DOCCOMP: QUERY-CONDITIONED SEMANTIC COMPONENT SELECTION FOR EFFICIENT DOCUMENT VQA

Anonymous Authors

ABSTRACT

Document visual question answering (DocVQA) requires high-resolution inputs to read small text and structured content, but full-page processing turns even question-irrelevant regions into visual tokens. Existing visual-token pruning reduces the sequence only after visual processing, whereas aggressive region selection can lose answer-supporting context, and both can be tied to a specific VLM. To address this, we propose DocComp, a plug-and-play framework for query-conditioned semantic component selection that keeps only question-relevant layout regions before downstream VLM processing. DocComp decomposes a page into semantic layout components and ranks them against the question with a late-interaction visual retriever adapted from page-level to within-page retrieval. Retrieval confidence then sets a question-specific evidence budget in place of a fixed Top-k. Finally, a reconstruction layer serializes the selected evidence for the downstream image processor, giving compatibility across VLM families while the detector and generative VLM remain frozen. Component-level adaptation raises within-page R@1 from 37.78% for the base ColSmol page-level retriever to 92.56%. Across nine frozen backbones, DocComp reduces visual tokens by 62.34–63.26% for Qwen with ANLS changes of -1.9 to $+0.8$ points, and by 33.20% for InternVL with changes of -4.4 to $+3.0$ points, improving three of four models. At similar reductions, it exceeds VScan and VisionZip by 14.14 and 16.92 ANLS points, and, counting its own detection and retrieval, cuts page-amortized compute by 61.04% on the reliable subset. These results support pre-VLM semantic evidence selection as an effective alternative to processing the full page first and pruning afterward.

1 INTRODUCTION

Document visual question answering (DocVQA) requires reading small text and structured content in scanned pages, forms, and reports. Recent vision-language models (VLMs) support this task through high-resolution visual processing (Bai et al., 2025a;b; Zhu et al., 2025; Wang et al., 2024; Chen et al., 2024b; Bai et al., 2023; Mathew et al., 2021). A page can produce thousands of visual tokens even when the question concerns only a few fields, spending visual-encoding and language-model computation on unrelated regions.

This waste is large in practice. On the spatially reliable subset of our evaluation data, Qwen2.5-VL-7B encodes each page into about 1,072 visual tokens, yet an adaptive selector retains the annotated answer in 99.55% of questions while keeping only about two layout components. Existing visual retrievers do not locate these components reliably: a compact page-level retriever ranks the answer-bearing component first for only 37.78% of questions, and 2.2–3.0B-parameter retrievers stay below 74%, motivating adaptation to within-page component ranking.

Current efficiency methods do not exploit this sparsity before encoding. Visual-token compression shortens sequences in later model stages (Xing et al., 2025; Zhang et al., 2025a; Li et al., 2024; Chen et al., 2024a; Zhang et al., 2024; 2025d; Yang et al., 2025), but most methods first compute full-page features and rely on backbone-specific internal signals. Region-first approaches instead restrict what reaches the generator (Kwon et al., 2026; Li et al., 2026; López et al., 2025), yet a selected region must still preserve context such as field names and table headers.

Pre-VLM selection must therefore satisfy three requirements at once: find the answer-bearing regions within a page, keep enough of them to preserve context, and serialize them so that the savings

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.

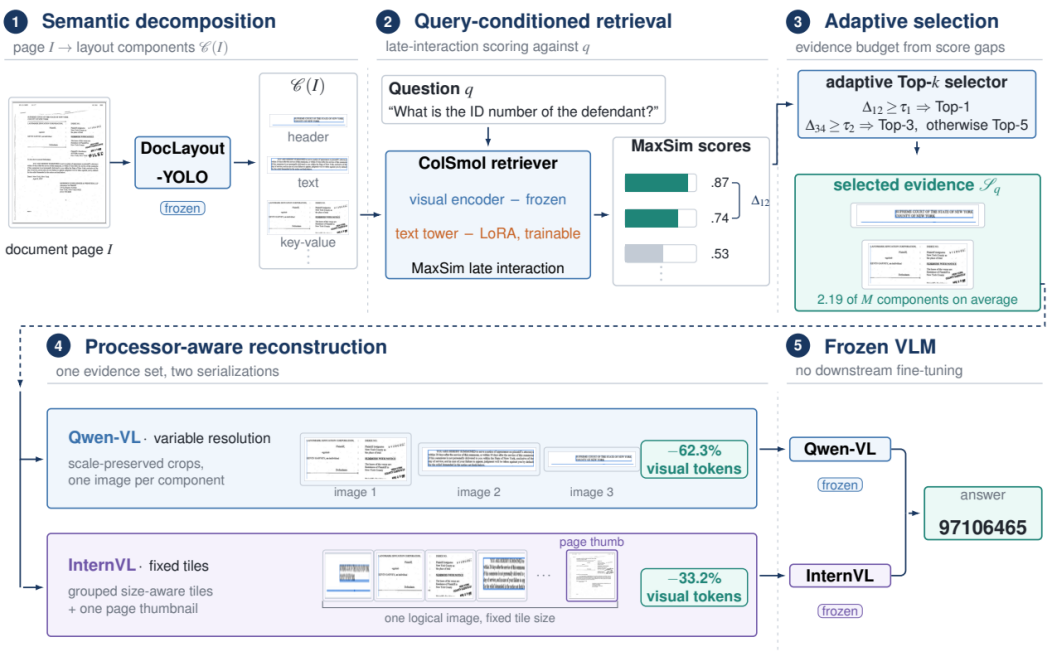


Figure 1: DocComp overview. A layout detector decomposes the page into semantic components; a retriever adapted from page-level to within-page ranking scores them against the question; score gaps set a question-specific evidence budget; and a processor-aware reconstruction layer serializes the selected evidence $\mathcal{S}_q$ for each frozen VLM. Stages 1–4 correspond to Sections 3.2–3.5. Token reductions are aggregate values over the 41,180-example answer-valid set (Table 1).

survive each VLM's image processor, whether variable-resolution Qwen-VL patch grids or fixed-resolution InternVL tiles (Bai et al., 2025a;b; Zhu et al., 2025; Wang et al., 2024; Chen et al., 2024b). DocComp addresses these with a layout-component retriever adapted from page-level retrieval (Faysse et al., 2025; Zhao et al., 2024), a question-specific evidence budget set by retrieval confidence, and processor-aware reconstruction, all without modifying the generative models (Fig. 1).

