LOUD AND CLEAR: MAKING VISUAL EVIDENCE COUNT IN MULTIMODAL DOCUMENT RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal document retrieval-augmented generation (RAG) has focused largely on retrieval, yet its answers depend on how well a vision-language model reads the retrieved pages. We find that evidence in retrieved pages remains underused even when the correct pages are retrieved. On SciEGQA oracle pages, five prior methods give the generator 0.7K to 2.7K visual tokens per query on average under a common image processor, yet raising the page-only budget from 1K to 32K improves accuracy by 15.22 points. At a fixed 16K budget, adding ground-truth crops improves accuracy by 3.75 points over page-only input. We study what the generator needs from retrieved pages through two requirements: **Loud**, making full use of visual token capacity, and **Clear**, refining the page so that its evidence becomes prominent. For Loud, increasing either the visual token budget or visual fidelity, the image resolution those tokens preserve, improves accuracy with the other held fixed, and full-page scaling outperforms repetition and tiling at matched budgets. For Clear, suppressing surrounding content improves accuracy without enlarging selected regions, and adding OCR of these regions to page images outperforms either input alone. We implement both as **Loud and Clear (LoC)**, which combines higher-resolution pages with OCR from query-relevant regions without ground-truth annotations or additional training. LoC outperforms eight prior methods on SciEGQA, ViDoRe-v3, and MMLongBench-Doc, improving over the strongest, ColQwen3, by 8.88, 3.09, and 4.22 points with the same retriever and generator. It also outperforms its page-only variant under the same visual token budget. These results locate a bottleneck beyond retrieval: how the retrieved pages are given to the generator.

1 INTRODUCTION

Multimodal document RAG extends retrieval-augmented generation (RAG) (Lewis et al., 2020) to visually rich documents (Yu et al., 2025; Tanaka et al., 2025), retrieving relevant pages and passing them to a vision-language model (VLM) as evidence conveyed through text, layout, tables, charts, figures, and typography (Mathew et al., 2021; 2022). Visual retrievers such as ColPali (Faysse et al., 2025) operate directly on page images, preserving layout and other visual signals without intermediate text extraction. A growing line of end-to-end systems has extended visual retrieval and generation to long, multi-page, and multi-document settings (Cho et al., 2025; Chen et al., 2025; Suri et al., 2025; Wang et al., 2025). Much of this work has focused on improving retrieval. How retrieved pages are then given to the generator remains comparatively underexplored, and many systems pass them at a default resolution without refining their content.

However, even when retrieval succeeds, the generator underuses the evidence in the retrieved pages. On SciEGQA (Yu et al., 2026b) oracle pages, five prior methods give the generator 0.7K to 2.7K visual tokens per query on average when adapted to a common image processor. Yet with the same pages and generator, scaling the page-only budget from 1K to 32K improves accuracy by 15.22 percentage points. How the page content is refined also matters: at a fixed 16K budget, pairing pages with ground-truth crops adds 3.75 points over pages alone. These gains with correct pages and a fixed generator reveal a bottleneck in how retrieved pages are given to the generator.

The systems that do change how retrieved pages are given to the generator leave open which changes help it use the same retrieved pages. They aggregate pages (Yu et al., 2025), fuse textual and visual

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

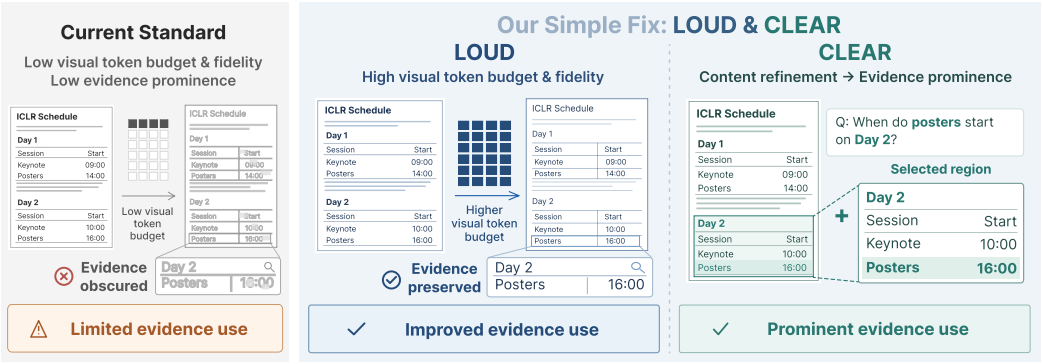


Figure 1: **Loud and Clear let the generator use evidence that a correctly retrieved page already contains.** For the Day 2 poster question, the current standard's low visual token budget obscures relevant rows. Loud preserves them with a larger budget and higher visual fidelity, while Clear makes the selected region prominent alongside the page. The retrieved page is unchanged throughout.

evidence (Suri et al., 2025), or replace full pages with retrieved regions (Li et al., 2026), each within its own pipeline with its own retriever and generator. The closest system, RegionRAG (Li et al., 2026), replaces pages with regions, whereas our controlled comparisons show that keeping the page in the input alongside its regions is more reliable across benchmarks.

To address this gap, we organize our diagnosis into two generator input requirements, Loud and Clear, and implement them as the **Loud and Clear (LoC)** framework (Figure 1). **Loud** fully exploits the generator's *visual token capacity*, its ability to read pages encoded as visual tokens. It requires a sufficient visual token budget and high *visual fidelity*, how much image resolution survives resizing into visual tokens. **Clear** applies *content refinement*, such as whiteout, crop, or OCR, to make each page's evidence prominent. Holding the retrieved pages and generator fixed, we compare alternative implementations of each requirement on SciEGQA (Yu et al., 2026b) and ViDoRe-v3 (Loison et al., 2026), using ground-truth region localization only for oracle diagnosis, not for LoC.

Our controlled comparisons select full-page scaling for Loud and page images with OCR of the selected regions for Clear. Full-page scaling raises both the visual token budget and visual fidelity, and each improves accuracy with the other held fixed. Spending the same budget on repeated pages, repeated blank pages, or spatial tiling does not match full-page scaling, so token count alone does not explain the gain. For Clear, adding OCR of the selected regions to page images outperforms either input alone under the same visual token budget. Whiting out content outside the selected regions also improves accuracy, whereas extracting or regrouping the regions gives no consistent advantage, which is consistent with keeping the retrieved page in the generator's input.

LoC combines full-page scaling with OCR from regions scored using the retriever's token embeddings, without ground-truth annotations or additional training. LoC outperforms eight prior methods, improving over the ColQwen3 (Huang & Tan, 2025) baseline with the same retriever and generator by 8.88, 3.09, and 4.22 points on SciEGQA, ViDoRe-v3, and MMLongBench-Doc (Ma et al., 2024b), respectively. LoC also surpasses its page-only variant under the same visual token budget and improves accuracy with two additional generators. Because ColQwen3 and both LoC variants share the retriever and generator, these gains come from how pages are given to the generator.

In summary, our main contributions are:

- **We identify the generator's visual input as an overlooked bottleneck in multimodal document RAG.** Auditing five prior methods on SciEGQA oracle pages under a common image processor, we find that they give the generator few visual tokens, whereas increasing the visual token budget and refining page content improve accuracy with the pages and generator held fixed.
- **We define Loud and Clear as two requirements for using evidence in retrieved pages.** Controlled comparisons show that visual token budget and visual fidelity independently improve accuracy, and identify full-page scaling and adding OCR of the selected regions to the page as consistent designs across benchmarks.
- **We propose LoC, a training-free framework that outperforms eight prior methods on three benchmarks.** Without ground-truth annotations, LoC outperforms its page-only variant under the same visual token budget and improves over the ColQwen3 baseline across three generators.

2 Related Work

2.1 Visual retrieval for multimodal document RAG

Recent multimodal document retrieval work has increasingly explored image-native models that preserve visual and structural information without relying on OCR-based content extraction. DSE (Ma et al., 2024a) encodes each document screenshot into a single dense vector, whereas ColPali (Faysse et al., 2025) uses patch-level embeddings with ColBERT's (Khattab & Zaharia, 2020) late-interaction scoring while keeping the page as the retrieval unit.

Retrieval and its evaluation have expanded beyond page-level relevance toward finer-grained evidence and document structure. LILaC (Yun et al., 2025) and LAD-RAG (Sourati et al., 2026) use document graphs for query-relevant evidence selection. On the evaluation side, ViDoRe (Faysse et al., 2025) evaluates page retrieval across domains and languages, while MMDocIR (Dong et al., 2025) adds layout-level evidence labels in long documents. ViDoRe-v3 (Loison et al., 2026) broadens the scope to answer generation and visual grounding.

Multimodal document RAG has advanced largely through retrieval, improving evidence selection and evaluation. Yet better evidence selection does not by itself establish whether the generator can use the evidence in a retrieved page, and how those pages are given to the generator remains comparatively underexplored. This gap motivates our study of the accuracy lost through the generator's input even when the correct pages are retrieved.

2.2 Visual token capacity in vision-language models

Modern VLMs use dynamic resolution and image tiling, so the number of visual tokens and their visual fidelity vary with the input (Wang et al., 2024b; Liu et al., 2024; Chen et al., 2024b). In multimodal document RAG, the visual fidelity of retrieved pages depends on how they are resized and encoded as visual tokens. Because processing more visual tokens increases inference cost, many methods improve the accuracy-cost trade-off by reducing or reallocating tokens.

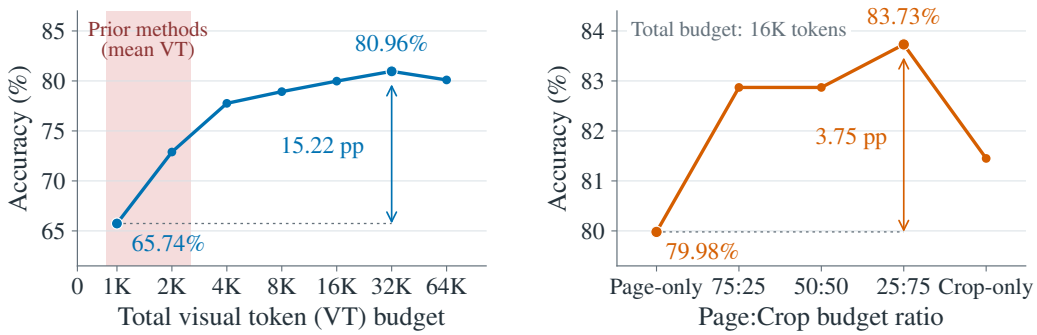
For token pruning, FastV (Chen et al., 2024a) and SparseVLM (Zhang et al., 2025) use attention-based pruning, with SparseVLM incorporating text guidance. VisionZip (Yang et al., 2025) and VisionTrim (Yu et al., 2026a) combine token selection and merging. Recent analysis shows that the layer depth at which visual tokens remain informative varies with the task's visual demands and model capacity (Wang et al., 2026). For budget allocation, HiRED (Arif et al., 2025) uses attention to allocate a fixed token budget across a full-image view and high-resolution tiles obtained by partitioning the same input image into a regular spatial grid. PromPrune (Lee et al., 2026) adapts the balance between locally salient and globally diverse tokens to each image. At the layer level, Optimal Vision Token Reduction (Zhao et al., 2025) searches for layer-wise retention schedules to reduce computation or improve accuracy under a prescribed budget.

These studies establish that visual token selection and allocation affect the accuracy and efficiency of VLM inference. They focus on reducing computation or allocating a prescribed visual token budget. This focus leaves open how much accuracy a low visual token budget sacrifices in multimodal document RAG. With the retrieved pages and generator held fixed, we vary the visual token budget and visual fidelity separately and compare alternative ways of spending a matched budget, such as spatial tiling and multi-scale views (Section 3.2). Our Loud study asks how to use additional visual token capacity to preserve evidence in retrieved pages and how much accuracy each use recovers.

