	45.6	40.4	56.5	43.2	60.4
MDocAgent	38.7	40.0	49.0	44.4	45.9	55.0	46.8	61.0
DocAgent	**43.7**	35.0	47.0	**51.8**	46.5	51.4	48.2	59.8
DAMG-RAG	39.7	**43.0**	**55.5**	48.1	**51.0**	**58.7**	**51.7**	**61.7**
Claude Sonnet 4.6								
Oracle	62.6	63.0	65.7	74.3	67.1	82.9	68.7	68.3
ViDoRAG	53.2	51.0	59.3	60.5	56.2	64.4	57.5	65.7
MDocAgent	51.3	53.0	52.3	50.6	48.6	58.7	52.2	64.5
DocAgent	57.8	54.0	57.4	60.9	61.3	60.2	61.1	62.9
DAMG-RAG	**60.5**	**62.0**	**62.1**	**64.0**	**63.1**	**70.3**	**64.7**	**66.6**

Table 2: End-to-end QA accuracy on MMLongBench and LongDocURL.

the multi-agent retrieval stage, we report DAMG-RAG with two backbone settings, where all agents are instantiated with GPT-4o or Claude Sonnet 4.6, respectively. The maximum number of agent interaction cycles is set to $T_{\max} = 3$. All prompts used in our framework are provided in Appendix A.1.

4.2 Main Results

Table 2 summarizes end-to-end QA accuracy on MMLongBench and LongDocURL under two backbone settings. The results highlight both the effectiveness of DAMG-RAG over retrieval-based baselines and its remaining gap to gold-evidence reasoning.

Overall Comparison with Agentic RAG Baselines. DAMG-RAG achieves the best average performance among agentic multimodal RAG baselines under both backbone settings. With GPT-4o, it obtains 51.7% average accuracy on MMLongBench, outperforming the strongest baseline DocAgent by 3.5 points. With Claude Sonnet 4.6, DAMG-RAG reaches 64.7%, exceeding DocAgent by 3.6 points and MDocAgent by 12.5 points. On LongDocURL, DAMG-RAG also achieves the strongest retrieval-based performance under both backbones. The smaller gains on LongDocURL suggest that broad aggregation over very long documents remains challenging.

Comparison with the Evidence-Page Oracle. The evidence-page oracle removes the retrieval bottleneck by directly providing gold evidence pages to the LVLM. With Claude Sonnet 4.6, DAMG-RAG reaches 64.7% on MMLongBench and 66.6% on LongDocURL, approaching the oracle scores of 68.7% and 68.3%, respectively. The oracle scores remain far from saturated, showing that evidence-grounded reasoning over long, visually rich documents is still challenging.

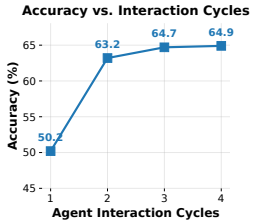
Setting	Aca.	Bro.	Gui.	Ind.	Rep.	Tut.	All
DAMG-RAG	60.5	62.0	62.1	64.0	63.1	70.3	**64.7**
w/o Exp.	58.7	63.0	60.1	63.2	58.9	68.1	62.5
w/o Nav.	54.0	62.0	56.0	53.8	58.3	65.2	59.2
w/o C.E.	59.4	57.0	59.0	61.2	56.6	67.2	60.4
w/o T.H.	54.8	54.0	57.1	56.3	59.4	63.8	57.9

Table 3: Ablation study of retrieval tools and the global state on MMLongBench with Claude Sonnet 4.6. Exp., Nav., C.E., and T.H. denote exploration tools, navigation tools, collected evidence, and tool history, respectively.

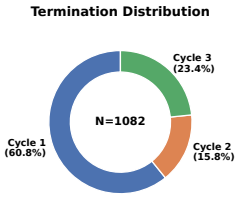
4.3 Ablation Studies

We conduct ablation studies on MMLongBench to assess the contribution of two key components in DAMG-RAG: the retrieval toolset and the adaptive atomic unit construction strategy. All ablations use Claude Sonnet 4.6 as the backbone model. Table 3 reports the tool ablation results, while Table 4 compares different atomic unit construction strategies.

Effect of Retrieval Tools and State Components. As shown in Table 3, removing any component degrades performance, confirming the contribution of both the retrieval tools and the search state. Removing exploration tools reduces accuracy from 64.7% to 62.5%, while removing navigation tools causes a larger drop to 59.2%, highlighting the importance of combining evidence discovery with structure-aware document navigation. The state components are also critical: removing collected evidence lowers accuracy to 60.4%, and removing tool history leads to the largest drop to 57.9%. These results



(a) Accuracy vs. cycles.



(b) Termination distribution.

Figure 3: Effect of agent interaction cycles on DAMG-RAG. Left: accuracy under different maximum cycle budgets. Right: termination distribution under the default setting of $T_{\max} = 3$.

suggest that DAMG-RAG relies not only on diverse retrieval operations, but also on an explicit search state that records verified evidence and previous tool usage to support evidence-conditioned retrieval across rounds.

Setting	Aca.	Bro.	Gui.	Ind.	Rep.	Tut.	All
Adaptive	60.5	62.0	62.1	64.0	63.1	70.3	**64.7**
Chunk-only	61.7	51.0	57.5	64.7	59.3	66.7	61.7
Page-only	56.9	58.0	60.7	60.5	59.9	70.1	61.8

Table 4: Ablation study of atomic unit construction strategies on MMLongBench with Claude Sonnet 4.6.

