MoLoRAG: Bootstrapping Document Understanding via Multi-modal Logic-aware Retrieval

Xixi Wu[1], **Yanchao Tan**[2], **Nan Hou**[1], **Ruiyang Zhang**[3], **Hong Cheng**[1](✉)
[1]The Chinese University of Hong Kong
[2]Fuzhou University [3]University of Macau
{xxwu, nhou, hcheng}@se.cuhk.edu.hk
yctan@fzu.edu.cn, yc47931@um.edu.mo

Abstract

Document Understanding is a foundational AI capability with broad applications, and Document Question Answering (DocQA) is a key evaluation task. Traditional methods convert the document into text for processing by Large Language Models (LLMs), but this process strips away critical multi-modal information like figures. While Large Vision-Language Models (LVLMs) address this limitation, their constrained input size makes multi-page document comprehension infeasible. Retrieval-augmented generation (RAG) methods mitigate this by selecting relevant pages, but they rely solely on semantic relevance, ignoring logical connections between pages and the query, which is essential for reasoning.

To this end, we propose **MoLoRAG**, a logic-aware retrieval framework for multi-modal, multi-page document understanding. By constructing a page graph that captures contextual relationships between pages, a lightweight VLM performs graph traversal to retrieve relevant pages, including those with logical connections often overlooked. This approach combines semantic and logical relevance to deliver more accurate retrieval. After retrieval, the top-K pages are fed into arbitrary LVLMs for question answering. To enhance flexibility, MoLoRAG offers two variants: a training-free solution for easy deployment and a fine-tuned version to improve logical relevance checking. Experiments on four DocQA datasets demonstrate average improvements of **9.68%** in accuracy over LVLM direct inference and **7.44%** in retrieval precision over baselines. Codes and datasets are released at https://github.com/WxxShirley/MoLoRAG.

1 Introduction

Document Understanding is a foundational AI capability with extensive real-world applications, such as interpreting medical reports and assisting with academic literature. This ability holds significant potential to improve productivity and support decision-making (Ding et al., 2022; Ma et al., 2024a; Suri et al., 2024; Zhang et al., 2024). A key task for evaluating document understanding is Document Question Answering (DocQA), which requires models to automatically answer questions

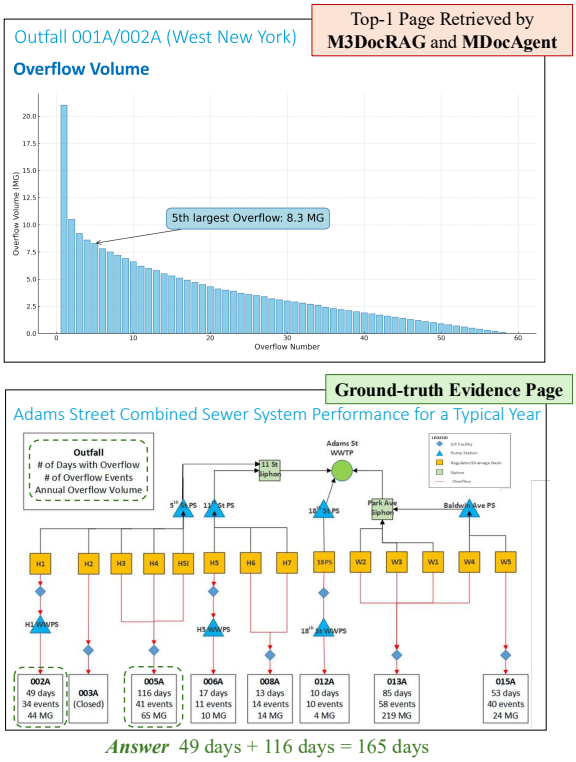

Figure 1: **Illustration of a retrieval example on LongDocURL (Deng et al., 2024).** Both M3DocRAG (Cho et al., 2024) and MDocAgent (Han et al., 2025) rely solely on **semantic relevance** between the query and the page for retrieval. As a result, they retrieve a page containing keywords from the question but lacking the necessary information to answer it. In contrast, the ground-truth evidence page, successfully retrieved by our MoLoRAG, is **logically relevant** to the question, providing detailed statistics for each outfall and enabling accurate derivation of the correct answer.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 14036–14057
November 4-9, 2025

based on the content of a document.

Classic approaches to DocQA typically follow a two-step pipeline: the document is first converted into text using Optical Character Recognition (OCR) (Memon et al., 2020; Wang et al., 2024a; Wei et al., 2024), and then Retrieval-augmented Generation (RAG) techniques identify relevant paragraphs to feed into Large Language Models (LLMs) for question answering. However, the text extraction process often strips away essential multimodal information, such as tables, figures, and document layouts, resulting in incomplete document understanding. Large Vision-Language Models (LVLMs) address this limitation by processing image-format document snapshots, enabling multimodal comprehension. Nevertheless, LVLMs, such as LLaMA-Vision (Grattafiori et al., 2024) and LLaVA-Next (Li et al., 2024a), are constrained to single-image inputs, rendering them ineffective for long, multi-page documents.

Recent research has explored methods to address these challenges. For example, M3DocRAG (Cho et al., 2024) leverages a document encoder, i.e., ColPali (Faysse et al., 2024), to encode individual pages and retrieve relevant ones based on vector similarity. This approach reduces the number of input pages, alleviating the comprehension burden for LVLMs. MDocAgent (Han et al., 2025) extends this by introducing parallel pipelines for text and image retrieval, with specialized agents for each modality to enable collaborative reasoning. While effective, these methods focus primarily on **semantic relevance**, matching queries to pages based on embedding similarity. For example, as shown in Figure 1, when asked to determine the total overflow days for two outfalls, the top-1 page retrieved by both methods contains keywords from the question but lacks detailed information about each outfall. In contrast, a **logically relevant** page, such as the ground-truth evidence page, provides detailed statistics for each outfall, enabling reasoning (e.g., summing the overflow days) to derive the correct answer.

In DocQA, accurate retrieval is critical, as it directly impacts downstream answering. Without precise retrieval, LVLMs are prone to errors or hallucinations stemming from irrelevant or incomplete inputs. Addressing this challenge requires retrieval methods that go beyond surface-level semantic matching to capture deeper logical relationships. Building on this insight, we propose **MoLoRAG**, a graph-based retrieval framework tailored for **M**ulti-m**o**dal **Lo**gic-aware document understanding. Document pages naturally exhibit structured relationships, e.g., cross-references and shared entities. Leveraging this property, we first construct a page graph to represent the dependencies between pages. A lightweight VLM serves as the retrieval engine, reasoning over this graph through traversal to identify logically relevant pages. Finally, both semantic and logical relevance are combined into a unified similarity score to re-rank pages, enabling a more comprehensive retrieval process.

To further enhance its utility, MoLoRAG introduces two variants to offer flexibility for different deployments. The first is a training-free variant, which leverages a pre-trained VLM, e.g., Qwen2.5-VL-3B (Bai et al., 2025), to perform graph traversal and retrieval directly, providing an off-the-shelf solution that is easy to deploy. The second is a fine-tuned variant, which involves training the retrieval engine on a curated dataset to improve its reasoning capabilities. This fine-tuned version functions as a more intelligent retrieval engine, capable of capturing nuanced relationships between queries and document pages. Moreover, MoLoRAG demonstrates strong compatibility with arbitrary LVLMs. Once the retrieval step is complete, only the top-K page snapshots are passed to an LVLM for question answering, filtering out irrelevant content and ensuring concise, high-quality inputs. To summarize, our contributions are as follows:

- **Logic-aware Retrieval Framework** We highlight the importance of page retrieval in DocQA and propose MoLoRAG, a novel retrieval method that incorporates logical relevance. By representing the document as a page graph and enabling a VLM to perform multi-hop reasoning through graph traversal, our method identify both semantically and logically relevant pages.
- **Comprehensive Experiments** We conduct extensive experiments on four DocQA datasets, comparing MoLoRAG with LLM-based, LVLM-based, and Multi-agent methods. Results demonstrates its superior retrieval accuracy, significant performance improvements over baselines, and flexible compatibility with arbitrary LVLMs.
- **Released Model and Dataset** We release the fine-tuned retriever engine model weights[1]

[1] https://huggingface.co/xxwu/MoLoRAG-QwenVL-3B

and the curated training dataset[2], empowering further development of intelligent and logic-aware retrieval engines.

2 Related Works

Document Question Answering DocQA is a core task for evaluating document understanding. Early benchmarks (Mathew et al., 2021; Tito et al., 2023; Tanaka et al., 2023) focused on single-page or short documents with low information density, where questions targeted individual elements like text or figures. Recent benchmarks (Ma et al., 2024b; Deng et al., 2024) shift toward lengthy, information-rich documents, introducing challenges like cross-page and multi-modal reasoning. DocQA methods can be broadly categorized into two branches based on backbone models: LLM-based and LVLM-based. LLM-based methods rely on OCR techniques to extract text from the document, enabling text-based question answering. LVLM-based methods, on the other hand, leverage their inherent multi-modal capabilities to process document images directly. With the advancements in LVLMs, the latter approach now dominates recent solutions. A notable advancement is MDocAgent (Han et al., 2025), which represents a new category by combining both LLMs and LVLMs into a multi-agent framework for collaborative question answering. However, challenges like input size limitations still necessitate effective retrieval strategies to reduce input burden and enhance performance.

Retrieval-augmented Generation RAG enhances LLMs by supplementing them with external knowledge, improving performance in domain-specific or knowledge-intensive tasks (Gao et al., 2024; Lewis et al., 2021; Asai et al., 2024). The emergence of LVLMs has further expanded RAG to multi-modal contexts, enabling the retrieval of relevant images to handle knowledge-seeking queries (Chen et al., 2024, 2022). Despite these advancements, existing RAG methods fail to address the unique challenges of DocQA, involving highly interleaved textual and visual elements. For page retrieval in DocQA, existing method like M3DocRAG (Cho et al., 2024) rely on document encoders for semantic-based retrieval, neglecting the logical relevance essential for accurate question answering.

Graph-based RAG GraphRAG is an advanced RAG paradigm that leverages graph-structured knowledge and retrieval for improved contextual reasoning (Zhang et al., 2025; Xiang et al., 2025). Existing methods are categorized into two types: Knowledge-based, which constructs knowledge graphs through entity recognition and relation extraction (He et al., 2024), and Index-based, which creates a two-layer graph linking high-level topic nodes to detailed text nodes for efficient retrieval (Sarthi et al., 2024; Edge et al., 2025; Liu et al., 2025; Li et al., 2024b). However, current GraphRAG approaches are limited to text and cannot handle the document with multi-modal information. We are the first to extend GraphRAG to the document domain by constructing a page graph that enables reasoning over its structure.

3 Methodology

In this section, we present the details of MoLoRAG, a novel graph-based retrieval framework designed to facilitate multi-modal and multi-page document understanding. The overall framework is illustrated in Figure 2.

3.1 Preliminary

Given a question q expressed in natural language and a document $\mathcal{D} = \{p_1, p_2, \ldots, p_N\}$, where each p_i represents an individual page in the form of an RGB image, and N is the total number of pages. The goal of DocQA is to generate an answer a that accurately addresses q using the information contained within $\mathcal{D}$. To solve this task, MoLoRAG adopts an LVLM-based two-stage framework:

- **Retrieval:** Given the extensive nature of document $\mathcal{D}$, the first step involves retrieving the top-K most relevant pages for the question, denoted as $\mathcal{P}^r = \{p_1^r, \ldots, p_K^r\}$, where $K \ll N$ (e.g., $K = 3$). Unlike traditional retrieval methods that rely solely on semantic relevance, MoLoRAG incorporates both semantic and logical relevance to enhance retrieval accuracy and contextual understanding for effective reasoning.

