# DOCSEAM: BIDIRECTIONAL CROSS-PAGE ENCODING FOR MULTI-PAGE DOCUMENT UNDERSTANDING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Multi-page document vision-language models encode each page independently, creating a feature-level discontinuity at page boundaries: visual tokens are formed without access to content on adjacent pages, and the language model must reason over these page representations formed in isolation. We introduce DocSEAM (Document Selective Enrichment via Asymmetric Masking), a lightweight module inserted between a pretrained vision encoder and the language model that injects the inductive bias of local cross-page context into visual representations. For each page, DocSEAM forms a local triplet with its two neighbors, applies asymmetric masked self-attention with directional salience gates to fuse information from neighbor pages, and returns only the enriched center-page tokens, leaving the visual token count unchanged. After a supervised cross-page warm-up and multi-page fine-tuning with DocSEAM frozen, our 9B-parameter model achieves state-of-the-art results among open-weight models on five multi-page document VQA benchmarks. We further show that DocSEAM makes the model robust to arbitrary re-pagination and significantly improves cross-page span localization, confirming that the gains stem from improved cross-page representations.

## 1 INTRODUCTION

Multi-page document vision-language models (VLMs) (Bai et al., 2025b;a; Team, 2026; Zhu et al., 2025) typically encode each page independently and concatenate the resulting visual tokens into a single sequence for the language model. This design is convenient and scalable, but it imposes an implicit assumption: the representation of a page can be fully determined without seeing its neighbors. In practice, this assumption is routinely violated. Paragraphs continue across page breaks, tables carry rows onto the next page, and section headings introduce content that begins on the following leaf. When a visual encoder commits to a representation of the last row of a table *before* seeing its continuation on the next page, the language model inherits a feature-level discontinuity that it must repair from already-formed tokens.

Can the language model repair this discontinuity on its own? In many cases, yes: the LLM can attend across pages, compare redundant evidence, and compensate for missing context. But we show empirically that this compensation is fragile. As shown in Fig. 1, when we re-paginate the same documents so that page breaks fall *inside* tables and paragraphs rather than between them, page-wise VLMs suffer a consistent performance drop, even though the semantic content is unchanged (More analysis in Section 4.3). This pagination sensitivity suggests that the bottleneck is not in the language model's reasoning capacity but in the *visual representations themselves*: by the time cross-page reasoning begins, the representational commitment has already been made.

We address this with a simple inductive bias: *a page's visual representation should be informed by its immediate neighbors before reaching the language model.* We propose **DocSEAM** (**Doc**ument **S**elective **E**nrichment via **A**symmetric **M**asking), a lightweight module inserted between a pretrained VLM's vision encoder and its language model. For each page $p$, DocSEAM forms a local triplet $(p{-}1, p, p{+}1)$, applies a small number of self-attention layers over the concatenated triplet, and returns only the enriched center-page tokens. The module adds $\sim$235M parameters ($\sim$2.6% of a 9B backbone), does not modify the vision encoder or change the visual token budget, and is initialized as a no-op so that training starts from the pretrained model's known-good operating point.



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.

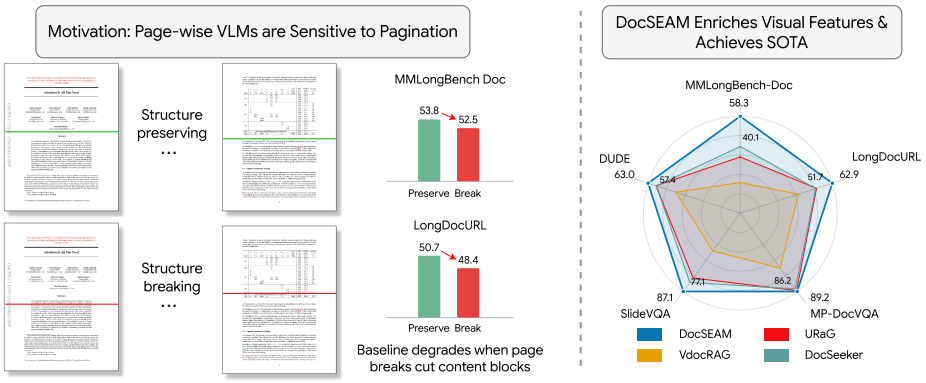


Figure 1: **Left:** Page-wise VLMs are sensitive to pagination. Re-paginating the same documents with double the pages reduces baseline performance; breaking document structure (cutting through tables and paragraphs) causes a further drop. Our model reduces this gap to near zero (§4.3). **Right:** DocSEAM achieves state-of-the-art results among end-to-end open-weight VLMs of scale less than 10B parameters on five multi-page document VQA benchmarks.

Two design choices encode document-specific structure. First, an asymmetric attention mask allows the center page to attend to both neighbors while blocking direct interaction between the previous and next pages; this forces all cross-page information to flow *through* the center page, preventing two-hop leakage between non-adjacent pages within a single layer. Second, directional salience gates, consisting of four learnable gates per token, modulate every cross-page attention cell, allowing the model to learn *where* cross-page communication is useful.

An interesting consequence of this inductive bias is that cross-page localization emerges much earlier in the network. In standard VLMs, we observe that the language model can locate content spanning a page boundary only after many layers of attention, as analyzed via attention tracing in Section 4.2. With DocSEAM, the enriched visual tokens already encode cross-page continuity *before* they reach the language model, so correct cross-page localization appears in early LLM layers. We show the advantages of this inductive bias in Section 4.2.

Our **contributions** are threefold: First, we identify pagination sensitivity as a symptom in conventional document VLMs and design a controlled re-pagination diagnostic to measure it. Second, we propose DocSEAM, a simple post-encoder module that introduces the inductive bias of local cross-page context into multi-page document representations via asymmetric attention and directional salience gates. Third, we design an effective two-stage training strategy in which DocSEAM is first warmed up on supervised cross-page alignment tasks and then *frozen* during downstream fine-tuning, allowing the learned cross-page prior to transfer seamlessly across diverse tasks. Together, these contributions enable DocSEAM to achieve state-of-the-art performance across five representative multi-page document VQA benchmarks (Fig. 1) among end-to-end open-weight VLMs with fewer than 10B parameters, a practically relevant model scale for document understanding.

## 2 Related Work

**Multi-page Document Visual Question Answering.** Document VQA was originally posed as a single-page problem (Mathew et al., 2021). Subsequent benchmarks extended it to multi-page settings: MP-DocVQA (Tito et al., 2023), DUDE (Van Landeghem et al., 2023). Current approaches fall into three families. *Retrieval-augmented methods* such as VisRAG (Yu et al., 2025), SV-RAG (Chen et al., 2025) and VDocRAG (Tanaka et al., 2025) select relevant pages before reading them, but still encode each page independently. *Agentic methods* like MACT (Yu et al., 2026) decompose reasoning into multi-step workflows (analysis, localization, reasoning) or multi-agent pipelines, yet the per-page visual representation is never modified. *End-to-end methods* such as GRAM (Blau et al., 2024) DocSeeker (Yan et al., 2026), mPLUG-DocOwl2 (Hu et al., 2025), URaG (Shi et al., 2026), and Leopard (Jia et al., 2025) compress or reallocate visual tokens but still treat each page as an independently-encoded unit. Across all three families, a visual token near a page break is computed as if the page were self-contained; whatever cross-page reasoning

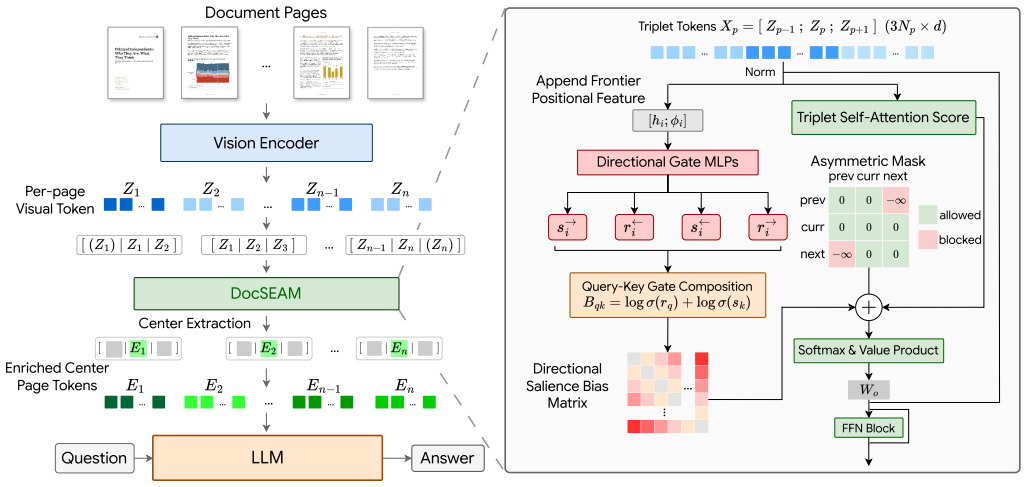


Figure 2: **DocSEAM overview.** For each page $p$, DocSEAM forms a local triplet $(Z_{p-1}, Z_p, Z_{p+1})$, applies asymmetric self-attention with directional salience gates, and returns only the enriched center-page tokens $E_p$. The language model receives the same number of visual tokens as in the stateless baseline.

happens, it happens *after* that representational commitment. DocSEAM modifies the page representation *before* it reaches the language model, retriever, or agent, and does so densely (every token) and directionally (tied to the specific neighbor).