Effect of Atomic Unit Construction. We further evaluate the impact of adaptive atomic unit construction. As shown in Table 4, the adaptive strategy achieves the best overall accuracy of 64.7%, outperforming both chunk-only (61.7%) and page-only (61.8%) construction. This confirms that a fixed construction strategy is suboptimal across heterogeneous document structures.

The subset results further explain this trend. Chunk-only construction performs competitively on text-dominant subsets such as Academic and Industry, where fine-grained textual semantics are crucial. In contrast, visually complex subsets such as Brochure benefit more from adaptive construction, since the router can invoke VLM-based grouping to recover clearer semantic boundaries from fragmented layouts.

4.4 Analysis

Effect of Agent Interaction Cycles. Figure 3(a) shows the effect of the maximum interaction-cycle budget. Accuracy improves substantially from 50.2% with one cycle to 63.2% with two cycles, indicating that DAMG-RAG can use the initial retrieval state to plan effective follow-up searches.

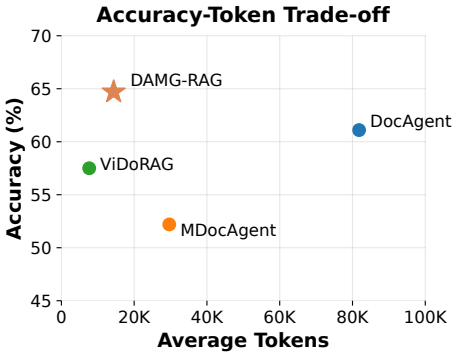
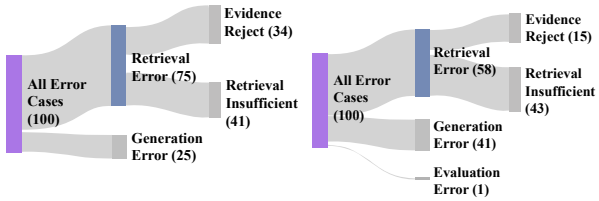


Figure 4: Accuracy and average token consumption of DAMG-RAG and agentic baselines under the Claude Sonnet 4.6 setting.

The gain becomes marginal beyond three cycles, increasing from 64.7% to only 64.9% when the budget is raised to four cycles, suggesting that three cycles are sufficient for most cases requiring targeted evidence expansion. Figure 3(b) reports the termination distribution under the default maximum budget of three cycles. 60.8%, 15.8%, and 23.4% of questions terminate after the first, second, and third cycles, respectively. This distribution shows that DAMG-RAG invokes additional retrieval selectively: simple questions stop once sufficient evidence is collected, while harder questions continue when the memory state remains incomplete.

Accuracy-Token Trade-off. Figure 4 compares DAMG-RAG with representative agentic baselines in terms of answer accuracy and average token consumption. Token consumption includes all agent prompts, retrieved evidence contexts, intermediate reasoning outputs, and final answer generation. DAMG-RAG achieves the strongest accuracy–efficiency trade-off, reaching 64.7% accuracy with only 15.4K tokens on average. Compared with DocAgent, which obtains 61.1% accuracy with 81.8K tokens, DAMG-RAG uses only 18.8% of its token budget while achieving higher accuracy. This demonstrates that DAMG-RAG improves long-document QA through more selective and state-guided evidence organization, rather than by expanding the context size, which is consistent with prior findings that long-context models do not always reliably use relevant information from long inputs (Liu et al., 2024).

Failure Mode Analysis. Following prior RAG evaluation work that separates retrieval quality from generation faithfulness (Es et al., 2024), we manually inspect 100 failed cases for each back-



(a) GPT-4o. (b) Claude Sonnet 4.6.

Figure 5: Failure mode analysis of DAMG-RAG under different backbone models.

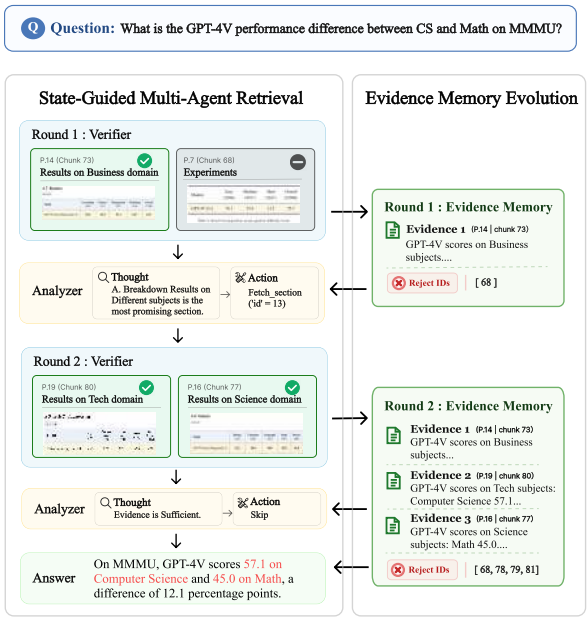


Figure 6: Case study of DAMG-RAG using accumulated evidence to guide cross-table retrieval.

bone and classify them by comparing the retrieved evidence with the gold evidence pages. We consider three error types: generation errors, where sufficient evidence is retrieved but the final answer is incorrect; evidence rejection, where relevant candidates are retrieved but not retained by the evidence verifier; and retrieval insufficiency, where the retrieved evidence does not cover the information required by the question.

As shown in Figure 5, Claude Sonnet 4.6 produces fewer retrieval-side errors, whereas GPT-4o exhibits a higher proportion of evidence rejection errors. This suggests that stronger backbones improve not only answer generation, but also the reliability of evidence verification in the multi-agent retrieval workflow.