- **Generation:** The retrieved pages $\mathcal{P}^r$, along with the input question q, are then fed into an LVLM to generate the answer a. For LVLMs that cannot directly process multiple images, we use a processing function $\text{Process}(\cdot)$ to prepare $\mathcal{P}^r$, e.g., concatenating multiple images into a single composite one. Formally, this stage is expressed as:

[2] https://huggingface.co/datasets/xxwu/MoLoRAG

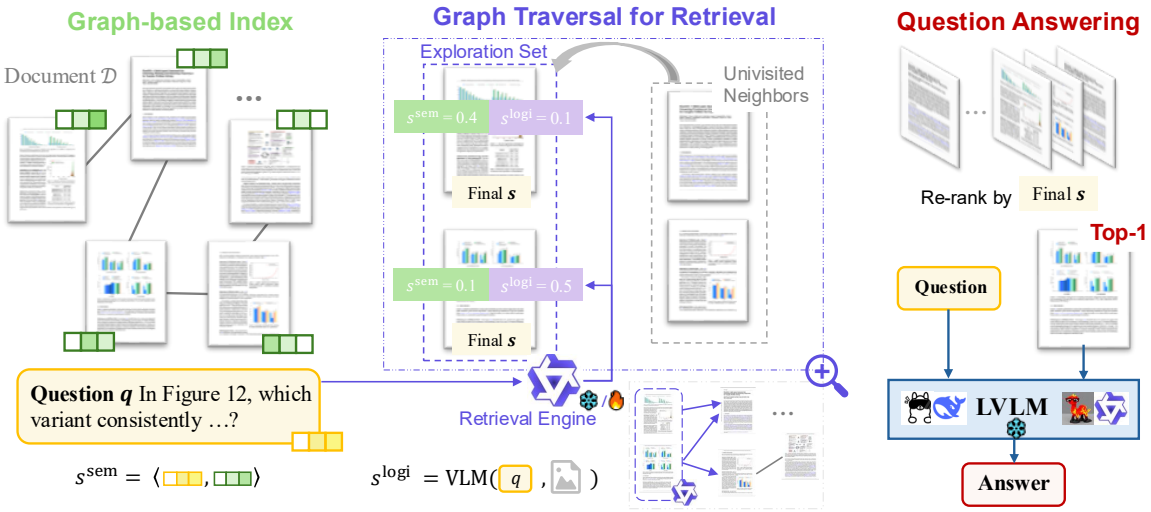

Figure 2: **Illustration of MoLoRAG framework.**

$$a = \text{LVLM}(q, \text{Process}(\mathcal{P}^r)).$$

3.2 Logic-aware Page Retrieval

In this subsection, we detail the retrieval process of MoLoRAG. We first construct a page graph as a graph-based index, depicting the relationships between pages within a document. Then, a VLM serves as the retrieval engine, performing reasoning over this graph through traversal to adaptively identify pages that are both semantically and logically relevant to the given question.

Graph-based Index Firstly, each document page p_i is encoded into a latent embedding that captures its distinct multi-modal content, represented as $E_{p_i} = \text{DocEncoder}(p_i) \in \mathbb{R}^{k \times d}$, where k denotes the number of visual tokens per page and d is the embedding dimension. Following Cho et al. (2024); Han et al. (2025), we choose ColPali (Faysse et al., 2024) as the document encoder due to its demonstrated effectiveness in preserving multi-modal semantics.

Using these embeddings, we construct a page graph $G(\mathcal{V}, \mathcal{E})$ to represent relationships between pages. In this graph, each node $p_i \in \mathcal{V}$ corresponds to a page from the document $\mathcal{D}$, and edges $\mathcal{E}$ are established between pairs of pages based on their similarity. Specifically, an edge (p_i, p_j) is added if the similarity between their embeddings exceeds a threshold θ, expressed as: $\mathcal{E} = \{(p_i, p_j) | \langle E_{p_i}, E_{p_j} \rangle \geq \theta\}$ where $\langle \cdot, \cdot \rangle$ denotes the inner product as the similarity measure. While such graph construction mechanism is simple, it offers the advantages of being **efficient**, **automatic**, ensuring **scalability** to large document, and leveraging prior knowledge encoded in the embedding.

Graph Traversal for Retrieval With the page graph constructed, we leverage a VLM as the retrieval engine to evaluate the relevance of each visited page in relation to the given question. This approach overcomes the limitations of traditional semantic-only retrieval by incorporating logical checking into the process. By utilizing the reasoning capabilities of the VLM, our method effectively identifies important pages that may otherwise be overlooked. The graph traversal process is outlined as follows, aligning with the pseudo-code in Algorithm 1 in the Appendix.

– **Initialization** For the question q and a page p_i from the document, the document encoder computes a semantic relevance score as $s_i^{\text{sem}} = \langle \text{DocEncoder}(q), E_{p_i} \rangle$. Based on these scores, the top-w nodes (pages) with the highest semantic scores are selected as the initial exploration set.

– **Relevance Scoring** For page p_i in the exploration set, the VLM assigns a logical relevance score s_i^{logi} using the prompt provided in Appendix B, reflecting the deeper logical connection of the page to the question. The final relevance score s_i is then updated as $s_i = \text{Combine}(s_i^{\text{sem}}, s_i^{\text{logi}})$, where $\text{Combine}(\cdot)$ integrates semantic and logical relevance scores, e.g., taking their weighted average.

– **Iterative Traversal** The traversal proceeds iteratively: at each step, we define the candidate set as the unvisited neighbors of the current exploration set. Each page in this candidate set is evaluated and its relevance score is updated using the same combination of semantic and logical relevance. The pages are ranked by their final relevance scores, and only the top-w nodes are retained as the new

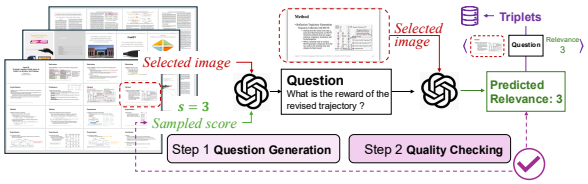

Figure 3: **Illustration of training data generation for MoLoRAG+.**

exploration set for the next iteration. The traversal continues until either the candidate set is empty or the maximum hop limit is reached. Both the exploration set size w and the hop limit n_{hop} constrain the traversal space, ensuring efficiency by avoiding the exhaustive process of sequentially traversing every page in the document.

Once the traversal is complete, all visited nodes are **re-ranked** based on their final relevance scores, and the top-K pages are selected for the subsequent question-answering phase.

3.3 Training-required Variant

While the MoLoRAG framework allows the use of a pre-trained VLM as an off-the-shelf solution for fast deployment, we propose an enhanced variant, **MoLoRAG+**[3]. This variant fine-tunes the VLM (retrieval engine) to bolster its reasoning capabilities during graph traversal, enabling the model to assign more accurate logical relevance scores.

Data Preparation (Figure 3) The success of fine-tuning relies on the availability of high-quality training data (Sun et al., 2024). To achieve this, we utilize GPT-4o (OpenAI, 2024) as a data generation engine to create reliable triplets in the format $\langle$`Question`, `Image`, `Relevance_Score`$\rangle$, where the `Relevance_Score` quantifies the alignment between the question and the image content. These triplets serve as supervision signals for fine-tuning, enabling the model to better estimate logical relevance. The data creation process begins by randomly selecting a page snapshot (image) from the document and sampling a relevance score from a pre-defined range. Using both the selected image and the sampled relevance score as context, GPT-4o generates a question that reflects the degree to which the selected image can answer it. GPT-4o then predicts the relevance score between the generated question and the image, enabling an automated **quality-checking**: only samples where the predicted score and the target score closely match (e.g., within a tolerance of ≤ 1) are retained. To further ensure accuracy, these filtered samples undergo manual verification, ensuring that the final dataset fully aligns with the task's requirements. Note that the data engine can be replaced with any arbitrary LVLMs instead of proprietary models like GPT-4o to reduce costs. An additional analysis is provided in Appendix C.

Model Training Using the curated dataset, we fine-tune the backbone VLM with supervised fine-tuning (SFT) techniques (Hu et al., 2022). Detailed training configurations are provided in Appendix C. After fine-tuning, the updated VLM replaces the original pre-trained model as the retrieval engine. By incorporating enhanced logical checking capabilities, this variant is expected to deliver more accurate retrieval performance.

3.4 Summary

In the proposed MoLoRAG framework, the top-K scored pages are fed into an LVLM during the question-answering phase, ensuring that only the most relevant information is utilized. Its key strengths include compatibility with arbitrary LVLMs, making it particularly adaptable for models limited to processing a single image by transforming an otherwise infeasible task into a practical solution. By incorporating both semantic and logical relevance, the framework enhances retrieval accuracy (Section 4.3). Furthermore, the graph-based traversal mechanism effectively narrows the search space, prioritizing relevant pages and significantly accelerating the retrieval process compared to exhaustive page-by-page traversal (Appendix D.4). Collectively, these features position MoLoRAG as a powerful solution for the DocQA task.

4 Experiments

4.1 Experimental Setup

Datasets We utilize four datasets from three benchmarks for evaluation, including **MMLongBench** (Ma et al., 2024b), **LongDocURL** (Deng et al., 2024), and **PaperTab** and **FetaTab** from the UDA-Benchmark (Hui et al., 2024). Dataset statistics are shown in Table 1. These datasets span a wide range of topics (e.g., administrative files, tutorials, research reports) and feature diverse multi-modal elements (e.g., chart, text, and table). Additionally, they vary in average document length and information density, ensuring a comprehensive evaluation. Other benchmarks like DocVQA (Mathew et al., 2021; Tanaka et al., 2023; Masry

[3]We use **MoLoRAG+** to denote the fine-tuned version.

Table 1: **Statistics of experimental datasets.**

Dataset	# Question	# Document	Avg. Pages	Avg. Tokens
PaperTab	393	307	11.0	12,685.4
FetaTab	1,016	871	15.8	16.524.5
MMLongBench	1,082	135	47.5	24,992.6
LongDocURL	2,325	396	85.6	56,715.1

et al., 2022) are omitted due to their shorter document lengths and lower information density.

Evaluation Metrics For MMLongBench and LongDocURL, we follow their original evaluation protocol, using a generalized **Accuracy** with rule-based evaluation to handle various answer types. Additionally, we report **Exact Match (EM)** as a supplementary metric, as the answers in these datasets are typically short and concise. For PaperTab and FetaTab, where ground-truth answers are formulated as long sentences or multiple choices, we follow MDocAgent to employ GPT-4o as the evaluator. Specifically, it evaluates **Binary Correctness** by determining whether the generated answer matches the ground-truth answer, assigning a binary score of 0 or 1. We also evaluate retrieval-stage accuracy using metrics like Recall@K, with further details provided in Appendix D.1.

Baselines We consider the following baselines: (1) **LLM w. Text RAG** first converts the document into texts using OCR and then applies retrieval techniques to the text, with LLMs serving as the backbone for question answering. (2) **LVLM Direct Inference** directly feeds LVLMs with full document snapshot images for question answering. For LVLMs that only support single-image input, we follow Ma et al. (2024b) by concatenating all images into a single combined one. (3) **M3DocRAG** (Cho et al., 2024) uses ColPali as a page retriever to identify relevant pages and feeds only the retrieved pages to the LVLM for further processing. (4) **MDocAgent** (Han et al., 2025) is a strong baseline for document understanding that employs a multi-agent system. A text agent and an image agent independently handle their respective modalities and collaborate to synthesize the final answer. Due to space limits, implementation details of each method are provided in Appendix D.2.

Choices of LLMs For LLMs, we consider Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), Qwen2.5-7B-Instruct (Qwen, 2025), LLaMA3.1-8B-Instruct (Grattafiori et al., 2024), GPT-4o, and DeepSeek-V3 (DeepSeek-AI, 2025). These models vary in series, scales, reasoning capabilities, and open-source availability, offering a diverse evaluation of LLM-based methods.