**Visual Encoders for Documents.** A parallel line of work improves how a single page is converted into visual tokens. Native-resolution ViTs (Dehghani et al., 2023) avoid aspect-ratio distortion on dense text; DAVE (Huang et al., 2026) pretrains an encoder specifically for documents and web UIs; DeepEncoder V2 (Wei et al., 2026) learns intra-page reading order via causal queries; UNIT (Zhu et al., 2024) unifies image and text recognition within one vision encoder and mPLUG-DocOwl2's H-Reducer compresses tokens while preserving layout. All of these improve *intra-page* representation quality. DocSEAM is complementary: it sits after whatever per-page encoder produced the token grid and opens a channel between grids that were previously computed in isolation.

**Stateful and Multi-image Encoding.** Most multi-image VLMs (Jiang et al., 2024; Li et al., 2024; 2025) use late fusion: images are encoded independently and cross-image reasoning is deferred to the LLM. Streaming VLMs propagate visual memory forward (Wang et al., 2025; Qian et al., 2024; Zhang et al., 2025), but this memory is causal and mainly targets temporal compression. SVE (Wang et al., 2026) is the closest architectural prior: it injects causal cross-image interaction inside the ViT stack so the current image can condition on a previous image. DocSEAM is complementary rather than a direct variant. It leaves the vision encoder unchanged, operates post-encoder on the token grid consumed by the LLM, and uses a bidirectional three-page neighborhood because documents are fully observed at inference. Its gates are also document-specific, learning which boundary tokens should send or receive cross-page context while suppressing margins, headers, and page numbers.

## 3 METHOD

Let a document consist of $P$ page images $\{I_1, \ldots, I_P\}$. A pretrained VLM vision encoder produce one visual-token sequence per page:

$$Z_p \in \mathbb{R}^{N_p \times d}, \tag{1}$$

where $N_p$ is the number of visual tokens for page $p$ and $d$ is the hidden dimension consumed by the language model. In the standard multi-page pipeline, the sequences $Z_1, \ldots, Z_P$ are concatenated and fed directly into the language model. Each page's representation is formed in complete isolation: the visual tokens at the bottom of page $p$ are produced without access to any pixel on page $p{+}1$, and vice versa.

DocSEAM replaces each $Z_p$ with an *enriched* sequence of the same shape:

$$E_p = \mathrm{DocSEAM}(Z_{p-1},\, Z_p,\, Z_{p+1}) \;\in\; \mathbb{R}^{N_p \times d}. \tag{2}$$

The final visual-token sequence is $[E_1; \ldots; E_P]$, so the language model receives exactly the same number of visual tokens as in the baseline. DocSEAM is inserted after the vision encoder and before the language model's multimodal input assembly. This placement has three practical advantages: it avoids modifying the pretrained vision tower, operates directly on the features the language model will consume, and makes the DocSEAM / no-DocSEAM comparison a single-site architectural difference.

### 3.1 Document-Local Triplets

As shown in Fig. 2, for each page $p$, DocSEAM forms a local triplet by concatenating the token grids of the previous, current, and next page:

$$X_p = [\, Z_{p-1};\; Z_p;\; Z_{p+1} \,]. \tag{3}$$

At document boundaries we use *self-fallback*, where $Z_0 = Z_1$, and $Z_{P+1} = Z_P$, keeping the computation shape uniform without introducing a learned null-page embedding.

### 3.2 Asymmetric Triplet Attention

Each DocSEAM layer performs self-attention over the concatenated triplet $X_p$. Tokens from all three pages act as queries, keys, and values, but an asymmetric mask (Fig. 2) restricts which page groups may communicate. Let the three groups in the triplet be PREV, CURR, and NEXT. The allowed attention flow is:

$$\begin{aligned} \textsc{prev} &\;\to\; \{\textsc{prev}, \textsc{curr}\}, \\ \textsc{curr} &\;\to\; \{\textsc{prev}, \textsc{curr}, \textsc{next}\}, \\ \textsc{next} &\;\to\; \{\textsc{curr}, \textsc{next}\}. \end{aligned} \tag{4}$$

Direct PREV $\leftrightarrow$ NEXT interaction is blocked. The two neighbor pages are separated by two page transitions; they should exchange information only through the center-page representation. With $L$ stacked layers, the center page can still integrate information from neighbor states that interacted with it in earlier layers.

As shown in Fig. 2, a DocSEAM layer consists of layer normalization, a fused QKV projection, scaled dot-product attention with the asymmetric additive mask, an output projection, and a feed-forward residual block. The output projection and the second feed-forward projection are *zero-initialized*, so at initialization the residual branch contributes zero and DocSEAM is an exact identity, $E_p = Z_p$.

This property ensures that training begins from the pretrained model's original behavior; the cross-page contribution is learned as a residual on top of a known-good operating point.

### 3.3 Directional Salience Gates

Tokens near page boundaries are not equally informative: headers, footers, page numbers, and blank margins often lie near edges but rarely provide continuation evidence. DocSEAM therefore learns *directional salience gates* that selectively modulate cross-page attention cells.

For each token $i$ with hidden representation $h_i$, the module computes four scalar logits using small independent MLPs:

$$s_i^{\rightarrow}, \quad r_i^{\leftarrow}, \quad s_i^{\leftarrow}, \quad r_i^{\rightarrow}, \tag{5}$$

where $s^{\rightarrow}$ and $s^{\leftarrow}$ denote sending information to the next and previous pages, respectively, while $r^{\leftarrow}$ and $r^{\rightarrow}$ denote receiving information from the previous and next pages.

Each gate logit is produced by a two-layer MLP that receives both the layer-normalized visual token representation and a sinusoidal positional embedding $\phi_i \in \mathbb{R}^{18}$ encoding token distance to the active content frontiers:

$$s_i^{(\cdot)} = f_\theta^{(\cdot)}\big([\mathrm{LN}(h_i);\, \phi_i]\big). \tag{6}$$

This design provides a geometric prior invariant to blank page margins. Details of the positional encoding are given in Appendix B.

The gates enter attention as additive log-biases before the softmax. For a current-page query $q$ reading a previous-page key $k$:

$$B_{qk} = \log \sigma(r_q^{\leftarrow}) + \log \sigma(s_k^{\rightarrow}). \tag{7}$$

For a current-page query reading a next-page key:

$$B_{qk} = \log \sigma(r_q^{\rightarrow}) + \log \sigma(s_k^{\leftarrow}). \tag{8}$$

The same directional convention applies when neighbor pages query center-page keys. Same-page attention cells receive no salience bias. The full attention computation in one DocSEAM layer is:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_h}} + M + B\right) V, \tag{9}$$

where $M$ is the asymmetric triplet mask in Equation 4 and $B$ is the directional salience bias. Since the exponentiated bias factorizes as

$$\exp(B_{qk}) = \sigma(r_q)\, \sigma(s_k), \tag{10}$$

cross-page communication is strong only when both the receiver's demand and the sender's utility are jointly high. We refer to this as the *dual-agreement* property: a cross-page link is activated only when both sides of the boundary agree that the interaction is useful. When either gate is near zero, $B_{qk} \to -\infty$ and the corresponding attention weight is suppressed regardless of the QK similarity.

### 3.4 Training

Training proceeds in two stages.

**Stage A: Cross-page Boundary Alignment.** Stage A trains only DocSEAM while keeping the vision tower and language model frozen. We draw supervised examples from MP-DocStruct1M (Hu et al., 2025), pairing consecutive-page tasks (requiring cross-page evidence) with single-page controls (preserving single-page reading).

The Stage A objective combines next-token cross-entropy loss with a lightweight gate alignment loss:

$$\mathcal{L}_A = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{gate}} \mathcal{L}_{\mathrm{gate}}, \tag{11}$$

where $\mathcal{L}_{\mathrm{gate}}$ provides geometric guidance by penalizing activations over blank margins. $\mathcal{L}_{\mathrm{CE}}$ remains the primary driver; see Appendix B for $\mathcal{L}_{\mathrm{gate}}$ details. Self-supervised boundary reconstruction was discarded as it led to gate collapse: global self-attention within each page allows masked tokens to be reconstructed without cross-page context (Appendix I).

**Stage B: Multi-page Document Fine-tuning.** In Stage B, DocSEAM is *frozen* and the rest of the model is fine-tuned with LoRA (Hu et al., 2022) on a mixture of multi-page document understanding tasks. Freezing DocSEAM preserves the cross-page fusion policy learned in Stage A and isolates the downstream evaluation: any improvement over the stateless baseline is attributable to the enriched visual tokens, not to additional gradient flow through the module. The matched baseline is trained with identical data, hyperparameters, and LoRA configuration; the only difference is the absence of DocSEAM. Full dataset composition and hyperparameters for both stages are provided in Appendix A.

**Inference.** At inference, each page is encoded once by the vision tower. DocSEAM processes the page triplets in a single sliding-window pass, producing enriched tokens at no increase in the visual sequence length seen by the language model. The additional inference cost of DocSEAM is small: two transformer layers over triplets of post-merger tokens, amounting to less than 7% of the total forward-pass FLOPs for a typical 20-page document.