4.5 Case Study

Figure 6 shows a cross-table question requiring CS–Math comparison. After retrieving a Business-domain table, DAMG-RAG infers that the evidence is organized by subject and triggers section-level retrieval to locate related tables. In the subsequent round, it retrieves the Tech & Engineering and Science tables, obtains the Computer Science and Math scores, and finally completes the comparison. This case illustrates how the global search state enables targeted evidence expansion beyond the initial retrieval result.

5 Conclusion

We presented DAMG-RAG, a document-aware multi-granularity RAG framework for visually rich document understanding. DAMG-RAG constructs layout-aware atomic retrieval units and organizes them into a document tree that connects atomic, page, and section-level structures. During inference, a state-aware multi-agent workflow maintains explored regions and verified evidence, and dynamically selects exploration and navigation tools for targeted evidence collection. DAMG-RAG achieves consistent improvements on MMLongBench and competitive-to-stronger results on LongDocURL, while using substantially fewer tokens. Further ablation and analysis demonstrate the importance of both adaptive atomic-unit construction and state-guided evidence exploration.

Limitations

DAMG-RAG depends on the quality of document parsing in the structuring stage. OCR errors, inaccurate layout analysis, missing visual elements, or incorrect outline extraction may lead to suboptimal atomic units and document-tree links. Although the framework is parser-agnostic, reliable parsing remains necessary for robust retrieval.

The multi-agent workflow is also sensitive to the capability of the underlying LVLM. Weaker backbones may reject relevant evidence, overlook missing information, or derive incorrect answers from retrieved evidence. The different failure patterns across backbones suggest that evidence verification and state-guided planning remain model-dependent.

DAMG-RAG focuses on targeted evidence exploration rather than exhaustive document-level aggregation. For very long documents or questions requiring broad synthesis across distant regions, the current toolset and interaction budget may be insufficient. Future work could incorporate stronger global aggregation mechanisms and more adaptive budget allocation.

References

Adobe. Adobe PDF Extract API. https://developer.adobe.com/document-services/docs/overview/pdf-extract-api/. Adobe PDF Services documentation. Accessed: 2026-05-25.

Anthropic. 2026. Claude Sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6. Accessed: 2026-05-09.

Aniket Bhattacharyya, Anurag Tripathi, Ujjal Das, Archan Karmakar, Amit Pathak, and Maneesh Gupta. 2025. Information extraction from visually rich documents using LLM-based organization of documents into independent textual segments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17241–17256, Vienna, Austria. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhong-Zhi Li, Jian Xu, Xiao-Hui Li, Yuan Gao, Jun Song, Bo Zheng, and Cheng-Lin Liu. 2025. LongDocURL: A comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1135–1159, Vienna, Austria. Association for Computational Linguistics.

Yaxin Du, Junru Song, Yifan Zhou, Cheng Wang, Jiahao Gu, Zimeng Chen, Menglan Chen, Wen Yao, Yang Yang, Ying Wen, and Siheng Chen. 2026. G^2-Reader: Dual evolving graphs for multimodal document QA. *arXiv preprint arXiv:2601.22055*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2024. RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158. Association for Computational Linguistics.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*.

Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. 2025. MDocAgent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Zhang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Hugging Face. ColQwen2. https://huggingface.co/docs/transformers/en/model_doc/colqwen2. Transformers documentation. Accessed: 2026-05-25.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023b. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 374–382. Association for Computing Machinery.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems*. Datasets and Benchmarks Track.

OpenAI. 2026. text-embedding-3-small. https://platform.openai.com/docs/models/text-embedding-3-small. OpenAI embedding model documentation. Accessed: 2026-05-20.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery.

PyMuPDF Developers. 2026. PyMuPDF. https://pymupdf.readthedocs.io/. Python library for data extraction, analysis, conversion, rendering, and manipulation of PDF and other documents. Accessed: 2026-05-11.

Aditya Sharma, Michael Saxon, and William Yang Wang. 2024. Losing visual needles in image haystacks: Vision language models are easily distracted in short and long contexts. *arXiv preprint arXiv:2406.16851*.

Joongmin Shin, Chanjun Park, Jeongbae Park, Jaehyung Seo, and Heuiseok Lim. 2025. MultiDocFusion: Hierarchical and multimodal chunking pipeline for enhanced RAG on long industrial documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20985–21004, Suzhou, China. Association for Computational Linguistics.

Ray Smith. 2007. An overview of the Tesseract OCR engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE Computer Society.

Zhivar Sourati, Zheng Wang, Marianne Menglin Liu, Yazhe Hu, Mengqing Guo, Sujeeth Bharadwaj, Kyu Han, Tao Sheng, Sujith Ravi, Morteza Dehghani, and Dan Roth. 2025. LAD-RAG: Layout-aware dynamic RAG for visually-rich document understanding. *arXiv preprint arXiv:2510.07233*.

Li Sun, Liu He, Shuyue Jia, Yangfan He, and Chenyu You. 2025. DocAgent: An agentic framework for multi-modal long-context document understanding. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17701–17716, Suzhou, China. Association for Computational Linguistics.

Nobuhiro Ueda, Yuyang Dong, Krisztián Boros, Daiki Ito, Takuya Sera, and Masafumi Oyamada. 2026. SCAN: Semantic document layout analysis for textual and visual retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 1618–1637, Rabat, Morocco. Association for Computational Linguistics.