Choices of LVLMs We classify LVLMs into three categories based on their input capacity: (1) Large Input Size models, such as Qwen2.5-VL-3B and Qwen2.5-VL-7B, which can process extensive context sizes, e.g., 30 images. (2) Medium Input Size models, such as DeepSeek-VL-16B (Lu et al., 2024), which can handle a moderate number of inputs, e.g., 5 images. (3) Single Input models, such as LLaVA-Next-7B, which are limited to processing one image at a time. For each LVLM backbone, we assess its compatibility with various methods and sensitivity to context size, providing guidelines for effectively leveraging LVLMs in document understanding tasks.

4.2 Overall Performance

In this subsection, we present the overall performance of MoLoRAG alongside all baseline methods. To evaluate performance under varying retrieval availability, we consider top-K values of $K = 1, 3, 5$. Results for top-3 retrieval are shown in Table 2, while additional results for $K = 1$ and $K = 5$ are provided in Tables 9 and 10 in Appendix, respectively. For LLM w. Text RAG, each retrieved element corresponds to a text chunk, whereas for LVLM-based methods, each retrieved element represents a document page in image format. Based on the experimental results, we summarize the key findings below:

1. LLMs struggle with document understanding compared to LVLM-based methods. Even advanced LLMs like DeepSeek-V3, fall short in performance compared to LVLM-based methods. This highlights the inherent limitations of LLMs in handling multi-modal document understanding tasks, even when paired with sophisticated retrieval methods. LVLMs, on the other hand, can natively handle multi-modal inputs, making them better suited for document understanding. A fine-grained analysis across different evidence modalities (e.g., text, tables, figures) in Appendix D.7 reveals LLMs' weak performance with non-text modalities, while LVLMs excel across diverse modalities.

2. MoLoRAG consistently boosts LVLM performance. Integrating LVLMs with MoLoRAG significantly improves their question answering capabilities. For example, DeepSeek-VL-16B, which performs poorly with concatenated document images (e.g., 8.40% on MMLongBench due to content overload), achieves a substantial improvement when paired with MoLoRAG, reaching 20.43%.

Table 2: **Overall performance comparison (in %) under the retrieved top-3 setting.** The "Direct" mode processes up to 30 document pages, while "MoLoRAG+" refers to the variant with a fine-tuned retrieval engine. Results for the top-1 and top-5 settings are in Tables 9 and 10, respectively. The best performance is **highlighted**.

Type	Model	Method	MMLongBench	LongDocURL	PaperTab	FetaTab	Avg.
LLM-based	Mistral-7B	Text RAG	24.47	25.06	11.45	41.14	25.53
	Qwen2.5-7B	Text RAG	25.52	27.93	12.72	40.06	26.56
	LLaMA3.1-8B	Text RAG	22.56	29.80	13.49	45.96	27.95
	GPT-4o	Text RAG	27.23	32.74	14.25	50.20	31.11
	DeepSeek-V3	Text RAG	**29.82**	**34.73**	**17.05**	**52.36**	**33.49**
LVLM-based	LLaVA-Next-7B	Direct	7.15	10.78	3.05	11.61	8.15
		M3DocRAG	**10.10**	**13.85**	5.34	**13.98**	**10.82**
		MoLoRAG	9.37	13.49	4.83	13.78	10.37
		MoLoRAG+	9.47	13.58	**5.60**	13.48	10.53
	DeepSeek-VL-16B	Direct	8.40	14.72	6.11	16.14	11.34
		M3DocRAG	18.12	29.60	7.89	27.07	20.67
		MoLoRAG	20.43	29.98	9.67	38.98	24.77
		MoLoRAG+	**25.47**	**37.21**	**10.94**	**41.54**	**28.79**
	Qwen2.5-VL-3B	Direct	26.65	24.89	25.19	51.57	32.08
		M3DocRAG	29.11	44.40	24.68	53.25	37.86
		MoLoRAG	32.11	**45.79**	24.43	57.68	40.00
		MoLoRAG+	**32.47**	45.27	**27.23**	**58.76**	**40.93**
	Qwen2.5-VL-7B	Direct	32.77	26.38	29.77	64.07	38.25
		M3DocRAG	36.18	49.03	28.50	63.78	44.37
		MoLoRAG	39.28	51.71	**32.32**	69.09	48.10
		MoLoRAG+	**41.01**	**51.85**	31.04	**69.19**	**48.27**
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		38.53	46.91	30.03	66.34	45.45

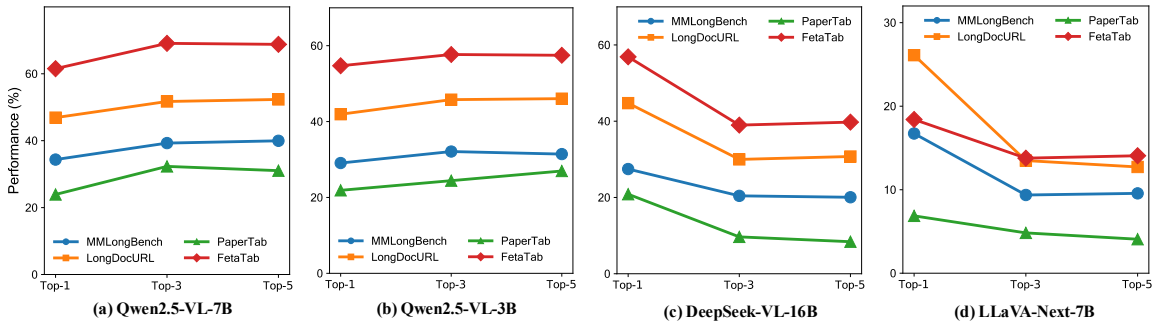

Figure 4: **Performance trends of MoLoRAG across different top-K retrieval settings for LVLMs.** LVLMs with extensive context support (e.g., Qwen2.5-VL series) benefit from retrieving more pages, improving performance with higher K. In contrast, LVLMs with limited context capacity (e.g., LLaVA-Next-7B) perform best with $K = 1$.

Similarly, high-capacity LVLMs like the Qwen2.5-VL series benefit from MoLoRAG's ability to filter and prioritize relevant pages, further improving their already strong performance.

3. Fine-tuned MoLoRAG+ delivers further performance gains. The fine-tuned variant, MoLoRAG+, outperforms the training-free version, demonstrating the benefits of task-specific optimization. For example, with DeepSeek-VL-16B, MoLoRAG+ achieves 5.04% improvement on MMLongBench compared to the training-free MoLoRAG. This enhancement stems from its superior ability to assess logical relevance, enabling more accurate retrieval (details in Section 4.3).

4. The relationship between top-K retrieval and performance depends on LVLM capability. Figure 4 illustrates how performance of MoLoRAG varies with top-K retrieval settings across four datasets. For LVLMs with extensive context support, such as the Qwen2.5-VL series, increasing K (i.e., providing more pages) improves performance across all scenarios. However, for LVLMs with limited context capacity, such as DeepSeek-VL-16B and LLaVA-Next-7B, additional pages often exceed their processing capabilities, leading to degraded performance. For these models, $K = 1$ is typically the optimal choice.

4.3 Retrieval Performance Comparison

In this subsection, we evaluate the retrieval accuracy of various methods to highlight the effectiveness of MoLoRAG in identifying relevant pages.

Table 3: **Retrieval performance comparison (in %) under the top-K setting.**

Top-K	Method	MMLongBench				LongDocURL			
		Recall	Precision	NDCG	MRR	Recall	Precision	NDCG	MRR
1	M3DocRAG	43.31	56.67	56.67	56.67	46.84	64.66	64.66	64.66
	MDocAgent (Text)	29.30	38.99	38.99	38.99	42.03	58.37	58.37	58.37
	MDocAgent (Image)	43.79	57.49	57.49	57.49	46.80	64.57	64.57	64.57
	MoLoRAG	45.46	59.95	59.95	59.95	48.98	67.71	67.71	67.71
	MoLoRAG+	**51.32**	**66.86**	**66.86**	**66.86**	**50.82**	**70.08**	**70.08**	**70.08**
3	M3DocRAG	64.17	31.62	54.13	65.36	67.00	33.78	58.23	72.51
	MDocAgent (Text)	43.21	20.77	37.13	45.26	58.53	29.33	54.12	65.28
	MDocAgent (Image)	64.74	31.97	54.75	66.12	66.67	33.62	58.26	72.47
	MoLoRAG	67.22	40.81	57.34	68.56	**70.04**	36.41	61.56	75.78
	MoLoRAG+	**68.87**	**48.67**	**64.49**	**73.50**	68.92	**47.53**	**64.90**	**77.14**
5	M3DocRAG	72.00	22.58	54.06	66.92	74.32	23.34	58.05	73.83
	MDocAgent (Text)	50.60	15.48	37.19	46.98	65.41	20.41	53.97	66.55
	MDocAgent (Image)	71.45	22.37	54.58	67.53	74.60	23.50	58.06	73.90
	MoLoRAG	**74.13**	35.83	57.29	69.63	**77.14**	26.13	61.30	76.88
	MoLoRAG+	72.37	**45.34**	**64.36**	**73.97**	73.69	**42.47**	**64.74**	**77.89**

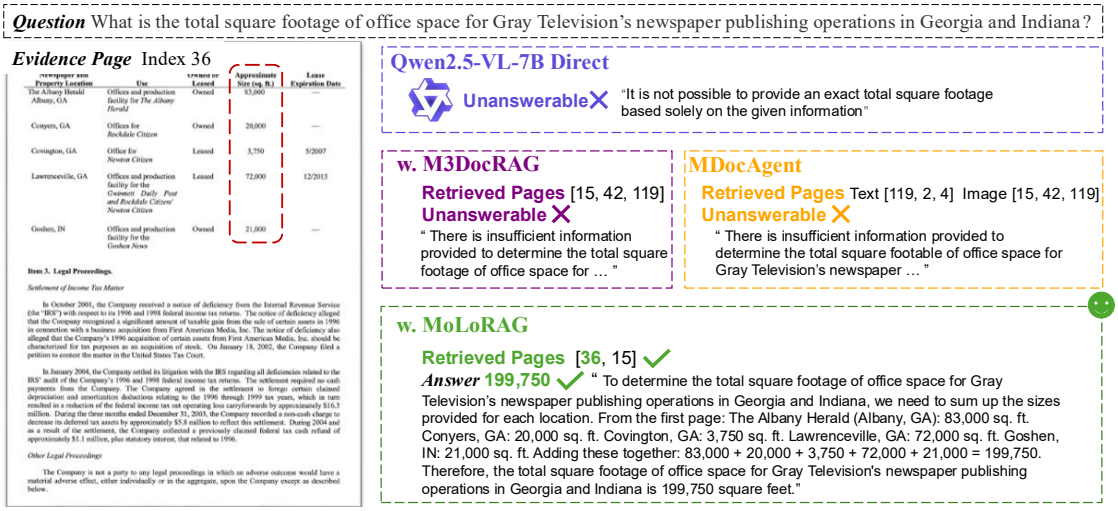

Figure 5: **Case study on LongDocURL.** MoLoRAG successfully **retrieves the correct evidence page** for the given question by leveraging logical relevance, enabling it to **provide the correct answer**. In contrast, both LVLM direct inference and other baseline methods fail to answer the question due to limited or irrelevant context.

Since only MMLongBench and LongDocURL provide ground-truth evidence pages for each query, our comparison is confined to these two datasets. We employ standard metrics, including Recall, Precision, NDCG, and MRR (details in Appendix D.1), where higher values indicate better retrieval performance. We compare MoLoRAG and its fine-tuned variant, MoLoRAG+, against two baseline methods: M3DocRAG (Cho et al., 2024) and MDocAgent (Han et al., 2025). MDocAgent performs separate text- and image-based retrieval, and results for both modalities are reported. The detailed results under top-K settings are presented in Table 3. MoLoRAG consistently outperforms baseline methods across metrics, with an average improvement of **9.94%** on MMLongBench and **7.16%** on LongDocURL. This advantage arises from MoLoRAG's integration of both semantic and logical relevance, unlike the baselines, which focus solely on semantic relevance. The fine-tuned variant, MoLoRAG+, further improves performance by leveraging task-specific optimization.