2.3 Content refinement in multimodal document RAG

On the generator side, many multimodal document RAG systems pass retrieved page images directly to a VLM without refining their content. M3DocRAG (Cho et al., 2025), SV-RAG (Chen et al., 2025), and VDocRAG (Tanaka et al., 2025) follow this page-based design. VisRAG (Yu et al., 2025) compares page concatenation, weighted selection over per-page answers, and native multi-image input. ViDoRAG (Wang et al., 2025) selects pages from coarse thumbnails, inspects the selected pages at high resolution, and uses a reference page to verify the final answer.

Other systems refine evidence before or during answer generation. VisDoMRAG (Suri et al., 2025) curates parallel textual and visual evidence and combines their reasoning outputs through modality



(a) Loud: Increasing visual token budget **(b) Clear: Content refinement**

Figure 2: **Even with correct pages, larger visual token budgets and content refinement raise accuracy.** We diagnose this on SciEGQA with oracle pages and fixed generation and evaluation settings. **(a) Loud:** The red band spans five prior methods' mean usage under the Qwen3.6 image processor (0.72K–2.68K visual tokens per query). Increasing the page-only budget from 1K to 32K adds 15.22 pp. The x-axis is piecewise linear. **(b) Clear:** At a fixed 16K visual token budget, the best observed mixture of pages and ground-truth-matched crops adds 3.75 pp over page-only input.

fusion. RegionRAG (Li et al., 2026) replaces full-page inputs to the generator with retrieved query-relevant semantic regions. MDocRAG-RL (Wang, 2026) lets the generator crop selected regions from retrieved pages and add the re-encoded regions as new visual observations. Related work in text RAG also processes retrieved context before generation, summarizing retrieved documents (Xu et al., 2024) or removing prompt tokens (Jiang et al., 2024).

These multimodal systems show that generation outcomes can depend on how retrieved pages are given to the generator. However, gains within these different pipelines do not by themselves establish which content refinement helps the generator use the same retrieved pages under a fixed visual token budget. We therefore compare content refinements of the same retrieved pages with the generator and visual token budget held fixed (Section 3.3), and use the findings to develop our Clear design.

3 Loud and Clear: Improving Evidence Use in Retrieved Pages

This section asks what the generator needs from retrieved pages, holding the pages and the generator fixed. In Sections 3.1 to 3.3, the retrieved pages of each query are its oracle pages, the ground-truth pages that contain the evidence. Section 3.1 diagnoses how evidence is underused, Sections 3.2 and 3.3 compare implementations of Loud and Clear, and Section 3.4 combines them into LoC.

3.1 Evidence in retrieved pages remains underused

To test how well the generator uses evidence in retrieved pages, we run a controlled diagnosis on SciEGQA (Yu et al., 2026b), which provides ground-truth evidence boxes. We vary the visual token budget and content refinement while fixing the retrieved pages, generator, prompt, decoding, and evaluation. We adjust the spatial resolution of the input views to target a visual token budget across all images per query, as detailed in Appendix D.

Prior document RAG methods leave much of the generator's visual token capacity unused. Specifically, five prior methods give the generator 0.72K–2.68K visual tokens per query on average on the same retrieved pages when adapted to the Qwen3.6 image processor, as detailed in Appendix A. However, Figure 2(a) shows that increasing the visual token budget from 1K to 32K with page-only inputs improves accuracy from 65.74% to 80.96%, a gain of 15.22 points. Because the retrieved pages and generator are unchanged, this gain shows that a low visual token budget can leave available evidence underused. These results motivate **Loud**, making full use of visual token capacity.

Moreover, content refinement can further improve generation accuracy at a fixed visual token budget. At 16K, the best observed mixture, assigning 25% of the budget to full-page views and 75% to ground-truth-matched crops (ground-truth crops for short), achieves 83.73% accuracy compared with 79.98% for page-only input, a gain of 3.75 percentage points (Figure 2(b)). This gain motivates **Clear**, refining the page content so that the evidence it already contains becomes prominent.

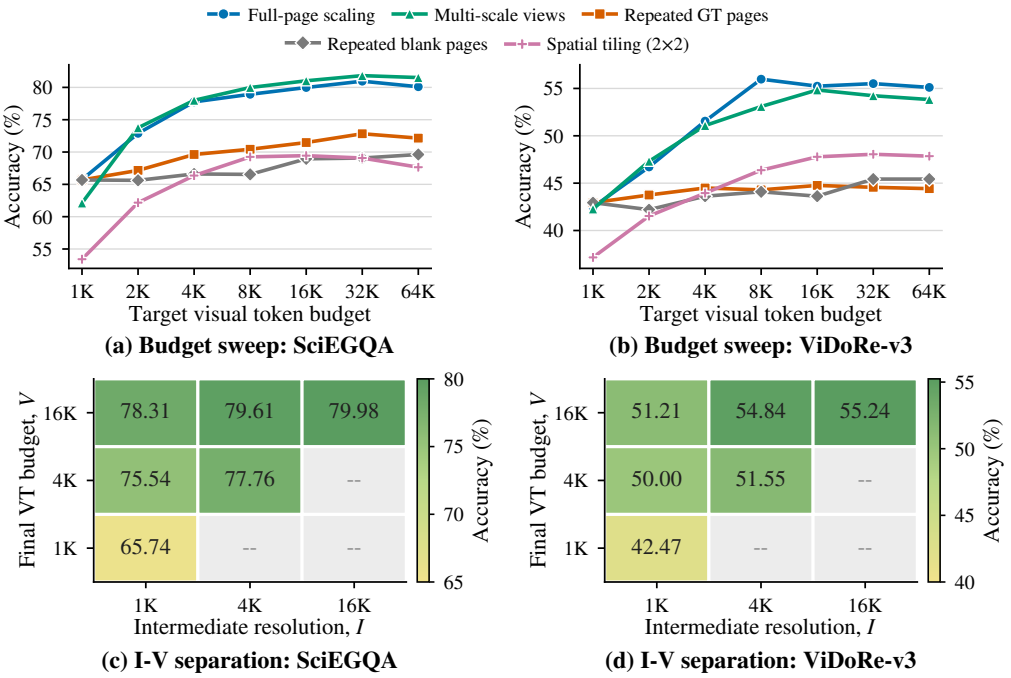


Figure 3: **How the extra tokens are spent matters, and full-page scaling spends them most consistently.** Panels (a) and (b) show that full-page scaling outperforms repetition and spatial tiling at 2K–64K on both benchmarks, while the advantage of multi-scale views varies across benchmarks. Panels (c) and (d) show that visual fidelity, controlled by intermediate resolution I, and the final visual token budget V each improve accuracy with the other held fixed under two-stage resizing. Budgets sum visual tokens across all input images, and 1K denotes 1,000 tokens. Gray cells with $I > V$ are untested. Color scales are independently normalized by benchmark.

3.2 LOUD: MAKING FULL USE OF VISUAL TOKEN CAPACITY

To determine how to make full use of visual token capacity, we compare five ways of increasing the visual token budget (Appendix B). On SciEGQA (1,623 queries) and five public English-language datasets from ViDoRe-v3 (1,488 queries), we hold the retrieved pages and generation and evaluation settings fixed. Each budget is a query-level target for total visual tokens produced by the generator's image processor across all input images, excluding the question and instruction text.

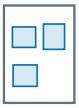
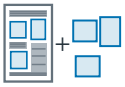
The first two test whether a larger budget helps without raising visual fidelity. **Repeated blank pages** add blank images while leaving the page views unchanged. **Repeated pages** instead allocate additional tokens to repeated views of the same pages. The remaining conditions test how the resolution and organization of page views affect the benefit of additional tokens. **Full-page resolution scaling** (full-page scaling for short) increases the resolution of each page. **Multi-scale views** add full-page views at multiple resolutions. **Spatial tiling** uses four uniform 2×2 tiles per page as separate input views, without retaining a full-page view.

How the additional tokens are spent matters (Figure 3(a) and (b)). Full-page resolution scaling outperforms repeated pages and repeated blank pages at 2K–64K, and spatial tiling at every budget on both benchmarks. Multi-scale views outperform full-page scaling on SciEGQA at 2K–64K, whereas on ViDoRe-v3 they lead only at 2K and full-page scaling performs better at 4K–64K.

Because full-page resolution scaling raises both the visual fidelity of the input and the final visual token budget, we separate the two with a two-stage resizing procedure (Appendix B.3). We first resize each page to an intermediate resolution I, expressed in token-equivalent units, which controls visual fidelity before final resizing. We then resize this intermediate image to a target final visual token budget V. Holding I fixed while increasing V enlarges the same intermediate image without raising visual fidelity or recovering information lost in the first resize.

Visual fidelity and the final visual token budget each improve accuracy when the other is held fixed (Figure 3(c) and (d)). With V fixed at 4K or 16K, accuracy increases monotonically as the inter-

Table 1: **At a fixed budget, suppressing surrounding content and adding selected-region OCR to page images both beat page-only input.** Accuracy (%) is measured using the same retrieved pages and generator. Page + Crop allocates 4K VT to pages and 12K to crops. Blue marks selected regions and gray marks surrounding content. Bold marks the best per dataset and refinement group.

	Visual refinements					*OCR-based refinements*	
Refinement	**Page**	**Whiteout Page**	**Crop**	**Canvas**	**Page + Crop**	**Crop OCR**	**Page + Crop OCR**
Input view							
Budget	16K VT	16K VT	16K VT	16K VT	16K VT	OCR $\leq$ 16K	OCR+VT $\leq$ 16K
SciEGQA	79.98	82.01	81.45	81.15	**83.73**	53.97	**82.87**
ViDoRe-v3	55.24	**56.05**	51.55	55.65	52.62	53.16	**56.12**

mediate resolution I increases on both benchmarks. Likewise, with I fixed at 1K or 4K, accuracy increases monotonically as the final visual token budget V increases, although the same intermediate image is only enlarged. Thus, higher visual fidelity alone does not explain the gains: a larger visual token budget can improve the use of evidence already preserved in the intermediate image.

We adopt full-page resolution scaling for Loud. It increases both visual fidelity and the visual token budget while outperforming repetition and spatial tiling. Because the advantage of multi-scale views varies across benchmarks, we use a single scaled view per page as a simple common choice.

Finding 1. Loud

Allocating more visual tokens can improve the use of the same evidence even at fixed visual fidelity, and at a fixed budget, higher visual fidelity improves it further. The benefit of additional tokens depends on how they are spent.

3.3 CLEAR: MAKING EVIDENCE PROMINENT

Loud addresses visual token capacity, but leaves open how to make answer-relevant content prominent within the retrieved pages. We compare content refinements that suppress surrounding content with those that present the selected regions separately, either as images or as text. Using the SciEGQA and ViDoRe-v3 English cohorts and the fixed settings of Section 3.2, we select regions by matching MinerU (Wang et al., 2024a) outputs to ground-truth evidence boxes. Table 1 compares content refinements at a 16K visual token budget (Appendix C provides construction details, additional comparisons, and uncertainty estimates). For conditions with OCR, the visual token budget includes OCR text tokens and visual tokens. In tables, VT denotes visual tokens from images.

Suppressing surrounding content offers a direct way to make the selected regions prominent within the page. Whiteout Page (Whiteout for short) replaces pixels outside the selected regions with white, preserving the selected pixels at their original positions, the page dimensions, and visual token budget. Whiteout improves accuracy over Page by 2.03 percentage points on SciEGQA and 0.81 on ViDoRe-v3 without enlarging the selected regions. The observed gains are larger when the selected regions occupy a smaller fraction of the pages (Table 10). Figure 4(a) illustrates Whiteout correcting an error where Page lists terms from the other table rather than the target table. Whiteout removes the other table and yields the correct terms from the target table.

