

AdaptiveDocRAG: Efficient Document Understanding with Multimodal Large Language Models

Surya Selvam

School of ECE, Purdue University
selvams@purdue.edu

Anand Raghunathan

School of ECE, Purdue University
raghunathan@purdue.edu

Abstract

Document understanding requires answering questions over long, multi-page documents that interleave text, figures, and tables. Recent methods adopt the Document Retrieval-Augmented Generation (DocRAG) paradigm, which indexes each document and feeds the top- K most relevant pages to a Multimodal Large Language Model (MLLM) per query. Because each page can be represented as extracted text or a rendered image, DocRAG systems expose two key knobs: an *indexing modality* choice when the documents are ingested and a *context modality* at query time. Existing systems either fix both knobs to the visual modality or use both modalities throughout, often incurring higher indexing and query costs than needed for a given document or query. We propose **AdaptiveDocRAG**, which adapts both knobs as well as the size of the visual context to optimize the trade-off between accuracy, query response time, and indexing time. **AdaptiveDocRAG** achieves this through three lightweight modules. First, *document-adaptive indexing modality selection* uses a text-density heuristic to route text-dominant documents to cheaper textual indexing. Second, *query-adaptive context modality selection* uses a learned per-query router to select textual or visual context. Third, *query-aware visual context compression* reuses the retriever’s per-patch query–page similarity to crop relevant sub-page regions and fuse them with top-ranked text chunks. On MMLongBench-Doc and NQ-Text with Qwen3-VL-8B, AdaptiveDocRAG matches or improves upon the accuracy of visual-only DocRAG (up to +1.1 points) while using $\sim 50\%$ of the MLLM tokens, resulting in $1.30\text{--}1.79\times$ end-to-end speedup and $1.34\times$ faster indexing.

1 Introduction

Document question answering (DocQA) supports enterprise search, literature review, and the analysis of legal and financial documents (Mathew et al.,

2021; Tito et al., 2023). Real-world documents are long and multimodal, interleaving text, figures, and tables in complex layouts that span hundreds of pages. Answering questions over them requires reasoning jointly across modalities and pages, a substantially harder challenge than single-page or text-only QA (Ma et al., 2024; Deng et al., 2025).

Recent Multimodal Large Language Models (MLLMs) are well suited to DocQA because they process and reason jointly over text and images (Anthropic, 2026; OpenAI, 2026; Google DeepMind, 2026). However, directly feeding entire documents to MLLMs does not scale in practice: context windows are limited, and both inference latency and token cost scale linearly with the number of pages. DocRAG (Cho et al., 2024; Yu et al., 2025) addresses this by indexing each document once with an embedding model and feeding only the top- K pages most relevant pages to a query to the MLLM, thereby reducing MLLM usage cost and latency.

A DocRAG system has three stages with distinct costs. *Indexing* runs once per document and embeds each page into a vector database; its cost scales with the number of pages and dominates time-to-first-response on a freshly ingested document. *Retrieval* runs once per query and scores each indexed page against the query to identify the top- K most relevant pages; while its cost depends on the database size, it is a very small part of query response latency in our workloads. *Question answering* runs once per query and feeds the top- K pages to the MLLM; its cost scales with K and dominates query response latency and token usage.

Each page in DocRAG is dual-modal: it can be represented as extracted text or as a rendered page image. This exposes two design knobs that together define the DocRAG modality design space. The *indexing modality* determines whether pages are embedded as text or as images, affecting both indexing latency and retrieval quality. The *context*

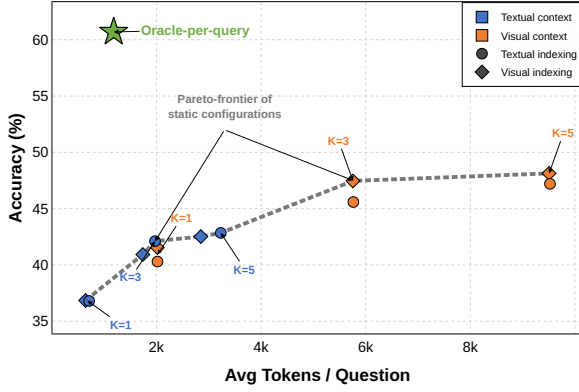


Figure 1: Accuracy vs. tokens-per-question on the Long-DocURL dataset across the 12 static DocRAG configurations ($2 \text{ indexing modalities} \times 2 \text{ context modalities} \times K \in \{1, 3, 5\}$). Marker shape denotes the indexing modality, and marker color denotes the context modality. The star denotes the per-query oracle.

modality determines whether the retrieved pages are passed to the MLLM as text or as images, affecting token count, latency, and answer accuracy.