4.4 Case Study

Figure 5 presents a case study on LongDocURL. LVLM direct inference marks the question as "unanswerable" due to limited input context. Baselines such as M3DocRAG and MDocAgent rely solely on semantic relevance for retrieval, failing to locate the evidence page, which leads to incorrect answers. In contrast, MoLoRAG accurately retrieves the evidence page by considering logical relevance, enabling the LVLM to leverage this knowledge and correctly answer the question. Another case involving **cross-page understanding** is illustrated in Figure 8 in the Appendix.

Due to space limits, **ablation study** and **efficiency analysis** are moved to Appendix D.6 and D.4.

5 Conclusion

In this paper, we tackle the DocQA task by addressing the limitations of prior methods that rely only on semantic relevance for retrieval. By incorporating logical relevance, our VLM-powered retrieval engine performs multi-hop reasoning over page graph to identify key pages. Extensive experiments demonstrate that MoLoRAG delivers superior retrieval accuracy, achieves SOTA performance, and ensures seamless compatibility with LVLMs.

Acknowledgments

This research is supported in part by project #MMT-p2-23 of the Shun Hing Institute of Advanced Engineering, The Chinese University of Hong Kong, by grants from the Research Grants Council of the Hong Kong SAR, China (No. CUHK 14217622). This research is also supported in part by the National Natural Science Foundation of China (No.62302098), and the Fujian Provincial Natural Science Foundation of China (2025J01540). The authors would like to express their gratitude to the reviewers for their valuable feedback, which has improved the clarity and contribution of the paper.

Limitations

MoLoRAG primarily focuses on closed-domain document understanding, where the relevant document is provided. Extending this approach to an **open-domain** setting, where the document corpus consists of extensive and diverse documents, is a challenge. This is because modeling relationships not only within individual document but also across different documents, as well as performing graph traversal between both document- and page-level nodes, becomes complex.

In this paper, we did not use any non-public data, unauthorized software, or APIs, and there are no privacy or other related ethical concerns associated with our work.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W. Cohen. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. *Preprint*, arXiv:2210.02928.

Zhanpeng Chen, Chengjin Xu, Yiyan Qi, and Jian Guo. 2024. Mllm is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training. *arXiv preprint arXiv:2407.21439*.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3docrag: Multimodal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*.

DeepSeek-AI. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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. 2024. Longdocurl: a comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. *Preprint*, arXiv:2412.18424.

Yihao Ding, Zhe Huang, Runlin Wang, YanHang Zhang, Xianru Chen, Yuzhong Ma, Hyunsuk Chung, and Soyeon Caren Han. 2022. V-doc: Visual questions answers with documents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21492–21498.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yulong Hui, Yao Lu, and Huanchen Zhang. 2024. UDA: A benchmark suite for retrieval augmented generation in real-world document analysis. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. 2024a. Llava-next: Stronger llms supercharge multimodal capabilities in the wild.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. 2024b. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12758–12786, Miami, Florida, USA. Association for Computational Linguistics.

Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. *Preprint*, arXiv:2502.12442.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Xueguang Ma, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Wenhu Chen, and Jimmy Lin. 2024a. Visa: Retrieval augmented generation with visual source attribution. *arXiv preprint arXiv:2412.14457*.

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. 2024b. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Preprint*, arXiv:2407.01523.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. 2020. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Qwen. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations (ICLR)*.

Jianwei Sun, Chaoyang Mei, Linlin Wei, Kaiyu Zheng, Na Liu, Ming Cui, and Tianyi Li. 2024. Dial-insight: Fine-tuning large language models with high-quality domain-specific data preventing capability collapse. *Preprint*, arXiv:2403.09167.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A Rossi, and Dinesh Manocha. 2024. Visdom: Multi-document qa with visually rich elements using multimodal retrieval-augmented generation. *arXiv preprint arXiv:2412.10704*.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidevqa: A dataset for document visual question answering on multiple images. In *AAAI*.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. Hierarchical multimodal transformers for multi-page docvqa. *Preprint*, arXiv:2212.05935.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. 2024a. Mineru: An open-source solution for precise document content extraction. *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. *Preprint*, arXiv:2409.12191.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *Preprint*, arXiv:2409.01704.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2025. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. *arXiv preprint arXiv:2506.05690*.

Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. 2024. Ocr hinders rag: Evaluating the cascading impact of ocr on retrieval-augmented generation. *arXiv preprint arXiv:2412.02592*.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. 2025. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*.

Algorithm 1 Graph Traversal for Retrieval

Require: Question q, DocEncoder$(\cdot)$, Page graph G, Exploration size w, Hop limit n_{hop}, Document $\mathcal{D}$

Ensure: Re-ranked pages $\mathcal{D}^r$

$s_i^{\text{sem}} \leftarrow \langle \text{DocEncoder}(q), \text{DocEncoder}(p_i) \rangle$ for each $p_i \in \mathcal{D}$ ▷ Semantic relevance scoring
$\mathcal{B} \leftarrow \text{TopK}(\{s_i^{\text{sem}}\}, w)$ ▷ Exploration set initialization
$\mathcal{D}^r \leftarrow \emptyset$
$\mathcal{S} \leftarrow \mathcal{B}$ ▷ Visited marking
for $p_i \in \mathcal{B}$ **do**
  $s_i^{\text{logi}} \leftarrow \text{VLM}(q, p_i)$ ▷ Logical relevance scoring
  $s_i \leftarrow \text{Combine}(s_i^{\text{sem}}, s_i^{\text{logi}})$ ▷ Score update
  $\mathcal{D}^r \leftarrow \mathcal{D}^r \cup \{(p_i, s_i)\}$
end for
for Hop $= 1$ to n_{hop} **do**
  $\mathcal{C} \leftarrow \emptyset$ ▷ Candidates initialization
  for $p_i \in \mathcal{B}$ **do**
    for $p_j \in \text{Neighbor}(G, p_i), p_j \notin \mathcal{S}$ **do**
      $s_j^{\text{logi}} \leftarrow \text{VLM}(q, p_j)$
      $s_j \leftarrow \text{Combine}(s_j^{\text{sem}}, s_j^{\text{logi}})$
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{s_j\}$
      $\mathcal{D}^r \leftarrow \mathcal{D}^r \cup \{(p_j, s_j)\}$
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{p_j\}$
    end for
  end for
  $\mathcal{B} \leftarrow \text{TopK}(\mathcal{C}, w)$ ▷ Exploration set update
end for
Sort $\mathcal{D}^r$ by descending s ▷ Pages re-ranking
return $\mathcal{D}^r$

A Algorithm Pseudo-Code

The graph traversal algorithm for retrieval is presented in Algorithm 1. This algorithm operates by efficiently identifying relevant pages through a combination of semantic and logical relevance scores. By leveraging an exploration size w and a hop limit n_{hop}, the traversal is restricted to exploring only the most promising paths in the page graph, ensuring scalability and avoiding the need to process all pages. The output, $\mathcal{D}^r$, is a re-ranked set of document pages that are both semantically and logically relevant, with its size typically smaller than the total number of document pages N. From this re-ranked set, the top-K pages are selected and passed to the next stage for question answering.

B Prompt

In this section, we present all the prompts used within the MoLoRAG framework, including querying the VLM to assign logical relevance scores, using LLMs or LVLMs for question answering, and prompting GPT-4o to curate the dataset for training MoLoRAG+.

Assessing Logical Relevance The prompt for querying the VLM to assign a logical relevance score between the observed image and the question is provided below:

Prompt for Assessing Logical Relevance

 # GOAL #
You are an Retrieval Expert, and your task is to evaluate how **relevant** the input document page is to the given query. Rate the relevance on a scale of 1 to 5, where:
5 Highly relevant - contains complete information needed to answer the query
4 Very relevant - contains most of the information needed
3 Moderately relevant - contains some useful information
2 Slightly relevant - has minor connection to the query
1 Irrelevant - contains no information related to the query

 # INSTRUCTION #
Please first read the given query, think about what knowledge is required to answer that query, and then carefully go through the document snapshot for judgment.

 # QUERY #
{{Question}}
Please generate just a single number (1-5) representing your relevance judgment. Your answer should be a single number without any extra contents.

LLM Question Answering For LLM-based question answering, all retrieved text chunks are concatenated into the `context`, and the LLM is queried to answer the given question based on the provided context as:" {{Context}} Answer the question based on the above context: {{Question}}".

LVLM Question Answering For LVLM-based question answering, the input includes document images and the question: "{{Image}} Based on the document, please answer the question:

{{Question}}".

Curating Training Data The prompt for guiding GPT-4o to generate training data is shown below:

Prompt for Curating Training Data

 # GOAL #

Given the input image, your task is to generate a question related to it. The relevance score is {{relevance_score}}, where a higher score indicates a closer connection between the question and the image. For example, a relevance score of 5 means the answer is **DIRECTLY** contained in the image, while a score below 3 indicates that the answer **CANNOT** be derived from it, with lower scores signifying less relevance.

 # REQUIREMENT #

The question must be based on the content of the input image, except when the relevance score is ≤ 2. For relevance scores of 4 or higher, create clear and straightforward questions with answers that are explicitly present in the image. For relevance scores of 3, generate questions that may require some inference but are still somewhat related to the content. For relevance scores of 2 or lower, formulate questions that are unanswerable based on the snapshot.

You may consider various elements, including text, layout, and figures. For this generation, please concentrate on {{focus}} if applicable and remember that the relevance score is {{relevance_score}}.

Your output should be formatted as follows: { "query": "Your generated question", "relevance_score": "relevance score", "answer": "Corresponding answer or inference" }

C Supplementary Materials for MoLoRAG+

This section provides detailed information on the training data, learning configurations, and alternative data engine for MoLoRAG+.

Training Data In the first stage, **Question Generation**, we prompt GPT-4o to generate approximately 5,500 samples using the illustrated prompt. Document snapshots are randomly selected from MMLongBench (Ma et al., 2024b) and LongDocURL (Deng et al., 2024), as these datasets contain multimodal, information-rich documents. The relevance score s is sampled from $\{1, 2, 3, 4, 5\}$ with equal distribution to ensure a balanced representation across different levels of logical relevance, preventing over-fitting to specific scores. For each sampled document snapshot p_i, the generated question is expressed as: $q' = \text{GPT-4o}(\text{prompt}, s, p_i)$, where p_i denotes the randomly selected document snapshot, and s is the sampled relevance score. To ensure data quality, each generated question q' and its corresponding document snapshot p_i are fed back to GPT-4o to assign a predicted logical relevance score as: $s' = \text{GPT-4o}(\text{prompt}, q', p_i)$. Samples are retained only if the predicted score closely matches the original score, i.e., $|s - s'| \leq 1$. After this automated filtering, the remaining samples undergo manual verification. This process results in a final high-quality training set containing 3,519 samples, formatted as triplets: $\langle \text{Question } q', \text{Image } p_i, \text{Relevance Score } s' \rangle$. In each sample, s', the predicted relevance score, is the expected output for model training.

Learning Configuration We utilize the LLaMA-Factory package[4] to fine-tune the backbone VLM, Qwen2.5-VL-3B (Bai et al., 2025), using the LoRA technique for parameter-efficient training. The fine-tuning process is configured with the following hyperparameters: a LoRA Rank of 8, a learning rate of 1×10^{-4}, a warmup ratio of 0.1, and gradient accumulation steps set to 8. The model is trained for a total of 2 epochs, ensuring efficient and effective parameter adaptation.