## 4 Experiments

**Backbone.** We instantiate DocSEAM on Qwen3.5-9B (Team, 2026), a native-resolution VLM with a NaViT-style vision tower and a $2\times2$ pixel-shuffle merger. DocSEAM adds $L{=}2$ transformer layers

Table 1: Multi-page document VQA results. All models in the open-weight block use comparable backbone sizes (≤10B). DUDE and MP-DocVQA are measured by ANLS; SlideVQA, MMLongBench-Doc, and LongDocURL by accuracy. †: proprietary model. Best open-weight result in **bold**.

| Method | Venue | Size | DUDE | MP-DocVQA | Slide VQA | MMLong Bench | Long DocURL |
|---|---|---|---|---|---|---|---|
| *Proprietary* | | | | | | | |
| GPT-4o† | – | – | 54.1 | 67.4 | – | 42.8 | 64.5 |
| Gemini 3 Pro† | – | – | 70.1 | – | – | 59.5 | 72.0 |
| *Open-weight (≤10B)* | | | | | | | |
| M3DocRAG 2024 | arXiv'24 | 9B | – | 84.4 | 55.7 | 21.0 | 35.1 |
| DocOwl2 2025 | ACL'25 | 8B | 46.8 | 69.4 | – | 13.4 | 5.3 |
| Vis-RAG 2025 | ICLR'25 | 4B | – | 70.9 | 50.7 | 18.8 | 41.9 |
| SV-RAG 2025 | ICLR'25 | 4B | 45.0 | 71.0 | 23.0 | – | – |
| VDocRAG 2025 | CVPR'25 | 8B | 44.0 | 62.6 | 42.0 | 18.4 | 39.8 |
| Docopilot 2025 | CVPR'25 | 8B | – | 81.3 | – | 28.8 | – |
| InternVL3 2025 | arXiv'25 | 8B | 47.4 | 80.8 | 38.7 | 24.1 | 54.4 |
| DocSeeker 2026 | CVPR'26 | 7B | 57.4 | 86.2 | 77.1 | 40.1 | 51.7 |
| URaG 2026 | AAAI'26 | 7B | 57.6 | 88.2 | 72.1 | 33.8 | 52.2 |
| Qwen3.5 2026 | – | 9B | 46.8 | 84.2 | 76.6 | 57.1 | 54.5 |
| Qwen3.5 + SFT | – | 9B | 54.1 | 87.5 | 85.1 | 44.5 | 42.8 |
| DocSEAM + Qwen2.5 VL | – | 7B | 58.0 | 88.3 | 80.0 | 39.5 | 53.3 |
| DocSEAM + Qwen3.5 | – | 4B | 57.5 | 86.3 | 79.6 | 38.2 | 53.3 |
| **DocSEAM + Qwen3.5** | – | 9B | **63.0** | **89.2** | **87.1** | **58.3** | **62.9** |

(∼235M parameters) between the Vision Encoder and the 9B language model. To demonstrate generality, we also report results with Qwen3.5-4B, Gemma3-4B and Qwen2.5-VL-7B backbones.

**Benchmarks.** We evaluate on five multi-page document VQA benchmarks: DUDE (Van Landeghem et al., 2023), MP-DocVQA (Tito et al., 2023), SlideVQA (Tanaka et al., 2023), MMLongBench-Doc (Ma et al., 2024), and LongDocURL (Deng et al., 2025). DUDE and MP-DocVQA use ANLS, while SlideVQA, MMLongBench-Doc, and LongDocURL use accuracy. We also report cross-page parsing results on MPDocBench-Parse (Zhou et al., 2026). Dataset details and task characteristics are provided in Appendix C.

**Baselines.** Our primary comparison is a *matched stateless baseline*: the same Qwen3.5-9B backbone, the same LoRA configuration, trained on the combined Stage A and Stage B data in a single SFT stage. We show in Section 4.4 that adding Stage A data to the baseline yields negligible change, so this setup is strictly more favorable to the baseline than using Stage B data alone. We additionally compare against other multi-page document understanding methods.

### 4.1 Main Results

Table 1 presents the main results. DocSEAM + Qwen3.5-9B achieves state-of-the-art performance among open-weight models on all five benchmarks: 63.0 ANLS on DUDE (+5.4 over URaG), 89.2 ANLS on MP-DocVQA (+1.0 over URaG), 87.1 accuracy on SlideVQA (+10.0 over DocSeeker), 58.3 accuracy on MMLongBench-Doc (surpassing even Qwen3.5 zero-shot and all other open-weight models), and 62.9 accuracy on LongDocURL (+10.1 over URaG).

Notably, standard SFT on short-answer document QA data actually *degrades* performance on MMLongBench-Doc ($57.1 \rightarrow 44.5$) and LongDocURL ($54.5 \rightarrow 42.8$) compared to the Qwen3.5 zero-shot baseline, likely because fine-tuning on short, extractive answers biases the model away from the multi-step reasoning these benchmarks require. DocSEAM not only reverses this degradation but surpasses the zero-shot baseline on both benchmarks (58.3 and 62.9), demonstrating that cross-page enrichment provides complementary benefits that survive task-specific fine-tuning. The

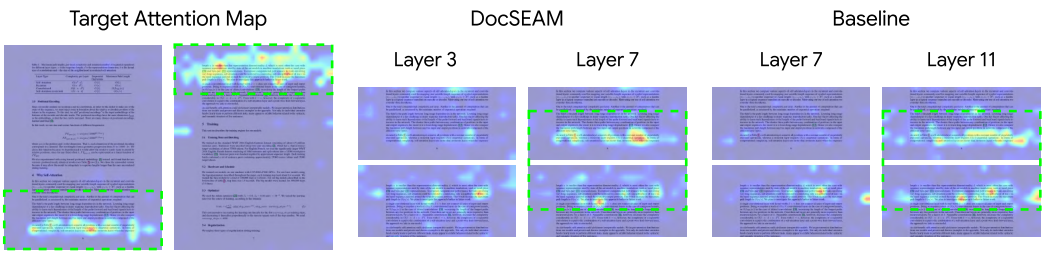


Figure 3: LLM layer-wise attention trace for cross-page span localization. Given a text span query crossing a page boundary, DocSEAM concentrates attention on the ground-truth evidence in early LLM layers (Layer 7), whereas the baseline forms a comparable pattern only in late layers (Layer 11). For readability, layer heatmaps display cropped top/bottom boundary regions of consecutive pages. Green and red dashed boxes indicate correct and missing attention, respectively.

Table 2: Cross-page span localization diagnostic. Exact-match accuracy (%) on three probe types. The cross-page lookup requires identifying *both* pages of a boundary-spanning span.

| Model | Cross-page lookup | Single-page lookup | Existence check |
|---|---|---|---|
| Qwen3.5 (zero-shot) | 50.2 | 86.0 | 91.8 |
| Qwen3.5 + SFT (baseline) | 65.0 | 70.0 | **95.6** |
| **Qwen3.5 + DocSEAM (ours)** | **85.8** | **94.2** | 94.0 |

gains over the matched SFT baseline are substantial across all benchmarks: +8.9 on DUDE, +2.0 on SlideVQA, +13.8 on MMLongBench-Doc, and +20.1 on LongDocURL.

DocSEAM is backbone-agnostic. When applied to Qwen2.5-VL-7B, the same backbone used by DocSeeker and URaG, DocSEAM achieves 58.0 on DUDE and 88.3 on MP-DocVQA, surpassing both published methods with a general-purpose module rather than task-specific training. Even the smaller DocSEAM + Qwen3.5-4B variant reaches 57.5 on DUDE and 86.3 on MP-DocVQA, performing on par with DocSeeker (57.4, 86.2) and URaG (57.6, 88.2) despite using roughly half the backbone parameters. We further validate DocSEAM on cross-page document parsing (MPDocBench-Parse) and stratified evidence-topology analysis (LongDocURL) in Appendix E and D, respectively, confirming that gains generalize beyond VQA and concentrate on adjacent-page evidence.

### 4.2 Cross-Page Representation Analysis

To understand *how* DocSEAM improves downstream performance, we examine the cross-page representations through attention visualization, a large-scale localization benchmark, and gate activation patterns.

**Attention trace: early-layer cross-page localization.** We visualize the layer-wise attention maps of the language model when queried with a cross-page span localization prompt. Figure 3 shows a representative case: the model is asked which pages contain a quoted text span that crosses a page boundary. With DocSEAM, attention concentrates on the correct boundary-spanning evidence as early as Layer 7, whereas the baseline model without cross-page enrichment fails to form a comparable pattern until Layer 11. This confirms that DocSEAM encodes cross-page continuity into the visual tokens *before* the language model begins deep reasoning. An extended visualization comparing attention maps across deeper LLM layers is provided in Appendix J.

**Large-scale span localization benchmark.** To confirm that this early-layer behavior is systematic rather than cherry-picked, we construct a quantitative benchmark of 1,500 samples from MPDocBench-Parse (Zhou et al., 2026) with three task types (500 each): *cross-page text lookup* (identify *both* pages of a boundary-spanning span), *single-page text lookup* (identify the page of a page-local span), and *single-page existence* (judge whether a span appears on a given page).

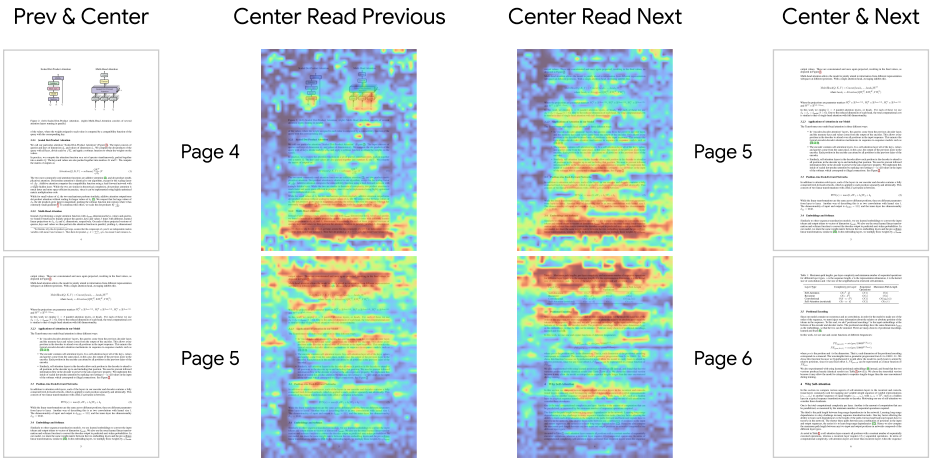


Figure 4: **Cross-page attention within DocSEAM.** Across a triplet from *Attention Is All You Need* (Pages 4–6). **(a–b)** Page 5 reads backward: attention flows to continuous text at the bottom of Page 4 and grounds on Figure 2. **(c–d)** Page 5 reads forward: despite Table 1 intervening at the top of Page 6, DocSEAM attends to both the table and the text continuation beneath it. Blank margins receive near-zero attention throughout.