Across nine frozen Qwen and InternVL backbones, DocComp removes 62–63% of Qwen visual tokens with ANLS changes between -1.9 and $+0.8$ points, and 33% of InternVL tokens while improving three of four models. At similar reductions, it exceeds VScan and VisionZip (Zhang et al., 2025a; Yang et al., 2025) by 14.1 and 16.9 ANLS points.

Our contributions are as follows.

- **Component-level supervision and retrieval adaptation.** We derive training pairs from word-level answer annotations and detector proposals, and fine-tune a page-level visual retriever to discriminate among components on the same page.
- **Plug-and-play selection and reconstruction.** We combine an adaptive component policy with a processor-aware input interface, allowing one trained retriever to serve different VLM families without modifying or fine-tuning their generative backbones.
- **Evaluation of accuracy and computational cost.** We assess answer quality and visual-token usage across nine frozen Qwen and InternVL backbones, analyze retrieval and reconstruction behavior, and compare with post-hoc compression while accounting for detection and retrieval overhead.

2 Related Work

Document visual question answering and high-resolution VLMs. DocVQA spans single- and multi-page inputs (Mathew et al., 2021; Tito et al., 2023). LayoutLMv3 combines OCR, layout, and image features (Huang et al., 2022), while Donut and Pix2Struct generate text directly from pixels (Kim et al., 2022; Lee et al., 2023). Dynamic-resolution VLMs preserve small text by tiling or natively encoding high-resolution pages (Bai et al., 2025b; Zhu et al., 2025), so the visual-token count grows with page area rather than with what the question requires.

Document-specific token reduction. Several document models reduce this cost inside their own architecture. TextMonkey resamples redundant high-resolution tokens (Liu et al., 2024), DocKylin removes low-information pixels and filters redundant tokens (Zhang et al., 2025c), and mPLUG-DocOwl2 compresses each page into 324 tokens guided by a low-resolution global view (Hu et al., 2025). These compressors are trained with a specific language model and compress the page independently of the question. DocComp keeps the downstream VLM frozen and decides, for each question, which regions to encode.

Post-hoc visual-token compression. TokenPacker compresses projector features (Li et al., 2024); FastV, SparseVLM, and PyramidDrop prune internal tokens (Chen et al., 2024a; Zhang et al., 2025d; Xing et al., 2025); FasterVLM and VisionZip reduce post-encoder representations (Zhang et al., 2024; Yang et al., 2025); and VScan combines visual-stage reduction with optional language-layer pruning (Zhang et al., 2025a). These methods require little or no training, but most prune after the full page has been encoded, whereas DocComp selects evidence before downstream encoding.

Question-conditioned region and evidence selection. ViCrop crops the region highlighted by an MLLM's own attention or gradients and adds it to the full-image input (Zhang et al., 2025b), and Visual CoT trains models to predict a question-relevant box before answering (Shao et al., 2024). PinPoint learns instruction–region alignment for information-rich VQA (Kwon et al., 2026), retrieval-augmented DocVQA selects compact evidence sets (López et al., 2025), and RegionRAG retrieves semantic regions formed from salient patches (Li et al., 2026). Zooming methods add views to the full page or retrain the answering model; DocComp replaces the page with detected components, using component supervision and an adaptive evidence budget.

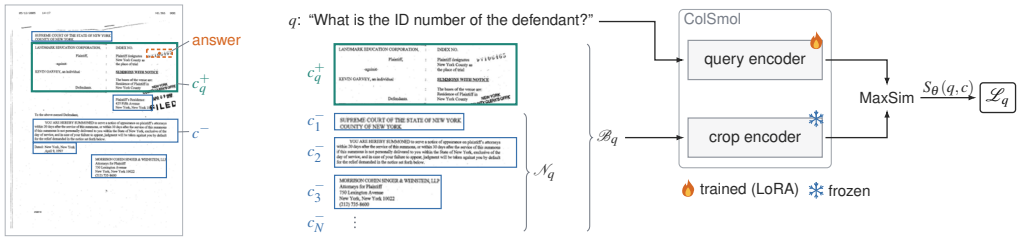
Visual document retrieval and semantic components. ColBERT introduced contextual late interaction (Khattab and Zaharia, 2020), and ColPali extended it to document images together with the ViDoRe page-retrieval benchmark (Faysse et al., 2025). VisRAG and M3DocRAG use such page retrievers for multi-page and multi-document QA (Yu et al., 2024; Cho et al., 2024), but their retrieval unit remains the page. Layout detectors expose coherent paragraphs, tables, figures, and headers (Zhao et al., 2024; Shen et al., 2021; Zhong et al., 2019). DocComp adapts page retrieval to these components and serializes the selected regions for variable-resolution or fixed-tile processors.

3 Method

3.1 Framework Overview

DocComp trains a component retriever, then uses it with frozen downstream VLMs (Fig. 1). Answer boxes supervise text-side LoRA during training. At inference, detector proposals are ranked against the question, selected adaptively, and reconstructed for the target processor; no answer annotations are used.