Existing DocRAG systems fix these knobs statically and adopt one of two strategies. Visual-only systems (Faysse et al., 2025; Cho et al., 2024; Yu et al., 2025) use the visual modality for both indexing and context, treating page images as a superset of text. Dual-modality systems (Han et al., 2025; Jain et al., 2025; Suri et al., 2025; Wang et al., 2025) index pages in both modalities and consume both at query time. Both strategies incur a cost that is often unnecessary: page images produce at least $3\times$ more tokens than their text, increasing both encoder compute at indexing and MLLM tokens at query time—yet many documents and queries (e.g., named-entity lookups, factoids) are answerable from text alone. A per-query oracle that selects the best static configuration for each question lies well above both fixed strategies (Figure 1), revealing substantial room for adaptation along these two axes. A third axis remains unaddressed: even when visual context is needed, prior work feeds the entire page image, though query relevance is typically concentrated in a small region. Together, these observations expose three inefficiencies in DocRAG systems.

We propose **AdaptiveDocRAG**, a query- and document-adaptive DocRAG framework that addresses these inefficiencies through the following techniques:

- *Document-adaptive indexing modality selection*, which uses text density to select be-

tween textual or visual indexing at ingestion time, reducing indexing time and time-to-first-response for text-heavy documents.

- *Query-adaptive context modality selection*, which uses a learned decision tree to select textual or visual context for each query, reducing MLLM token cost and response time.
- *Query-aware visual context compression*, which crops retrieved pages to query-relevant regions and fuses them with top-ranked text chunks, reducing MLLM token usage while retaining relevant visual evidence.

On MMLongBench-Doc and NQ-Text with Qwen3-VL-8B, AdaptiveDocRAG matches or improves upon visual-only DocRAG accuracy by up to 1.1 points while using approximately half as many MLLM tokens, running $1.30\text{--}1.79\times$ faster end-to-end, and indexing documents $1.34\times$ faster.

2 Related Work

DocRAG pipelines. Document retrieval has evolved from OCR-based pipelines with dense or late-interaction text retrievers (Khattab and Zaharia, 2020; Chen et al., 2024a) to vision-based retrievers that embed rendered page images directly. ColPali (Faysse et al., 2025) introduced patch-level late interaction over a vision-language backbone, and ColQwen (ColQwen2) extends it to Qwen2-VL. Multimodal DocRAG systems pair these retrievers with MLLMs for question answering. M3DocRAG (Cho et al., 2024) and VisRAG (Yu et al., 2025) use the visual modality for both indexing and context, treating page images as a superset of OCR text. MDocAgent (Han et al., 2025) and VisDoM (Suri et al., 2025) use parallel text and image RAG pipelines and consume both modalities at query time, increasing MLLM token use regardless of query type. SimpleDoc (Jain et al., 2025) and ViDoRAG (Wang et al., 2025) use iterative retrieval, whereas MoLoRAG (Wu et al., 2025) uses page-graph traversal and MAB-DQA (Xiang et al., 2026) decomposes queries via a multi-armed bandit; all require multiple MLLM calls per query. Adaptive RAG methods instead decide whether to retrieve or how many passages to retrieve (Jeong et al., 2024; Asai et al., 2024; Taguchi et al., 2025), but operate on a single modality. AdaptiveDocRAG differs by selecting the indexing modality per document, the context modality per query, and the extent of visual context per query.

Indexing	Time/doc (s)	Accuracy (%)
Textual	6.8	45.6
Visual	18.9	47.5

Table 1: Cost–accuracy trade-off of the indexing modality on LongDocURL.

Context	Accuracy (%)	Tokens	Resp. (s)
Textual	40.9	1,737	0.18
Visual	47.5	5,755	0.57

Table 2: Cost–accuracy trade-off of the context modality on LongDocURL.

Visual context compression. General-purpose token-pruning methods operate within the MLLM. FastV (Chen et al., 2024b) discards low-attention vision tokens in early decoder layers, LLaVA-PruMerge (Shang et al., 2025) compresses tokens in the vision encoder using CLS attention, and SparseVLM (Zhang et al., 2025) uses cross-modal attention for text-conditioned selection. RegionRAG (Li et al., 2026), the most closely related method to our third module, fine-tunes a vision-language retriever with region-level supervision, groups high-scoring patches into semantic regions, and crops them for MLLM input. AdaptiveDocRAG instead reuses an off-the-shelf retriever’s per-patch query–page similarity without additional training, combines cropped regions with top-ranked text chunks, and operates outside the MLLM.

3 Motivation

DocRAG indexes each page as extracted text or a rendered image, retrieves the top- K pages relevant to a query, and supplies their content to an MLLM for question answering. The indexing and context modalities can each be textual or visual, yielding four choices in addition to the retrieval depth K . We characterize the cost–accuracy trade-offs of these choices to identify opportunities for improvement. To isolate modality effects, we use ColQwen2 for both modalities and Qwen3-VL-8B as the MLLM on LongDocURL.