Table 2 shows that DocSEAM achieves 85.8% on cross-page lookup (+20.8 over the SFT baseline). The SFT baseline's single-page lookup drops to 70.0% compared to the zero-shot 86.0%, likely due to distribution overfitting; DocSEAM recovers and surpasses it (94.2%). These results confirm that DocSEAM provides the language model with representations encoding cross-page continuity.

**Gate activation patterns.** Figure 4 visualizes the cross-page attention flow within the DocSEAM module across three consecutive pages of *Attention Is All You Need* (Vaswani et al., 2017). Three emergent behaviors stand out. First, activation concentrates on the bottom and top of consecutive pages, where truncated content resides, while blank margins, headers, and page numbers receive near-zero attention, confirming content-aware routing. Second, in the previous-page interaction (Panels a, b), the Multi-Head Attention figure on Page 4 lights up together with the corresponding equation on Page 5, showing that the module grounds a diagram to its mathematical definition across the page boundary. Third, in the next-page interaction (Panels c, d), Page 6 exhibits two distinct bright regions: Section 3.5 continues from the bottom of Page 5, but Table 1 is inserted between the boundary and the continuation; DocSEAM connects both the table and the text below it, preserving alignment across the layout interruption. Additional visualizations are in Appendix K.

### 4.3 Pagination Invariance

A model with robust document understanding should be insensitive to *where* page breaks fall. We test this by splitting every page of a benchmark into two half-pages near the center in two modes: **structure-preserving** (cut at whitespace between paragraphs/tables) and **structure-breaking** (cut through paragraphs or table interiors). Both variants have identical page counts and pixel content; only the boundary location differs.

We define two metrics to quantify robustness. The *total degradation* $\Delta_{\text{orig}} = S_{\text{orig}} - S_{\text{break}}$ measures how much performance drops from the original to the worst-case re-pagination. The *pagination penalty* $\Delta_{\text{pag}} = S_{\text{preserve}} - S_{\text{break}}$ isolates sensitivity to boundary placement by comparing two re-paginated variants with identical page counts.

Table 3 shows that the baseline suffers a total degradation of 4.6 points on MMLongBench-Doc and 6.1 on LongDocURL, with pagination penalties of 1.3 and 2.3 respectively. DocSEAM reduces total degradation to 1.5 and 1.8, and pagination penalty to 0.8 and 0.7. Notably, DocSEAM's absolute scores under structure-breaking re-pagination on LongDocURL still exceed the baseline's *original* scores, demonstrating that cross-page enrichment not only improves robustness but also lifts the performance floor. This closes the loop on the motivating observation from the introduction: DocSEAM's cross-page enrichment makes the model substantially more pagination-invariant.

Table 3: **Pagination invariance** under controlled re-pagination. $\Delta_{\text{orig}} = S_{\text{orig}} - S_{\text{break}}$; $\Delta_{\text{pag}} = S_{\text{preserve}} - S_{\text{break}}$.

| Metric | Model | MMLong Bench | Long DocURL |
|---|---|---|---|
| Original ($S_{\text{orig}}$) | Baseline | 57.1 | 54.5 |
| | **+DocSEAM** | **58.3** | **62.9** |
| Preserved ($S_{\text{pres}}$) | Baseline | 53.8 | 50.7 |
| | **+DocSEAM** | **57.6** | **61.8** |
| Broken ($S_{\text{break}}$) | Baseline | 52.5 | 48.4 |
| | **+DocSEAM** | **56.8** | **61.1** |
| Degradation ($\Delta_{\text{orig}}$) | Baseline | 4.6 | 6.1 |
| | **+DocSEAM** | **1.5** | **1.8** |
| Penalty ($\Delta_{\text{pag}}$) | Baseline | 1.3 | 2.3 |
| | **+DocSEAM** | **0.8** | **0.7** |

Table 4: **Key ablations** on MMLongBench-Doc and LongDocURL (accuracy %). †Baseline SFT trained on Stage A+B data.

| Variant | MMLong Bench | Long DocURL |
|---|---|---|
| Qwen3.5 + SFT (Stage B only) | 44.3 | 42.6 |
| Qwen3.5 + SFT† (Stage A+B data) | 44.5 | 42.8 |
| *Capacity controls* | | |
| + Self-as-neighbor | 45.1 | 43.5 |
| + Dup. vision-enc. layers | 45.8 | 44.1 |
| + Single-page transformer | 46.2 | 44.5 |
| *Module design* | | |
| + Causal (prev-only) triplet | 53.1 | 55.8 |
| + Full 3-page attn., no gates | 55.2 | 57.8 |
| + Asym. mask, no gates | 55.9 | 58.7 |
| + Full 3-page attn. + 4 dir. gates | 56.8 | 60.5 |
| + Asym. mask + 4 dir. gates (ours) | **58.3** | **62.9** |
| *Training strategy* | | |
| Baseline two-stage (no DocSEAM) | 42.1 | 40.3 |
| No Stage A (direct Stage B) | 52.7 | 56.1 |
| Stage A: CE only | 55.8 | 59.5 |
| Stage A: CE + aux. losses (ours) | **58.3** | **62.9** |

The full re-pagination protocol, including the cutting algorithm and sample counts, is described in Appendix G.

### 4.4 Ablation Studies

Table 4 presents key ablations on the two benchmarks most sensitive to cross-page reasoning. **Capacity controls:** the "self-as-neighbor" variant uses the identical DocSEAM architecture and FLOPs but feeds copies of the center page instead of real neighbors; the negligible gain (+0.6 on MMLongBench) confirms improvements come from cross-page context, not added parameters. Two further controls match DocSEAM's parameter count without cross-page information: duplicating the last layers of the vision encoder and adding single-page transformer layers; both confirm the same conclusion (Appendix F). **Bidirectional vs. causal:** restricting to a previous-page-only window degrades performance, validating that documents benefit from seeing both neighbors. **Data attribution:** to ensure gains are architectural rather than data-driven, our SFT baseline is trained on the *combined* Stage A and Stage B data in a single stage; adding Stage A data yields negligible change over Stage B-only SFT (44.5 vs. 44.3 on MMLongBench), confirming that the cross-page parsing data alone does not explain DocSEAM's improvements. **Training strategy:** applying the same two-stage procedure to the baseline (without DocSEAM) actually *hurts* downstream performance (42.1 vs. 44.5), confirming that Stage A is beneficial specifically because it pre-trains the cross-page gates, not because of the additional training steps. Without Stage A warm-up, DocSEAM still helps but is under-trained; self-supervised boundary reconstruction hurts, confirming that post-encoder masking admits a same-page shortcut. Full ablations including gate input variants, number of DocSEAM layers, and backbone generality are in Appendix F.

## 5 Conclusion

We identified pagination sensitivity as a systematic weakness of page-wise document VLMs and proposed DocSEAM, a lightweight post-encoder module that enriches each page's visual tokens with bidirectional neighbor context via asymmetric masked attention and directional salience gates, without modifying the vision encoder or increasing the visual token budget. DocSEAM achieves state-of-the-art results among open-weight models ($\leq$10B) on five multi-page document VQA benchmarks, improves cross-page span localization by over 20 points, and reduces pagination sensitivity to near zero. Ablations confirm that gains stem from cross-page context rather than added capacity, and gate analysis reveals emergent directional specialization without explicit spatial supervision, suggesting that document VLMs benefit from architectural bias aligned with the local continuity of real documents.

### AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing, including grammar checking, proofreading, and improving text clarity and sentence structure. We have not used generative AI tools for core research tasks, such as generating research ideas, designing methodologies, conducting experiments, analyzing data, or writing the initial text drafts, and any other tasks requiring disclosure are not applicable to this work. We have reviewed all AI-assisted work. Specifically, all AI-suggested rephrasings and corrections were manually inspected, verified, and edited by the authors to ensure accuracy and faithful representation of our scientific findings. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, 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, 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, 2025b. URL https://arxiv.org/abs/2502.13923.

Tsachi Blau, Sharon Fogel, Roi Ronen, Alona Golts, Roy Ganz, Elad Ben Avraham, Aviad Aberdam, Shahar Tsiper, and Ron Litman. Gram: Global reasoning for multi-page vqa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15598–15607, 2024.

Jian Chen, Ruiyi Zhang, Yufan Zhou, Tong Yu, Franck Dernoncourt, Jiuxiang Gu, Ryan A. Rossi, Changyou Chen, and Tong Sun. SV-RAG: LoRA-contextualizing adaptation of MLLMs for long document understanding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FDaHjwInXO.

Jaemin Cho, Ozan İrsoy, Debanjan Mahata, Yujie He, and Mohit Bansal. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding, 2024.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Peter Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Lucic, and Neil Houlsby. Patch n' pack: Navit, a vision transformer for any aspect ratio and resolution. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=VpGFHmI7e5.