Alternative Data Engine We primarily use GPT-4o as the data engine for curating training data. However, our data generation pipeline is flexible and **can accommodate other LVLMs**. To demonstrate this flexibility, we replace the original engine with the open-source model Qwen2.5-VL-32B (Bai et al., 2025), while keeping all prompts and processes consistent. This variant is denoted as MoLoRAG$^{+}_{\text{Qwen}}$. We compare the retrieval performance of MoLoRAG, MoLoRAG+, and MoLoRAG$^{+}_{\text{Qwen}}$ on MMLongBench (Table 4), where the backbones are pre-trained Qwen2.5-VL-3B, Qwen2.5-VL-3B fine-tuned with GPT-4o data, and fine-tuned with Qwen2.5-VL-32B data, respectively. The results indicate that the Qwen2.5-VL-32B distilled model performs **comparably** to its GPT-4o counterparts. This is attributed to (1) the simplicity of logical relevance scoring, enabling effective high-quality data generation by Qwen2.5-

[4] https://github.com/hiyouga/LLaMA-Factory/

Table 4: **Retrieval performance comparison (in %)** between MoLoRAG, MoLoRAG$^{+}_{\text{Qwen}}$, and MoLoRAG+ on MMLongBench.

Top-K	Model	Recall	Precision	NDCG	MRR
1	MoLoRAG	45.46	59.95	59.95	59.95
	MoLoRAG$^{+}_{\text{Qwen}}$	**51.62**	**67.56**	**67.56**	**67.56**
	MoLoRAG+	51.32	66.86	66.86	66.86
3	MoLoRAG	67.22	40.81	57.34	68.56
	MoLoRAG$^{+}_{\text{Qwen}}$	**71.79**	37.94	64.24	**74.57**
	MoLoRAG+	68.87	**48.67**	**64.49**	73.50
5	MoLoRAG	74.13	35.83	57.29	69.63
	MoLoRAG$^{+}_{\text{Qwen}}$	**78.72**	29.06	64.01	**75.69**
	MoLoRAG+	72.37	**45.34**	**64.36**	73.97

VL-32B, and (2) the shared family of the data engine and distilled model, which facilitates capability transfer. Additionally, this variant reduces data construction costs due to its open-source nature, highlighting the effectiveness of our data construction pipeline.

D Supplementary Materials for Experiments

D.1 Details of Metrics

Evaluation for Question Answering We utilize **Accuracy** and **Exact Match** as evaluation metrics for MMLongBench (Ma et al., 2024b) and LongDocURL (Deng et al., 2024). Accuracy is rule-based to accommodate various answer types, with detailed explanations provided in Appendix B.3 of Ma et al. (2024b). Exact Match measures the percentage of predictions where the generated answer **exactly** matches the ground-truth answer. For the remaining datasets, PaperTab and FetaTab (Hui et al., 2024), we employ GPT-4o as the evaluator to assign a **Binary Correctness** score $\in \{0, 1\}$, where 1 indicates that the generated answer aligns with the ground-truth answer. We report the averaged values across all test samples.

Evaluation for Retrieval We employ standard retrieval metrics, including Recall@K, Precision@K, NDCG@K, and MRR@K, where K represents the number of retrieved elements. For a specific data sample with ground-truth evidence pages denoted as $\mathcal{P}^{\text{gt}} = \{p_1^{\text{gt}}, \ldots, p_n^{\text{gt}}\}$ and the top-K retrieved pages $\mathcal{P}^r = \{p_1^r, \ldots, p_K^r\}$, these metrics are computed as follows:

- **Recall** measures the proportion of ground-truth pages that are successfully retrieved within the top-K results:

$$\text{Recall@}K = \frac{\sum_{i=1}^{K} \mathbb{I}(p_i^r \in \mathcal{P}^{\text{gt}})}{n},$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the condition is true and 0 otherwise.

- **Precision** assesses the accuracy of the retrieved pages by calculating the proportion of retrieved pages that are relevant:

$$\text{Precision@}K = \frac{\sum_{i=1}^{K} \mathbb{I}(p_i^r \in \mathcal{P}^{\text{gt}})}{K}.$$

- **NDCG** (Normalized Discounted Cumulative Gain) evaluates the ranking quality of the retrieved pages by considering the positions of relevant pages within the top-K results. It is computed in three steps:

$$\text{DCG@}K = \sum_{i=1}^{\min(n,K)} \frac{\mathbb{I}(p_i^r \in \mathcal{P}^{\text{gt}})}{\log_2(i+1)},$$

$$\text{IDCG@}K = \sum_{i=1}^{\min(n,K)} \frac{1}{\log_2(i+1)},$$

$$\text{NDCG@}K = \frac{\text{DCG@}K}{\text{IDCG@}K}$$

- **MRR** (Mean Reciprocal Rank) measures the reciprocal rank of the first relevant page within the top-K retrieved pages. It is defined as:

$$\text{MRR@}K = \begin{cases} \frac{1}{i} & \text{if } p_i^r \text{ is the } \textbf{first} \text{ relevant} \\ 0 & \text{otherwise} \end{cases}$$

where i denotes the position of the **first relevant page** that satisfies $p_i^r \in \mathcal{P}^{\text{gt}}$. If no relevant page is retrieved within the top-K, the MRR score for that sample is 0.

For each metric, we average the scores over all test samples and report the values in Tables 3, 7 and 8, respectively.

D.2 Implementation Details

This subsection outlines the detailed configurations for each method to ensure clarity and reproducibility. All experiments were conducted on 3 NVIDIA A6000 48G GPUs.

- **LLM w. Text RAG** We use the `PyPDFLoader` from the LangChain package[5] to extract text from the raw document. Each document is divided into chunks of 1,000 tokens. The QwenRAG API[6] is employed as the text RAG engine. Specifically, we use the `text-embedding-v1` model to encode text chunks and perform retrieval. For each top-K setting, the top-ranked K text chunks are combined as context, which is then passed to the LLM for answering.

- **LVLM Direct Inference** To ensure scalability, we truncate documents to retain only the first 30 pages. For Qwen2.5-VL series, as these models support extensive image contexts, all images are fed directly for processing. For DeepSeek-VL-16B, although it supports multi-image inputs, it requires significant memory for loading. Therefore, we concatenate document images into 5 larger images, ensuring compatibility with GPU memory. For LLaVA-Next-7B, as it accepts only a single image, all available pages are combined into one single image for processing.

- **M3DocRAG** This baseline is implemented according to its original paper and official repository. The document encoder used is `colpali` for encoding and retrieval. While the original paper uses Qwen2-VL-7B (Wang et al., 2024b) as the backbone LVLM, we extend the evaluation by integrating the method with various LVLMs to assess compatibility.

- **MDocAgent** This baseline is implemented following its official repository: `colpaligemma-3b-mix-448-base` for image retrieval and `colbertv2.0` for text retrieval. For all five agents in this framework, we consistently use the original LLaMA3.1-8B (Grattafiori et al., 2024) as the LLM for the text agent, while employing a consistent LVLM, i.e., Qwen2.5-VL-7B (Bai et al., 2025), for remaining agents.

- **MoLoRAG** For document encoding, `colpali` is used as the document encoder. Pages are connected in the graph if their similarity score exceeds a threshold of 0.4. During graph traversal, the number of hops n_{hop} is set to 4, and the exploration set size w is set to 3. Semantic relevance and LVLM-generated logical relevance are combined using an average score. All visited pages are re-ranked based on this combined score for final retrieval. Qwen2.5-VL-3B is employed as the retrieval engine due to its strong performance and lightweight architecture, ensuring both effectiveness and efficiency. The retrieved top-K images are fed into LVLMs based on their input format capabilities: for LLaVA-Next-7B, the images are concatenated into a single composite image, while for all other LVLMs, the images are processed separately.

[5] https://python.langchain.com/

[6] https://dashscope.console.aliyun.com/overview

D.3 Discussion of Advanced OCR Methods

We expand our discussion on the LLM w. Text RAG baseline by using more advanced OCR tools, including `MinerU` (Wang et al., 2024a) and `GOT-OCR-2.0` (Wei et al., 2024), as alternatives to `PyPDFLoader`, while ensuring consistency in the retrieval engine and LLM calling process. The results across different top-K settings on MMLongBench and PaperTab are presented in Table 5. Our findings indicate that advanced tools generally enhance QA performance due to improved OCR capabilities, with the benefits becoming more pronounced as K increases. For instance, replacing `PyPDFLoader` with `MinerU` yields performance gains of up to 4% across various LLMs. However, **LLMs w. Text RAG baselines still show a performance gap compared to strong LVLMs**, primarily due to the inevitable loss of multi-modal information.

D.4 Efficiency Analysis

In this subsection, we provide efficiency analysis of our MoLoRAG with baseline methods from three aspects: (1) Retrieval scalability, (2) Inference efficiency, and (3) Total time costs.

Retrieval Scalability The retrieval stage of MoLoRAG requires the VLM to evaluate the logical relevance score of each visited page. To efficiently manage the traversal scope, we introduce hyper-parameters such as hop limit and exploration set size, which help narrow the querying space, ensuring scalability and accelerating the retrieval process. To illustrate the scalability of MoLoRAG, Figure 6 shows the average number of queried pages alongside the total number of pages for all testing samples across different datasets. The figure also indicates the percentage of queried pages. As the document size increases, the average percentage

Table 5: **Comparison of various OCR methods on the performance of LLM w. Text RAG.**

Model	OCR	Top-1		Top-3		Top-5	
		MMLongBench	PaperTab	MMLongBench	PaperTab	MMLongBench	PaperTab
Qwen2.5-7B	PyPDFLoader	22.11	5.34	25.52	12.72	26.09	16.79
	GOT-OCR-2.0	22.66	6.11	25.84	13.74	26.70	16.79
	MinerU	21.99	7.63	24.54	16.28	27.53	19.08
GPT-4o	PyPDFLoader	24.07	8.65	27.23	14.25	28.74	20.36
	GOT-OCR-2.0	23.86	8.91	27.78	13.74	29.47	18.07
	MinerU	25.89	9.92	28.74	18.32	30.98	22.90
DeepSeek-V3	PyPDFLoader	25.94	10.18	29.82	17.05	31.23	23.92
	GOT-OCR-2.0	25.35	9.92	28.39	18.83	30.75	22.90
	MinerU	26.21	11.96	30.11	21.12	32.36	26.97
Qwen2.5-VL-7B w. MoLoRAG		34.35	23.92	39.28	32.32	39.97	31.04

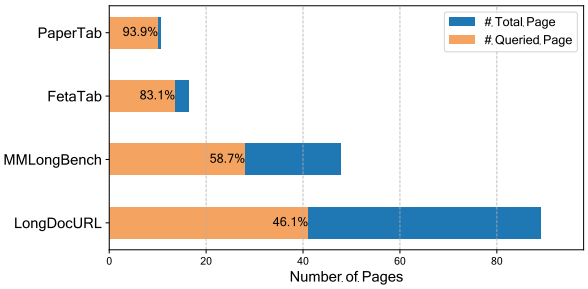

Figure 6: **Illustration of retrieval scalability.** We present the average number of pages queried by MoLoRAG and the total number of pages across all test samples for each dataset.

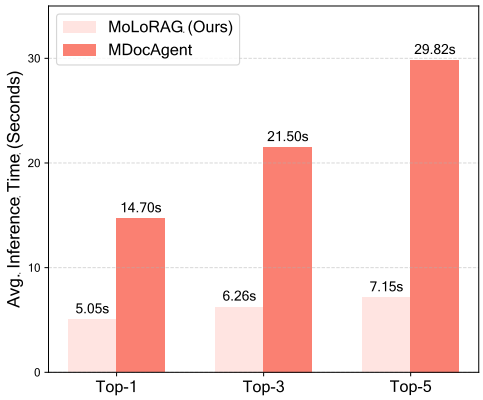

Figure 7: **Inference time comparison between MoLoRAG and MDocAgent (Han et al., 2025) on the MMLongBench.**

of queried pages decreases significantly. For instance, fewer than 50% of the pages are queried for LongDocURL, demonstrating MoLoRAG's ability to effectively control graph traversal and focus on relevant pages. This reduction highlights the scalability of the method, as it maintains high retrieval accuracy while minimizing computational overhead even for large documents.