Observation 1: Indexing modality presents a speed–accuracy trade-off. On LongDocURL, textual indexing is $2.8\times$ faster per document than visual indexing but sacrifices 1.9 percentage points in accuracy at $K = 3$ (Table 1). This trade-off suggests that textual indexing is suitable for text-heavy documents, whereas visual indexing remains

Visual context	Accuracy (%)	Tokens
Full page	47.5	5,755
Top 8 sub-pages	35.6	3,877
Top 4 sub-pages	27.8	1,997

Table 3: Effect of reducing visual context to query-relevant sub-pages on accuracy and MLLM input tokens.

important when layout and visual content provide retrieval evidence.

Observation 2: Context modality presents a token–accuracy trade-off. Visual context uses $3.3\times$ as many MLLM input tokens as textual context but improves accuracy by 6.6 percentage points at $K = 3$ (Table 2). This loss exceeds the 1.9-point loss from textual indexing, indicating that visual evidence matters more for question answering than for retrieval. Nevertheless, text-answerable queries can use textual context, whereas queries involving charts, tables, and diagrams benefit from page images. These results motivate document- and query-adaptive modality selection.

Observation 3: Fixed visual-context budgets present a token–accuracy trade-off. Reducing visual context from full pages to the top 8 and top 4 query-relevant sub-pages lowers MLLM input tokens by 33% and 65%, respectively, but decreases accuracy by 11.9 and 19.7 percentage points at $K = 3$ (Table 3). This aggregate loss is unevenly distributed across queries: among questions answered correctly with full-page visual context, 70% and 54% remain correct with only the top 8 and top 4 query-relevant sub-pages, respectively. These results expose a cost–accuracy trade-off under fixed sub-page budgets, motivating query-aware visual context compression that selects the amount of visual evidence supplied to the MLLM for each query.

Observation 4: No static configuration matches a per-query oracle. No static configuration addresses all three trade-offs. Figure 1 compares the 12 static configurations (2 indexing modalities $\times$ 2 context modalities $\times$ $K \in \{1, 3, 5\}$) with a per-query oracle. The oracle achieves 60.7% accuracy with 1,180 tokens, whereas the best static configuration (visual context, $K = 5$) reaches 48.1% with 9,507 tokens. This gap motivates a framework that is adaptive to documents and queries.

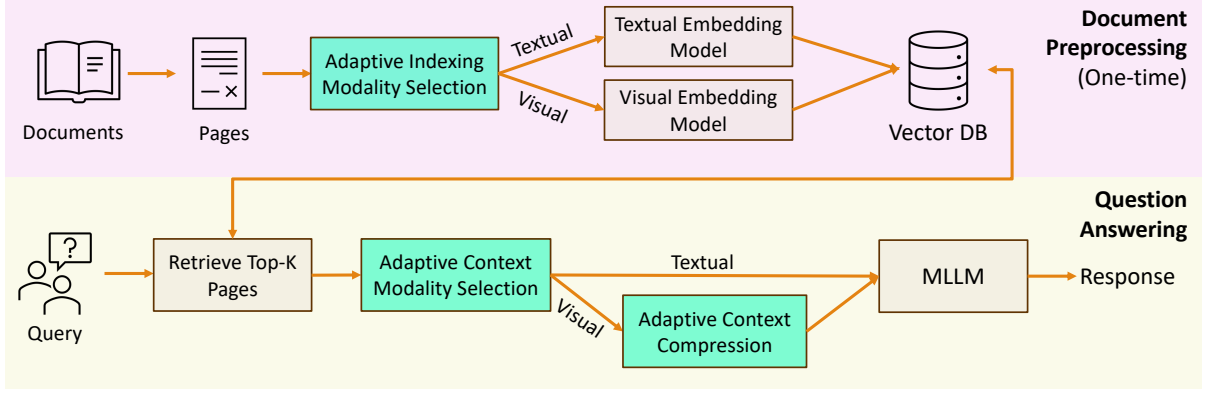


Figure 2: Overview of AdaptiveDocRAG with three adaptive modules (green boxes) that choose the indexing modality (per document), context modality (per query), and visual context compression.

4 AdaptiveDocRAG Methodology

AdaptiveDocRAG is a query- and document-adaptive DocRAG framework with three lightweight modules that address the trade-offs identified above (Figure 2). We first select an indexing modality for each document using text density. For each query, we then use a learned decision tree to select textual or visual context from features of the query and retrieved pages. When visual context is selected, we further compress it by retaining query-relevant sub-page regions and top-ranked text chunks. None of the modules requires fine-tuning of the retriever or MLLM, and each adds negligible computation relative to MLLM inference.