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

Yuchen Duan, Zhe Chen, Yusong Hu, Weiyun Wang, Shenglong Ye, Botian Shi, Lewei Lu, Qibin Hou, Tong Lu, Hongsheng Li, et al. Docopilot: Improving multimodal models for document-level understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4026–4037. IEEE, 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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5817–5834, 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 *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Brandon Huang, Hang Hua, Zhuoran Yu, Rogerio Feris, Roei Herzig, et al. Dave: A vlm vision encoder for document understanding and web agents. In *International Conference on Learning Representations*, volume 2026, pp. 129478–129499, 2026.

Mengzhao Jia, Wenhao Yu, Kaixin Ma, Tianqing Fang, Zhihan Zhang, Siru Ouyang, Hongming Zhang, Dong Yu, and Meng Jiang. Leopard: A vision language model for text-rich multi- image tasks. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=R2rasAEPVi.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=skLtdUVaJa. Featured Certification, Outstanding Certification.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=zKv8qULV6n.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2024.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *Advances in Neural Information Processing Systems*, 37:119336–119360, 2024.

Yongxin Shi, Jiapeng Wang, Zeyu Shan, Dezhi Peng, Zening Lin, and Lianwen Jin. Urag: unified retrieval and generation in multimodal llms for efficient long document understanding. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i30.39729. URL https://doi.org/10.1609/aaai.v40i30.39729.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: a dataset for document visual question answering on multiple images. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i11.26598. URL https://doi.org/10.1609/aaai.v37i11.26598.

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

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834, 2023.

Jordy Van Landeghem, Rubèn Tito, Łukasz Borchmann, Michał Pietruszka, Pawel Joziak, Rafal Powalski, Dawid Jurkiewicz, Mickaël Coustaty, Bertrand Anckaert, Ernest Valveny, et al. Document understanding dataset and evaluation (dude). In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19528–19540, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yuxuan Wang, Yiqi Song, Cihang Xie, Yang Liu, and Zilong Zheng. Videollamb: Long streaming video understanding with recurrent memory bridges. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24170–24181, October 2025.

Zirui Wang, Junwei Yu, Adam Yala, David M. Chan, Joseph E. Gonzalez, and Trevor Darrell. Stateful visual encoders for vision-language models. In *ECCV Workshop X-Reason: Visual Perception and Reasoning in the Interactable World*, 2026. URL https://openreview.net/forum?id=5J7cbBFHVH.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026.

Hao Yan, Yuliang Liu, Xingchen Liu, Yuyi Zhang, Minghui Liao, Jihao Wu, Wei Chen, and Xiang Bai. Docseeker: Structured visual reasoning with evidence grounding for long document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41140–41149, June 2026.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. Visrag: Vision-based retrieval-augmented generation on multimodality documents. In *International Conference on Learning Representations*, volume 2025, pp. 21074–21098, 2025.

Xinlei Yu, Chengming Xu, Zhangquan Chen, Yudong Zhang, Shilin Lu, Cheng Yang, Jiangning Zhang, Shuicheng Yan, and Xiaobin Hu. Visual document understanding and reasoning: A multi-agent collaboration framework with agent-wise adaptive test-time scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12300–12311, 2026.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21059–21069, October 2025.

Bangbang Zhou, Hangdi Xing, Yifan Chen, Jianjun Xu, Qi Zheng, Feiyu Gao, Zhibo Yang, Shuai Bai, Ming Yan, Jieping Ye, et al. Mpdocbench-parse: Benchmarking practical multi-page document parsing. *arXiv preprint arXiv:2605.22100*, 2026.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Yi Zhu, Yanpeng Zhou, Chunwei Wang, Yang Cao, Jianhua Han, Lu Hou, and Hang Xu. Unit: Unifying image and text recognition in one vision encoder. *Advances in Neural Information Processing Systems*, 37:122185–122205, 2024.

# A Training Details

**Stage A dataset.** We derive 74k training examples from MP-DocStruct1M (Hu et al., 2025), organized into four groups: (1) **Consecutive parsing** (50k): the model reads text spanning the boundary between two consecutive pages and produces the merged passage. (2) **Consecutive lookup** (12k): given a text snippet crossing a page boundary, the model identifies which pages contain it. (3) **Single-page parsing control** (6k): the model reads a single page's text inside a multi-page context. (4) **Single-page lookup control** (6k): the model localizes page-local evidence inside a multi-page context. All examples are presented as document-local page neighborhoods.

**Stage A hyperparameters.** $L$=2 DocSEAM layers, AdamW with learning rate $10^{-4}$, cosine decay, batch size 64 (8 GPUs $\times$ gradient accumulation 8), 500 training steps.

**Stage B dataset.** The training mixture includes multi-page document VQA datasets (MP-DocVQA (Tito et al., 2023), DUDE (Van Landeghem et al., 2023), SlideVQA (Tanaka et al., 2023)) and MP-Docreason51K multipage reasoning data from mPlug-DocOwl2 (Hu et al., 2025).

**Stage B hyperparameters.** LoRA (Hu et al., 2022) rank 16 applied to all linear layers of the language model. Visual patch merger is also fine-tuned; vision tower is frozen. AdamW with learning rate $2\times10^{-5}$, cosine schedule, batch size 64, 2 epochs.

# B Gate and Loss Details

**Content-frontier positional encoding.** The positional feature $\phi_i \in \mathbb{R}^{18}$ encodes each token's distance to the top and bottom content frontiers (first and last rows containing document ink). The distance to each frontier is capped at a fraction $\alpha = 0.20$ of the page height and mapped via a 4-frequency sinusoidal expansion:

$$\psi(d) = \big[d,\, \sin(\pi d),\, \cos(\pi d),\, \ldots,\, \sin(8\pi d),\, \cos(8\pi d)\big] \in \mathbb{R}^9, \tag{12}$$

yielding $\phi_i = [\psi(d_{\text{top}}(r_i));\, \psi(d_{\text{bot}}(r_i))]$. Physical page edges often contain semantically irrelevant white space; measuring distance relative to detected ink frontiers rather than raw margins provides the gate with a geometric prior invariant to blank page margins.

**Auxiliary gate loss.** The gate alignment loss consists of three terms:

$$\mathcal{L}_{\text{gate}} = \mathcal{L}_{\text{frontier}} + \lambda_{\text{margin}}\mathcal{L}_{\text{margin}} + \mathcal{L}_{\text{budget}}, \tag{13}$$

where $\mathcal{L}_{\text{frontier}}$ applies a weighted binary cross-entropy to align directional send/receive channels with content frontiers, $\mathcal{L}_{\text{margin}}$ suppresses gate leakage over whitespace margins, and $\mathcal{L}_{\text{budget}} = \max(0, \bar{g} - \rho)^2$ maintains the average cross-page communication density around a target budget $\rho = 0.20$.

# C Benchmark Details

**Document VQA benchmarks.** We evaluate on five multi-page document VQA benchmarks that cover complementary document types and reasoning regimes. **DUDE** (Van Landeghem et al., 2023) is built from multi-industry, multi-domain, multi-page visually rich documents and includes diverse question and answer types, including layout-navigating and multi-step reasoning questions. We report ANLS following the benchmark protocol. **MP-DocVQA** (Tito et al., 2023) extends single-page document VQA to multi-page scanned industry documents by providing each question with its surrounding document pages, up to 20 pages per document. We report ANLS. **SlideVQA** (Tanaka et al., 2023) evaluates question answering over slide decks, where the model must identify relevant evidence slides and answer the question. We report exact-match accuracy. **MMLongBench-Doc** (Ma et al., 2024) targets long-context document understanding over lengthy PDF documents with evidence drawn from text, tables, charts, images, and layout structure. It includes both single-page and cross-page questions, as well as unanswerable questions for hallucination evaluation. We report accuracy. **LongDocURL** (Deng et al., 2025) evaluates long multimodal document understanding through tasks covering document understanding, numerical reasoning, and cross-element locating. We report accuracy.

**Parsing benchmark.** Beyond VQA, we evaluate on **MPDocBench-Parse** (Zhou et al., 2026), a multi-page document parsing benchmark designed for document-level end-to-end parsing. Its evaluation protocol covers content fidelity and logical structure, including text, table, and formula recognition, truncated text and table merging, figure extraction, reading order, and heading hierarchy recovery. Because DocSEAM is designed to improve cross-page continuity, we focus on the two metrics most directly tied to page-boundary reasoning: truncated text merging and truncated table merging.

## D LongDocURL Stratified Analysis

LongDocURL provides `evidence_pages` annotations for each question, enabling us to stratify results by evidence topology without constructing new data.

Table 5: LongDocURL stratified by evidence topology. $\Delta$: improvement of DocSEAM over the matched baseline.

| Evidence topology | Baseline | DocSEAM | $\Delta$ |
|---|---|---|---|
| Single page | 57.5 | 63.0 | +5.5 |
| Adjacent two pages | 58.0 | 69.5 | +11.5 |
| Multiple ($\geq$3, $\exists$ adj.) | 48.8 | 60.1 | +11.3 |
| Non-adjacent only | 48.8 | 54.7 | +5.9 |
| Overall | 54.5 | 62.9 | +8.4 |

The largest gains occur on **adjacent two-page** questions (+11.5), precisely the setting where DocSEAM's triplet window directly enriches both evidence pages. Multiple-page questions with at least one adjacent pair also benefit substantially (+11.3). Single-page questions show moderate gain (+8.0). Non-adjacent questions improve modestly (+5.7), consistent with the local window providing less direct benefit for distant evidence. The gradient $\Delta_{\text{adjacent}} \gg \Delta_{\text{non-adjacent}} > \Delta_{\text{single}}$ matches the inductive bias of DocSEAM and is strong evidence that the improvement is mechanistic rather than a uniform capacity boost.