Extracting or regrouping the selected regions offers no consistent advantage over keeping them in their original positions. We present the regions as separate images (Crop), group them into a canvas per page (Canvas), or pair them with full-page views (Page + Crop). Crop underperforms Whiteout on both benchmarks, although Canvas closes much of the gap on ViDoRe-v3. Page + Crop is the best visual refinement on SciEGQA but falls below Page alone on ViDoRe-v3. Inspected cases show that the original page keeps heading associations and captions that refined inputs can lose (Figure 6).

Adding OCR of the selected regions to page images outperforms either input alone. Crops present the selected regions as images, leaving the generator to read the text within them. We therefore ex-

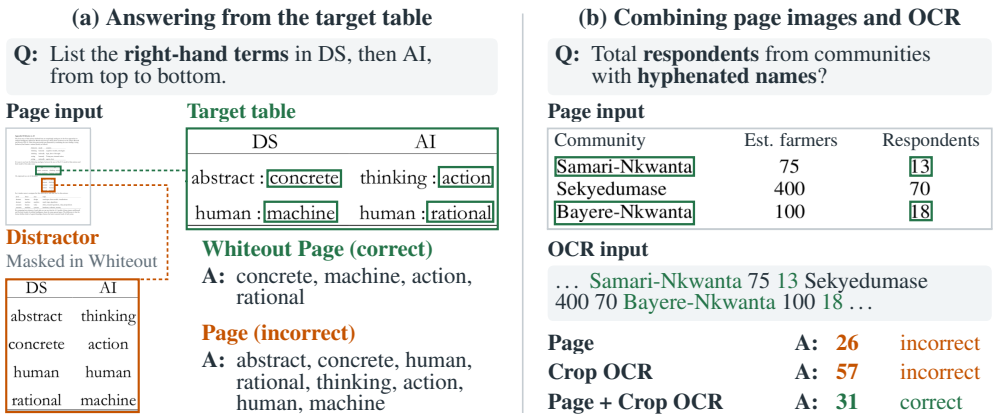


Figure 4: **Content refinement corrects answers whose evidence is already in the retrieved pages.** **(a)** Whiteout Page masks the distractor table and corrects the answer without enlarging selected regions or changing VT. **(b)** Only Page + Crop OCR gives the correct total, 31 (13 + 18). For readability, table excerpts are enlarged and annotated, and OCR text is rewrapped without reordering.

amine whether making this text explicit through OCR helps the generator use the selected regions. Crop OCR supplies this text alone, whereas Page + Crop OCR combines it with page images, allocating the 16K visual token budget to OCR first and the remainder to page images. OCR alone scores below Page on both benchmarks, but their combination exceeds both, improving over Page by 2.89 percentage points on SciEGQA and 0.88 on ViDoRe-v3. The ViDoRe-v3 gain is small, and its paired bootstrap interval includes zero. Page + Crop OCR also scores above Page + Crop on ViDoRe-v3 and slightly below it on SciEGQA. Figure 4(b) illustrates how selected text can complement page images. OCR provides the relevant community names and counts in a continuous text sequence, while the page retains their row and column layout. Page and Crop OCR return incorrect respondent totals of 26 and 57, respectively, whereas their combination gives the correct total of 31.

We adopt Page + Crop OCR as our Clear design, making selected text explicit while retaining the pages as a visual reference. Both Whiteout Page and Page + Crop OCR improve over Page on the two benchmarks. However, with identical OCR text, Whiteout and Canvas each raise accuracy on ViDoRe-v3 but lower it on SciEGQA relative to Page + Crop OCR (Table 8). Neither offers a consistent advantage, so we retain the simpler Page + Crop OCR input.

Finding 2. Clear

Content refinement that makes evidence prominent can improve the use of information already present in the retrieved pages. Making selected text explicit while retaining page images is one effective way to do so, with the combined input outperforming either input alone.

3.4 The LoC framework

LoC applies the Loud and Clear designs to end-to-end RAG by combining scaled retrieved pages with OCR from query-relevant regions. Without additional training, LoC reuses the retriever's token embeddings to select regions and changes only how the retrieved pages are given to the generator. Following the **Loud** design, we use full-page scaling to keep visual fidelity high within an expanded visual token budget. Within this budget, we combine page images with OCR text from selected regions, applying the **Clear** design so that explicit text complements rather than replaces page images.

To form this input, LoC selects query-relevant regions in the top-k pages from our baseline retriever, ColQwen3 (Huang & Tan, 2025). For each page, we map MinerU's region bounding boxes (Wang et al., 2024a) to the page embedding's visual token grid. We then score each region using ColBERT's MaxSim-based late interaction (Khattab & Zaharia, 2020), summing each query token's maximum similarity to visual tokens within that region. For OCR, we select each page's top-m regions by score, or all detected regions if fewer than m. All retrieved pages remain as visual context.

LoC then fits selected-region OCR and page images into one expanded visual token budget. We extract OCR text at the original page resolution as in Section 3.3 and group it by page in reading order. We set the visual token budget for actual retrieval to match the token expansion used by the

Table 2: **LoC reaches the highest accuracy on all three benchmarks.** The first three methods use their released generator configurations. All remaining methods use Qwen3.6-35B-A3B. ColQwen3 and both LoC variants share the retriever and generator, and LoC w/o Clear gives the whole budget to page images. The margin over ColQwen3, the strongest prior baseline on each benchmark, is 8.88, 3.09, and 4.22 percentage points on SciEGQA, ViDoRe-v3, and MMLongBench-Doc.

Method	SciEGQA	ViDoRe-v3	MMLongBench-Doc
VDocRAG (Tanaka et al., 2025)	16.14	23.64	19.07
SV-RAG-InternVL2 (Chen et al., 2025)	16.94	24.58	19.98
EVisRAG-7B (Sun et al., 2026)	20.46	23.10	32.63
RegionRAG (Li et al., 2026)	28.77	32.10	24.38
ViSAR (Mialland et al., 2026)	60.20	43.32	39.51
ColPali (Faysse et al., 2025)	64.08	41.64	39.78
M3DocRAG (Cho et al., 2025)	67.65	43.32	41.34
ColQwen3 (baseline) (Huang & Tan, 2025)	69.19	46.61	41.61
LoC w/o Clear	75.48	49.16	43.54
LoC	**78.07**	**49.70**	**45.83**

Page + Crop OCR condition, which is about 5.12 times that of the same oracle pages at the baseline input resolution on average across the two benchmarks. To approximate this token expansion in actual retrieval, we scale the baseline input width and height by $s = 2.25 \approx \sqrt{5.12}$ and using the resulting total visual token count as each query's visual token budget. As in Section 3.3, we allocate this budget to OCR text first and resize the original page images to fit the remainder. The generator then receives the page images and their OCR text, followed by the QA instruction and question.

4 Applying LoC to End-to-End RAG Systems

We use LoC to evaluate whether the Loud and Clear designs selected in Section 3 remain effective in actual retrieval settings. We compare end-to-end performance with prior RAG systems, examine the effects of visual token budgets and content refinement, and test these designs across generators.

Setup. We evaluate on SciEGQA (Yu et al., 2026b) (1,623 queries), ViDoRe-v3 (Loison et al., 2026) (1,489 English queries across all five public English-language datasets), and MMLongBench-Doc (Ma et al., 2024b) (1,091 queries). For SciEGQA and MMLongBench-Doc, retrieval covers all pages of the document provided with each question. For ViDoRe-v3, retrieval covers the full corpus of the corresponding domain. We use Qwen3.6-35B-A3B as the default generator with a fixed QA instruction, zero temperature, and disabled thinking. For LoC, we use $k = 4$ retrieved pages, $m = 5$ regions per page, and the default scale $s = 2.25$ described in Section 3.4. We report answer accuracy under a common evaluation protocol, and Appendix D specifies the QA instruction, the judge model and prompt, and the configuration of each compared method.

End-to-end RAG performance. LoC achieves the highest accuracy on all three benchmarks (Table 2). We use the released generator configurations for three prior methods and the common generator for the remaining five. In its released setting, RegionRAG selects four regions across all candidate pages, giving the generator less page content than four pages. ColQwen3 and LoC use the same retrieved pages, page order, and generator. Relative to ColQwen3, the strongest prior baseline on each benchmark, LoC improves accuracy by 8.88, 3.09, and 4.22 points on SciEGQA, ViDoRe-v3, and MMLongBench-Doc, respectively. Paired query-level bootstrap intervals for all three gains exclude zero (Appendix E). These gains indicate that our controlled findings in Section 3 provide practical guidance for improving evidence use in end-to-end RAG.

We evaluate LoC w/o Clear, which omits selected-region OCR text while retaining full-page scaling and allocating the entire expanded visual token budget to page images. Both LoC variants share ColQwen3's retrieved pages, page order, and generator. With Loud alone, LoC w/o Clear already outperforms ColQwen3 on all three benchmarks. Under the same visual token budget, LoC improves accuracy over this variant by 2.59, 0.54, and 2.29 percentage points on SciEGQA, ViDoRe-v3, and MMLongBench-Doc, respectively. These gains show that Clear further improves evidence use.

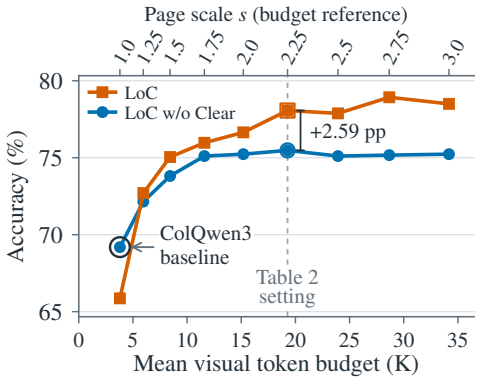


Figure 5: **Page scaling saturates, and Clear adds further gains.** SciEGQA accuracy at reference page scales $s = 1.0$–3.0, with retrieved pages, page order, and generator fixed. At the smallest budget, OCR priority leaves fewer page visual tokens, possibly lowering accuracy.

Table 3: **The gain over ColQwen3 carries to two additional generators.** ColQwen3 uses $s = 1.0$ and LoC uses $s = 2.25$. LoC improves over ColQwen3 on all three benchmarks with every generator we evaluate.

Method	SciEGQA	ViDoRe-v3	MMLong Bench-Doc
Qwen2.5-VL-7B			
ColQwen3	48.98	39.69	32.36
LoC	**56.44**	**41.91**	**36.57**
Qwen2.5-VL-32B			
ColQwen3	60.01	47.15	36.57
LoC	**67.04**	**48.22**	**39.96**
Qwen3.6-35B-A3B			
ColQwen3	69.19	46.61	41.61
LoC	**78.07**	**49.70**	**45.83**

Effects of visual token budget and content refinement. On SciEGQA, page scaling saturates as the budget grows, whereas Clear keeps adding gains above the smallest budget. We sweep the reference page scale s over 1.0–3.0 in 0.25 steps, fixing the retrieved pages, page order, and generator and matching the per-query visual token budget between the two variants. Figure 5 shows that LoC w/o Clear improves from the ColQwen3 baseline of 69.19% at $s = 1.0$ to 75.11% at $s = 1.75$, then remains near 75% as the budget grows further. Furthermore, LoC outperforms this variant at every tested scale from $s = 1.25$ to 3.0, with gains of 2.59–3.76 percentage points from $s = 2.25$ (Table 2) to $s = 3.0$. Thus, combining selected-region OCR with page images adds gains even where further page scaling yields little improvement.

At the smallest budget, however, LoC scores 65.87%, below the 69.19% of LoC w/o Clear. Allocating the budget to OCR first reduces the mean page visual token count from 3.80K to 1.48K, which may lower the visual fidelity of the page images that complement the selected text. This pattern is consistent with the controlled comparisons in Section 3.3, where selected-region OCR alone underperformed Page, while their combination outperformed both. Together, these observations support keeping visual fidelity high when part of the budget goes to selected-region OCR.