Xueyao Wan and Hang Yu. 2025. MMGraphRAG: Bridging vision and language with interpretable multimodal knowledge graphs. *arXiv preprint arXiv:2507.20804*.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, and 1 others. 2024a. MinerU: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. 2025. ViDoRAG: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024c. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2024. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *Proceedings of the First Conference on Language Modeling*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Murong Yue. 2025. A survey of large language model agents for question answering. *arXiv preprint arXiv:2503.19213*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

A Implementation Details

A.1 Prompt Templates

We provide the complete prompts used by each agent module to support reproducibility. These prompts specify the input fields, decision constraints, and output schemas used in our experiments. Figures 13–16 present the prompts for the main agent modules.

A.2 Details of Tools

Table 5 summarizes the retrieval tools used by DAMG-RAG, including their types, target levels, and descriptions. We provide implementation details for each tool below.

Keyword Search. The keyword search tool takes a set of query keywords, each optionally associated with a priority level. We assign each keyword q_i a confidence weight w_i, where higher-priority keywords receive larger weights. To reduce the effect of overly common terms, we further compute an inverse-frequency factor over the candidate atomic units:

$$\gamma_i = 1 + \alpha \log \frac{N+1}{\mathrm{df}(q_i)+1},$$

where N is the number of candidate units, $\mathrm{df}(q_i)$ is the number of units containing q_i, and α controls the strength of the frequency correction. The final term weight is given by

$$\tilde{w}_i = w_i \cdot \gamma_i.$$

For each candidate unit d, the keyword score combines weighted unique matches and weighted total occurrences:

$$s_{\mathrm{kw}}(d) = \sum_{q_i \in M(d)} \tilde{w}_i + \lambda \sum_i c(q_i, d)\tilde{w}_i,$$

where $M(d)$ denotes the set of matched keywords in d, $c(q_i, d)$ is the occurrence count of q_i, and λ is a small coefficient for repeated matches. This design prioritizes candidates that cover more distinct high-confidence query terms while still giving a minor preference to repeated evidence mentions.

Semantic Search. The semantic search tool retrieves atomic units using dense embedding similarity. Given a query embedding $\mathbf{e}_q$ and a candidate unit embedding $\mathbf{e}_d$, the relevance score is computed by cosine similarity:

$$s_{\mathrm{sem}}(q, d) = \frac{\mathbf{e}_q^\top \mathbf{e}_d}{\|\mathbf{e}_q\|_2 \|\mathbf{e}_d\|_2}.$$

This tool is used for exploratory retrieval when the analyzer needs to discover semantically related evidence beyond exact lexical matches.

Fetch Identity. The identity fetching tool takes an entity name or identity string as input and matches it against the identity annotations of atomic units. Matched units are ranked by weighted identity overlap, making the tool suitable for entity-centric evidence retrieval.

Fetch Page. The page fetching tool takes a page range as input and returns the corresponding page-level content. It provides broader visual and textual context when the analyzer needs to inspect neighboring pages or verify evidence within a page region.

Fetch Section. The section fetching tool takes a section ID as input and returns the pages aligned with that section. It supports structure-aware navigation from a partial clue to the corresponding section-level context.

A.3 Baseline Reproduction Details

We reproduce representative agentic multimodal RAG baselines under the same backbone model and evaluation protocol. For adapted baselines, we describe the modifications below.

MDocAgent. We use ColPali (Faysse et al., 2024) for image-based semantic matching and `text-embedding-3-small` for text-based semantic retrieval. The top-k value for each modality is set to 3.

ViDoRAG. The original ViDoRAG setting is designed for open-domain document collections. We adapt it to the single-document QA setting used in our experiments. We use ColQwen2, a ColPali-style visual retriever based on Qwen2-VL (Faysse et al., 2024; Wang et al., 2024b; Hugging Face), for image-based semantic matching, and `text-embedding-3-small` for text-based semantic retrieval, consistent with the retrieval backends used in DAMG-RAG.

Evidence-page Oracle. We use an oracle evidence-page setting, where the golden evidence pages are directly provided to the backbone model for answer generation. This setting removes retrieval errors and serves as an oracle reference.

A.4 Artifact Licenses and Terms

We use publicly available datasets, models, tools, and APIs for research and evaluation purposes, following their respective licenses, terms of use, and intended access conditions, and cite their original sources throughout the paper.

B Evaluation Details

B.1 LLM-as-a-Judge Evaluation Protocol

The benchmarks used in our experiments mainly contain objective short-answer questions, where

Tool	Type	Target Level	Description
Semantic Search	Exploration	Atomic unit	Dense retrieval for concept-level evidence needs.
Keyword Search	Exploration	Atomic unit	Lexical retrieval for salient entities, terms, values, or domain keywords.
Fetch Identity	Navigation	Atomic unit	Direct access to identified elements such as figures, tables, charts.
Fetch Page	Navigation	Page	Page-level access when evidence is localized by page number or nearby context is needed.
Fetch Section	Navigation	Section	Section-level access either to search within a target outline node or to expand retrieval toward a broader section suggested by accumulated evidence.

Table 5: Retrieval tool space used by DAMG-RAG. Each tool is characterized by its retrieval type, target document level, and role in evidence exploration.

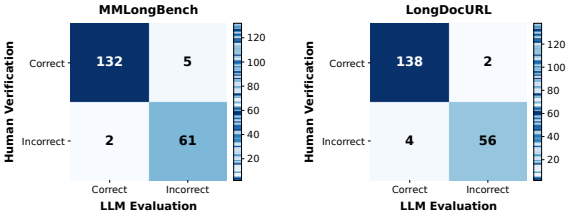


Figure 7: Reliability of LLM-based evaluation against human verification. The two confusion matrices report agreement on MMLongBench and LongDocURL, respectively.