## E Multi-Page Document Parsing

To test whether DocSEAM's benefit extends beyond QA, we evaluate on MPDocBench-Parse (Zhou et al., 2026), a recent benchmark designed to assess multi-page document parsing. We focus on the two metrics most directly related to cross-page continuity: **truncated text merging** (T-Text) and **truncated table merging** (T-Table), where existing models nearly universally fail.

Table 6 shows that DocSEAM substantially improves cross-page parsing. On truncated text merging, DocSEAM reduces Normalized Edit Distance (NED) from 0.341 to 0.172, surpassing all end-to-end VLMs and approaching the best pipeline-based model (PaddleOCR-VL-1.5, 0.147) which employs explicit rule-based merging heuristics. On truncated table merging, DocSEAM improves TEDS from 55.8 to 62.9, a +7.1 point gain over the baseline.

This cross-task validation is important: the same architectural module that improves QA also improves parsing, suggesting that DocSEAM learns a general property of cross-page document representation rather than a task-specific optimization. Notably, the pipeline-based models achieve strong T-Table scores through hand-designed merging rules that explicitly detect truncated tables and stitch them; DocSEAM approaches this performance through learned representation enrichment alone, with no explicit merging logic.

## F Full Ablation Details

Table 7 extends the main-paper ablations (Table 4) with additional variants.

Table 6: MPDocBench-Parse results on cross-page parsing. T-Text and T-Table measure merging of paragraphs and tables split across pages. NED↓ for T-Text; TEDS↑ for T-Table. *Uses explicit rule-based merging heuristics.

| Model | Overall↑ | T-Text↓ | T-Table↑ |
|---|---|---|---|
| *End-to-end specialized* | | | |
| FireRed-OCR | 69.3 | 0.178 | 62.6 |
| DeepSeek-OCR2 | 75.6 | 0.284 | 63.0 |
| dots.ocr | 74.2 | 0.303 | 60.7 |
| *General VLMs* | | | |
| Gemini 3.1 Pro | 71.8 | 0.232 | 61.3 |
| Qwen3-VL-235B | 74.0 | 0.187 | 61.3 |
| InternVL3.5-38B | 57.0 | 0.515 | 51.7 |
| *Pipeline-based** | | | |
| PaddleOCR-VL-1.5* | 80.8 | **0.147** | 83.1 |
| MinerU2.5* | 76.7 | 0.328 | **88.1** |
| *Ours* | | | |
| Qwen3.5 + SFT (base.) | 55.2 | 0.341 | 55.8 |
| **Qwen3.5 + DocSEAM** | 65.6 | 0.172 | 62.9 |

Table 7: Full ablation results on MMLongBench-Doc and LongDocURL (accuracy %).

| Variant | MMLong Bench | Long DocURL |
|---|---|---|
| Qwen3.5 + SFT (no DocSEAM) | 44.5 | 42.8 |
| *Capacity controls* | | |
| + Self-as-neighbor | 45.1 | 43.5 |
| + Dup. vision-enc. layers | 45.8 | 44.1 |
| + Single-page transformer | 46.2 | 44.5 |
| *Module design* | | |
| + Causal (prev-only) triplet | 53.1 | 55.8 |
| + Full 3-page attn., no gates | 55.2 | 57.8 |
| + Asym. mask, no gates | 55.9 | 58.7 |
| + Full 3-page attn. + 4 dir. gates | 56.8 | 60.5 |
| + Asym. mask + shared gate | 57.1 | 60.2 |
| + Asym. mask + 4 dir. gates (ours) | **58.3** | **62.9** |
| *Number of DocSEAM layers* | | |
| 1 layer | 54.1 | 60.8 |
| 2 layers (ours) | **58.3** | **62.9** |
| 4 layers | 55.1 | 61.7 |
| *Training strategy* | | |
| Baseline two-stage (no DocSEAM) | 42.1 | 40.3 |
| No Stage A (direct Stage B) | 52.7 | 56.1 |
| Stage A: boundary reconstruction | 47.3 | 48.9 |
| Stage A: CE only | 55.8 | 59.5 |
| Stage A: CE + auxiliary losses (ours) | **58.3** | **62.9** |

**Capacity controls.** All three controls match DocSEAM's ∼235M parameter budget without cross-page information. *Self-as-neighbor* uses the identical architecture but feeds copies of the center page as neighbors (+0.6 on MMLongBench). *Duplicated vision-encoder layers* extends the per-page visual encoder to the same depth. *Single-page transformer* adds independent transformer layers over each page's tokens. None exceeds +1.7 over the baseline, confirming that cross-page context, not capacity, drives the improvement.

**Module design.** Restricting DocSEAM to a causal (previous-only) window loses 5.2 points on MMLongBench, validating bidirectional enrichment. The asymmetric mask consistently outperforms full 3-page attention under matched gating (58.3 vs. 56.8 with four gates; 55.9 vs. 55.2 without gates), supporting the design choice of blocking direct prev↔next interaction. Four directional gates outperform a shared gate by 1.2 on MMLongBench and 2.7 on LongDocURL.

**Depth.** One DocSEAM layer already yields +9.6 over the baseline on MMLongBench; two layers provide the best result. Four layers slightly regress, likely due to over-smoothing of page-specific features.

**Training strategy and data attribution.** To ensure fair attribution, Table 8 specifies the data and trainable components for each compared variant.

Table 8: Training configuration for each ablation variant.

| Variant | Stage A data | Stage B data | Trainable in Stage A |
|---|---|---|---|
| SFT baseline (B only) | ✗ | ✓ | - |
| SFT baseline (A+B)† | combined, single stage | | LoRA |
| Baseline two-stage | ✓ | ✓ | LoRA (no DocSEAM) |
| DocSEAM (ours) | ✓ | ✓ | DocSEAM only |

The SFT baseline used throughout the paper trains on the combined Stage A and Stage B data in a single stage (†). This is strictly more favorable to the baseline than using Stage B data alone, yet yields nearly identical performance (44.5 vs. 44.3 on MMLongBench), confirming that Stage A data provides no benefit without DocSEAM. The *baseline two-stage* variant applies the same two-stage schedule to the baseline without DocSEAM; performance *degrades* (42.1 vs. 44.5), showing that the two-stage procedure is beneficial specifically because it pre-trains the cross-page gates.

For Stage A objectives: skipping Stage A leaves the gate policy under-trained (52.7); self-supervised boundary reconstruction hurts (47.3), as post-encoder masking admits a same-page shortcut (Section 3.4); and auxiliary gate-shaping losses provide marginal benefit over CE alone (58.3 vs. 57.8).

Table 9: DocSEAM consistently improves over matched baselines across VLM backbones.

| Backbone | LongDocURL | | | MMLongBench | | |
|---|---|---|---|---|---|---|
| | Base | Base + SFT | +DocSEAM | Base | Base + SFT | +DocSEAM |
| Qwen2.5-VL-7B | 37.8 | 42.7 | 53.3 | 25.4 | 28.8 | 39.5 |
| Qwen3.5-9B | 54.5 | 42.8 | 62.9 | 57.1 | 44.5 | 62.9 |
| Gemma3-4B | 34.7 | 39.5 | 42.6 | 20.1 | 25.2 | 33.6 |

**Backbone generality.** DocSEAM yields consistent gains across three backbone families. On Qwen2.5-VL-7B, the same backbone used by DocSeeker and URaG, DocSEAM surpasses both published methods with a general-purpose module.

**Inference overhead.** Each page undergoes one additional forward pass through the DocSEAM module (∼235M parameters) within a sliding-window triplet. On a 20-page document with Qwen3.5-9B, this adds ∼8% wall-clock latency; the dominant cost remains the LLM's autoregressive decoding.

## G Re-pagination Protocol

We evaluate on the complete test sets of MMLongBench-Doc (1,082 samples) and LongDocURL (2,325 samples) with no exclusions. Every page in every sample is split into two half-pages using the procedure in Algorithm 1.

**Algorithm 1** Controlled re-pagination

**Require:** Page image $I$ of height $H$, mode $\in \{\textsc{Preserve}, \textsc{Break}\}$
**Ensure:** Two half-page images $I_{\text{top}}, I_{\text{bottom}}$
Compute row-wise ink density $\rho(y)$ and content bounding box
**if** mode $=$ PRESERVE **then**
  Find all whitespace bands (contiguous rows with $\rho < 0.004$, height $\geq 8$ px) in the middle 70% of the page
  Select the widest band closest to $H/2$; cut at its midpoint $y^*$
**else if** mode $=$ BREAK **then**
  Detect horizontal table rules (long dark runs, $\geq 3$ clustered)
  **if** table detected **then**
    Cut at an *internal* row rule nearest $H/2$
  **else**
    Find consecutive same-paragraph text lines (gap $\leq 1.8\times$ median spacing, horizontal overlap $\geq 45\%$)
    Cut between the pair nearest $H/2$
  **end if**
**end if**
$I_{\text{top}} \leftarrow I[0 : y^*], \quad I_{\text{bottom}} \leftarrow I[y^* : H]$

Both modes produce exactly $2N$ half-pages from $N$ originals, preserving the total pixel content, NaViT resolution, and token allocation. Questions and answers are unchanged; the only variable is the vertical position of the artificial boundary. $\Delta_{\text{pag}} = S_{\text{preserve}} - S_{\text{break}}$ isolates boundary-placement sensitivity by comparing two re-paginated variants with identical page counts. $\Delta_{\text{orig}} = S_{\text{orig}} - S_{\text{break}}$ additionally captures the effect of the increased page count.