Inference Efficiency After retrieval, MoLoRAG requires only **a single query** to the LVLM for question answering by providing the relevant pages, ensuring efficiency comparable to LVLM Direct Inference. In contrast, the best-performing baseline, MDocAgent (Han et al., 2025), requires **five separate queries** to both LLMs and LVLMs, as it functions as a unified multi-agent system. Figure 7 illustrates the average inference times of MoLoRAG (Qwen2.5-VL-7B as the backbone) and MDocAgent, clearly highlighting MoLoRAG's significant advantage in inference efficiency, making it a more practical and scalable solution.

Total Time Costs In addition to inference time efficiency, we present the total time (retrieval and inference) for various methods to provide a clearer illustration. Note that the retrieval time includes both indexing and retrieval processes. DeepSeek-V3 and GPT-4o are invoked via APIs in a sequential manner to ensure a fair comparison, while the remaining open-source LLMs are deployed locally using vLLM on a single NVIDIA A6000-48G GPU. For LVLM Direct, the number of pages considered is up to 30, with times reported using 3 A6000 GPUs. All other time costs are measured on a single A6000 GPU, except for MDocAgent, which utilizes 2 A6000 GPUs. The results are shown in Table 6.

For LLM with Text RAG, variations in latency occur due to differences in invocation methods and model architecture. The index stage that invokes Qwen-RAG via APIs significantly affects overall latency. In LVLM-based methods, direct inference involves managing 30 pages, which can slow processing. While M3DocRAG is the most efficient method, it focuses solely on semantic relevance, limiting retrieval accuracy. The robust baseline, MDocAgent, employs parallel text and image retrieval; however, its multi-agent framework can

Table 6: **Total time costs comparison (in seconds) across different methods on LongDocURL.**

Type	Model	Method	Top-1 Retrieval Time	Top-1 Inference Time	Top-1 Total	Top-5 Retrieval Time	Top-5 Inference Time	Top-5 Total
LLM-based	Mistral-7B	Text RAG	48.5s	2.4s	50.9s	48.5s	4.1s	52.6s
	Qwen2.5-7B	Text RAG	48.5s	5.1s	53.6s	48.5s	6.9s	55.4s
	LLaMA3.1-8B	Text RAG	48.5s	11.8s	60.3s	48.5s	18.3s	66.8s
	GPT-4o	Text RAG	48.5s	1.4s	49.9s	48.5s	1.5s	50.0s
	DeepSeek-V3	Text RAG	48.5s	7.5s	56.0s	48.5s	21.4s	69.9s
LVLM-based	Qwen2.5-VL-3B	Direct	-	34.3s	34.3s	-	34.3s	34.3s
		M3DocRAG	10.7s	5.7s	16.4s	10.7s	8.3s	19.0s
		MoLoRAG	38.4s	5.7s	44.1s	38.4s	8.2s	46.6s
	Qwen2.5-VL-7B	Direct	-	47.1s	47.1s	-	47.1s	47.1s
		M3DocRAG	10.7s	6.8s	17.5s	10.7s	10.2s	20.9s
		MoLoRAG	38.4s	6.8s	45.2s	38.4s	10.1s	48.5s
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		31.8s	16.8s	48.6s	31.8s	35.5s	67.3s

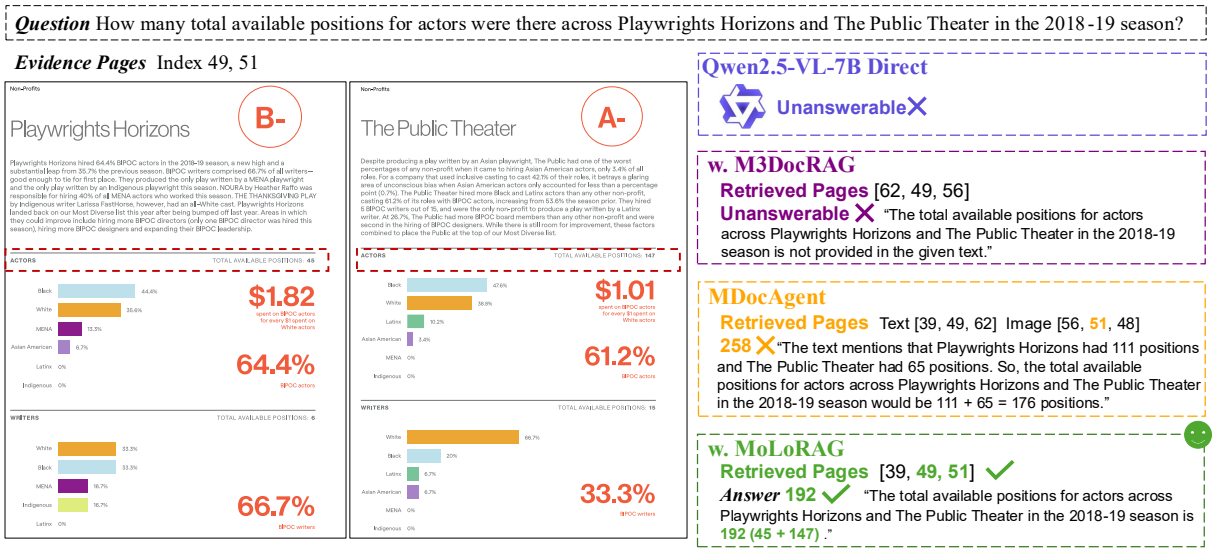

Figure 8: **Case study under the top-3 setting on LongDocURL involving cross-page reasoning.** The given question requires evidence from two separate pages to derive the correct answer. LVLM direct inference fails due to its limited context capacity (e.g., processing up to 30 pages). M3DocRAG is unable to retrieve the relevant content, leading to failure. MDocAgent relies on only one evidence page and falls into hallucination, producing an incorrect answer. In contrast, MoLoRAG successfully **retrieves both relevant pages**, enabling the LVLM to leverage this information to **answer the question correctly**.

hinder overall efficiency as context complexity increases. Despite the associated retrieval time costs, MoLoRAG maintains a notably fast inference stage and scales well with larger K values, ensuring a balance between efficiency and performance.

D.5 Case Study

We present another case involving cross-page reasoning, as shown in Figure 8. The question asks: "How many total available positions for actors were there across Playwrights Horizons and The Public Theater in the 2018–19 season?" Answering this requires evidence from two separate pages: the available positions for actors at Playwrights Horizons and The Public Theater.

LVLM direct inference fails due to limited context capacity, as it can only process up to 30 pages, while the evidence pages are located at indices 49 and 51. Both M3DocRAG and MDocAgent fail to retrieve these evidence pages because they rely solely on semantic relevance. Specifically, MDocAgent retrieves only one evidence page, resulting in hallucination and an incorrect answer. In contrast, MoLoRAG effectively retrieves both relevant pages, enabling the LVLM to access the necessary information and correctly compute the total available positions, demonstrating its superior cross-page reasoning capability.

D.6 Ablation Study

In this subsection, we present the ablation study to evaluate the effectiveness of individual components

Table 7: **Retrieval performance comparison of MoLoRAG$_{\text{Logi}}$ and MoLoRAG+ with the same fine-tuned retrieval engine.** MoLoRAG$_{\text{Logi}}$, which relies solely on logical relevance, consistently underperforms compared to MoLoRAG+, demonstrating the effectiveness of combining semantic and logical relevance.

Dataset	K	Method	Recall	NDCG	MRR
MMLong	1	MoLoRAG$_{\text{Logi}}$	46.55	58.90	58.90
		MoLoRAG+	**51.32**	**66.86**	**66.86**
	3	MoLoRAG$_{\text{Logi}}$	60.45	58.02	64.62
		MoLoRAG+	**68.87**	**64.49**	**73.50**
LongDocURL	1	MoLoRAG$_{\text{Logi}}$	47.65	65.56	65.56
		MoLoRAG+	**50.82**	**70.08**	**70.08**
	3	MoLoRAG$_{\text{Logi}}$	62.71	61.27	71.53
		MoLoRAG+	**68.92**	**64.90**	**77.14**

within MoLoRAG, focusing on two key aspects: the combination of semantic and logical relevance, and the graph construction process.

Combination of Semantic and Logical Relevance We consider a variant of MoLoRAG that relies solely on logical relevance by setting $s_i = s_i^{\text{logi}}$ within the framework. This variant is referred to as MoLoRAG$_{\text{logi}}$. In this setup, the retrieval engine is the fine-tuned Qwen2.5-VL-3B, which already demonstrates strong task understanding. We compare the retrieval performance of MoLoRAG$_{\text{logi}}$ with MoLoRAG+ in Table 7. From the results, it becomes evident that relying solely on logical relevance does not achieve optimal performance. This is because VLMs may exhibit over-confidence and, in some cases, fall into hallucinations when relying exclusively on reasoning capabilities. Additionally, logical relevance scores are discrete, often leading to multiple pages with identical scores, making it difficult to rank and distinguish between them. Consequently, it is more reliable to combine both logical and semantic relevance.

Effectiveness of Graph Construction We evaluate another variant of MoLoRAG, named MoLoRAG$_{\text{Full}}$, in which the VLM **traverses all pages within the document** for a given question, assigning a relevance score to each page. Each page's relevance score is updated by combining semantic and logical relevance scores. After traversal, only the top-K pages are retained as the final retrieval result. Unlike MoLoRAG, this variant **eliminates the graph-based indexing mechanism** and instead **sequentially processes all pages in the document for selection**. We compare the retrieval performance of this variant with MoLoRAG+ in Table 8. While MoLoRAG$_{\text{Full}}$ achieves a slight improvement due to the expanded set of queried pages, it requires nearly twice the computational time (Figure 6), significantly reducing efficiency. Furthermore, this marginal performance difference highlights the effectiveness of our graph construction, which provides a high-quality candidate set for retrieval.

Table 8: **Retrieval performance comparison between MoLoRAG$_{\text{Full}}$ and MoLoRAG+ using the same fine-tuned retrieval engine.** MoLoRAG$_{\text{Full}}$ sequentially queries each page of the entire document and combines their relevance scores for final re-ranking. Although this variant achieves a marginal improvement over MoLoRAG+, it requires nearly **twice the computational time** (Figure 6), significantly reducing efficiency.

Dataset	K	Method	Recall	NDCG	MRR
MMLong	1	MoLoRAG$_{\text{Full}}$	**51.63**	**67.21**	**67.21**
		MoLoRAG+	51.32	66.86	66.86
	3	MoLoRAG$_{\text{Full}}$	**73.64**	64.31	**75.20**
		MoLoRAG+	68.87	**64.49**	73.50
LongDocURL	1	MoLoRAG$_{\text{Full}}$	**51.24**	**70.68**	**70.68**
		MoLoRAG+	50.82	70.08	70.08
	3	MoLoRAG$_{\text{Full}}$	**72.30**	64.32	**78.53**
		MoLoRAG+	68.92	**64.90**	77.14

D.7 Fine-grained Analysis

This subsection provides a detailed performance analysis across evidence modalities (e.g., text, tables, figures) and question contexts (e.g., single-page and multi-page understanding). The analysis focuses on the MMLongBench and LongDocURL datasets, which offer official splits based on modalities and locations. For MMLongBench, results for top-1, top-3, and top-5 settings are presented in Tables 11, 12, and 13, respectively. For LongDocURL, the corresponding results are shown in Tables 14, 15, and 16. From these results, we conclude that **LVLMs w. MoLoRAG excel across diverse modalities**. While LLMs w. Text RAG methods perform reasonably well on text-based modalities, they struggle significantly with non-textual content like figures (e.g., 9.83% on Figure). This highlights the critical need for robust multi-modal understanding. In contrast, LVLMs integrated with MoLoRAG demonstrate strong and balanced performance across all modalities. For example, Qwen2.5-VL-7B with MoLoRAG+ achieves 37.43% on Text and 36.94% on Figure (MMLongBench in the top-1 setting), showing the effectiveness of retrieval-based multi-modal reasoning.