Generalization across generators. The gains of LoC hold across generators. We evaluate the Loud and Clear designs with two additional generators, Qwen2.5-VL-7B and Qwen2.5-VL-32B (Bai et al., 2025). For each generator, ColQwen3 and LoC use the same retrieved pages and page order, and LoC retains the default scale of $s = 2.25$. Table 3 shows that LoC improves accuracy over ColQwen3 on all three benchmarks with each generator. These gains show that the combined LoC input improves evidence use with each evaluated generator.

5 Conclusion and Discussion

Evidence in correctly retrieved pages can remain underused by the generator. With pages and the generator fixed, our controlled comparisons motivate Loud, sufficient visual tokens with high visual fidelity, and Clear, content refinement that makes evidence prominent. LoC implements these designs through full-page scaling and selected-region OCR, without ground-truth annotations or additional training. It outperforms eight prior methods on three benchmarks and improves over the ColQwen3 baseline across three generators. These findings highlight the importance of how retrieved pages are presented to the generator alongside which pages are retrieved.

Limitations. LoC's larger visual token budgets introduce a trade-off between accuracy and inference latency (Appendix F). Clear relies on document parsing and OCR quality, and allocating tokens to OCR can reduce accuracy under tight budgets. Our evaluation focuses on English documents and Qwen-family generators and uses a single reference-based judge. Adaptive budget allocation and broader evaluation across languages, model families, and judges are directions for future work.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and editing, literature search, code development, figure and table preparation, and interpretation of experimental results. The authors reviewed and verified all AI-assisted material and take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide the controlled experimental settings and input construction details in Appendices B, C, and D. The answer evaluation protocol, including the full judge prompt, and the end-to-end RAG configurations are described in Appendix D.

REFERENCES

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i2.32171. URL https://doi.org/10.1609/aaai.v39i2.32171.

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Jian Chen, Ruiyi Zhang, Yufan Zhou, Tong Yu, Franck Dernoncourt, Jiuxiang Gu, Ryan Rossi, Changyou Chen, and Tong Sun. Sv-rag: Lora-contextualizing adaptation of mllms for long document understanding. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 4447–4462, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0d40c7b4d78bf7f08ba532542818f6c0-Paper-Conference.pdf.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 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: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXXI*, pp. 19–35, Berlin, Heidelberg, 2024a. Springer-Verlag. ISBN 978-3-031-73003-0. doi: 10.1007/978-3-031-73004-7_2. URL https://doi.org/10.1007/978-3-031-73004-7_2.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, Ji Ma, Jiaqi Wang, Xiaoyi Dong, Hang Yan, Hewei Guo, Conghui He, Botian Shi, Zhenjiang Jin, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, Pinlong Cai, Licheng Wen, Xiangchao Yan, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites, 2024b. URL https://arxiv.org/abs/2404.16821.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3docvqa: Multi-modal multi-page multi-document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 6237–6247, October 2025.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595.

Sören Domrös, Daniel Lucas, Reinhard von Hanxleden, and Klaus Jansen. On order-preserving, gap-avoiding rectangle packing. In *Proceedings of the 13th International Conference on Information Visualization Theory and Applications (IVAPP'21)*, pp. 38–49. SciTePress, 2021. doi: 10.5220/0010186400380049. URL https://www.scitepress.org/Papers/2021/101864/101864.pdf.

Kuicai Dong, Yujing Chang, Derrick Goh Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. MMDocIR: Benchmarking multimodal retrieval for long documents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30971–31005, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1576. URL https://aclanthology.org/2025.emnlp-main.1576/.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, CELINE HUDELOT, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 61424–61449, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/99e9e141aafc314f76b0ca3dd66898b3-Paper-Conference.pdf.

Xin Huang and Kye Min Tan. Beyond text: Unlocking true multimodal, end-to-end rag with tomoro colqwen3, 2025. URL https://tomoro.ai/insights/beyond-text-unlocking-true-multimodal-end-to-end-rag-with-tomoro-colqwen3. Model card: https://huggingface.co/TomoroAI/tomoro-colqwen3-embed-8b.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

O. Khattab and Matei A. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. URL https://api.semanticscholar.org/CorpusID:216553223.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Jaehoon Lee, Mingi Jung, Soohyuk Jang, Seungryong Yoo, Dahuin Jung, and Sungroh Yoon. Balancing saliency and coverage: Semantic prominence-aware budgeting for visual token compression in vlms, 2026. URL https://arxiv.org/abs/2603.14892.

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

Yinglu Li, Zhiying Lu, Zhihang Liu, Yiwei Sun, Chuanbin Liu, and Hongtao Xie. Regionrag: Region-level retrieval-augmented generation for visual document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6662–6670, 2026.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

António Loison, Quentin Macé, Antoine Edy, Victor Xing, Tom Balough, Gabriel de Souza P. Moreira, Bo Liu, Manuel Faysse, Celine Hudelot, and Gautier Viaud. ViDoRe v3: A comprehensive evaluation of retrieval augmented generation in complex real-world scenarios. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16570–16600, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.755. URL https://aclanthology.org/2026.acl-long.755/.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. Unifying multimodal retrieval via document screenshot embedding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6492–6505, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.373. URL https://aclanthology.org/2024.emnlp-main.373/.

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. Mmlongbench-doc: benchmarking long-context document understanding with visualizations. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024b. Curran Associates Inc. ISBN 9798331314385.

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2200–2209, January 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C.V. Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1697–1706, January 2022.

Adrien Mialland, Marc Plantevit, Julien Gallois, and Céline Robardet. Visar: Training-free adaptive-k retrieval for visual document question answering, 2026. URL https://arxiv.org/abs/2609.02486.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Zhivar Sourati, Zheng Wang, Marianne Menglin Liu, Yazhe Hu, Mengqing Guo, Sujeeth Bharadwaj, Kyu J. Han, Tao Sheng, Sujith Ravi, Morteza Dehghani, and Dan Roth. LAD-RAG: Layout-aware dynamic RAG for visually-rich document understanding. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15945–15968, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.724. URL https://aclanthology.org/2026.acl-long.724/.

Yubo Sun, Chunyi Peng, Yukun Yan, Shi Yu, Zhenghao Liu, Sen Mei, Chi Chen, and Maosong Sun. Visrag2.0: Mitigating visual hallucinations via evidence-guided multi-image reasoning in visual retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2510.09733.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A. Rossi, and Dinesh Manocha. VisDoM: Multi-document QA with visually rich elements using multimodal retrieval-augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the*

2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 6088–6109, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.310. URL https://aclanthology.org/2025.naacl-long.310/.

Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Vdocrag: Retrieval-augmented generation over visually-rich documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24827–24837, June 2025.

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. Mineru: An open-source solution for precise document content extraction, 2024a. URL https://arxiv.org/abs/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. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024b. URL https://arxiv.org/abs/2409.12191.

Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao. ViDoRAG: Visual document retrieval-augmented generation via dynamic iterative reasoning agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9113–9134, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.464. URL https://aclanthology.org/2025.emnlp-main.464/.

Yahong Wang, Juncheng Wu, Zhangkai Ni, Longzhen Yang, Yihang Liu, Chengmei Yang, Ying Wen, Lianghua He, Xianfeng Tang, Hui Liu, and Yuyin Zhou. When token pruning is worse than random: Understanding visual token information in vllms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31910–31919, June 2026.

Zhongyu Wang. MDocRAG-RL: Empowering multi-modal document RAG via complex visual reasoning with reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8641–8651, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.420. URL https://aclanthology.org/2026.findings-acl.420/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In *International Conference on Learning Representations*, volume 2024, pp. 43478–43502, 2024.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19792–19802, June 2025.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. Visiontrim: Unified vision token compression for training-free MLLM acceleration. In *The Fourteenth International*

Conference on Learning Representations, 2026a. URL https://openreview.net/forum?id=57IXIg6nZ0.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 21074–21098, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/3640a1997a4c9571cea9db2c82e1fc35-Paper-Conference.pdf.

Wenhan Yu, Zhaoxi Zhang, Wang Chen, Guanqiang Qi, Weikang Li, Lei Sha, Deguo Xia, and Jizhou Huang. Sciegqa: A dataset for scientific evidence-grounded question answering and reasoning, 2026b. URL https://arxiv.org/abs/2511.15090.

Joohyung Yun, Doyup Lee, and Wook-Shin Han. LILaC: Late interacting in layered component graph for open-domain multimodal multihop retrieval. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20540–20559, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1037. URL https://aclanthology.org/2025.emnlp-main.1037/.

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. SparseVLM: Visual token sparsification for efficient vision-language model inference. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74840–74857. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhang25s.html.

Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N. Metaxas, and Licheng Yu. Accelerating multimodal large language models by searching optimal vision token reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29869–29879, June 2025.

APPENDIX

A Underused Visual Token Capacity in Prior Methods **16**

B Extended Analysis of Loud **18**

B.1 Conditions 18

B.2 Visual token accounting and resolution allocation 18

B.3 Separating intermediate resolution from final budget 19

B.4 Complete budget sweep 20

C Extended Analysis of Clear **21**

C.1 Construction 21

C.2 Additional comparisons 21

C.3 Region selection and answer content 22

C.4 Examples of page context 23

D Implementation Details **24**

D.1 Controlled conditions and evaluation 24

D.2 Answer evaluation protocol 24

D.3 End-to-end RAG configurations 26

E Additional Results on End-to-End RAG Systems **28**

E.1 Paired comparisons and uncertainty 28

E.2 ViDoRe-v3 domain results 28

E.3 Document-cluster sensitivity 28

F Generation-stage Latency across Visual Token Budgets **30**

A UNDERUSED VISUAL TOKEN CAPACITY IN PRIOR METHODS

We measure the visual token usage of prior document RAG methods on the SciEGQA pages used in Section 3. We adapt each method's image resizing and slicing settings to the common image processor used in our controlled experiments, allowing their input sizes to be measured on the same pages and token grid. Table 4 reports the resulting usage. The means across the five generator-side methods range from 718.08 to 2,679.87 visual tokens per query, giving the range of 0.72K–2.68K reported in Section 3.1, which the abstract and introduction round to 0.7K–2.7K. The table also reports the median and range across individual queries.

Table 4: **Under a common image processor, five prior methods give the generator 0.72K–2.68K visual tokens per query on average.** Visual token usage on the 1,623 SciEGQA queries. Counts sum all views from all retrieved pages for each query. The SV-RAG row uses the later Self-Visual-RAG implementation described below. ColPali is a retriever-side setting and is excluded from the range reported in the main text.

Method	Mean	Median	Min–max
M3DocRAG	2,281.80	1,900	1,350–3,922
VDocRAG	1,906.41	1,700	1,300–3,400
VisRAG	718.08	630	486–1,260
SV-RAG (later implementation)	2,679.87	2,548	1,372–5,096
VisDoM	740.65	616	600–1,260
ColPali (retrieval)	234.40	196	196–392

Common measurement setting. The audit uses the same 1,623 queries and their retrieved pages as Appendix D.1. Each query has one or two retrieved pages, with 1,941 page occurrences and 837 unique pages in total. We start from the original page images rendered at 300 DPI, before the resizing used in the controlled budget sweep. We fix the queries, page identities and order, original images, and final image processor while varying the method's settings. Each method's settings are applied to all retrieved pages for a query, without applying its original retrieval top-k or matching a query-level budget from the controlled sweep. We run the processor on CPU without loading model weights, performing retrieval, or generating answers.