For page I, question q, and frozen VLM F_ϕ, let $\mathcal{C}(I)$ be the component candidates and $\mathcal{S}_q \subseteq \mathcal{C}(I)$ the selected subset. Reconstruction G_p produces $\widetilde{\mathcal{I}}_q = G_p(\mathcal{S}_q, I)$, and the VLM predicts $\hat{a} = F_\phi(\widetilde{\mathcal{I}}_q, q)$. Selection is shared across downstream models. The reconstruction interface handles variable-resolution patch grids and fixed-resolution tile groups. We measure the resulting token cost rather than assuming it is proportional to crop area.



(a) answer-box assignment (b) query-local group (c) retriever adaptation

Figure 2: Component-level supervision (boxes illustrative). (a) Answer boxes select the positive c_q^+. (b) Up to 32 same-page components that do not overlap the answer form $\mathcal{N}_q$. (c) Only text-side LoRA is trained (Eq. 4); answer boxes are not used at inference.

3.2 Component-Level Supervision Construction

We build query–component training groups by assigning existing word-level answer annotations to detected layout regions (Fig. 2). This converts spatial QA annotations into supervision over the same type of candidates used at inference.

Full-page inputs are too coarse for selective processing, whereas arbitrary patches fragment tables, key–value groups, and reading order. We therefore use DocLayout-YOLO (Zhao et al., 2024) to decompose I into M candidate components

$$\mathcal{C}(I) = \{c_i = (I[b_i], b_i, t_i)\}_{i=1}^{M}, \tag{1}$$

where $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$ is a bounding box, $I[b_i]$ is its image crop, and t_i is the detector's layout category. Components correspond to coherent units such as paragraphs, tables, figures, and headers. Each box is expanded by a fixed context margin and clipped to the page, reducing boundary errors such as separated table headers. Inference uses detector candidates only, without OCR, document parsing, or task-specific proposals.

Relevance is question-specific rather than determined by layout category. Answer annotations label visual crops; they do not replace pixels with recognized text.

A positive component must cover at least 80% of the answer anchors and contain every answer-box center. The coverage criterion requires substantial overlap with the answer, while the center condition prevents a proposal from qualifying when an annotated word center lies outside it. Each training row contains one qualifying positive. Ground-truth fallback crops are disabled, so supervision does not introduce answer crops unavailable from the detector at inference. These labels indicate answer-containing regions, not an exhaustive set of supporting evidence. A neighboring heading or another component may still be needed to interpret the answer.

For each training row, let c_q^+ be the positive and $\mathcal{N}_q$ contain up to 32 same-page negatives. Answer-overlapping components are excluded from the negative set to avoid assigning them contradictory supervision. The resulting group is

$$\mathcal{B}_q = \{c_q^+\} \cup \mathcal{N}_q, \qquad |\mathcal{N}_q| \le 32. \tag{2}$$

Same-page negatives share document style and subject, requiring discrimination among locally similar fields rather than document topics.

Dataset sizes and split checks appear in Section 4.1; annotation curation is described in Appendix A.

3.3 Adapting ColSmol for Within-Page Retrieval

Page retrieval ranks documents that may differ in topic and appearance. Within-page retrieval instead compares regions from the same document, including visually similar fields with different meanings. We retain ColSmol's late-interaction architecture and adapt its scoring function to these query-local candidate groups (Faysse et al., 2025). For question q and component crop c_i, the model produces token-level query embeddings $Q \in \mathbb{R}^{L_q \times d}$ and visual embeddings $V_i \in \mathbb{R}^{L_i \times d}$. Token-wise L2

normalization gives the late-interaction score

$$s_i = S_\theta(q, c_i) = \sum_{j=1}^{L_q} \max_{k \in \{1,\dots,L_i\}} \widehat{Q}_j^\top \widehat{V}_{i,k}. \tag{3}$$

MaxSim sums each query token's strongest visual match. We retain this scoring rule and adapt the representations through component supervision.

We fine-tune ColSmol with text-side low-rank adaptation (Hu et al., 2022). The objective increases the positive component's score relative to the other candidates in its query-local group,

$$\mathcal{L}_q = \log \sum_{c \in \mathcal{B}_q} \exp\left(\frac{S_\theta(q, c)}{\tau}\right) - \frac{S_\theta(q, c_q^+)}{\tau}, \qquad \tau = 1, \tag{4}$$

where the batch loss averages $\mathcal{L}_q$. Negatives come only from the assigned query-local group, without cross-query or model-mined negatives.

Only text-side LoRA is optimized, adding 17.37M parameters to the 460.30M-parameter retriever. Base weights, the detector, and generative VLMs remain frozen. Training uses retrieval supervision rather than answer-generation loss, allowing the same checkpoint to serve all downstream backbones.

3.4 Query-Adaptive Component Selection

The ranked scores identify likely evidence, but they do not determine how much evidence a question requires. A fixed top-k policy imposes the same evidence budget on every question. A large fixed budget can retain unnecessary regions for simple questions, while a small budget can omit context needed across multiple regions. DocComp instead applies a frozen Top-1/3/5 policy to the descending raw MaxSim scores. Let $\Delta_{12} = s_{(1)} - s_{(2)}$ and $\Delta_{34} = s_{(3)} - s_{(4)}$, with a gap set to $+\infty$ when the lower-ranked score required to compute it does not exist. The requested component count is

$$K_q = \begin{cases} 1, & \Delta_{12} \geq 4.0, \\ 3, & \Delta_{12} < 4.0 \text{ and } \Delta_{34} \geq 1.5, \\ 5, & \text{otherwise.} \end{cases} \tag{5}$$

We then take $k_q = \min(K_q, M)$ and $\mathcal{S}_q = \{c_{(1)}, \dots, c_{(k_q)}\}$. The thresholds are selected on validation and frozen across test examples and downstream VLMs. Candidate shortage may yield Top-2 or Top-4.

Fixed-budget comparisons and evidence-retention results are reported in Section 5.3.

3.5 Plug-and-Play VLM Integration

Reconstruction converts the shared selection into processor-compatible inputs without accessing VLM attention or hidden states.