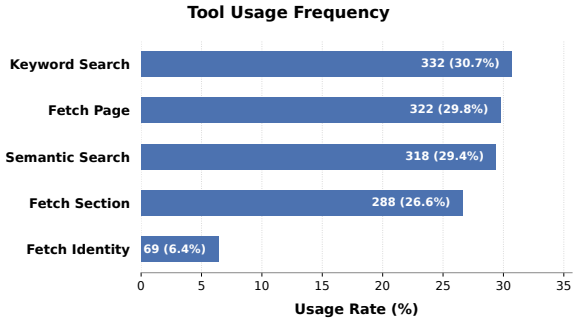


Figure 8: Tool usage frequency in our retrieval process. The figure reports the proportion of retrieval steps that invoke each tool.

answers are fixed entities, numbers, dates, or short phrases. The official evaluation follows an extract-then-match procedure: it extracts the final answer from the prediction and compares it with the reference answer using string-based matching. While this setting is suitable for objective answers, exact string matching can be too sensitive to aliases, formatting differences, or harmless explanatory text.

We therefore keep the same extract-then-compare principle but replace string-based matching with an LLM-as-a-judge equivalence check. The evaluator first normalizes both the prediction and the reference answer into concise final answers, and then judges whether they are factually equivalent. This criterion is still strict: it is intended to approximate exact-match evaluation at the semantic level rather than reward partially correct or overclaimed answers.

We further compare the evaluator's judgments with human annotations; the matrix is shown in Figure 7. The full evaluator prompt is provided in Figure 17.

C Additional Experimental Analysis

C.1 Tool Usage Analysis

Figure 8 summarizes the tool usage frequency of DAMG-RAG on MMLongBench. All five tools are invoked during evaluation, indicating that the analyzer does not collapse to a single preferred retrieval operation. Keyword Search, Semantic Search, and Fetch Page are the most frequently selected tools. This pattern supports our design motivation: effective evidence exploration requires both open-ended search over candidate units and structure-aware navigation within the document tree.

C.2 Atomic Construction Strategy Distribution

This section reports the distribution of chunk-based, page-based, and VLM-grouping construction strategies across different document types. As shown in Figure 9, the selected strategies differ substantially between the two benchmarks. MMLongBench-Doc contains a larger proportion of chunk-based cases (48.9%), suggesting that many documents are relatively text-centered and

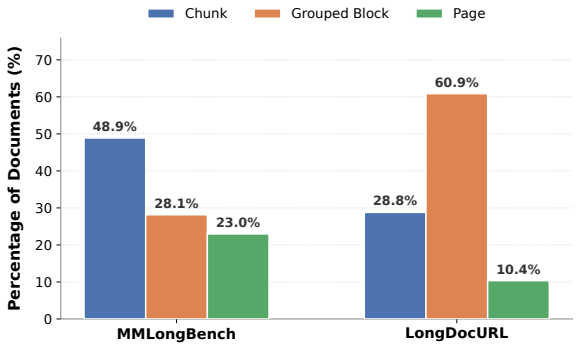


Figure 9: Distribution of atomic-unit construction strategies on MMLongBench and LongDocURL. The three strategies correspond to chunk-based construction, VLM-based grouping, and page-based construction, respectively.

can be represented by fine-grained textual units. In contrast, LongDocURL is dominated by VLM-grouped blocks (60.9%), indicating that more documents require visually grounded grouping to preserve coherent evidence regions. The low usage of page-based construction on LongDocURL (10.4%) further suggests that whole-page units are often too coarse for such documents. Overall, these results support the need for adaptive atomic unit construction rather than applying a fixed granularity to all documents.

D Example

Figures 10–12 show an example of DAMG-RAG answering the question: *"How many quizzes are there in the entire course?"* The answer is not contained in a single location, but must be aggregated from quiz mentions across different course sections.

In Round 1, the system identifies initial evidence about several quizzes. More importantly, the analyzer observes that the document consistently refers to quizzes with the surface form "Quiz #", such as "Quiz #1" and "Quiz #3". It therefore adapts the next retrieval step by searching for the same string pattern with later quiz numbers, rather than using only the original question as the query.

This evidence-conditioned strategy retrieves additional quiz mentions in Round 2. The verifier accepts chunks containing Quiz #5, Quiz #6, and Quiz #7, and these findings are added to the memory pool. After this round, the verified evidence covers Quiz #1 through Quiz #7. The analyzer then performs a final check for uncovered Quarter 4 content and possible later quizzes.

In Round 3, the retrieved chunks are judged irrelevant to the quiz-counting question. Since no new valid quiz evidence is found and the maximum number of rounds is reached, DAMG-RAG stops retrieval and answers based on the verified memory pool.

This example highlights the central behavior of DAMG-RAG: retrieval decisions are not fixed before search, but are progressively conditioned on verified evidence and document-specific regularities discovered during exploration. The system first infers the document-specific naming pattern "Quiz #", uses it to conduct targeted searches, and then verifies whether later-course content adds new quiz evidence.

Question: How many quizzes are there in the entire course?