The processor uses the same Qwen3.6-35B-A3B checkpoint revision as Appendix D.1, with Transformers (Wolf et al., 2020) 5.12.1 and PyTorch (Paszke et al., 2019) 2.11.0. Its `Qwen2VLImageProcessor`, accessed through `Qwen3VLProcessor`, uses 16×16 patches and a 2×2 spatial merge. A processed image view of height H and width W therefore contributes $(H/32)(W/32)$ visual tokens. We count the expanded image tokens and check them against the image grids, summing across all views and pages for each query before averaging over queries. We exclude question text, image delimiters, and sequence padding but include padding encoded within an image view. Processor measurements are reused for views with the same dimensions and area constraints, since these determine the token count.

Method selection and adaptation. We use the public settings associated with each system's default or main model setting. The model names below identify the source of each setting. The common processor has default minimum and maximum areas of 65,536 and 16,777,216 pixels per view, with the method-specific overrides given below.

- **M3DocRAG.** We use the Qwen2-VL-7B settings (Cho et al., 2025), constructing one view per page with dimensions corresponding to 144 DPI and the most frequent page dimensions in the full document. The document-level dimensions are determined from all pages, including those outside the retrieved set. We resize the original raster images to these dimensions and retain the original area limits of 3,136 and 12,845,056 pixels per view. This matches the target dimensions and token counts of the rendering setting without assuming pixel-identical PDF rerendering.
- **VDocRAG.** We use the Phi-3-V settings (Tanaka et al., 2025) with `num_crops=16`, the checkpoint setting used without a generator override. Each page produces one global view and up to 16 local tiles. The original 336×336 views are mapped to 320×320 views on the common

processor grid, yielding 100 tokens per view. Only the actual views are counted, excluding empty batch crop slots and any duplicate of the tiled canvas.

- **VisRAG.** We use the MiniCPM-V 2.6 multi-image settings (Yu et al., 2025), retaining its slicing procedure with a base scale of 448 and patch size of 14. Each page produces one global view and up to nine local views. We represent the native resampler's fixed 64 tokens per view by a 64-token cap under the common processor, with area limits of 1,024 and 65,536 pixels per view. The resulting common count can be below 64 for a view, and is at most 640 per page. This adaptation imposes a token-count constraint and does not reproduce the learned resampler or its information preservation.
- **SV-RAG.** We use the InternVL2-4B settings from the later, author-linked Self-Visual-RAG implementation of SV-RAG (Chen et al., 2025).[1] Its answer-generation call sets `max_num=20`, yielding up to 20 local 448×448 tiles and an additional thumbnail when more than one local tile is present. Each tile or thumbnail contributes 196 tokens under the common processor. The tile limit belongs to this later implementation and is not a verified setting of the original paper.
- **VisDoM.** We use the Qwen2-VL-7B visual-branch settings (Suri et al., 2025), constructing one full-page view with minimum and maximum areas of 262,144 and 655,360 pixels. These correspond to 256–640 visual tokens per page on the common grid. We count the visual branch once per page, without adding another image pass for the text branch or modality fusion.

Retriever-side reference. For ColPali (Faysse et al., 2025), we resize each page to 448×448 without preserving its aspect ratio, yielding 196 tokens per page under the common processor. Its query-level count in Table 4 sums these page counts over the same retrieved pages. This row describes a retriever-side configuration, so it is kept separate from the five generator-side methods used to obtain the 0.72K–2.68K range in the main text.

Interpretation. These measurements characterize the adapted settings under a shared processor, rather than the native token counts or QA performance of the original systems. Mapping each method's settings to a common token grid does not preserve their original resolution, token count, and information content simultaneously, particularly when a learned compression module is replaced by an input-area constraint. The usage measurements establish the input levels of the selected methods on these pages. The separate controlled results in Figure 2(a) establish the utility of greater visual token capacity in our setting. The audit alone does not attribute a performance loss to any prior system as it was originally designed and trained.

[1] The author's project page links to the Self-Visual-RAG implementation.

B Extended Analysis of Loud

This appendix details the conditions, visual token accounting, and two-stage resizing procedure used in Section 3.2. The datasets, generation settings, and answer evaluation protocol shared by all conditions are specified in Appendices D.1 and D.2.

B.1 Conditions

We keep the oracle pages fixed and vary their visual representation to the generator. Each query is assigned a target budget $B \in \{1, 2, 4, 8, 16, 32, 64\}$K, where 1K = 1,000 tokens. The target applies to the total visual token count across all input views of the query. These conditions use only page images, without crops, OCR text, or other additional inputs.

Full-page resolution scaling. We represent each page with a single full-page view and adjust its resolution according to the query-level target budget. At each budget, we resize directly from the original page image, rather than from a lower-budget input.

Repeated pages and repeated blank pages. Repeated pages uses multiple views of each retrieved page, whereas repeated blank pages combines page views with additional blank images. The target visual token budget applies to all image views of a query. These conditions compare repetition and blank-image addition with full-page resolution scaling at the same target budgets.

Multi-scale views. For a query with P oracle pages, we construct exactly three full-page views for each page, yielding $3P$ input images at every target budget. The three views use relative linear scales $s_\ell \in \{1, 2, 4\}$ and are constructed independently from the original page image. Because visual token count scales approximately with image area, their target token shares are proportional to s_ℓ^2, giving the ratio $1 : 4 : 16$ across the three scales.

For a query-level target budget B, each page is first assigned B/P tokens, and scale ℓ receives

$$b_{p,\ell} = \frac{B}{P} \frac{s_\ell^2}{1^2 + 2^2 + 4^2}. \tag{1}$$

Thus, the three scales receive $1/21$, $4/21$, and $16/21$ of each page's target allocation. Each view is independently quantized to the processor's 32-pixel grid, and no residual query-level correction is applied after this quantization. Consequently, realized visual token counts can deviate from B, particularly for queries with many oracle pages at small budgets. We retain all oracle pages and all three scales, ordering the views page-wise from coarse to fine, so the coarse view of each page precedes its finer views.

Spatial tiling. We divide each page into a uniform 2×2 grid and supply the four tiles as separate input views, without retaining a full-page view. The tiles from all pages share the query-level target budget B, so each tile's resolution decreases as the number of pages grows.

B.2 Visual token accounting and resolution allocation

The generator's image processor uses 16×16 pixel patches and merges 2×2 spatial patches into one visual token. For a static image whose processed dimensions are $H_j \times W_j$, both divisible by 32, the token count is

$$T_j = \frac{H_j W_j}{32^2}, \qquad \widehat{B}(q) = \sum_{j=1}^{M(q)} T_j, \tag{2}$$

where $M(q)$ is the number of input views for query q. Equivalently, the processor grid (t_j, h_j, w_j) contributes $t_j h_j w_j / 4$ tokens, with $t_j = 1$ for these static images. The count includes blank image regions but excludes the question, instruction text, and image boundary markers.

For full-page resolution scaling, the reference dimensions are obtained by applying the processor's resizing rule with a per-page area ceiling of 1,003,520 pixels. These dimensions serve only as allocation references: the final pixels are resized directly from the original page image. We apply

a common continuous scale to the reference dimensions within a query, then map the dimensions to the processor's 32-pixel grid. To approach the target, the allocator chooses between the resulting sizes and their adjacent grid transitions in the required direction. This preserves the common-scale construction up to quantization rather than assigning arbitrary independent resolutions to different pages. The selected combination minimizes the absolute token error among these candidates. Thus, B is a target, and the realized count $\widehat{B}(q)$ can lie slightly above or below it.

Table 5 reports the mean realized visual token counts for all five Loud constructions. Despite processor-grid quantization, the condition-level means closely match their target budgets across both benchmarks.

Table 5: **All Loud constructions closely match their target visual-token budgets on average.** Values report the mean realized visual tokens per query. Across all constructions and target budgets, the largest absolute deviation of a condition mean from its target is 35.12 tokens, and the largest relative deviation is 1.32%.

Construction	1K	2K	4K	8K	16K	32K	64K
SciEGQA							
Full-page resolution scaling	1002.8	1999.5	4009.1	7978.5	15993.2	32002.7	63977.1
Repeated pages	1002.8	2011.9	3996.2	7998.8	15990.0	32001.6	64002.7
Repeated blank pages	1002.8	2011.9	3996.2	7998.8	15990.0	32001.6	64002.7
Multi-scale views	987.4	1973.6	4000.1	8016.7	16032.2	32030.5	64032.9
Spatial tiling (2×2)	997.8	1999.0	3996.2	8004.4	15994.0	32017.6	64024.3
ViDoRe-v3							
Full-page resolution scaling	1000.0	2000.4	4000.1	7998.9	15996.9	32003.0	63997.1
Repeated pages	1000.0	2002.5	3999.6	7998.7	15997.7	31999.7	63997.2
Repeated blank pages	1000.0	2002.5	3999.6	7998.7	15997.7	31999.7	63997.2
Multi-scale views	987.3	1985.0	3984.1	8006.2	16007.7	32022.9	64035.1
Spatial tiling (2×2)	999.7	1999.7	3999.6	7999.7	15998.4	32001.1	64001.5

At the query level, the realized budgets are also closely matched for four of the five constructions. For full-page resolution scaling, repeated pages, repeated blank pages, and spatial tiling, no query deviates from its target by more than 5% in any evaluated benchmark or budget. Multi-scale views show larger deviations on ViDoRe-v3 at the smallest budgets because each page's allocation is divided across three scales and each view is independently quantized to the processor grid. At 1K, 8.13% (121/1,488) of ViDoRe-v3 queries deviate from the target by more than 5%; this fraction falls to 3.02% at 2K and 0.54% at 4K, and is zero from 8K onward.

B.3 Separating intermediate resolution from final budget

Full-page resolution scaling changes both visual fidelity and the final visual token count. We separate these factors using an intermediate resolution I, which controls the visual fidelity retained before final resizing, and a final budget V. I specifies image dimensions and does not directly measure semantic information. Let P_i be an original page image, and let $d_i(b)$ denote its processed dimensions in the page-only allocation for target b. With $R(X; d)$ denoting resizing image X to dimensions d, we construct

$$Z_i^{(I)} = R(P_i; d_i(I)), \qquad X_i^{(I,V)} = R(Z_i^{(I)}; d_i(V)). \tag{3}$$

The 1K and 4K intermediate images are constructed independently from the original pages. Conditions with the same I reuse the same intermediate image tensors. Both stages use the existing processor's bicubic resizing with antialiasing enabled, and the second resize is skipped when its input already has exactly the required final dimensions.

The final dimensions $d_i(V)$ are taken from the existing page-only allocation; they are not recomputed from the intermediate image. Consequently, conditions with the same V have identical page order, per-page final dimensions, and actual visual token counts for each query. The input contains all page views followed by the QA instruction and question. Reference answers are used for evaluation only and never appear in the generator input.

Controlled comparisons. At fixed V, increasing I changes the visual fidelity retained before final resizing while preserving the final token grid. At fixed I, increasing V enlarges the same intermediate image without recovering information lost in the first resize, testing whether a larger visual token budget improves the use of the preserved evidence.

We evaluate the off-diagonal conditions $(I, V) = (1\text{K}, 4\text{K})$, $(1\text{K}, 16\text{K})$, and $(4\text{K}, 16\text{K})$. The diagonal entries in Figure 3 use the corresponding page-only results. Configurations with $I > V$ are outside this degrade-then-upsample design and are not evaluated.

Table 6: **Visual fidelity and the final visual token budget each improve accuracy with the other held fixed.** Accuracy (%) under two-stage resizing. Intermediate resolution controls retained source visual fidelity. Intermediate resolution and final visual token budget are expressed in query-level token-equivalent units, with 1K = 1,000 tokens. Diagonal entries ($I = V$) reuse the page-only budget sweep, while off-diagonal entries ($I < V$) use two-stage resizing.