The interface accounts for how each processor converts crops into visual tokens. For variable-resolution processors, selected components are supplied as separate images at approximately the full-page character scale. For fixed-tile processors, component tiles are grouped into one logical image with an optional full-page thumbnail. These operations adapt the input format while keeping component selection unchanged. The evaluated operating points are specified in Section 4.3, and geometric construction details are given in Appendix C.

4 Experimental Setup

4.1 Data and Spatial Annotation Curation

We use BoundingDocs, a multi-source DocVQA collection (Giovannini et al., 2025). Answer boxes supervise component assignment during training and validation but are never supplied to the downstream VLM.

OCR string matching can produce false or misaligned answer boxes. We curate a 35,349-example spatially reliable subset from the 41,180 answer-valid examples for retrieval, evidence retention, and the controlled adaptation ablation (Appendix A). Pipeline profiling (Section 5.7) also uses this subset. Main QA and token results use all 41,180 examples, with questions and textual answers unchanged.

The retriever manifests contain 479,947 training and 60,278 validation rows. Audited train, validation, and reliable-test artifacts have zero overlap at the question, QA–page, page, or document levels.

4.2 Models and Training

We evaluate five Qwen backbones (Qwen2.5-VL-3B/7B and Qwen3-VL-2B/4B/8B) (Bai et al., 2025a;b) and four InternVL backbones (InternVL2.5-2B/8B and InternVL3-2B/8B), with development ablations on InternVL3-2B (Zhu et al., 2025; Chen et al., 2024b). All use native chat templates and deterministic generation without fine-tuning. ColSmol query and visual embeddings are converted to FP32 before L2 normalization and MaxSim scoring.

4.3 Compared Settings

Full-page processes the original page using the downstream model's native processor. Qwen uses DocComp-SP, which supplies scale-preserved component images. InternVL uses DocComp-Eff. (One-Tile+Thumb), which normalizes each selected component to one tile, and DocComp-Bal. (Hybrid-T2+Thumb), which retains native-scale tiles for larger components. Both InternVL settings append a full-page thumbnail to the same logical image group.

For the balanced setting, let n_i be the native tile count of selected component c_i. Components with $n_i \leq T$ are normalized to one tile, while larger components retain their native tiles. We choose $T = 2$ on a fixed 2,000-example development analysis and freeze it for all full-scale InternVL runs. The selector score-gap thresholds remain 4.0 and 1.5 across downstream models. Fixed-budget comparisons appear in Fig. 3.

4.4 Evaluation Metrics and Computational Accounting

All methods use the same question prompts and generation settings within each downstream backbone. Answer quality is measured with average normalized Levenshtein similarity (ANLS) (Mathew et al., 2021). Main answer-quality and processor-token results use all 41,180 answer-valid examples with exact-ID joins. Spatial retrieval and evidence-retention metrics use the 35,349-example reliable subset, with R@1/3/5, MRR, and 10,000 paired bootstrap replicates.

Visual-token reduction is computed from the processor outputs as

$$R_{\mathrm{VT}} = 1 - \frac{\sum_{n=1}^{N} \mathrm{VT}(\widetilde{\mathcal{I}}_{q_n})}{\sum_{n=1}^{N} \mathrm{VT}(I_n)}. \tag{6}$$

This aggregate ratio accounts for differences in page resolution. Reconstruction variants share detector outputs, retriever scores, selected components, and component order.

The post-hoc baseline comparison uses the same 41,180 answer-valid IDs for Full-page, DocComp, VScan, and VisionZip with Qwen2.5-VL-7B. Each method keeps the compression setting it uses on the reliable subset, so visual-token reductions are similar rather than identical. Full-set FLOPs are analytical estimates. DocComp's full-set estimate covers VLM inference only and excludes detection and retrieval, while baseline estimates exclude pruning, scoring, selection, and merging overhead. I/O and CPU preprocessing are excluded for all methods, and the analytical estimates omit patch embedding and merger/projector operations. Separately, pipeline profiling on the 35,349-example reliable subset includes detection, component encoding, retrieval, and VLM inference under uncached, cached, and page-amortized accounting. Appendix B distinguishes these cost scopes.

Table 1: Source-wise ANLS and visual-token reduction on the complete 41,180-example answer-valid evaluation set. Qwen models use DocComp-SP. For InternVL, DocComp-Eff. denotes One-Tile+Thumb and DocComp-Bal. denotes Hybrid-T2+Thumb. VT↓ is computed on the same examples and shared within each processor generation or policy. Bold is reserved for visual-token reduction.