4.1 Document-Adaptive Indexing Modality Selection

We select an indexing modality once per document using its text density. We use text density as a proxy for the extent to which a document’s retrieval evidence is available in extracted text. For text-heavy documents, textual embeddings are likely to capture much of the retrievable content while avoiding the cost of image encoding. Conversely, documents with little extracted text often contain visual elements, such as tables, figures, and diagrams, whose information is better preserved by visual indexing. During preprocessing, we divide each document D into pages and extract text from each page using PDF text extraction when applicable and using OCR otherwise. We define the text density d_{TD} as the average character count per page in D . Given a threshold τ , we assign the indexing modality as:

$$\mathcal{M}(D) = \begin{cases} \text{TEXTUAL} & d_{TD} \geq \tau \\ \text{VISUAL} & d_{TD} < \tau \end{cases} \quad (1)$$

4.2 Query-Adaptive Context Modality Selection

For each query, we select extracted text when it is likely sufficient to answer the question and rendered page images when answering requires layout or other visual evidence. Because both representations are retained at ingestion, this selection is independent of the indexing modality. We use a lightweight decision tree to make this choice from features of the query and its top- K retrieved pages. The routing decision asks whether the retrieved evidence is sufficient in textual form or whether answering the query requires visual evidence. To make this prediction, we use three complementary features:

Query intent. Some queries can be answered by locating textual facts, whereas others require visual reasoning over charts, tables, or layouts. We construct two reference sets of example questions: one containing questions answerable from text alone, such as named-entity lookups, and another containing questions requiring visual reasoning, such as chart interpretation. For each query seen at run-time, we compute its cosine similarity to the two sets; these similarity scores characterize whether the query is more likely to require textual or visual evidence.

Retrieved-page content. For each retrieved page, we compute text density from its extracted text and image entropy from its rendered image. Text density quantifies the amount of textual evidence available in the page, whereas image entropy characterizes the complexity of its visual content. These measurements, together with the document’s selected indexing modality $\mathcal{M}(D)$, characterize

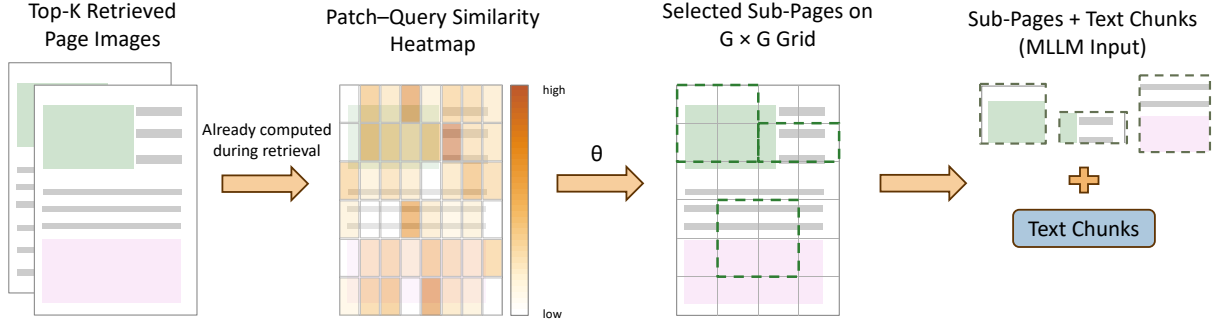


Figure 3: Query-Aware Visual Context Compression: thresholding normalized sub-page scores aggregated from patch-query similarities selects sub-pages, which are fused with top- T text chunks to form the MLLM input.

whether the retrieved evidence is better represented as text or images.

Query-page alignment. For each retrieved page, we compute the cosine similarity between the query and the page’s extracted text. This similarity indicates whether the page contains query-relevant textual evidence.

Let $\mathcal{P}_q = \{p_1, \dots, p_K\}$ denote the top- K retrieved pages for query q , and $\phi(q, \mathcal{P}_q)$ the feature vector defined above. We use the decision tree T to assign the context modality:

$$\mathcal{C}(q) = T(\phi(q, \mathcal{P}_q)) \in \{\text{TEXTUAL}, \text{VISUAL}\}. \quad (2)$$

Router training. We construct oracle modality labels on a held-out training set by running both textual and visual contexts for each query with the same retriever. We label each query with the higher-scoring modality, assigning ties to the lower-cost textual modality. We train the decision tree on these labels and the routing signals above. The decision tree provides interpretable routing decisions at negligible cost relative to MLLM inference.

4.3 Query-Aware Visual Context Compression

Visual context is token-intensive because each retrieved page is represented by many image tokens. However, a query often requires evidence from only a small part of a page, such as a chart, table, or figure. We therefore identify query-relevant sub-page regions and pair them with relevant text chunks, rather than passing all retrieved page images in full to the MLLM (Figure 3).

Patch-query similarity. To identify these regions, we divide each retrieved page into a $G \times G$

grid of sub-pages. We use the query-patch similarities produced by the late-interaction retriever to assign each sub-page c on page p a score $s(p, c)$ by summing its patch similarities.

Sub-page selection. To select the most relevant regions, we normalize sub-page scores jointly across all $K \times G^2$ candidates:

$$\hat{s}(p, c) = \frac{s(p, c) - s_{\min}}{s_{\max} - s_{\min}}, \quad (3)$$

where $s_{\min}$ and $s_{\max}$ are the minimum and maximum scores across the candidate sub-pages. We retain sub-page (p, c) when $\hat{s}(p, c) \geq \theta$ and pass its cropped image to the MLLM; applying the same threshold to every candidate makes the number of retained sub-pages query-adaptive rather than fixed. For a query about a localized element, only a few sub-pages score above θ and are retained, whereas queries requiring evidence across the page retain more sub-pages.