Intermediate resolution, I	Final visual token budget, V	SciEGQA	ViDoRe-v3
1K	1K	65.74	42.47
1K	4K	75.54	50.00
4K	4K	77.76	51.55
1K	16K	78.31	51.21
4K	16K	79.61	54.84
16K	16K	79.98	55.24

B.4 Complete budget sweep

Table 7 provides the numerical values underlying the five curves in Figure 3(a) and (b). For full-page resolution scaling, we use the complete 1,623-query SciEGQA cohort and 1,488-query ViDoRe-v3 cohort at every budget. Answers reaching the 1,024-token output limit remain in the evaluation.

Table 7: **Multi-scale views lead on SciEGQA at 2K–64K, whereas full-page resolution scaling leads on ViDoRe-v3 at 4K–64K.** Accuracy (%) across target visual token budgets for the five Loud conditions. Each target is summed across all input images. Bold indicates the highest observed accuracy at each budget within a benchmark. Values correspond to Figure 3.

Condition	1K	2K	4K	8K	16K	32K	64K
SciEGQA							
Full-page resolution scaling	**65.74**	72.89	77.76	78.93	79.98	80.96	80.10
Repeated pages	65.68	67.16	69.62	70.43	71.47	72.83	72.15
Repeated blank pages	65.68	65.62	66.61	66.54	68.95	69.07	69.62
Multi-scale views	62.11	**73.75**	**78.00**	**79.98**	**81.02**	**81.82**	**81.52**
Spatial tiling (2×2)	53.42	62.17	66.36	69.25	69.44	69.07	67.65
ViDoRe-v3							
Full-page resolution scaling	42.47	46.71	**51.55**	**55.98**	**55.24**	**55.51**	**55.11**
Repeated pages	**42.94**	43.75	44.49	44.29	44.76	44.56	44.42
Repeated blank pages	**42.94**	42.20	43.62	44.09	43.62	45.43	45.43
Multi-scale views	42.27	**47.31**	51.08	53.09	54.84	54.23	53.83
Spatial tiling (2×2)	37.16	41.53	43.95	46.37	47.78	48.05	47.85

Results across budgets. Within the evaluated sweep, multi-scale views achieve higher accuracy than full-page resolution scaling on SciEGQA at 2K–64K, whereas full-page resolution scaling achieves higher accuracy on ViDoRe-v3 at 4K–64K. Full-page resolution scaling reaches its observed maximum at 32K on SciEGQA and 8K on ViDoRe-v3. Because the advantage of multi-scale views on SciEGQA does not carry over to ViDoRe-v3, we use a single scaled full-page view per page as a simple common choice across the two benchmarks.

C Extended Analysis of Clear

C.1 Construction

Selected regions. For each MinerU region, we identify the ground-truth evidence box that maximizes the pair of intersection-over-union (IoU) and min-area coverage lexicographically. We retain the region if either its IoU with this matched box is at least 0.5 or their intersection covers at least half of the smaller box. We add 10 pixels of padding in a per-page reference coordinate system obtained by the Qwen image processor's smart resizing, with a maximum area of 1,003,520 pixels, and map the padded box back to the original page image. Region matching is performed before padding; padding is applied only after a MinerU region has been selected. The fallback to raw ground-truth boxes applies only when no matched region exists anywhere in the query.

Visual refinements. Whiteout replaces pixels outside the selected regions with white, preserving Page's image dimensions, selected pixels, page order, and visual token counts. Crop presents the selected regions as individual images. Page + Crop allocates 4K VT to full-page views and 12K VT to crop images. Canvas packs the same crops separately for each page using the order-preserving rectangle-packing algorithm in the Eclipse Layout Kernel (ELK) (Domrös et al., 2021), implemented in elkjs 0.12.0. Crops are ordered by their vertical and horizontal coordinates and packed without rotation or added spacing. We render the raw crops directly at the allocated final resolution to avoid enlarging an intermediate low-resolution canvas. This grouping changes the original page geometry.

Main shared-budget OCR comparison. We extract text before generation resizing with the PaddleOCR components (Cui et al., 2025) bundled in MinerU 2.5.4, using PP-OCRv5 detection and PP-OCRv4 text recognition. Detected strings are concatenated in detection order and grouped by page and crop, without reconstructing table structure or correcting text using reference answers. The main Crop OCR and Page + Crop OCR conditions in Table 1 share text extracted from the same padded crops. OCR-only inputs are truncated to at most 16,000 text tokens. Combined inputs allocate tokens to OCR first and resize the visual views into the remaining budget, enforcing $T_{\mathrm{OCR}} + V \leq 16{,}000$. This cap includes OCR text and visual tokens, excluding the question and other prompt text. If OCR fills the budget, no images are included, as occurs for three ViDoRe-v3 queries in the main Page + Crop OCR condition of Table 1.

C.2 Additional comparisons

Visual refinements with identical OCR. Table 8 isolates the effect of visual refinement using identical OCR text extracted from the padded crops described in Appendix C.1. The Page + Crop OCR row is the same condition reported in Table 1; the other rows change only the visual refinement.

Table 8: **With the same padded-crop OCR, no visual refinement leads on both benchmarks.** Accuracy with identical crop OCR and different visual inputs. All refinements share the same OCR text under an OCR + VT $\leq$ 16K budget. Bold marks the highest accuracy per dataset.

	Accuracy (%)	
Refinement	SciEGQA	ViDoRe-v3
Page + Crop OCR	**82.87**	56.12
Whiteout Page + Crop OCR	82.50	57.12
Canvas + Crop OCR	81.45	**57.26**

Adding OCR to fixed page images. Table 9 tests a different question: whether selected-region OCR provides additional value when the page images themselves are left unchanged. We therefore preserve the 16K Page images and their visual tokens exactly, and add the padded-crop OCR as additional text. Unlike the shared-budget comparisons above, the OCR tokens in this setting are not counted against the 16K visual token budget of the page images.

Table 9: **Crop OCR helps even when the page images are preserved exactly.** Accuracy (%) when adding crop OCR while preserving the Page images and their VT exactly. OCR uses additional tokens beyond the 16K visual token budget. Accuracy increases over Page by 2.09 and 0.61 percentage points on SciEGQA and ViDoRe-v3, respectively.

Condition	Budget	SciEGQA	ViDoRe-v3
Page	16K VT	79.98	55.24
Page + Crop OCR	16K VT + OCR	82.07	55.85

Paired uncertainty estimates. For the shared-budget comparisons in Table 1, we report paired-query 95% bootstrap percentile intervals using 20,000 resamples. Each comparison uses the same 1,623 SciEGQA or 1,488 ViDoRe-v3 queries and their saved strict-correct judgments; ViDoRe-v3 accuracy is pooled over queries. For Page→Whiteout, the accuracy changes are $+2.03$ percentage points (pp), $[0.49, 3.51]$, on SciEGQA and $+0.81$ pp, $[-1.14, 2.69]$, on ViDoRe-v3. For Page→Page+Crop OCR, they are $+2.89$ pp, $[1.23, 4.56]$, and $+0.88$ pp, $[-1.14, 2.82]$, respectively. For Crop→Canvas, they are -0.30 pp, $[-1.54, 0.86]$, and $+4.10$ pp, $[2.35, 5.85]$, respectively. Displayed changes subtract rounded accuracies, while intervals use unrounded paired scores. The original analysis uses a shared random-number stream initialized with seed 20260915 and linear percentile interpolation. These comparisons form an exploratory, post hoc subset of 38 dataset-specific contrasts. The intervals are pointwise, without multiple-comparison correction, and condition on saved outputs and judgments; they do not measure generation-rerun or judge variability.

C.3 Region selection and answer content

Selected area and suppression. Table 10 groups the Page–Whiteout comparison by the fraction of page area occupied by selected regions. The observed gains are larger in the groups with smaller selected fractions on both benchmarks. The groups are not matched for question type, document structure, or difficulty, so this association neither isolates an effect of area nor supports a rule that chooses the input by the area of its selected regions.

Table 10: **Whiteout gains are larger when the selected regions cover less of the page.** Page-to-Whiteout accuracy changes by selected area. Selected area is the sum of selected-box union areas across a query's pages divided by their total area. Each difference uses paired outcomes within the indicated group. The middle group includes 10% and excludes 30%. The groups are not matched in other respects, so the association is not a causal effect of area.

	SciEGQA		ViDoRe-v3	
Selected area	N	Δ accuracy (pp)	N	Δ accuracy (pp)
$< 10\%$	312	$+4.49$	223	$+1.79$
10–30%	808	$+2.60$	717	$+1.12$
$\geq 30\%$	503	-0.40	548	0.00

Competing passages. In SciEGQA query `2412.20176-p1-249`, the question asks for the numbers of economic concepts and data sources listed in the selected passage. The page's abstract names three concepts, whereas the selected introduction passage lists four, including the Poverty Gap Index. Both passages name three data sources. Page answers "3 economic concepts and 3 data sources," matching the abstract, while Whiteout answers "4 economic concepts and 3 data sources," matching the selected passage and reference answer. The selected passage retains identical pixels and position, and both inputs use 16,006 VT. The answer correspondence is observable, but it does not establish where the model attended internally.

Selecting a table entry with OCR. SciEGQA query `2501.00176-p15-468` asks for the latitude of the site with the highest MLAT among rows citing *The Times of India*. Bhownuggur has MLAT 12.9 and latitude N21°46′, whereas Jacobabad has the highest qualifying MLAT, 19.9, and latitude N28°17′. Page answers with Bhownuggur's latitude. Crop OCR and Page + Crop OCR

both return Jacobabad's latitude, matching the reference. The OCR includes both rows' sites, coordinates, MLAT values, and references. The combined input uses 15,012 VT, compared with 15,984 for Page. This case illustrates a more accurate answer selection when the relevant text is supplied explicitly, without establishing that Page failed to read the individual values. Together with the corrected respondent total in Figure 4(b), it illustrates how supplying selected text alongside page images can improve answers from tables. These inspected cases illustrate possible improvements, not how often such improvements occur across all queries.

C.4 Examples of page context

Heading–item associations. The ViDoRe-v3 question in Figure 6(a) asks for five Verification items and excluded products. The original page separates Product Tracing and Verification into columns. Crop and Canvas responses include Product Tracing items such as "Lot-level" among the Verification items, while Page + Crop answers correctly. Inspection of the complete Canvas input confirms that all five requested items and the items from other columns remain present. The example illustrates a change in heading–item association rather than deletion of the required text, without isolating layout from changes in image size or grouping.

Surrounding information with identical OCR. The SciEGQA question in Figure 6(b) asks for the figure number of a plot and the aggregation figure cited in nearby text. Page + Crop OCR answers Figure 4 and Figure 5 correctly, while Whiteout + the same OCR answers only Figure 5. The plot and nearby paragraph remain in Whiteout, but the Figure 4 caption falls outside the selected regions and is removed. The shared OCR also lacks this caption. Both inputs use 15,730 VT and the same 223 OCR tokens. This example illustrates information that the original page retains and that the shared selected-region OCR text alone does not replace.

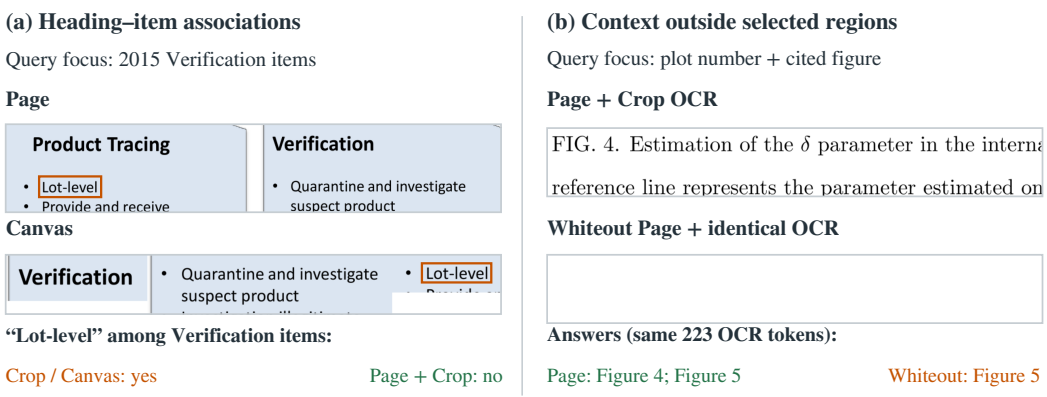


Figure 6: **The original page keeps context that crops, canvases, and whiteout can lose. (a)** "Lot-level" belongs to Product Tracing in the page, but appears among Verification items in Crop and Canvas responses. Page + Crop excludes it from those items. **(b)** The source caption labeled "FIG. 4" is present in Page but removed in Whiteout. Their identical OCR does not contain it. The displayed answer excerpts concern the requested figure numbers. Document excerpts are enlarged for readability, and colored outlines are explanatory annotations.