Round 1 : Verifier

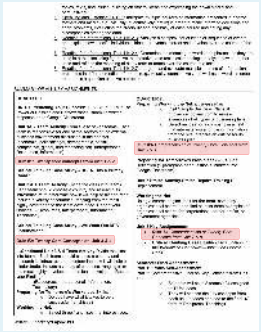
(Chunk 1)

Quiz #3- Comprehension of Twenty Core Concepts from Unit 5 & 6 Preparing for Tomorrow's Workplace Skills: Self Reflection (1 paragraph per question submitted into Google Classroom) o Question 1: Would you be a good manager? Do a self-assessment that includes your current technical, human relations, and conceptual skills. What skills do you already possess, and which do you need to add? Where do your strengths lie? Based on this exercise, develop a description of an effective manager. (Resources, Information) ....

(Chunk 2)



(Chunk 3)



Round 1 : Evidence Memory

Evidence 1 (Chunk3)

"SUMMARY OF ALL KEY ASSIGNMENTS — QUARTER 1: Quiz #1: Twenty core concepts from Unit 1 & 2; Quiz #2: Twenty Core Concepts on Unit 3 & 4. QUARTER 2: Quiz #3- Comprehension of Twenty Core Concepts from Unit 5 & 6; Unit 8 Key Assignments: Quiz #4- Comprehension of Twenty Core Concepts from Unit 7 & 8.",

Evidence 2 (Chunk1)

"Quiz #3- Comprehension of Twenty Core Concepts from Unit 5 & 6 (listed under UNIT 7: Designing Organizational Structures section, Quarter 2 content)."

Round 1 : Analyzer

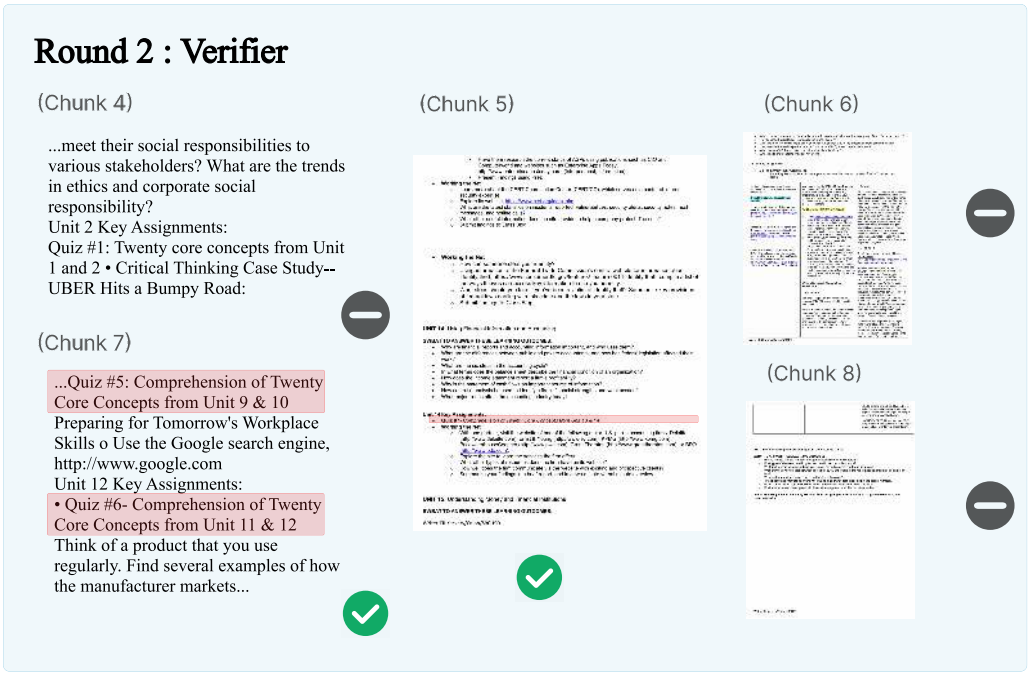
Thought

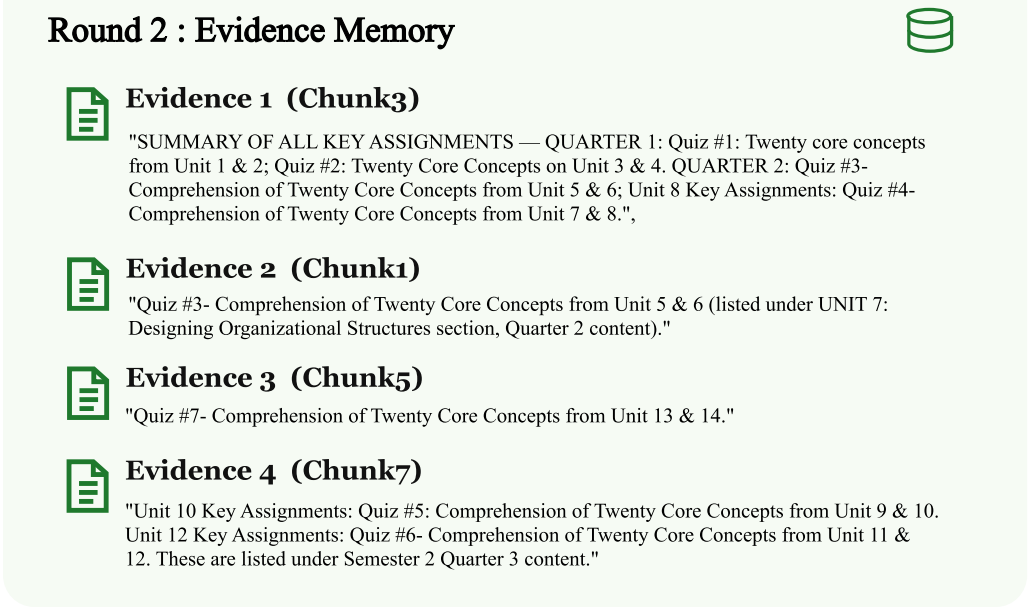
The current evidence from page 3 lists Quiz #1 through Quiz #4 in a 'Summary of All Key Assignments' covering Quarter 1 and Quarter 2. However, this summary may only cover Semester 1. The course has Semester 2 (Quarters 3 & 4) as well, per the TOC. I need to check if there are additional quizzes in Semester 2 content. TOC nodes '8' (SEMESTER 2: QUARTER 3) and '13' (SEMESTER 2: QUARTER 4) are the relevant sections. I'll use get_toc_scope_pages to pull those sections, and also run a keyword search for 'quiz' to catch any quiz references across the full document."