Text chunk fusion. After selecting the visual regions, we add complementary text because the crops may omit textual evidence needed to interpret the visual content, including evidence located elsewhere in the retrieved pages. We therefore rank fixed-length chunks from the retrieved pages by their similarity to the query and append the top- T chunks to the selected crops.

5 Experimental Results

5.1 Experimental Setup

Datasets. We evaluate AdaptiveDocRAG on a 2,204-question test set combining MMLongBenchDoc (Ma et al., 2024) (1,091 questions across 135 chart/table/figure-heavy documents) and NQ-Text from the UDA benchmark suite (Hui et al.,

Method	Indexing Time (s/doc)	$K = 1$		$K = 3$		$K = 5$	
		Accuracy (%)	Tokens	Accuracy (%)	Tokens	Accuracy (%)	Tokens
Textual RAG	2.10	46.2	792	53.1	2,256	55.7	3,656
Visual RAG	5.83	46.4	1,929	54.7	5,613	56.6	9,194
AdaptiveDocRAG ($\theta = 0.5$)	4.34	45.4	963	54.5	2,724	57.4	4,360
AdaptiveDocRAG ($\theta = 0.3$)	4.34	45.7	996	54.7	2,909	57.7	4,721
MDocAgent	7.93	53.1	7,169	58.4	17,663	61.4	27,688
SimpleDoc	124.9	60.0	13,820	63.8	15,561	65.4	17,152
Oracle	7.93	60.5	823	67.3	2,416	68.6	4,005

Table 4: AdaptiveDocRAG Compared with baselines on MMLongBench-Doc and NQ-Text with Qwen3-VL-8B.

2024) (1,113 questions across 274 text-heavy documents). We use LongDocURL (Deng et al., 2025) (2,325 questions across 396 documents) for the cost-accuracy analysis in Section 3 and for hyperparameter tuning.

Models. We use ColQwen2 (ColQwen2; Faysse et al., 2025) as the retriever for both textual and visual indexing, so comparisons isolate the effect of modality rather than retriever architecture. All experiments use Qwen3-VL-8B-Instruct (Qwen Team, 2025) as the MLLM and BGE-M3 (Chen et al., 2024a) for context-routing features and top- T text-chunk selection.

Implementation details. We render page images at 144 DPI and extract embedded text using either PyMuPDF (Artifex Software, 2024), or using Tesseract OCR (Smith, 2007) when embedded text is unavailable. The MLLM is served with vLLM (Kwon et al., 2023) on a single A100 80GB GPU. We use $\tau = 2,500$ chars/page for indexing modality routing, grid size $G = 2$, sub-page similarity threshold $\theta = 0.3$, and $T = 10$ text chunks for visual context compression.

Metrics. Accuracy is measured by ANLS on MMLongBench-Doc and by an LLM-as-judge score on NQ-Text, using Qwen3-VL-8B-Instruct as the judge and reporting scores on a percentage scale. Cost is measured by indexing time per document and MLLM tokens per question.

5.2 Main Results

We compare AdaptiveDocRAG with Textual RAG, Visual RAG, a per-query Oracle, MDocAgent (Han et al., 2025), and SimpleDoc (Jain et al., 2025). Textual RAG fixes both indexing and context to text, while Visual RAG fixes both to page images. Oracle selects, for each query and K , the best

of the four static (indexing, context) choices. For MDocAgent and SimpleDoc, we use ColQwen2 and Qwen3-VL-8B-Instruct to match AdaptiveDocRAG. Table 4 reports average accuracy and total MLLM tokens (context + response) per question at $K \in \{1, 3, 5\}$. Table 8 reports the effect of replacing the textual retriever at fixed $K = 5$. AdaptiveDocRAG matches Visual RAG’s accuracy at $K = 3$ using 52% of its tokens, and Pareto-dominates it at $K = 5$ (+1.1 points at 51% of its tokens). The more aggressive $\theta = 0.5$ configuration provides a lower-token operating point: at $K = 5$, it exceeds Visual RAG by 0.8 points while using 47% of its tokens. At $K = 1$, AdaptiveDocRAG trails Visual RAG by 0.7 points, where a single routing or compression error cannot be offset by other retrieved pages. AdaptiveDocRAG remains within $1.21\times$ of Oracle’s token cost at every K . At $K \in \{3, 5\}$, MDocAgent and SimpleDoc obtain higher accuracy with approximately $4\text{--}6\times$ more tokens.