Model	Method	Overall	DeepForm	DUDE	FATURA	FUNSD	Kleister Charity	Kleister NDA	MP-VQA	SP-VQA	VRDU Ad Buy	VRDU Reg. Form	XFUND
N		41,180	13,362	525	10,243	163	3,760	134	8,608	40	2,317	394	1,634
Qwen-VL family													
Qwen2.5-3B	Full-page	.7677	.7911	.7860	.7382	.8107	.7057	.3362	.8151	.9149	.7850	.7979	.6452
	DocComp-SP	.7502	.7554	.7441	.7420	.8298	.7283	.3253	.7873	.8885	.8220	.8708	.5086
Qwen2.5-7B	Full-page	.8165	.8297	.8126	.7803	.8462	.8527	.3471	.8257	.9241	.8421	.8250	.7997
	DocComp-SP	.8249	.8587	.7604	.8034	.8400	.8100	.3423	.8040	.8991	.9068	.8787	.7558
Qwen2.5-VL VT↓ vs. Full-page (%)		**63.26**	**56.32**	**61.34**	**91.33**	**47.01**	**75.77**	**67.38**	**57.13**	**62.20**	**48.41**	**70.75**	**54.67**
Qwen3-2B	Full-page	.7474	.7007	.7974	.7419	.8330	.7943	.3134	.8071	.9126	.7685	.7933	.7071
	DocComp-SP	.7284	.6802	.7313	.7406	.8063	.7674	.2946	.7719	.8329	.7705	.8816	.6538
Qwen3-4B	Full-page	.7675	.7299	.8094	.7895	.8108	.7047	.2428	.8098	.9207	.8524	.7798	.7572
	DocComp-SP	.7585	.7261	.7496	.7672	.7999	.7363	.3320	.7911	.8887	.8824	.8636	.6790
Qwen3-8B	Full-page	.8016	.7836	.8061	.7873	.8509	.8107	.2315	.8340	.9314	.8988	.7783	.7506
	DocComp-SP	.7991	.8177	.7504	.7684	.8394	.8107	.2383	.8007	.8568	.9203	.9055	.6624
Qwen3-VL VT↓ vs. Full-page (%)		**62.34**	**56.32**	**61.35**	**90.97**	**46.93**	**75.76**	**67.26**	**57.18**	**62.12**	**48.41**	**70.79**	**54.67**
InternVL family													
InternVL2.5-2B	Full-page	.6399	.4750	.7527	.6706	.7942	.6288	.3078	.8244	.9573	.7370	.7906	.6440
	DocComp-Eff.	.5468	.3524	.5734	.6480	.6880	.5292	.2219	.7783	.8044	.3949	.7723	.4808
	DocComp-Bal.	.5956	.4135	.6442	.6491	.7551	.5241	.2395	.8112	.9057	.7089	.8626	.5444
InternVL2.5-8B	Full-page	.7577	.6857	.7871	.7706	.8328	.7898	.3822	.8314	.9261	.8355	.7843	.6962
	DocComp-Eff.	.6991	.6214	.7322	.7376	.7596	.7496	.2724	.7917	.8435	.5937	.7962	.6301
	DocComp-Bal.	.7872	.8109	.7696	.7377	.8145	.7507	.2750	.8201	.8747	.8797	.8636	.7081
InternVL3-2B	Full-page	.7497	.7314	.7756	.7370	.8416	.7336	.4070	.8122	.9390	.7434	.8032	.6886
	DocComp-Eff.	.7158	.6718	.6925	.7821	.7751	.7374	.2888	.7623	.8662	.5514	.8108	.6083
	DocComp-Bal.	.7679	.7607	.7468	.7816	.8579	.7332	.2801	.7908	.9417	.7919	.8658	.6764
InternVL3-8B	Full-page	.8215	.8479	.7834	.7825	.8521	.8373	.4144	.8278	.9598	.9026	.7561	.7213
	DocComp-Eff.	.7887	.7922	.7425	.8092	.8002	.8583	.2988	.7876	.8502	.6914	.7714	.6701
	DocComp-Bal.	.8356	.8677	.7748	.8094	.8303	.8556	.2872	.8181	.9029	.9140	.7950	.7446
DocComp-Eff. VT↓ vs. Full-page (%)		**69.27**	**73.75**	**65.33**	**67.99**	**71.17**	**50.42**	**31.56**	**68.38**	**67.55**	**79.75**	**78.16**	**42.35**
DocComp-Bal. VT↓ vs. Full-page (%)		**33.20**	**25.40**	**29.99**	**67.41**	**15.05**	**35.38**	**27.61**	**25.62**	**45.92**	**27.03**	**46.40**	**18.46**

Table 2: Within-page component retrieval (35,349 reliable examples). Adapt.: component-level adaptation. Params include the LoRA adapter; metrics in %; bold is best.

Retriever	Adapt.	Params	R@1	R@3	R@5	MRR
Base ColSmol	No	460.3M	37.78	66.27	80.70	55.88
DocComp	Yes	477.7M	**92.56**	**99.17**	**99.67**	**95.90**
ColPali	No	2.964B	73.58	89.24	95.42	82.53
ColQwen2	No	2.246B	71.19	89.70	95.22	81.32

Table 3: Adaptation effect with frozen Qwen2.5-VL-7B (35,349 examples, Top-3, SP; VT: visual tokens). Bold marks the best answer scores among compressed inputs.

Input / retriever	ANLS	EM	F1	Mean VT	VT↓
Full-page	.8175	.7335	.8297	1,071.69	–
Base ColSmol, Top-3	.5898	.5060	.5797	410.05	61.74%
DocComp, Top-3	**.8179**	**.7206**	**.8199**	498.74	53.46%

5 Results and Analysis

5.1 Component Retrieval and Evidence Coverage

On identical candidates and positives, DocComp raises R@1 from 37.78% to 92.56% and exceeds the larger ColPali and ColQwen2 models by 18.98 and 21.37 points (Table 2). The larger models use official processors and MaxSim without component adaptation. This tests adapted compact retrieval against off-the-shelf larger models, not equally adapted alternatives.

All paired document-cluster bootstrap 95% intervals for the retrieval gains exclude zero (Appendix E). Candidate coverage is 100% on this curated subset, so the comparison measures ranking, not general detector recall.

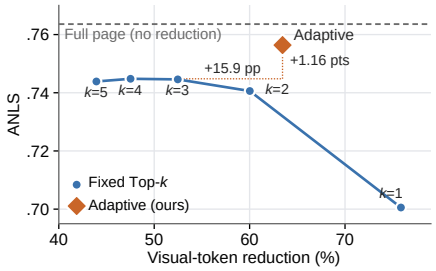


Figure 3: Fixed Top-k vs. adaptive selection with Qwen3-VL-4B; dotted guides mark the gain over Top-4.

Table 4: Qwen2.5-VL-7B on all 41,180 answer-valid examples at similar visual-token reductions. Bold: best ANLS among compressed inputs.