Table 9: **Overall performance comparison (in %) under the retrieved top-1 setting.** The best performance across all methods is **highlighted**.

Type	Model	Method	MMLongBench	LongDocURL	PaperTab	FetaTab	Avg.
LLM-based	Mistral-7B	Text RAG	21.66	17.63	5.09	24.11	17.12
	Qwen2.5-7B	Text RAG	22.11	20.75	5.34	22.64	17.71
	LLaMA3.1-8B	Text RAG	18.70	22.57	7.12	28.25	19.16
	GPT-4o	Text RAG	24.07	22.84	8.65	34.15	22.43
	DeepSeek-v3	Text RAG	**25.94**	**24.32**	**10.18**	**34.55**	**23.75**
LVLM-based	LLaVA-Next-7B	Direct	7.15	10.78	3.05	11.61	8.15
		M3DocRAG	16.32	25.25	6.62	15.26	15.86
		MoLoRAG	16.73	26.11	**6.87**	**18.41**	**17.03**
		MoLoRAG+	**17.15**	**27.00**	6.36	17.52	17.01
	DeepSeek-VL-16B	Direct	8.40	14.72	6.11	16.14	11.34
		M3DocRAG	26.23	42.21	16.54	48.43	33.35
		MoLoRAG	27.47	44.75	20.87	56.89	37.50
		MoLoRAG+	**28.98**	**45.17**	**21.88**	**58.27**	**38.58**
	Qwen2.5-VL-3B	Direct	26.65	24.89	**25.19**	51.57	32.08
		M3DocRAG	26.77	39.82	19.85	45.77	33.05
		MoLoRAG	29.08	41.95	21.88	54.72	36.91
		MoLoRAG+	**30.03**	**43.17**	23.16	**55.41**	**37.94**
	Qwen2.5-VL-7B	Direct	32.77	26.38	**29.77**	**64.07**	38.25
		M3DocRAG	32.29	43.32	19.34	50.98	36.48
		MoLoRAG	34.35	46.89	23.92	61.52	41.67
		MoLoRAG+	**36.37**	**47.86**	27.48	62.50	**43.55**
Multi-agent	MDocAgent (LLaMA3.1-8B + Qwen2.5-VL-7B)		31.73	44.42	21.63	57.78	38.89

Table 10: **Overall performance comparison (in %) under the retrieved top-5 setting.** The best performance across all methods is **highlighted**.

Type	Model	Method	MMLongBench	LongDocURL	PaperTab	FetaTab	Avg.
LLM-based	Mistral-7B	Text RAG	23.43	26.43	13.23	48.62	27.93
	Qwen2.5-7B	Text RAG	26.09	31.36	16.79	49.21	30.86
	LLaMA3.1-8B	Text RAG	24.25	33.27	17.81	54.53	32.47
	GPT-4o	Text RAG	28.74	36.98	20.36	57.78	35.97
	DeepSeek-v3	Text RAG	**31.23**	**39.04**	**23.92**	**62.01**	**39.05**
LVLM-based	LLaVA-Next-7B	Direct	7.15	10.78	3.05	11.61	8.15
		M3DocRAG	**10.43**	12.65	**4.58**	12.80	**10.12**
		MoLoRAG	9.56	12.72	4.07	**14.07**	10.11
		MoLoRAG+	9.19	**13.59**	4.33	13.09	10.05
	DeepSeek-VL-16B	Direct	8.40	14.72	6.11	16.14	11.34
		M3DocRAG	18.87	29.27	8.14	27.26	20.89
		MoLoRAG	20.07	30.76	8.40	39.76	24.75
		MoLoRAG+	**24.86**	**38.02**	**9.67**	**41.44**	**28.50**
	Qwen2.5-VL-3B	Direct	26.65	24.89	25.19	51.57	32.08
		M3DocRAG	28.38	44.67	27.48	55.22	38.94
		MoLoRAG	31.43	**46.05**	26.97	57.48	40.48
		MoLoRAG+	**32.41**	45.13	**27.48**	**58.07**	**40.77**
	Qwen2.5-VL-7B	Direct	32.77	26.38	29.77	64.07	38.25
		M3DocRAG	37.19	50.33	30.53	64.37	45.61
		MoLoRAG	39.97	52.33	31.04	68.80	48.04
		MoLoRAG+	**40.47**	**52.33**	**31.55**	**69.39**	**48.44**
Multi-agent	MDocAgent (LLaMA3.1-8B + Qwen2.5-VL-7B)		38.34	48.07	29.77	63.78	44.99

Table 11: **Fine-grained performance analysis (Accuracy in %) across evidence modality and evidence locations on MMLongBench under the retrieved top-1 setting.** "UNA" refers to unanswerable questions.

Type	Model	Method	Modality					Location			Overall	
			Text	Table	Figure	Chart	Layout	Single	Multiple	UNA	Acc	EM
LLM-based	Mistral-7B	Text RAG	11.32	6.60	5.20	6.97	7.21	10.54	4.26	75.78	21.66	19.96
	Qwen2.5-7B	Text RAG	13.31	9.56	6.65	6.18	8.01	12.09	5.41	73.09	22.11	20.61
	LLaMA3.1-8B	Text RAG	16.53	10.93	7.80	9.84	**12.92**	13.26	7.96	47.98	18.70	16.36
	GPT-4o	Text RAG	15.04	12.95	6.39	8.05	10.33	13.34	5.97	**77.13**	24.07	22.46
	DeepSeek-v3	Text RAG	**17.76**	**14.70**	**9.83**	**9.99**	10.72	**16.80**	**7.97**	76.68	**25.94**	**23.84**
LVLM-based	LLaVA-Next-7B	Direct	6.54	1.53	7.38	2.12	4.38	4.17	5.07	**16.59**	7.15	5.73
		M3DocRAG	18.04	**10.47**	20.03	13.35	14.76	19.93	11.26	15.70	16.32	13.03
		MoLoRAG	18.12	9.71	19.68	**13.57**	**16.57**	19.80	**12.13**	16.14	16.73	12.94
		MoLoRAG+	**18.94**	9.85	**21.61**	13.22	15.94	**22.92**	10.50	13.90	**17.15**	**13.03**
	DeepSeek-VL-16B	Direct	8.86	6.57	13.39	5.23	13.63	9.86	9.36	3.14	8.40	6.01
		M3DocRAG	30.84	23.95	31.55	27.29	30.16	41.55	15.76	7.62	26.23	20.98
		MoLoRAG	31.67	**28.54**	30.80	26.42	**32.84**	43.65	16.63	**8.07**	27.47	21.81
		MoLoRAG+	**35.14**	27.86	**35.32**	**27.42**	30.74	**47.62**	**17.37**	4.93	**28.98**	**22.83**
	Qwen2.5-VL-3B	Direct	34.11	23.37	**33.75**	24.72	**29.56**	36.30	**23.42**	9.87	26.65	20.98
		M3DocRAG	31.88	22.70	27.87	23.85	22.22	37.90	15.40	20.63	26.77	21.90
		MoLoRAG	33.07	**29.18**	29.06	23.44	24.88	41.58	17.02	**21.52**	29.08	23.84
		MoLoRAG+	**35.26**	28.11	32.09	**25.03**	26.66	**43.68**	18.11	19.73	**30.03**	**24.49**
	Qwen2.5-VL-7B	Direct	37.14	25.52	31.95	28.00	27.26	40.21	**23.88**	30.94	32.77	27.36
		M3DocRAG	31.87	23.88	30.03	26.48	26.74	42.66	12.37	**40.81**	32.29	27.63
		MoLoRAG	33.80	**32.12**	29.77	29.23	30.34	46.61	15.37	37.22	34.35	29.67
		MoLoRAG+	**37.43**	31.34	**36.94**	**29.95**	**33.11**	**50.14**	18.25	34.08	**36.37**	**30.87**
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		35.08	30.13	29.61	27.47	24.72	45.26	14.86	29.15	31.73	27.45

Table 12: **Fine-grained performance analysis (Accuracy in %) across evidence modality and evidence locations on MMLongBench under the retrieved top-3 setting.** "UNA" refers to unanswerable questions.

Type	Model	Method	Modality					Location			Overall	
			Text	Table	Figure	Chart	Layout	Single	Multiple	UNA	Acc	EM
LLM-based	Mistral-7B	Text RAG	15.97	14.77	8.41	10.70	12.38	16.16	8.86	68.61	24.47	22.00
	Qwen2.5-7B	Text RAG	17.96	15.42	9.58	10.35	10.69	17.72	9.34	70.40	25.52	23.29
	LLaMA3.1-8B	Text RAG	20.40	20.02	10.44	15.42	13.82	18.74	**13.62**	45.29	22.56	19.22
	GPT-4o	Text RAG	19.68	19.14	10.10	13.58	12.25	20.20	10.52	**70.40**	27.23	24.31
	DeepSeek-v3	Text RAG	**25.37**	**22.23**	**13.34**	**19.60**	**17.27**	**24.85**	13.03	69.06	**29.82**	**26.62**
LVLM-based	LLaVA-Next-7B	Direct	6.54	1.53	7.38	2.12	4.38	4.17	5.07	16.59	7.15	5.73
		M3DocRAG	**8.74**	**6.59**	**11.72**	1.87	**8.54**	7.27	**8.55**	**17.49**	**10.10**	**8.23**
		MoLoRAG	6.84	5.55	10.72	2.15	7.66	7.82	6.43	16.14	9.37	7.49
		MoLoRAG+	7.49	2.49	11.24	**2.89**	8.08	**7.86**	6.25	16.59	9.41	7.30
	DeepSeek-VL-16B	Direct	8.86	6.57	13.39	5.23	13.63	9.86	9.36	3.14	8.40	6.01
		M3DocRAG	19.75	14.31	27.55	18.38	25.02	25.91	15.84	3.59	18.12	13.49
		MoLoRAG	21.57	18.79	29.00	17.55	24.54	29.60	18.15	2.69	20.43	16.27
		MoLoRAG+	**27.58**	**23.33**	**34.45**	**21.56**	**32.67**	**39.40**	**18.74**	**4.04**	**25.47**	**19.59**
	Qwen2.5-VL-3B	Direct	34.11	23.37	33.75	24.72	29.56	36.30	23.42	9.87	26.65	20.98
		M3DocRAG	35.11	25.58	32.23	24.04	28.06	39.62	20.62	18.83	29.11	23.66
		MoLoRAG	**38.94**	**30.48**	**36.05**	24.33	28.73	44.29	**23.52**	19.28	32.11	26.16
		MoLoRAG+	38.29	29.39	35.34	**26.48**	**33.53**	**45.12**	23.17	**19.28**	**32.47**	**26.52**
	Qwen2.5-VL-7B	Direct	37.14	25.52	31.95	28.00	27.26	40.21	23.88	30.94	32.77	27.36
		M3DocRAG	38.83	36.24	35.83	30.46	36.56	46.85	25.29	28.70	36.18	30.96
		MoLoRAG	41.67	37.89	37.56	**34.15**	32.44	50.07	26.12	34.98	39.28	33.18
		MoLoRAG+	**42.69**	**38.53**	**40.73**	33.26	**38.79**	**52.90**	**27.59**	**35.87**	**41.01**	**34.94**
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		43.14	38.72	37.90	32.55	31.17	53.45	23.82	28.25	38.53	33.27

Table 13: **Fine-grained performance analysis (Accuracy in %) across evidence modality and evidence locations on MMLongBench under the retrieved top-5 setting.** "UNA" refers to unanswerable questions.