At ingestion, AdaptiveDocRAG routes 38.9% of documents to textual indexing, yielding a $1.34\times$ indexing speedup over Visual RAG. Oracle and MDocAgent require both text and visual indices, raising indexing time to 7.93s/doc. SimpleDoc requires substantially more indexing time because it summarizes each page with an MLLM, raising indexing time to 124.9s/doc.

At query time, AdaptiveDocRAG selects textual context for 75.4% of queries. At $K = 3$, routing reduces MLLM tokens to 56% of Visual RAG before compression. Compression further reduces this cost to 52%, while the complete pipeline matches Visual RAG’s 54.7% accuracy. This compression path is available for visually indexed documents with visual context, for which patch-query similarity scores are available.

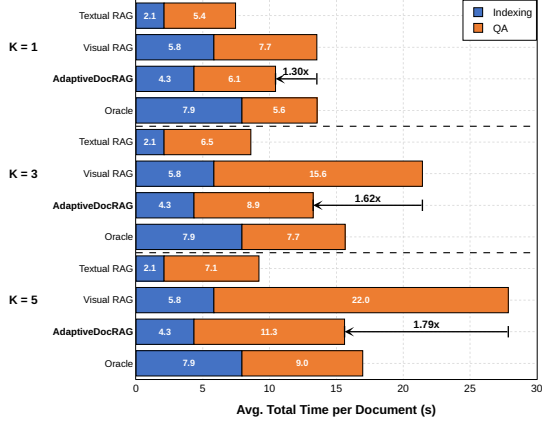


Figure 4: End-to-end wall time per document for Textual RAG, Visual RAG, Oracle, and AdaptiveDocRAG at $K \in \{1, 3, 5\}$.

Strategy	Accuracy (%)	Indexing Time (s/doc)
Textual Indexing	51.8	2.10
Visual Indexing	54.7	5.83
Adaptive Indexing	53.4	4.34

Table 5: Indexing modality ablation at $K = 3$ with context modality held at visual.

Figure 4 includes all adaptive-module overheads: routing features over the query and retrieved pages, tree inference, sub-page selection, and text-chunk ranking—totaling tens of milliseconds per query. AdaptiveDocRAG is $1.30\times / 1.62\times / 1.79\times$ faster than Visual RAG at $K \in \{1, 3, 5\}$. MDocAgent and SimpleDoc are omitted from the chart for legibility; at $K = 3$, they are $4.5\times$ and $17.0\times$ slower than AdaptiveDocRAG, respectively.

5.3 Ablation Studies

We ablate the three adaptive modules and evaluate sensitivity to the textual retriever.

For indexing, we hold context modality fixed to visual at $K = 3$ and vary only the indexing modality (Table 5). Adaptive indexing achieves 53.4% accuracy, within 1.3 points of visual indexing, while reducing indexing time from 5.83s/doc to 4.34s/doc. This reduction comes from routing 39% of documents to textual indexing; the remaining gap occurs when textual retrieval misses pages found by visual retrieval.

With adaptive indexing fixed, query-adaptive context routing provides the largest token reduction (Table 6). It improves accuracy from 53.4% to 55.2% relative to adaptive indexing with visual

Strategy	Accuracy (%)	Tokens
Adaptive Indexing + Textual Context	52.6	2,135
Adaptive Indexing + Visual Context	53.4	5,614
Adaptive Indexing + Adaptive Context	55.2	3,119
AdaptiveDocRAG	54.7	2,909
Oracle (per-query)	67.3	2,416

Table 6: Context modality and compression ablation at $K = 3$ on top of adaptive indexing.

Strategy	Accuracy (%)	Tokens
$\theta = 0, T = 0$ (Full Visual Page)	46.1	8,560
$\theta = 0.1, T = 10$	46.0	8,271
$\theta = 0.3, T = 0$ (sub-pages only)	40.7	5,724
$\theta = 0.3, T = 10$	46.1	6,552
$\theta = 0.5, T = 10$	44.6	4,651
$\theta = 0.7, T = 10$	39.8	3,006

Table 7: Visual context compression ablation at $K = 5$ on queries routed to visual context with visual indexing ($n = 419$).

context and reduces tokens from 5,614 to 3,119 per query. Compression further reduces tokens to 2,909 per query with a 0.5-point accuracy decrease, so the full pipeline remains 1.3 points above adaptive indexing with visual context while using 52% of its tokens.

We evaluate compression on the 419 queries routed to visual context from visually indexed documents at $K = 5$ (Table 7). At the default setting ($\theta = 0.3, T = 10$), compression matches full-page accuracy while using 77% of the uncompressed token budget. Sub-page selection without text fusion loses 5.4 accuracy points, whereas adding $T = 10$ text chunks recovers the full-page accuracy. Raising θ to 0.5 reduces tokens to 54% of the uncompressed budget, with a 1.5-point accuracy decrease.