Method	ANLS	VT↓	TFLOPs/QA	FLOPs↓
Full-page	.8165	–	22.9275	–
VScan†	.6835	64.44%	13.3009	41.99%
VisionZip†	.6557	65.02%	13.2161	42.36%
DocComp-SP‡	**.8249**	63.26%	8.1323	64.53%

FLOPs are analytical estimates with different exclusions. †Excludes baseline pruning, scoring, selection, and merging overhead. ‡DocComp VLM-only cost, excluding detection and retrieval. I/O and CPU preprocessing are excluded throughout.

5.2 Effect of Component-Level Adaptation

With frozen Qwen2.5-VL-7B on the reliable subset, we fix detector candidates, the Top-3 policy, SP reconstruction, prompts, and decoding. Both retrievers select all candidates when fewer than three exist. Selection cardinality, rather than visual-token budget, is controlled.

DocComp improves ANLS by 22.81 points (95% document-bootstrap CI $[21.87, 23.71]$), recovering the observed Full-page ANLS at 53.46% fewer tokens (Table 3). EM and F1 remain lower by 1.28 and 0.97 points. DocComp uses 21.63% more tokens than Base, so this ablation does not isolate adaptation from selected region size. Audit details appear in Appendix D.

5.3 Adaptive Evidence Budgets

On the spatially reliable evaluation set, the adaptive selector retains an annotated evidence component in 99.55% of examples while selecting 2.19 components on average. This measures retention of annotated answer locations, not exhaustive supporting-evidence coverage. With Qwen3-VL-4B, adaptive selection reaches 0.7564 ANLS at 63.45% token reduction and Pareto-dominates fixed Top-2–5. Versus Top-4, it gains 1.16 points (page-cluster bootstrap 95% CI $[0.91, 1.41]$) while removing 15.93 additional percentage points of tokens (Fig. 3).

5.4 Answer Quality with Qwen

Table 1 reports Qwen results on all 41,180 answer-valid examples. DocComp-SP reduces visual tokens by 62.34–63.26% overall and 46.93–91.33% across sources. Qwen2.5-VL-7B improves by $+0.84$ ANLS points and the remaining four change by -0.25 to -1.90. The gains vary by source. VRDU Reg. Form improves for every Qwen backbone, while DUDE, MP-VQA, and XFUND decline for all five. FATURA is mixed despite the largest token reduction.

5.5 Reconstruction and Answer Quality with InternVL

DocComp-Eff. reduces InternVL tokens by 69.27% but loses 3.29–9.32 ANLS points. At 33.20% reduction, DocComp-Bal. improves InternVL2.5-8B and InternVL3-2B/8B by 2.96, 1.82, and 1.40 points, while InternVL2.5-2B drops 4.43 points (Table 1). The gap between the two settings is the cost of forcing every component into one tile: grouping already avoids per-component thumbnails, but only the hybrid policy preserves native tiles for large regions.

InternVL2.5-8B's micro gain is concentrated in DeepForm ($+12.53$), leaving its source macro-average essentially unchanged (-0.15). InternVL3-2B improves on both micro and macro ANLS, and InternVL3-8B improves micro ANLS with a -0.78-point macro change. Kleister NDA shows the largest consistent InternVL drop (-6.8 to -12.7 points on all four backbones); with $N = 134$ and the lowest absolute ANLS of any source, it is both small and hard for every model.

5.6 Variation across Document Sources

Across all nine backbones, the representative DocComp setting improves VRDU registration forms but reduces DUDE and MP-VQA ANLS. XFUND declines for all Qwen models and two InternVL models. These differences caution against uniform accuracy claims; source scores alone cannot establish whether lost context causes the regressions.

5.7 Post-Hoc Baselines and Pipeline Efficiency

At similar token reductions on all 41,180 examples, DocComp-SP exceeds VScan and VisionZip by 14.14 and 16.92 ANLS points while retaining slightly more tokens (Table 4). The analytical FLOPs estimates exclude baseline compression overhead and DocComp detection/retrieval, so they are not end-to-end comparisons.

Pipeline profiling on the reliable subset includes detection and retrieval, yielding 52.58%, 65.55%, and 61.04% compute reductions under uncached, cached, and page-amortized accounting. Cached and amortized latency improve by 61.58% and 29.33%, but uncached execution is 4.75% slower than Full-page. On the same subset, VScan and VisionZip reduce FLOPs by at most 42.03% and 42.40% even before their pruning overhead is counted, so DocComp's page-amortized advantage holds after charging it for its own front-end. Measured speedups nevertheless require preprocessing reuse, and these measurements have not been extended to the full set (Appendix B).

6 Conclusion

We presented DocComp, which moves visual-token reduction in front of the VLM by treating it as within-page retrieval over semantic layout components. Component-level supervision raises within-page R@1 from 37.78% for the base retriever to 92.56%, the adaptive budget retains annotated evidence in 99.55% of examples with about two components, and processor-aware reconstruction turns the same selection into 62–63% fewer Qwen tokens and 33% fewer InternVL tokens without modifying the frozen VLMs. At similar reductions, DocComp outperforms post-hoc pruning by 14–17 ANLS points. For document VQA, deciding which evidence to encode appears to be a stronger lever than pruning what has already been encoded. Practical latency gains depend on reusing page-level detection and component features.

Disclosure of LLM Usage

We used large language models (LLMs) to assist with manuscript organization and language editing, literature search, and implementing parts of the experimental code. LLM-suggested references were checked against the original papers, and LLM-assisted code was reviewed and tested by the authors. Every reported number comes from the authors' evaluation outputs; LLMs did not generate, alter, or estimate experimental results. The authors take full responsibility for the contents of this submission.

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

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 *ECCV*, 2024a.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. M3DocRAG: Multimodal retrieval is what you need for multi-page multi-document understanding. *arXiv preprint arXiv:2411.04952*, 2024.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. ColPali: Efficient document retrieval with vision language models. In *ICLR*, 2025.