D Implementation Details

This section describes the controlled conditions for the analyses in Section 3, the answer evaluation protocol shared by all experiments, and the end-to-end RAG configurations.

D.1 Controlled conditions and evaluation

Data and page views. The SciEGQA comparisons in Section 3 and Figure 2 use the same 1,623 queries (Yu et al., 2026b) in every condition, with the retrieved pages fixed to the oracle pages of each query. ViDoRe-v3 (Loison et al., 2026) contains English and French document datasets and supports monolingual evaluation as well as multilingual evaluation with queries in six languages. We use the English monolingual setting on all five public English-language datasets, with 1,489 queries in total. For the Loud and Clear comparisons in Section 3, we exclude query ID 239 in the Industrial reports dataset due to missing ground-truth bounding boxes, leaving a base cohort of 1,488 queries. MinerU (Wang et al., 2024a) parses each retrieved page into paragraph, table, figure, and caption regions. We retain regions matched to the dataset's ground-truth evidence boxes and crop them directly from the original page image. When no matched MinerU region exists anywhere in a query, we use raw ground-truth boxes as fallback regions, following Appendix C.1. All full-page views precede all localized views, such as crops and canvases, in the generator input of each query.

Visual token budgets. We measure the visual token budget by the total number of visual tokens produced across all input images by the generator's image processor. For image-only conditions, the budget is the total number of visual tokens produced across all input images by the generator's image processor. For conditions with OCR, we use a shared input budget that counts both OCR text tokens and visual tokens; VT refers specifically to visual tokens from page images (Appendix C). The target budget per query is $B \in \{1\text{K}, 2\text{K}, 4\text{K}, 8\text{K}, 16\text{K}, 32\text{K}, 64\text{K}\}$, where $1\text{K} = 1{,}000$ tokens. For the page–crop mixture sweep in Figure 2(b), the fraction assigned to full-page views is $\alpha \in \{0, .25, .50, .75, 1\}$, with $\alpha = 1$ and $\alpha = 0$ denoting page-only and crop-only inputs. Full-page views and crops receive target budgets of αB and $(1 - \alpha)B$, respectively. For each query, we apply one continuous resizing factor to each view group and quantize the resulting image sizes to the processor's spatial grid. We then choose the feasible sizes that minimize the absolute difference between the resulting visual token count and the target budget B. The budget is therefore a target rather than a hard cap, and small overshoots or undershoots are permitted. Across the 35 combinations of budget and mixture, the maximum absolute deviation of a condition's mean actual token count from its target is 25.72 tokens. For the 16K comparison in Figure 2(b), the mean actual counts are 15,993.23 for page-only input and 16,010.19 for the best observed mixture ($\alpha = .25$).

Generation. Generation uses Qwen3.6-35B-A3B (Qwen Team, 2026) revision `995ad96eacd98c81ed38be0c5b274b04031597b0` in BF16 with temperature zero and thinking disabled. We use vLLM (Kwon et al., 2023) version 0.24.0 with a maximum context length of 262,144 tokens and allow up to 1,024 generated tokens. The QA instructions and decoding configuration are fixed across all conditions. OCR conditions additionally prepend the extracted OCR text to the QA prompt. We use the ViDoRe-v3 generation prompt (Loison et al., 2026):

```
Keep the response concise when appropriate.
Output only the answer.

Question: {question}
Answer:
```

Evaluation. We score generated answers using the common evaluation protocol in Appendix D.2. The accuracy reported in the text and Figure 2 is the strict accuracy defined there.

D.2 Answer evaluation protocol

Judge and inputs. The controlled analyses in Section 3 and the end-to-end comparisons in Section 4 use the same reference-based judge, `gpt-5.6-sol`, with medium reasoning effort. The judge receives the question, reference answer, and predicted answer as text, without document images or method identifiers. We submit the prompt below as a user message with no additional system

instructions, and request structured output with exactly two fields, `judgment` and `reason`. The first field must be `correct`, `partial`, or `incorrect`, and the second contains a brief explanation that names the issue behind a partial or incorrect judgment.

Judgment criteria. The judge considers question suitability and reference accuracy equally. Question suitability requires a clear answer at the scope and specificity requested by the question, while reference accuracy requires semantic agreement with the reference, including the values, entities, relationships, conditions, and additional claims in the answer. A prediction is `correct` only when both criteria are fully satisfied. It is `partial` when one criterion is fully satisfied and the other is partially satisfied, with a specific reason for the shortfall. It is `incorrect` when either criterion is not satisfied at all or neither is fully satisfied. Equivalent wording and exactly equivalent numerical or unit representations are accepted, but rounding or approximation is accepted only when explicitly allowed by the question or reference answer.

Scores and aggregation. For query i, let y_i be its final judgment and define the strict score as

$$z_i = \mathbf{1}\{y_i = \texttt{correct}\}, \qquad \text{Accuracy}(\%) = \frac{100}{N}\sum_{i=1}^{N} z_i, \tag{4}$$

where N is the number of evaluated queries in that benchmark and condition. Both `partial` and `incorrect` therefore receive zero credit. Each query contributes one final judgment to the reported accuracy. For ViDoRe-v3, we pool the queries from the five English-language datasets before computing accuracy, giving each query equal weight. The end-to-end evaluation uses $N = 1{,}623$, $1{,}489$, and $1{,}091$ for SciEGQA, ViDoRe-v3, and MMLongBench-Doc, respectively. The controlled analyses use the query cohorts specified for their respective experiments. Displayed percentage-point differences are computed by subtracting accuracies rounded to two decimal places. Bootstrap intervals are computed from unrounded query-level scores.

Judge prompt. The following is the full prompt template, with line wrapping adjusted for readability. We replace {`evaluation_input`} with a JSON object containing the string fields `question`, `reference_answer`, and `predicted_answer`.

```
You are an expert judge evaluating a predicted answer using a question
and a reference answer.

Evaluate the predicted answer using two equally important criteria:

1. Question suitability:
   Whether the predicted answer directly and clearly answers what the
   question asks, at the
   requested scope and level of specificity, and communicates a
   discernible final answer.

2. Reference accuracy:
   Whether the predicted answer is semantically equivalent to the
   reference answer with respect
   to what the question asks. The values, entities, relationships,
   conditions, and any additional
   claims that contribute to the predicted answer must be accurate
   according to the reference
   answer.

The question determines what must be answered, while the reference
answer determines what is
accurate.

Judge semantic meaning rather than wording, structure, format, or
length. Accept semantically
equivalent expressions and exactly equivalent numerical or unit
representations.
```

```
Do not accept rounded or approximate values unless the question
explicitly requests rounding,
approximation, a range, or a specific level of precision, or the
reference answer explicitly
provides an approximation or range.

Evaluate only the predicted answer actually provided. Additional
content or apparent truncation
matters only when it affects question suitability or reference
accuracy.

### Evaluation Criteria

- correct:
  The predicted answer fully satisfies both question suitability and
  reference accuracy.

- partial:
  The predicted answer fully satisfies one criterion and partially
  satisfies the other. There must
  be a clear and specific reason why the partially satisfied criterion
  is not fully satisfied.

- incorrect:
  The predicted answer does not satisfy one criterion at all,
  regardless of how well it satisfies
  the other; or neither criterion is fully satisfied.

The reason must briefly identify the specific issue that prevents a
partial or incorrect predicted
answer from being correct.

### Input Data

EVALUATION_INPUT_JSON:
{evaluation_input}

### Output Format

Return only one JSON object with exactly these two fields:

{
  "judgment": "<correct, partial, or incorrect>",
  "reason": "<a very brief explanation>"
}
```

D.3 End-to-end RAG configurations

Checkpoint links identify the evaluated model revisions for Table 2.

Common generation settings. All methods using Qwen3.6-35B-A3B share the checkpoint revision and QA instruction in Appendix D.1. Generation uses BF16, temperature zero, disabled thinking, a maximum context length of 65,536 tokens, and an output limit of 3,072 tokens. The chat template assigns sequential image labels such as `Picture 1:` in input order.

Page retrieval baselines. ColQwen3 (Huang & Tan, 2025) uses Tomoro ColQwen3-Embed-8B, with a retriever limit of 1,280 visual tokens per page. ColPali (Faysse et al., 2025) uses the ColPali-v1.2 adapter with the ColPaliGemma-3B base and `colpali-engine` 0.3.13. Both baselines rank candidate pages by MaxSim and supply the top four full-page images to the generator in retrieval order. Their generation processors use bounds of 3,136–1,003,520 pixels per image.

M3DocRAG. M3DocRAG (Cho et al., 2025) uses its ColPali-v1.2 adapter revision, the same base, and `colpali-engine` 0.3.1. We follow its PDF processing procedure by rendering all pages at 144 DPI in RGB and resizing them to the document's modal page size before retrieval. The top four retrieved pages are passed to the common generator in retrieval order, with generation processor bounds of 3,136–12,845,056 pixels for each image.

RegionRAG. RegionRAG (Li et al., 2026) uses RegionRet with the ColQwen2.5 base. We use a saliency threshold of 0.25, a neighborhood radius of one, and maximum saliency for region scoring. The four highest-scoring regions are selected across the entire candidate corpus, and crops from the same page are grouped following the released implementation. Each crop is processed within 4,096–262,144 pixels. Thus, its retrieval count specifies regions rather than pages.

ViSAR. ViSAR (Mialland et al., 2026) uses the same ColPali-v1.2 adapter revision and ColPaliGemma-3B base as the ColPali baseline, with its released image prompt prefix and adaptive selector. Pages are rendered at 144 DPI without normalizing their sizes within a document. We use `top_T`=50, `block_size`=4, `weight_thr`=0, $\gamma = 100{,}000$, and $\epsilon = 0$. The selector determines the number of pages adaptively, capped at four for this comparison. Selected pages pass through its 55-million-pixel aggregate guard and then through the common generator's processor with per-image bounds of 3,136–12,845,056 pixels.

VDocRAG. VDocRAG (Tanaka et al., 2025) uses Phi-3-vision-128k-instruct with separate VDocRetriever and VDocGenerator LoRA adapters. Pages are resized to $1{,}344 \times 1{,}344$ pixels for its native processor, and the top three retrieved pages are passed to the generator in retrieval order. We retain the released concise-answer prompt and greedy decoding with a 64-token output limit.

SV-RAG-InternVL2. We evaluate the published R5 configuration of SV-RAG (Chen et al., 2025) with the trained Col-InternVL2-4B retrieval adapter and InternVL2-4B for generation without additional QA fine-tuning. The top five pages are selected, then supplied in corpus order following the released implementation. Each page is divided into up to twenty 448×448 tiles, with a thumbnail added when more than one tile is used. We retain the released QA prompt and use greedy decoding, with an output limit of 1,024 generated tokens per answer.