The main comparison and module ablations use ColQwen2 for both textual and visual retrieval, which keeps the retriever architecture fixed when evaluating modality decisions. Table 8 tests whether AdaptiveDocRAG is sensitive to the textual retriever. We replace only the textual retriever with BM25 or BGE-M3, while keeping ColQwen2 for visual retrieval and using the same routing model, thresholds, and compression settings. BGE-M3 matches ColQwen2 within 0.1 accuracy points and slightly reduces both indexing time and token use. BM25 reduces Textual RAG indexing time from 2.10s/doc to 0.08s/doc, but the corresponding AdaptiveDocRAG variant loses 1.9 accuracy points, indicating that lower ingestion cost comes

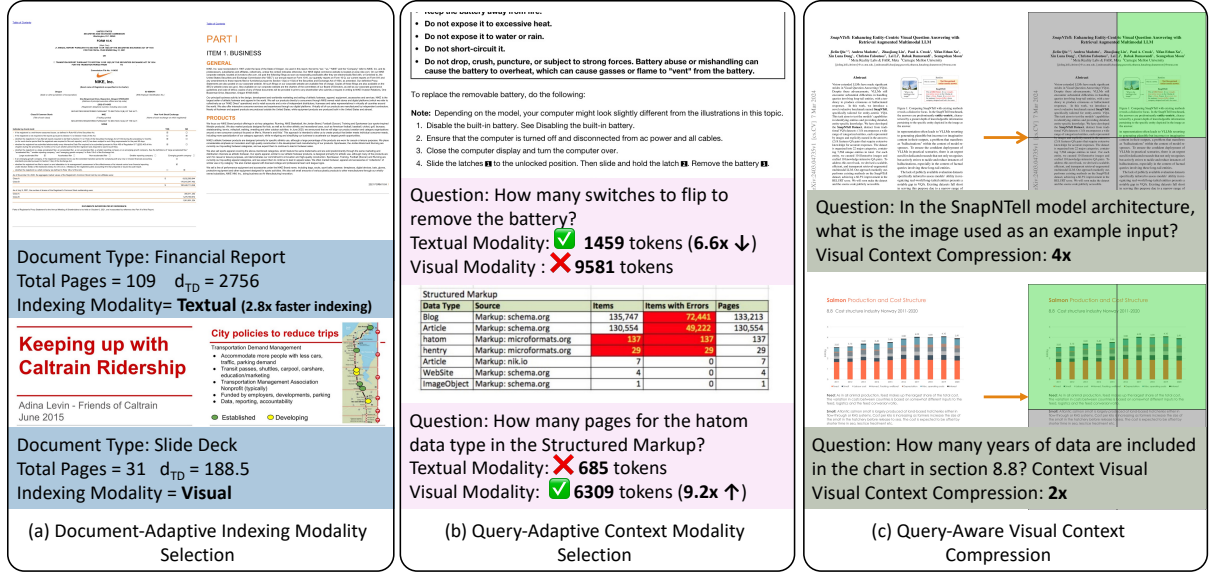


Figure 5: Qualitative examples of AdaptiveDocRAG’s three techniques: (a) document-adaptive indexing modality selection, (b) query-adaptive context modality selection, and (c) query-aware visual context compression.

Retrievers	Index (s/doc)	Acc. (%)	Tokens
<i>Textual RAG</i>			
BM25	0.08	52.7	3,580
BGE-M3	1.82	55.7	3,280
ColQwen2	2.10	55.7	3,656
<i>Visual RAG</i>			
ColQwen2	5.83	56.6	9,194
<i>AdaptiveDocRAG</i>			
BM25 + ColQwen2	3.59	55.8	4,621
BGE-M3 + ColQwen2	4.27	57.6	4,472
ColQwen2 + ColQwen2	4.34	57.7	4,721

Table 8: Textual retriever ablation at $K = 5$; AdaptiveDocRAG rows list textual + visual retrievers.

with weaker retrieval accuracy.

5.4 Qualitative Analysis

Figure 5 illustrates representative decisions from the three adaptive modules. Document-level text density routes a text-heavy SEC form 10-K to textual indexing, which is $2.8\times$ faster, and a sparse slide deck to visual indexing to preserve its layout (Fig. 5a). Query-level routing selects textual context for a procedural question, answering it correctly with $6.6\times$ fewer tokens, but selects visual context for a table lookup that requires row-column structure (Fig. 5b). For visual-context queries, sub-page selection reduces visual context by $4\times$ when relevant evidence is localized and by $2\times$ when it spans a larger portion of the page (Fig. 5c).

6 Conclusion

Static DocRAG pipelines incur avoidable costs by fixing the indexing modality, context modality, and extent of visual context independently of the document and query. We introduced AdaptiveDocRAG, which adapts these three design knobs with lightweight modules: a text-density rule selects textual or visual indexing, a learned decision tree selects textual or visual context for each query, and patch-query similarity selects query-relevant sub-page regions that are paired with top-ranked text chunks. In our evaluations, AdaptiveDocRAG matches or improves upon the accuracy of visual-only DocRAG while substantially speeding up indexing and query response.