Action

(1) Fetch_Section ("8","13")
(2) Keyword_Search ("Quiz#5","Quiz#6","quiz")

Figure 10: Round 1: Initial evidence discovery and identification of the document-specific quiz naming pattern.





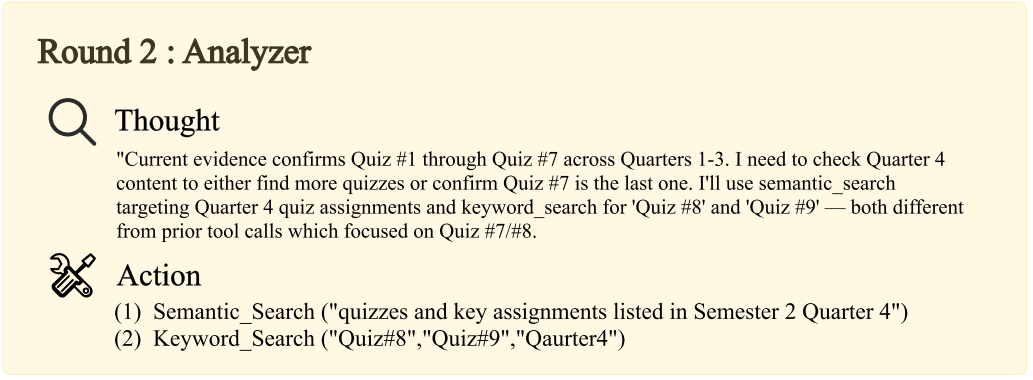
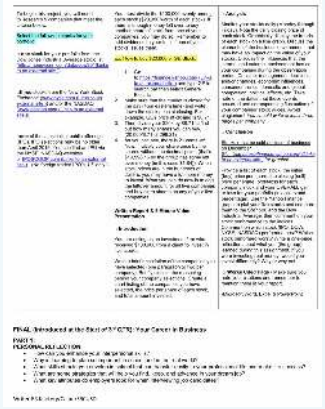


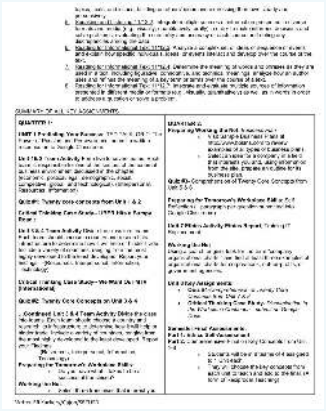
Figure 11: Round 2: Targeted retrieval using the inferred “Quiz #” pattern to collect additional quiz evidence.

Round 3 : Verifier

(Chunk 9)



(Chunk 10)



Round 3 : Evidence Memory

Evidence 1 (Chunk3)

"SUMMARY OF ALL KEY ASSIGNMENTS — QUARTER 1: Quiz #1: Twenty core concepts from Unit 1 & 2; Quiz #2: Twenty Core Concepts on Unit 3 & 4. QUARTER 2: Quiz #3- Comprehension of Twenty Core Concepts from Unit 5 & 6; Unit 8 Key Assignments: Quiz #4- Comprehension of Twenty Core Concepts from Unit 7 & 8.",

Evidence 2 (Chunk1)

"Quiz #3- Comprehension of Twenty Core Concepts from Unit 5 & 6 (listed under UNIT 7: Designing Organizational Structures section, Quarter 2 content)."

Evidence 3 (Chunk5)

"Quiz #7- Comprehension of Twenty Core Concepts from Unit 13 & 14."

Evidence 4 (Chunk7)

"Unit 10 Key Assignments: Quiz #5: Comprehension of Twenty Core Concepts from Unit 9 & 10. Unit 12 Key Assignments: Quiz #6- Comprehension of Twenty Core Concepts from Unit 11 & 12. These are listed under Semester 2 Quarter 3 content."

Round 3 : Answer

"There are at least 7 quizzes in the entire course: Quiz #1 (Units 1 & 2), Quiz #2 (Units 3 & 4), Quiz #3 (Units 5 & 6), Quiz #4 (Units 7 & 8), Quiz #5 (Units 9 & 10), Quiz #6 (Units 11 & 12) and Quiz #7 (Units 13 & 14).",