EVisRAG-7B. EVisRAG-7B (Sun et al., 2026) uses VisRAG-Ret and the EVisRAG-7B checkpoint released on October 2, 2025, based on Qwen2.5-VL-7B. The top three pages are provided in retrieval order, with processor bounds of 3,136–3,920,000 pixels per image. We retain the released evidence-guided prompt, place its text before the images, and use temperature zero, repetition penalty 1.05, and a 2,048-token output limit. We extract text after the first `<answer>` tag, stopping at `</answer>` or the response end. Responses without an opening tag are retained in full.

Additional generators. For Table 3, we use Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-32B-Instruct with their native processors, tokenizers, and chat templates. Both use 14-pixel patches merged 2×2, giving one image VT per 28×28 pixels. ColQwen3 supplies the same top-four pages in retrieval order, smart-resized with per-image bounds of 3,136–1,003,520 pixels. For LoC, each baseline dimension is multiplied by $s = 2.25$ and rounded to the nearest multiple of 28 using Python's `round`; the resulting image VT sum defines the query-specific budget B. OCR text, including its evidence labels, is tokenized first; pages receive the remaining budget through common-scale resizing with dimensions floored to the 28-pixel grid, leaving quantization slack unused. Thus, image VT plus OCR tokens is at most B; question, instruction, chat wrappers, and output tokens are excluded. Images precede OCR and the unchanged QA instruction, with native `Picture 1:` labels and the system message `You are a helpful assistant.` Both models use BF16, vLLM 0.24.0, TP=1, greedy decoding (temperature 0, top-p 1, seed 0), `generation_config=vllm`, a 65,536-token context, and a 3,072-token output limit.

Table 11: **Paired comparisons quantify the end-to-end gains.** Improved and Regressed count non-correct-to-correct and correct-to-non-correct changes, respectively. Changes and 95% confidence intervals are in percentage points. Loud, Clear, and Total denote the comparisons defined above.

Dataset	Comparison	N	Improved	Regressed	Δ	95% CI
SciEGQA	Loud	1623	176	74	+6.29	$[4.37, 8.19]$
	Clear	1623	114	72	+2.59	$[0.99, 4.25]$
	Total	1623	223	79	+8.88	$[6.84, 10.91]$
ViDoRe-v3	Loud	1489	103	65	+2.55	$[0.87, 4.30]$
	Clear	1489	93	85	+0.54	$[-1.21, 2.28]$
	Total	1489	114	68	+3.09	$[1.34, 4.84]$
MMLongBench-Doc	Loud	1091	57	36	+1.93	$[0.18, 3.67]$
	Clear	1091	71	46	+2.29	$[0.37, 4.22]$
	Total	1091	84	38	+4.22	$[2.29, 6.14]$

E Additional Results on End-to-End RAG Systems

E.1 Paired comparisons and uncertainty

The three conditions in Table 2 cover identical query sets, questions, and reference answers within each benchmark. Recomputing accuracy from their saved predictions and final judgments reproduces all nine reported accuracies. No query intersection filtering is needed, and answers reaching the output limit remain included. Unlike the controlled comparisons in Section 3, the end-to-end evaluation includes all 1,489 ViDoRe-v3 queries because it does not require ground-truth bounding boxes.

We compare ColQwen3 with LoC w/o Clear (Loud), LoC w/o Clear with LoC (Clear), and ColQwen3 with LoC (Total). Only `correct` receives credit; `partial` and `incorrect` both receive zero. For each comparison, we estimate a paired percentile bootstrap interval by resampling queries with both conditions' outcomes together. All bootstrap analyses in this appendix use 20,000 resamples, NumPy 2.3.5 with linear percentile interpolation, and fixed comparison-specific seeds derived from the base seed 20260925. Intervals are computed from unrounded query scores; displayed accuracy changes follow the rounding convention in Appendix D.2. These marginal intervals are not corrected for multiple comparisons and describe query-sampling uncertainty conditional on the saved answers and judgments, rather than variation across independent model runs.

The paired query-bootstrap intervals for LoC over ColQwen3 exclude zero on all three benchmarks (Table 11). For Clear, the intervals exclude zero on SciEGQA and MMLongBench-Doc, while the smaller gain on ViDoRe-v3 remains uncertain.

E.2 ViDoRe-v3 domain results

Clear's contribution varies across the five English-language domains (Table 12). Its net gain of eight correct answers combines gains in computer science ($+12$), pharmaceuticals ($+9$), and industrial reports ($+5$) with losses in HR (-12) and finance (-6). The full LoC method has higher observed accuracy than ColQwen3 in all five domains, with computer science, finance, and pharmaceuticals accounting for 44 of the 46 additional correct answers. Domain intervals are exploratory, use paired query resampling, and are not corrected for multiple comparisons.

E.3 Document-cluster sensitivity

We additionally resample whole documents with replacement, retaining all their query pairs and recomputing query-weighted accuracy differences. On SciEGQA (80 documents), the 95% intervals for Loud, Clear, and Total are $[4.28, 8.34]$, $[0.98, 4.15]$, and $[6.55, 11.31]$ percentage points, respectively. On MMLongBench-Doc (135 documents), they are $[0.00, 3.88]$, $[0.18, 4.47]$, and $[2.33, 6.19]$. Thus, the Clear and Total intervals remain above zero on both benchmarks, while the lower bound

Table 12: **Domain results explain the small pooled Clear gain on ViDoRe-v3.** Accuracy is reported in percent; Clear changes and intervals are in percentage points. The pooled row weights each query equally. Displayed changes subtract rounded accuracies, while intervals use unrounded query scores.

Domain	N	ColQwen3	LoC w/o Clear	LoC	Clear Δ	Clear 95% CI
Computer science	215	59.07	59.07	64.65	+5.58	$[0.47, 10.70]$
Finance	309	43.69	51.13	49.19	-1.94	$[-5.83, 1.94]$
HR	318	44.03	48.11	44.34	-3.77	$[-7.86, 0.31]$
Industrial	283	49.47	48.06	49.82	+1.76	$[-1.77, 5.30]$
Pharmaceuticals	364	41.76	43.41	45.88	+2.47	$[-0.82, 5.77]$
Pooled micro	1489	46.61	49.16	49.70	+0.54	$[-1.21, 2.28]$

for Loud on MMLongBench-Doc is approximately zero. We omit document-cluster resampling for ViDoRe-v3 because 130 queries reference multiple documents, so a unique non-overlapping document assignment would require additional assumptions.

Table 13: **Median generation-stage latency increases with the visual token budget in both SciEGQA-only sweeps.** Every row includes all 1,623 queries. Mean used budget counts image visual tokens plus OCR text tokens, excluding the question, instructions, and chat delimiters. K denotes 1,000 tokens. Unlike the target cap used in Figure 5, this column reports actual usage, which can be slightly lower for LoC due to token quantization. Mean output length is included because decoding workload also varies. Reused session records are reported separately in Table 14.

Variant	s	Mean used budget (K)	Mean output tokens	Median (s)	P95 (s)	Acc. (%)
SciEGQA-only sweep sessions						
LoC w/o Clear	1.25	5.959	69.37	34.02	191.71	72.15
LoC w/o Clear	1.50	8.429	56.75	79.59	241.53	73.81
LoC w/o Clear	1.75	11.594	49.83	121.65	324.95	75.11
LoC w/o Clear	2.00	15.190	53.78	205.42	363.99	75.23
LoC w/o Clear	2.50	23.927	39.09	416.45	503.76	75.11
LoC w/o Clear	2.75	28.662	32.42	572.89	646.46	75.17
LoC w/o Clear	3.00	34.177	33.42	745.66	792.53	75.23
LoC	1.00	3.762	82.36	16.44	41.29	65.87
LoC	1.25	5.901	84.43	17.83	137.58	72.70
LoC	1.50	8.354	63.23	56.00	237.05	75.05
LoC	1.75	11.498	58.87	96.75	298.27	75.97
LoC	2.00	15.079	70.98	159.90	418.36	76.65
LoC	2.50	23.776	68.18	343.44	666.16	77.88
LoC	2.75	28.500	70.27	482.30	729.37	78.93
LoC	3.00	33.998	70.99	658.35	1000.64	78.50

F Generation-stage Latency across Visual Token Budgets

The saved SciEGQA runs underlying Figure 5 allow us to examine how generation-stage latency varies with the visual token budget. We report LoC and LoC w/o Clear at nine scales, using the same 1,623 queries in every condition (29,214 query records). Accuracy is computed from the judged predictions of the corresponding timed runs. All conditions have complete timing records and no final generation failures or timeouts; slow queries and answers reaching the output limit remain included.

Measurement scope. The reported latency includes input preprocessing and engine request residence time under concurrent execution. Preprocessing covers image loading, conversion, resizing, and processor/tokenizer operations. Engine residence is measured from enqueueing a request to receiving its completed output, including engine queueing and shared execution. The sum excludes retrieval, region selection, OCR, model loading, producer ready-queue waiting, and some preparation and I/O operations. It therefore measures neither end-to-end RAG latency nor isolated GPU execution time. Tables 13 and 14 report the median and descriptive 95th percentile (P95), computed using linear interpolation.

Execution and comparability. The runs share the recorded GPU model and generation settings, but their workloads differ. Each worker uses one NVIDIA RTX PRO 6000 Blackwell Max-Q Workstation Edition GPU with Qwen3.6-35B-A3B, vLLM 0.24.0, BF16, and tensor parallelism of one. Generation settings follow Appendix D.3; continuous batching permits at most 128 sequences and 16,384 batched tokens, rather than a fixed batch size. Table 13 reports the SciEGQA-only sweeps. Table 14 reports the two reused $s = 2.25$ sessions, each of which processed 4,203 queries across benchmarks in a separate run, together with the reused $s = 1.0$ page baseline. The reused $s = 1.0$ page baseline also comes from a mixed-benchmark session and differs in input preparation and the absence of the explicit answer warm-up used by the other runs. Model loading precedes query timing, and no benchmark queries are excluded as warm-up. Host identity and instantaneous resource load were not recorded, so these are descriptive comparisons of archived runs, not controlled speedup measurements.

Table 14: **Execution differences limit direct latency comparisons across reused sessions.** Every row reports all 1,623 SciEGQA queries from its session. Column definitions follow Table 13. The two $s = 2.25$ variants were run in separate mixed-benchmark sessions. The $s = 1.0$ page baseline also comes from a mixed-benchmark session, with different input preparation and no explicit answer warm-up. These records do not establish a controlled latency ratio between the page baseline and LoC.

Variant	s	Mean used budget (K)	Mean output tokens	Median (s)	P95 (s)	Acc. (%)
Reused $s = 2.25$ sessions (separate run per variant)						
LoC w/o Clear	2.25	19.271	44.28	264.91	378.91	75.48
LoC	2.25	19.137	63.23	246.34	467.44	78.07
Reused page baseline (different preparation and warm-up)						
LoC w/o Clear	1.00	3.797	74.44	6.81	25.90	69.19

Observed tradeoff. Within each dedicated sweep in Table 13, median latency increases monotonically with the visual token budget, while accuracy gains diminish at larger budgets. For LoC w/o Clear, increasing s from 1.75 to 3.0 changes accuracy from 75.11% to 75.23%, while median latency rises from 121.65 to 745.66 seconds. For LoC, increasing s from 2.75 to 3.0 changes accuracy from 78.93% to 78.50%, while median latency rises from 482.30 to 658.35 seconds. Across the seven scales shared by the dedicated sweeps, LoC has lower median latency, but its P95 is higher at $s = 2.0$, 2.5, 2.75, and 3.0. For example, at $s = 3.0$, its median is 658.35 versus 745.66 seconds, whereas its P95 is 1,000.64 versus 792.53 seconds. Thus, the results show an observed accuracy–latency tradeoff without establishing a uniform latency advantage for LoC or a latency-optimal choice of s. Input modality, output length, batching, and queueing also vary across runs; the latency differences cannot be attributed solely to the visual token budget. End-to-end RAG latency and peak memory remain unmeasured.