Simone Giovannini, Fabio Coppini, Andrea Gemelli, and Simone Marinai. BoundingDocs: a unified dataset for document question answering with spatial annotations. *arXiv preprint arXiv:2501.03403*, 2025.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mPLUG-DocOwl2: High-resolution compressing for OCR-free multi-page document understanding. In *ACL*, 2025.

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

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for document AI with unified text and image masking. In *ACM Multimedia*, 2022.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *SIGIR*, 2020.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-free document understanding transformer. In *ECCV*, 2022.

Mincheol Kwon, Minseung Lee, Seonga Choi, Miso Choi, Kyeongjin Oh, Hyunyoung Lee, Cheonyoung Park, Yongho Song, Seunghyun Park, and Jinkyu Kim. Focus, don't prune: Identifying instruction-relevant regions for information-rich image understanding. In *CVPR*, pages 31900–31909, 2026.

Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *ICML*, 2023.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient visual projector for multimodal LLM. *arXiv preprint arXiv:2407.02392*, 2024.

Yinglu Li, Zhiying Lu, Zhihang Liu, Yiwei Sun, Chuanbin Liu, and Hongtao Xie. RegionRAG: Region-level retrieval-augmented generation for visual document understanding. In *AAAI*, volume 40, pages 6662–6670, 2026. doi: 10.1609/aaai.v40i8.37597.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. TextMonkey: An OCR-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024.

Eric López, Artemis Llabrés, and Ernest Valveny. Enhancing document VQA models via retrieval-augmented generation. *arXiv preprint arXiv:2508.18984*, 2025.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *WACV*, 2021.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *NeurIPS*, 2024.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. LayoutParser: A unified toolkit for deep learning based document image analysis. In *ICDAR*, 2021.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multi-page DocVQA. *Pattern Recognition*, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, et al. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *CVPR*, 2025.

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 *CVPR*, pages 19792–19802, 2025.

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. *arXiv preprint arXiv:2410.10594*, 2024.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Yaqi Xie, Katia Sycara, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *arXiv preprint arXiv:2505.22654*, 2025a.

Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. In *ICLR*, 2025b.

Jiaxin Zhang, Wentao Yang, Songxuan Lai, Zecheng Xie, and Lianwen Jin. DocKylin: A large multimodal model for visual document understanding with efficient visual slimming. In *AAAI*, 2025c.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, MinQi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [CLS] attention is all you need for training-free visual token pruning: Make VLM inference faster. *arXiv preprint arXiv:2412.01818*, 2024.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *ICML*, 2025d.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. DocLayout-YOLO: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*, 2024.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. PubLayNet: Largest dataset ever for document layout analysis. In *ICDAR*, 2019.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Dataset Curation and Retriever Training Details

BoundingDocs spatial curation. BoundingDocs localizes answers by comparing each ground-truth string with Amazon Textract OCR lines using Jaccard similarity and then retaining the word-level OCR boxes that reconstruct the answer. When the same string occurs multiple times, all matched occurrences may be stored as valid locations (Giovannini et al., 2025). This automatic process can yield false matches for repeated values and incomplete or misaligned boxes under OCR errors, which motivates the 35,349-example spatially reliable subset used for the grounding analyses in the main paper. The 479,947 training and 60,278 validation retriever rows cover 295,905 and 37,255 unique source-qualified QA–page keys.

The component assignment rules and grouped contrastive objective are described in Sections 3.2 and 3.3. Training examples use detector proposals, and ground-truth boxes provide supervision only.

B Evaluation Protocol and Profiling Details

Base and adapted rankings have exact-ID parity, no candidate mismatch, and no train/validation document overlap with the reliable test set. Retrieval improvements use 10,000 paired bootstrap replicates. VisionZip uses 35% retention, selected on a fixed 2,000-example development subset solely by budget matching against DocComp-SP. FLOPs use cached token counts and actual output lengths (1 MAC = 2 FLOPs). For pipeline profiling on the 35,349-example reliable subset, DocComp accounting includes detection, component encoding, retrieval, and VLM inference under uncached, cached, and page-amortized settings over 12,260 pages (2.883 QA/page). Page-amortized DocComp-SP uses 8.7802 TFLOPs/QA against 22.5337 for Full-page. Baseline FLOPs exclude pruning or merging overhead and are therefore lower bounds. Measured wall-clock latency depends on preprocessing reuse. Cached and page-amortized latency fall by 61.58% and 29.33% relative to Full-page, while uncached latency is 4.75% slower because detection and component encoding are not yet reused across questions.

Full-set baseline extension. Full-set baseline predictions combine the 35,349 reliable-subset predictions with inference on the remaining 5,831 IDs under unchanged compression settings. Each merged baseline covers all 41,180 IDs exactly once, with no duplicate, missing, or foreign IDs and no inference failures. Table 4 uses full-set token totals and analytical FLOPs rather than subset estimates; the DocComp entry excludes detector and selector cost. The pipeline and latency results above are reported on the reliable subset.

Fixed-tile canvas construction. Let the Full InternVL processor choose a grid of $C \times R$ tiles with native side length s. We reproduce the corresponding Full-grid canvas of size (Cs, Rs), project each selected component box into this canvas, crop the component at the Full-page scale, and pad only its right and bottom boundaries to multiples of s. The padded crop is then split into $s \times s$ tiles in row-major order and concatenated across components in document reading order.

Reconstruction pilot. On a fixed 2,000-example Qwen3-VL-4B pilot, supplying the selected components as separate scale-preserved images outperforms compositing them onto a joint canvas (0.7287 vs. 0.7101 ANLS).

C Processor-Specific Reconstruction Details

The following input constructions implement the plug-and-play interface of Section 3.5. They do not alter the selected component set.