## H AGGREGATE GATE ACTIVATION STATISTICS

The qualitative gate visualizations in Section 4.2 (Figure 4) show that DocSEAM's directional gates concentrate on boundary content and suppress blank margins. To verify that this behavior is systematic rather than cherry-picked, we compute aggregate gate activation statistics over 200 randomly sampled triplets from the MMLongBench-Doc benchmark.

For each triplet, we record the sigmoid-activated gate values ($\sigma(s) \in [0, 1]$) for all four directional gates and partition each page's tokens using an automatically computed ink mask (pixels darker than 95% white = content). We further divide each page into a **top band** (top 20% of rows), **bottom band** (bottom 20%), and **middle** region.

**Content vs. margin selectivity.** Across all gates and pages, mean activation on content regions is 0.27 vs. 0.19 on margins (1.4× ratio, Pearson $\rho = 0.25$ with the ink mask). While this ratio is moderate, the gates are not designed to be binary content detectors; their primary role is to *directionally* modulate cross-page attention flow, which the next analysis reveals.

**Directional boundary specialization.** The most striking pattern is that each gate learns to concentrate on the *directionally appropriate* page boundary, with strong specialization emerging primarily from the task loss despite only a weak geometric prior (Table 10).

Gates that mediate interaction with the **next** page (`send_to_next`, `recv_from_next`) activate 2.8-3.4× more strongly on the bottom boundary band than the top, precisely where content continues onto the next page. Conversely, gates mediating interaction with the **previous** page (`recv_from_prev`, `send_to_prev`) activate 3.1-3.9× more on the top boundary band. The middle region receives uniformly moderate activation (∼0.25) across all gates.

This directional specialization is guided by the Stage A frontier prior (Appendix B), which pairs each gate with its directionally appropriate ink frontier (top frontier for previous-page gates, bottom frontier for next-page gates, $\lambda_{\text{frontier}} = 10^{-3}$). The aggregate statistics confirm that this weak geometric signal, combined with the cross-entropy task loss, produces consistent and well-calibrated directional routing across diverse documents. The pattern is consistent across all three page posi-

Table 10: Directional gate activation by page region, averaged over 200 document triplets (center page shown). Gates interacting with the next page concentrate on the **bottom** band; gates interacting with the previous page concentrate on the **top** band. The strong directional asymmetry (2.8-3.9×) exceeds what the weak frontier prior ($\lambda = 10^{-3}$) alone could produce, indicating that the task loss is the primary driver.

| Gate direction | Top band | Middle | Bottom band | Relevant boundary | Relevant/ Opposite |
|---|---|---|---|---|---|
| *Next-page gates (should concentrate on bottom)* | | | | | |
| send_to_next | 0.13 | 0.25 | **0.46** | bottom | 3.4× |
| recv_from_next | 0.16 | 0.26 | **0.45** | bottom | 2.8× |
| *Previous-page gates (should concentrate on top)* | | | | | |
| recv_from_prev | **0.46** | 0.25 | 0.15 | top | 3.1× |
| send_to_prev | **0.47** | 0.24 | 0.12 | top | 3.9× |

tions in the triplet (previous, center, next); center-page values are shown in Table 10, with the full breakdown available in the released statistics file.

These aggregate measurements complement the qualitative visualizations in the main paper and provide quantitative evidence that the learned gate behavior is a systematic, directionally-specialized property of the module, not an artifact of selected examples.

# I Analysis of Post-Encoder Representation Leakage

As noted in Section 3.4, a natural candidate for pre-training DocSEAM in Stage A is self-supervised *boundary reconstruction*, which mask visual tokens near page margins and forcing the module to reconstruct them from neighboring pages. However, our empirical investigation revealed that this self-supervised task fails when inserted *post-encoder* due to an architectural phenomenon we term **post-encoder representation leakage**.

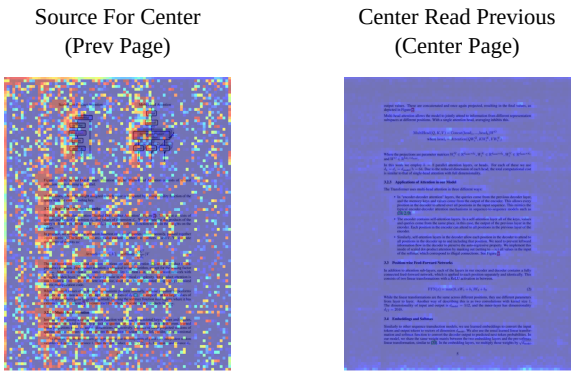


Figure 5: **Visualization of gate collapse due to post-encoder representation leakage.** When trained with self-supervised boundary reconstruction post-encoder, the receive gates on the center page (*right*) collapse to near zero (dark blue), shutting off cross-page attention even though the source features on the previous page (*left*) contain valid content signals. Because global self-attention inside the per-page vision transformer allows unmasked tokens to leak context to boundary tokens before feature extraction, the module trivially reconstructs masked boundary tokens from same-page context alone.

**Mechanism.** Modern VLM vision towers (such as NaViT or ViT backbones) employ global intra-page self-attention across all visual patches of a single page image. Consequently, by the time visual

representations reach the post-encoder stage, tokens at the top or bottom boundaries of page $p$ have already absorbed rich global context from unmasked interior tokens on the same page.

When a self-supervised reconstruction loss is applied on post-encoder visual features:

1. The module discovers that masked boundary features can be accurately predicted using unmasked features from the *same* page.
2. Neighbor pages, operating over distinct visual coordinate spaces, contribute zero incremental predictive value for same-page feature restoration and instead introduce inter-page noise.
3. As a result, the optimization process drives the receive gate logits ($r^{\leftarrow}, r^{\rightarrow}$) toward large negative values ($\sigma(r) \rightarrow 0$). As visualized in Figure 5, the receive gates completely close, collapsing cross-page attention to zero.

This finding confirms that self-supervised feature reconstruction is fundamentally mismatched for post-encoder injection in VLM architectures. It directly motivates our supervised Stage A training strategy: by using task-level supervised cross-entropy loss from the language model on cross-page parsing and alignment prompts, the LLM forces DocSEAM to extract genuine cross-page semantic dependencies that cannot be resolved from intra-page vision encoder representations alone.

## J Extended Layer-Wise Attention Trace Analysis

To complement the early-layer attention trace presented in Section 4.2, we visualize the progression of LLM layer-wise attention maps across both shallow and deep layers (Layers 3, 7, 11, and 19) in Figure 6.

**Observations:**

1. **Early Convergence (Layers 3-7):** At Layer 3, both models show initial exploratory attention across the page body. By Layer 7, DocSEAM already exhibits strong, localized activation on the target text span across the page boundary, whereas the baseline remains scattered without a clear focal point.
2. **Sustained Focus at Deep Layers (Layers 11-19):** In deeper layers (Layer 11 and Layer 19), DocSEAM maintains a high signal-to-noise ratio, keeping attention tightly bounded to the cross-page evidence required for answer generation. While the baseline eventually locates the region at Layer 11, its attention map at Layer 19 displays residual activation across irrelevant text regions, reflecting the additional computational burden placed on the LLM when resolving feature-level cross-page discontinuities without post-encoder enrichment.

## K Additional Directional Salience Case Study

Figure 7 presents an additional qualitative case study illustrating the attention routing mechanism of DocSEAM across a complex financial prospectus document containing a multi-page table split by a page break.

**Observations:**

1. **Cross-Boundary Table Alignment (Prev $\rightarrow$ Center):** The previous page (*Source For Center*, left) ends with an incomplete table detailing financial fund allocations. As shown in the *Center Read Previous* panel, when the center page attends backward, attention activates intensely (highlighted in bright red/yellow) over the table continuation at the top of the center page. This confirms that DocSEAM successfully learns to bridge tabular data across physical page breaks.
2. **Forward Directional Continuity (Center $\rightarrow$ Next):** When attending forward (*Center Read Next*), activation shifts down toward the lower body text of the center page and top of the next page where section paragraphs continue overleaf, demonstrating dynamic, direction-dependent attention allocation.