Limitations

Our evaluation is limited to MMLongBench-Doc and NQ-Text, a single MLLM (Qwen3-VL-8B), and thresholds tuned on LongDocURL. The extent to which the routing and compression decisions transfer across models, languages, and document domains remains to be evaluated.

Ethics Statement

By selecting the indexing modality per document and the context modality per query, AdaptiveDocRAG reduces MLLM token use and end-to-end latency. These reductions may lower the compute and deployment costs of multimodal document QA. Our evaluation uses publicly available research

benchmarks: MMLongBench-Doc, NQ-Text, and LongDocURL. Deployment over sensitive corpora may create privacy risks, and users may over-rely on automated answers. Appropriate data governance and human oversight are especially important in high-stakes settings.

Acknowledgment: This work was supported in part by CoCoSys, one of seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, and in part by the National Science Foundation under Award 2318101. The authors acknowledge Murugan Sankaradass for initial discussions regarding challenges in DocRAG.

References

- Anthropic. 2026. [Introducing Claude Opus 4.7](#). Anthropic news post. Released April 16, 2026.
- Artifex Software. 2024. [PyMuPDF: Python bindings for MuPDF](#). Software.
- 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 (ICLR)*.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. [M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024b. [An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models](#). In *Computer Vision – ECCV 2024*, pages 19–35. Springer.
- Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3DocRAG: Multi-modal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*.
- ColQwen2. 2024. ColQwen2. <https://huggingface.co/vidore/colqwen2-v1.0>.
- Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhongzhi 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 (ACL)*, pages 1135–1159. Association for Computational Linguistics.
- Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025. [ColPali: Efficient document retrieval with vision language models](#). In *The Thirteenth International Conference on Learning Representations (ICLR)*.
- Google DeepMind. 2026. [Gemini 3.1 Pro: A smarter model for your most complex tasks](#). The Keyword (Google blog). Released February 19, 2026.
- 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*.
- Yulong Hui, Yao Lu, and Huanchen Zhang. 2024. [UDA: A benchmark suite for retrieval augmented generation in real-world document analysis](#). In *Advances in Neural Information Processing Systems 37 (NeurIPS), Datasets and Benchmarks Track*.
- Chelsi Jain, Yiran Wu, Yifan Zeng, Jiale Liu, Shengyu Dai, Zhenwen Shao, Qingyun Wu, and Huazheng Wang. 2025. [SimpleDoc: Multi-modal document understanding with dual-cue page retrieval and iterative refinement](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 28410–28427. Association for Computational Linguistics.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. [Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 7036–7050. Association for Computational Linguistics.
- Omar Khattab and Matei Zaharia. 2020. [ColBERT: Efficient and effective passage search via contextualized late interaction over BERT](#). In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 39–48.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pages 611–626. ACM.
- Yinglu Li, Zhiying Lu, Zhihang Liu, Yiwei Sun, Chuanbin Liu, and Hongtao Xie. 2026. [RegionRAG: Region-level retrieval-augmented generation for visual document understanding](#). In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pages 6662–6670.

- 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 37 (NeurIPS), Datasets and Benchmarks Track*.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. [DocVQA: A dataset for VQA on document images](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.
- OpenAI. 2026. [Introducing GPT-5.5](#). OpenAI announcement. Released April 23, 2026.
- Qwen Team. 2025. [Qwen3-VL technical report](#). Preprint, arXiv:2511.21631.
- Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. [LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22857–22867.
- Ray Smith. 2007. [An overview of the Tesseract OCR engine](#). In *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, volume 2, pages 629–633. IEEE.
- Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A. Rossi, and Dinesh Manocha. 2025. [VisDoM: Multi-document QA with visually rich elements using multimodal retrieval-augmented generation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 6088–6109. Association for Computational Linguistics.
- Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. 2025. [Efficient context selection for long-context QA: No tuning, no iteration, just adaptive-k](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 20105–20130. Association for Computational Linguistics.
- Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. [Hierarchical multimodal transformers for multipage DocVQA](#). *Pattern Recognition*, 144:109834.
- 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 (EMNLP)*, pages 9113–9134, Suzhou, China. Association for Computational Linguistics.
- Xixi Wu, Yanchao Tan, Nan Hou, Ruiyang Zhang, and Hong Cheng. 2025. [MoLoRAG: Bootstrapping document understanding via multi-modal logic-aware retrieval](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14024–14045. Association for Computational Linguistics.
- Yixin Xiang, Yunshan Ma, Xiaoyu Du, Yibing Chen, Yanxin Zhang, and Jinhui Tang. 2026. [MAB-DQA: Addressing query aspect importance in document question answering with multi-armed bandits](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22973–22992, San Diego, California, United States. Association for Computational Linguistics.
- Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. [VisRAG: Vision-based retrieval-augmented generation on multi-modality documents](#). In *The Thirteenth International Conference on Learning Representations (ICLR)*.
- Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025. [Sparse-VLM: Visual token sparsification for efficient vision-language model inference](#). In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 74840–74857. PMLR.