C.1 Variable-Resolution Multi-Image Reconstruction

A single-image variable-resolution processor serializes each image as an aspect-ratio-preserving patch grid under a pixel budget. When component crops are submitted independently, per-image resizing constraints can enlarge small regions and alter their character scale. DocComp therefore preserves the scale chosen for the original full page. If the processor maps (H, W) to $(\bar{H}, \bar{W})$, the page-level scale is

$$\alpha_I = \bar{W}/W \approx \bar{H}/H. \tag{7}$$

Each component is resized once by α_I, padded—rather than stretched—to the nearest model-specific vision factor, and passed with further geometric resizing disabled. Components are supplied as separate images in reading order, preserving approximately the full-page character scale without per-crop minimum-pixel inflation. We instantiate this regime with Qwen-VL and denote the resulting operating mode **DocComp-SP**.

C.2 Fixed-Tile Grouped Reconstruction

A fixed-tile processor allocates visual tokens through discrete square-tile grids, often with a separate global thumbnail. Submitting each selected component independently can therefore create a new grid and thumbnail for every crop, making token cost depend on tile boundaries rather than only on crop area. DocComp instead groups selected evidence within one logical image. For InternVL, we reproduce the canvas that the Full processor's dynamic grid of fixed-size square tiles would produce, crop each selected component from it at the Full-page scale, pad the crop to whole tiles, and split it in row-major order (Appendix B).

DocComp concatenates all component tiles in document reading order and assigns them to one image placeholder. A one-tile thumbnail of the original full page can be appended as the final tile of the same logical image group. Component tiles preserve local evidence, while the thumbnail supplies coarse global context without reintroducing Full local tiles.

C.3 Hybrid Tile Allocation

Let n_i denote the scale-preserved tile count of component c_i. The hybrid policy uses $\widetilde{n}_i = 1$ when $n_i \leq T$ and $\widetilde{n}_i = n_i$ otherwise. Small components are normalized to one native tile, and larger components retain their scale-preserved tiles. Tiles are concatenated in reading order and followed by the full-page thumbnail. The choice of T and named evaluation settings are specified in Section 4.3.

D Downstream Adaptation Ablation Details

The comparison in Table 3 uses an exact join of 35,349 reliable QA IDs. There are no duplicate, missing, or extra IDs and no generation failures for either retriever. Both have 560 examples with fewer than three candidates and select an average of 2.9796 components. For 6,672 examples the adaptive selector already chose exactly three components, so their predictions are reused from the adaptive run; the remaining 28,677 were inferred with the fixed Top-3 policy.

For the ANLS difference (DocComp minus Base), 10,000 paired sample-bootstrap resamples give a 95% confidence interval of $[22.35, 23.27]$ percentage points (seed 20260921). A paired document-cluster bootstrap with 10,000 resamples over 4,727 document clusters gives $[21.87, 23.71]$ points (seed 20260922). These intervals test the adapted-versus-base difference, not equivalence to Full-page. Under normalized exact match, 8,435 examples change from incorrect with Base to correct with DocComp, while 848 change in the opposite direction. DocComp uses 88.68 more visual tokens per example than Base. Both token counts and all three answer metrics are therefore reported alongside the shared Top-3 policy.

E Retrieval Baselines and Statistical Analysis

We evaluate Base ColSmol, DocComp, ColPali v1.3, and ColQwen2 v1.0 on identical within-page candidates. A candidate is positive when its maximum answer-region coverage is at least 0.8. R@k counts any positive among the first k candidates, and reciprocal rank uses the first positive. When fewer than k candidates exist, all available candidates are considered. Scores are sorted in descending order with candidate-index tie breaking. All four models have 35,349 unique evaluation IDs with no duplicate, missing, or extra IDs and no candidate or positive-assignment mismatches. No encoding or scoring failures occurred.

Base ColSmol has 460,296,512 parameters. DocComp adds 17,367,040 adapter parameters for a total of 477,663,552 including the adapter. This count does not imply a measured latency or FLOPs advantage. The base revision is `650243e9bf299a5a082841ed2907da8b0b9ce553`. ColPali uses https://huggingface.co/vidore/colpali-v1.3, revision `b5c6dd62125326e6f0b540c1f7b36e901cdc0a11`, and ColQwen2 uses https://huggingface.co/vidore/colqwen2-v1.0, revision `2b6ac8fb37f46a49e4841e599583d00ae8a20117`. Both use their official processors and `score_multi_vector` MaxSim scoring with `colpali-engine` 0.3.10 and bfloat16

Table 5: Retrieval gains from DocComp with paired document-cluster bootstrap 95% confidence intervals, in percentage points.

Comparator	Metric	Difference	95% CI
Base ColSmol	R@1	+54.78	[+53.64, +55.94]
Base ColSmol	R@3	+32.90	[+31.89, +33.92]
Base ColSmol	R@5	+18.97	[+18.15, +19.80]
Base ColSmol	MRR	+40.02	[+39.17, +40.85]
ColPali	R@1	+18.98	[+18.02, +19.96]
ColPali	R@3	+9.93	[+9.36, +10.52]
ColPali	R@5	+4.25	[+3.93, +4.59]
ColPali	MRR	+13.37	[+12.70, +14.04]
ColQwen2	R@1	+21.37	[+19.90, +22.70]
ColQwen2	R@3	+9.47	[+8.85, +10.08]
ColQwen2	R@5	+4.45	[+4.13, +4.78]
ColQwen2	MRR	+14.58	[+13.60, +15.46]

model execution. For ColQwen2, crop dimensions smaller than 28 pixels are white-padded to 28 pixels without removing crop content.

DocComp is trained on the BoundingDocs training split, which is document-disjoint from the reliable test set. The larger public retrievers receive no additional component adaptation. Their training-data overlap with BoundingDocs is unknown because it cannot be audited at the document level. Accordingly, this experiment compares the adapted compact retriever with off-the-shelf larger retrievers, not equally adapted models.

We use 10,000 paired document-cluster bootstrap resamples with seed 20260922 over 4,727 clusters, keyed by source and document ID. Table 5 reports DocComp minus each comparator in percentage points.