Type	Model	Method	Modality					Location			Overall	
			Text	Table	Figure	Chart	Layout	Single	Multiple	UNA	Acc	EM
LLM-based	Mistral-7B	Text RAG	17.41	12.03	8.13	13.32	16.30	16.16	10.59	61.43	23.43	20.43
	Qwen2.5-7B	Text RAG	19.98	19.29	10.06	12.57	15.31	19.99	11.34	64.13	26.09	23.57
	LLaMA3.1-8B	Text RAG	24.61	22.71	12.21	18.42	**21.49**	22.60	**15.74**	41.26	24.25	21.07
	GPT-4o	Text RAG	22.38	24.50	12.30	15.42	16.17	23.37	13.54	**65.47**	28.74	25.51
	DeepSeek-v3	Text RAG	**27.54**	**28.33**	**15.53**	**21.39**	20.44	**28.90**	15.67	62.33	**31.23**	**27.54**
LVLM-based	LLaVA-Next-7B	Direct	6.54	1.53	7.38	2.12	4.38	4.17	5.07	16.59	7.15	5.73
		M3DocRAG	**8.59**	**6.23**	**12.16**	3.40	7.43	**8.65**	**7.92**	**17.49**	**10.43**	**7.95**
		MoLoRAG	7.33	5.52	11.36	3.12	8.34	7.95	7.12	16.14	9.56	7.67
		MoLoRAG+	6.46	2.26	10.33	**3.70**	**8.54**	8.02	4.79	17.49	9.19	7.30
	DeepSeek-VL-16B	Direct	8.86	6.57	13.39	5.23	13.63	9.86	9.36	3.14	8.40	6.01
		M3DocRAG	22.04	15.45	28.86	15.25	23.28	26.90	15.77	**4.93**	18.87	14.51
		MoLoRAG	22.38	18.64	28.07	13.83	23.68	29.60	16.52	4.04	20.07	15.80
		MoLoRAG+	**26.61**	**22.98**	**34.82**	**19.32**	**32.11**	**38.27**	**18.58**	4.04	**24.86**	**19.41**
	Qwen2.5-VL-3B	Direct	34.11	23.37	33.75	24.72	29.56	36.30	23.42	9.87	26.65	20.98
		M3DocRAG	35.79	26.04	32.06	24.15	29.33	39.08	22.16	14.35	28.38	22.46
		MoLoRAG	38.22	**31.23**	32.84	**27.85**	30.33	43.33	23.60	17.04	31.43	25.60
		MoLoRAG+	**38.38**	30.99	**36.04**	26.00	**33.94**	**44.82**	**23.67**	**18.83**	**32.41**	**26.52**
	Qwen2.5-VL-7B	Direct	37.14	25.52	31.95	28.00	27.26	40.21	23.88	30.94	32.77	27.36
		M3DocRAG	40.17	34.20	36.39	**35.34**	31.71	48.36	24.59	31.39	37.19	32.16
		MoLoRAG	**43.07**	38.10	38.92	35.22	35.08	**51.72**	**27.02**	33.63	39.97	34.20
		MoLoRAG+	41.57	**38.31**	**39.08**	31.64	**38.62**	51.16	26.96	**38.57**	**40.47**	**34.57**
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		41.92	42.00	34.02	33.45	29.77	49.97	25.25	32.29	38.34	32.99

Table 14: **Fine-grained performance analysis (Accuracy in %) across evidence modality and evidence locations on LongDocURL under the retrieved top-1 setting.**

Type	Model	Method	Modality				Location		Overall	
			Text	Table	Figure	Layout	Single	Multiple	Acc	EM
LLM-based	Mistral-7B	Text RAG	26.18	11.51	14.59	12.62	16.32	18.49	17.63	15.05
	Qwen2.5-7B	Text RAG	29.25	14.98	20.50	16.10	20.66	20.53	20.75	17.46
	LLaMA3.1-8B	Text RAG	30.63	16.26	22.18	16.52	21.32	23.49	22.57	18.11
	GPT-4o	Text RAG	32.16	16.35	23.21	17.44	21.91	23.40	22.84	18.45
	DeepSeek-V3	Text RAG	**33.28**	**18.17**	**25.47**	**19.84**	**24.00**	**24.34**	**24.32**	**20.09**
LVLM-based	LLaVA-Next-7B	Direct	16.79	5.28	12.12	7.39	7.87	13.43	10.78	9.29
		M3DocRAG	33.67	17.46	31.41	22.21	24.72	25.58	25.25	17.33
		MoLoRAG	34.61	18.32	32.37	24.13	24.99	**26.99**	26.11	17.89
		MoLoRAG+	**34.87**	**19.99**	**32.83**	**24.47**	**27.09**	26.80	**27.00**	**18.28**
	DeepSeek-VL-16B	Direct	19.98	8.26	13.81	13.65	11.18	17.87	14.72	11.35
		M3DocRAG	51.63	36.79	40.39	33.04	46.83	38.10	42.21	33.08
		MoLoRAG	54.64	39.97	41.52	34.64	48.77	**41.17**	44.75	35.18
		MoLoRAG+	**54.91**	**40.11**	**42.90**	**34.90**	**49.88**	41.16	**45.17**	**35.61**
	Qwen2.5-VL-3B	Direct	31.98	17.43	23.50	22.86	21.60	27.77	24.89	18.67
		M3DocRAG	49.08	32.91	38.70	31.24	42.77	37.17	39.82	32.22
		MoLoRAG	50.60	36.93	38.09	32.20	44.66	39.54	41.95	34.15
		MoLoRAG+	**52.06**	**37.56**	**40.40**	**32.97**	**46.83**	**39.91**	**43.17**	**34.80**
	Qwen2.5-VL-7B	Direct	32.37	19.88	27.09	23.25	24.20	28.15	26.38	19.74
		M3DocRAG	52.00	38.18	41.79	34.86	49.67	37.47	43.32	34.19
		MoLoRAG	55.91	42.77	44.06	36.58	52.88	41.46	46.89	37.25
		MoLoRAG+	**56.99**	**42.81**	**45.77**	**37.24**	**53.91**	**42.47**	**47.86**	**37.81**
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		54.52	40.41	43.20	31.25	48.88	40.26	44.42	36.30

Table 15: **Fine-grained performance analysis (Accuracy in %) across evidence modality and evidence locations on LongDocURL under the retrieved top-3 setting.**

Type	Model	Method	Modality				Location		Overall	
			Text	Table	Figure	Layout	Single	Multiple	Acc	EM
LLM-based	Mistral-7B	Text RAG	33.94	17.95	21.25	18.98	23.09	26.63	25.06	19.78
	Qwen2.5-7B	Text RAG	36.41	20.55	25.94	23.77	26.75	28.73	27.93	21.94
	LLaMA3.1-8B	Text RAG	37.22	22.99	29.64	22.53	29.42	30.08	29.80	22.75
	GPT-4o	Text RAG	40.91	26.80	33.14	26.57	33.19	32.13	32.74	25.20
	DeepSeek-V3	Text RAG	**41.89**	**30.84**	**35.49**	**28.15**	**35.77**	**33.67**	**34.73**	**26.84**
LVLM-based	LLaVA-Next-7B	Direct	16.79	5.28	12.12	7.39	7.87	13.43	10.78	9.29
		M3DocRAG	**20.64**	7.17	16.16	**10.75**	11.12	16.20	**13.85**	10.62
		MoLoRAG	20.52	6.45	15.64	10.58	10.94	15.68	13.49	**10.75**
		MoLoRAG+	19.94	**7.32**	**17.11**	10.64	**11.17**	**15.73**	13.58	10.49
	DeepSeek-VL-16B	Direct	19.98	8.26	13.81	13.65	11.18	17.87	14.72	11.35
		M3DocRAG	40.61	16.19	27.56	30.78	25.54	33.31	29.60	21.29
		MoLoRAG	40.62	17.57	28.69	28.86	27.07	32.67	29.98	21.81
		MoLoRAG+	**44.28**	**29.89**	**37.81**	**32.84**	**39.19**	**35.58**	**37.21**	**27.74**
	Qwen2.5-VL-3B	Direct	31.98	17.43	23.50	22.86	21.60	27.77	24.89	18.67
		M3DocRAG	54.07	37.97	42.07	36.97	46.39	42.64	44.4	34.97
		MoLoRAG	**55.99**	38.23	**42.95**	**37.01**	**48.09**	**43.76**	**45.79**	**36.17**
		MoLoRAG+	54.24	**39.03**	41.13	36.62	47.49	43.31	45.27	35.53
	Qwen2.5-VL-7B	Direct	32.37	19.88	27.09	23.25	24.20	28.15	26.38	19.74
		M3DocRAG	58.16	43.75	46.04	41.24	53.35	45.13	49.03	38.88
		MoLoRAG	**61.46**	45.66	**49.06**	**43.27**	**55.60**	**48.30**	51.71	**40.86**
		MoLoRAG+	61.43	**45.98**	49.01	42.56	55.01	49.01	**51.85**	40.13
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		56.81	42.25	44.07	35.48	49.46	44.51	46.91	37.63

Table 16: **Fine-grained performance analysis (Accuracy in %) across evidence modality and evidence locations on LongDocURL under the retrieved top-5 setting.**

Type	Model	Method	Modality				Location		Overall	
			Text	Table	Figure	Layout	Single	Multiple	Acc	EM
LLM-based	Mistral-7B	Text RAG	34.74	18.78	22.91	19.91	25.57	26.94	26.43	20.22
	Qwen2.5-7B	Text RAG	39.38	24.63	29.76	24.72	30.48	31.92	31.36	25.03
	LLaMA3.1-8B	Text RAG	40.04	26.02	32.74	26.01	32.56	33.85	33.27	25.42
	GPT-4o	Text RAG	44.20	31.54	40.20	29.32	37.86	36.00	36.98	28.13
	DeepSeek-V3	Text RAG	**45.71**	**34.39**	**41.58**	**31.26**	**40.09**	**38.08**	**39.04**	**29.38**
LVLM-based	LLaVA-Next-7B	Direct	16.79	5.28	12.12	7.39	7.87	13.43	10.78	9.29
		M3DocRAG	19.20	5.89	13.58	9.36	8.86	15.93	12.65	10.02
		MoLoRAG	19.39	5.94	13.51	10.13	9.18	15.79	12.72	10.19
		MoLoRAG+	**20.03**	**7.00**	**16.67**	**10.69**	**10.79**	**16.01**	**13.59**	**10.58**
	DeepSeek-VL-16B	Direct	19.98	8.26	13.81	13.65	11.18	17.87	14.72	11.35
		M3DocRAG	40.54	15.44	26.41	30.35	24.50	33.62	29.27	21.03
		MoLoRAG	42.08	16.97	28.58	31.09	26.40	34.76	30.76	22.19
		MoLoRAG+	**46.11**	**29.48**	**38.03**	**34.02**	**39.76**	**36.61**	**38.02**	**28.34**
	Qwen2.5-VL-3B	Direct	31.98	17.43	23.50	22.86	21.60	27.77	24.89	18.67
		M3DocRAG	54.76	37.06	39.66	38.47	45.49	43.95	44.67	34.84
		MoLoRAG	**55.29**	**39.68**	40.79	**38.88**	**47.07**	**45.25**	**46.05**	**35.74**
		MoLoRAG+	53.12	39.45	**40.95**	37.32	47.05	43.43	45.13	35.53
	Qwen2.5-VL-7B	Direct	32.37	19.88	27.09	23.25	24.20	28.15	26.38	19.74
		M3DocRAG	59.52	45.01	45.25	42.82	53.14	47.71	50.33	39.23
		MoLoRAG	**60.70**	**46.99**	47.95	44.74	**55.23**	49.65	**52.33**	**41.76**
		MoLoRAG+	60.54	46.61	**48.68**	**45.13**	54.86	**50.05**	52.33	40.65
Multi-agent	MDocAgent (LLaMA3.1-8B+Qwen2.5-VL-7B)		57.09	44.66	45.97	35.47	50.79	45.52	48.07	38.32