Figure 12: Round 3: Final verification for remaining course sections before termination.

```
You are the analyzer agent in a document-aware retrieval system.
Given the current retrieval state, decide whether the accumulated evidence is
sufficient to answer the query. If not, define the next atomic retrieval goal
and select suitable retrieval tools.

Input:
- query: the user question.
- evidence: accumulated evidence collected from previous rounds.
- tool_history: previous tool calls and their arguments.
- doc_metadata: global document information, such as filename and page count.
- toc_tree: section nodes with id, text, level, and page_count.
- round_information: the current retrieval round.

Decision procedure:
1. Assess whether the accumulated evidence is sufficient to answer the query.
   If it is sufficient, set stop=true and do not issue new tool calls.

2. If more evidence is needed, define a focused round_goal. For complex or
   multi-hop questions, the round_goal should target the next missing piece of
   information rather than the full query.

3. Select one retrieval mode:
   - Navigation mode: use this mode when the round_goal involves a concrete
     document target, such as a figure, table, page range, slide, or TOC
     section. Navigation tools can also be used to identify structurally
     related nodes or coarser-grained units, such as pages or sections, for
     further inspection.

   - Exploration mode: use this mode when the round_goal requires semantic or
     lexical discovery across the document.

4. Avoid repeating previous tool calls with identical arguments.

Output JSON:
{
  "round_goal": "Focused retrieval objective for this round",
  "stop": false,
  "rationale": "Brief explanation of the selected mode and tools",
  "mode": "navigation | exploration",
  "tool_calls": [
    {"method": "tool_name", "args": {"arg1": "..."}}
  ]
}
```

Figure 13: Prompt template for the analyzer agent.

```
You are an evidence verifier. Given the user query, the current round goal,
and retrieved document chunks, select chunks that directly support the goal
and extract grounded evidence.

Input:
- user_query: the original user question.
- round_goal: the focused retrieval objective for the current round.
- retrieved_chunks: text or image chunks with ids and page information.

Instructions:
1. Mark each chunk as useful or useless.
2. A useful chunk must contain explicit information needed for the round_goal.
3. Extract atomic evidence from useful chunks, preserving key facts, numbers,
   labels, units, captions, and visual details.
4. For image chunks, rely on visible information such as titles, legends,
   axes, table headers, annotations, and values.
5. Every output id must come from the current retrieved_chunks. Each evidence
   item must be supported by its own chunk id.

Output JSON:
{
  "useless_ids": ["..."],
  "useful_ids": ["..."],
  "evidence": [
    {
      "id": "...",
      "evidence_content": "...",
      "goal_support": "..."
    }
  ]
}
```

Figure 14: Prompt template for the verifier.

```
You are a document layout router.
Given representative page layout visualizations, assign the document to one
atomic-unit construction mode.

Modes:
- Mode A: Linear text. The document is text-heavy and follows a mostly linear
  reading order, such as academic reports.
- Mode B: Visual fragments. The document is information-dense but organized
  through icons, badges, small boxes, or non-linear visual components, such as
  brochures.
- Mode C: Sparse pages. The document has low information density, large
  whitespace, decorative backgrounds, or minimal text, such as slides.

Decision procedure:
1. Check visual density. If most representative pages are sparse, decorative,
   or contain very little text, assign Mode C.
2. Otherwise, distinguish linear text from visual fragments:
   - Assign Mode A if the main information carrier is paragraphs, tables, or
     structured text with a clear reading order.
   - Assign Mode B if the main information carrier is fragmented visual
     components, such as icons, badges, callout boxes, or non-linear layouts.

Output JSON:
{
  "mode": "A | B | C",
  "reason": "Brief explanation based on visual density and layout type."
}
```

Figure 15: Prompt template for the document router.

```
You are a layout-aware visual grouping agent for document reconstruction.
Given a page image with numbered local elements, group the element ids into
semantic blocks.

Input:
- page image with numbered local elements.
- visible_local_ids: ids visible on the page.

Grouping criteria:
1. Figure or table units: merge a figure or table with its caption, legend,
   labels, annotations, or footnotes.
2. Textual units: merge headings with their immediate body text, and merge
   lines that form the same paragraph or list.
3. Component blocks: merge elements inside the same border, shaded region,
   aligned column, card, or visually distinct block.
4. Captions and annotations: merge small nearby text with the visual element it
   describes.

Constraints:
- Every id in visible_local_ids must appear in exactly one group.
- Use only ids from visible_local_ids.
- If an element does not belong with others, keep it as a single-element group.
- If the whole page forms one semantic block, return {"groups": "all"}.
- If no valid local ids are visible, return {"groups": []}.
- Return JSON only.

Output JSON:
{
  "groups": [[1, 2, 3], [4], [5, 6]]
}
```

Figure 16: Prompt template for VLM-based visual grouping.

```
You are an answer correctness evaluator.

Input:
- question: the user question.
- answer_gt: the ground-truth answer.
- answer_pred: the model prediction.

Task:
First, extract a normalized answer from answer_pred. The normalized answer
should be either a concise concrete answer or "Not answerable" if the
prediction explicitly states that the question cannot be answered.

Apply the same normalization procedure to answer_gt. Ignore rationales,
citations, hedging, and meta commentary. If a prediction contains both an
unanswerability statement and a concrete answer or guess, treat it as a
concrete answer.

Correctness criteria:
1. If the normalized ground truth is "Not answerable", the prediction is
   correct only when the normalized prediction is also "Not answerable".
2. If the normalized ground truth is a concrete answer, the prediction is
   correct only when the normalized prediction is factually equivalent to it.
3. For list answers, the number of items must match exactly, and all key
   elements must be present. Order does not matter.
4. For numerical answers, reasonable rounding or precision differences are
   allowed.
5. When equivalence is uncertain, mark the prediction as incorrect.

Output JSON:
{
  "correct": true or false,
  "reason": "Brief explanation mentioning the normalized ground truth and prediction."
}
```

Figure 17: Prompt template for answer correctness evaluation.