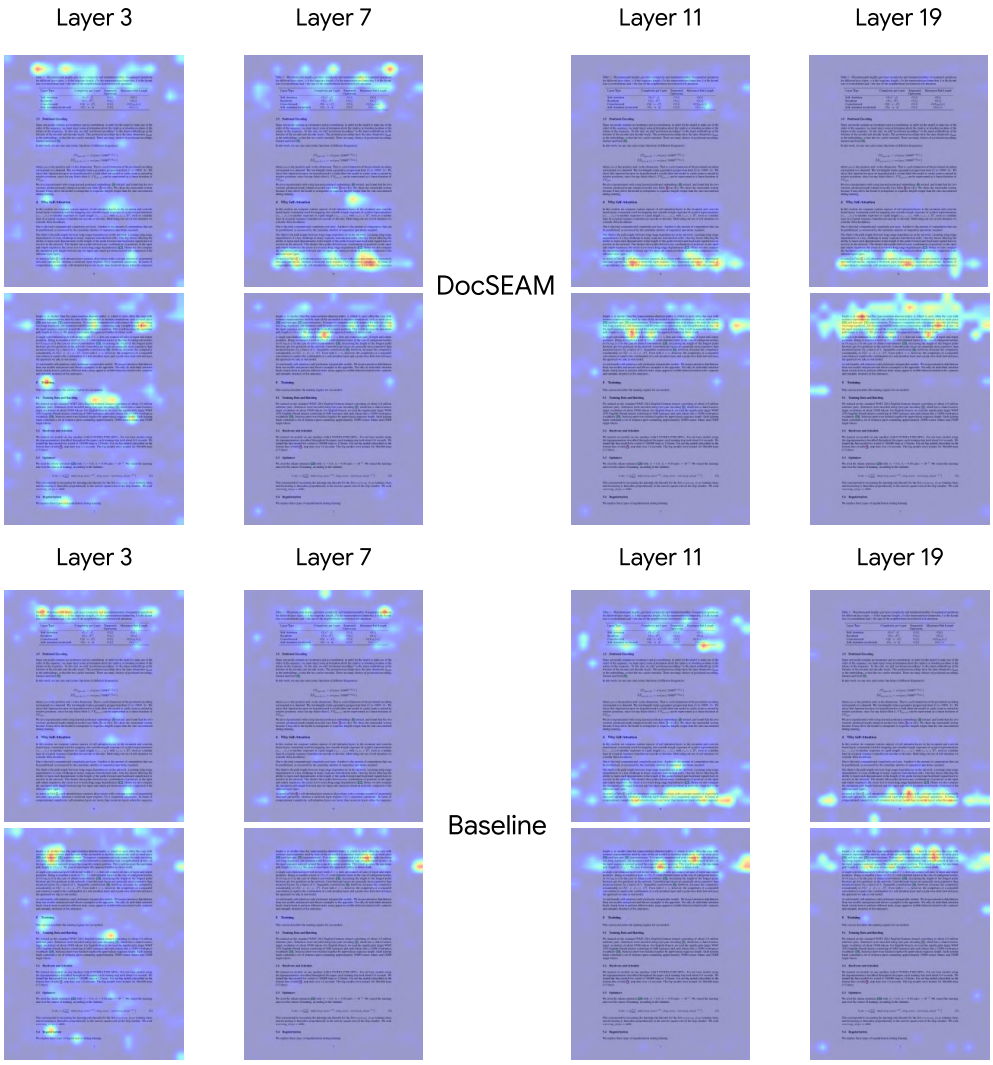


Figure 6: **Extended layer-wise attention trace across deep LLM layers.** Comparison of attention maps between DocSEAM (top) and the stateless baseline (bottom) at Layers 3, 7, 11, and 19 for a cross-page span localization query. With DocSEAM, attention sharply concentrates on the boundary-spanning ground-truth evidence as early as Layer 7 and maintains a sharp, focused pattern through Layer 19. In contrast, the baseline exhibits diffuse attention at shallow layers (Layers 3 and 7), only forms a relevant pattern around Layer 11, and retains noticeable background noise even at Layer 19.

3. **Selective Margin & Header Suppression:** Across all panels, blank side margins, running headers, and page numbers receive near-zero attention (dark blue). This validates that the directional salience gates act as content-driven routers rather than rigid positional heuristics.

## L Qualitative Answer Comparisons

To visually inspect how DocSEAM improves multi-page document reasoning over baseline models, we examine representative qualitative examples where answering the question requires integrating information across adjacent pages. Figures 8 and 9 present side-by-side comparisons of predic-

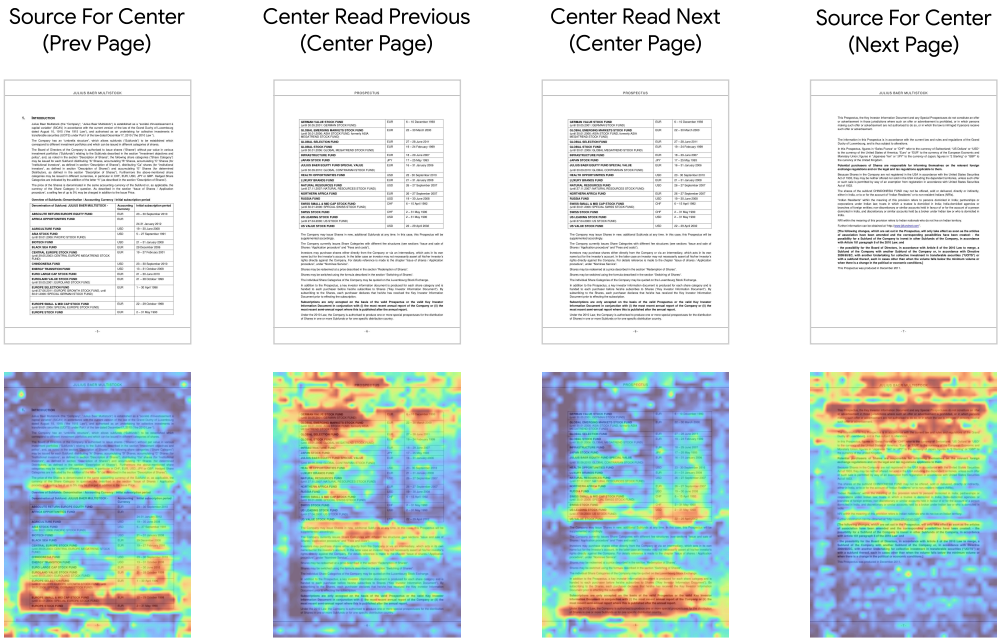


Figure 7: **Additional case study of directional salience gate activation across a table continuation.** Attention heatmaps across three consecutive document pages (Prev, Center, Next) containing a multi-page table split across a page boundary. DocSEAM dynamically activates cross-page attention to connect table rows truncated at the bottom of the previous page with their continuation at the top of the center page, while suppressing uninformative margins.

tions from DocSEAM, the matched baseline, and URaG, alongside ground-truth annotations and document evidence.

**Case 1: Cross-Page & Cross-Modal Evidence Integration (Figure 8).** In this example from a multi-page health report, the question asks for the demographic group accounting for the highest healthcare expenditure. The textual definition resides on Page 13, while the supporting quantitative chart is located on Page 14. DocSEAM effectively connects the visual tokens across the boundary between Pages 13 and 14, extracting the correct demographic (*"60 and over age group"*) and grounding its explanation with exact page references. In contrast, the stateless baseline confuses page citations (citing Page 9 and Page 10 instead of 13 and 14) and hallucinates multiple risk groups by failing to correlate the text continuation with the chart on the following page. URaG outputs a brief, ungrounded fragment (*"Individuals with chronic diseases"*).

**Case 2: Structural Tracking Across Section Boundaries (Figure 9).** This example evaluates structural section tracking over document page breaks. The prompt asks for the title of the section following Section 6 (*"Emergency Response"*). Section 6 concludes at the bottom of Page 50, and Section 7 begins at the top of Page 51. Because DocSEAM enriches boundary tokens with bidirectional neighbor context, it seamlessly connects the conclusion of Section 6 on Page 50 to the heading of Section 7 on Page 51, retrieving the complete section title (*"7 How do you recover and rehabilitate your property after a disaster?"*). Conversely, the baseline model only produces a generic label (*"Section 7"*), and URaG returns only a single word (*"Recovery"*), missing the complete heading required by the ground truth.

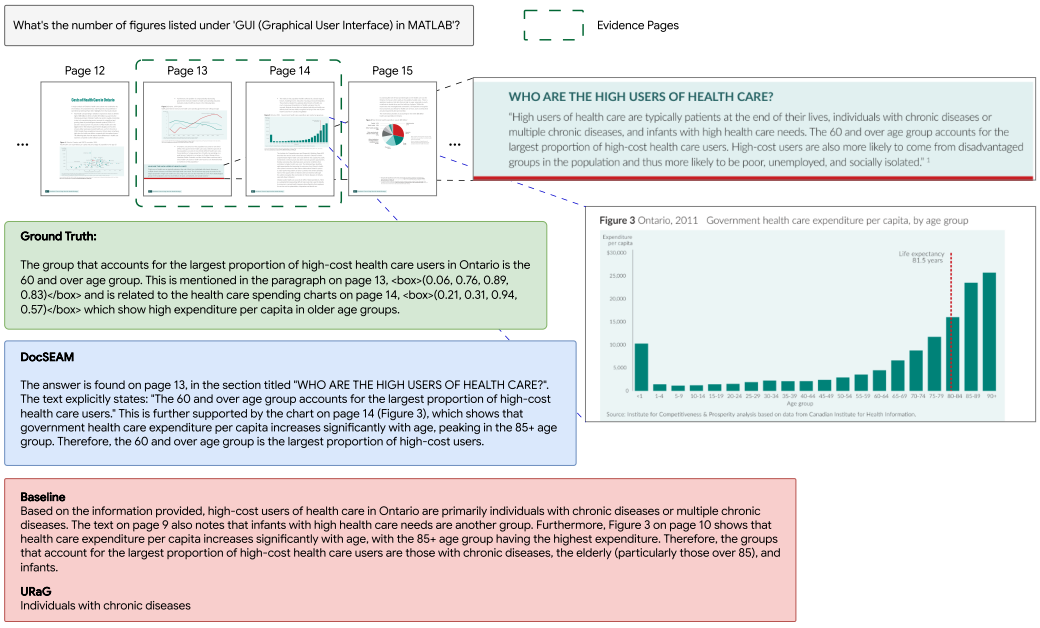


Figure 8: **Qualitative comparison on multi-page cross-modal QA.** Given a question spanning a text description (Page 13) and a supporting chart (Page 14), DocSEAM correctly identifies the target demographic group with precise page attribution, whereas the baseline hallucinates page citations and URaG produces an ungrounded fragment.

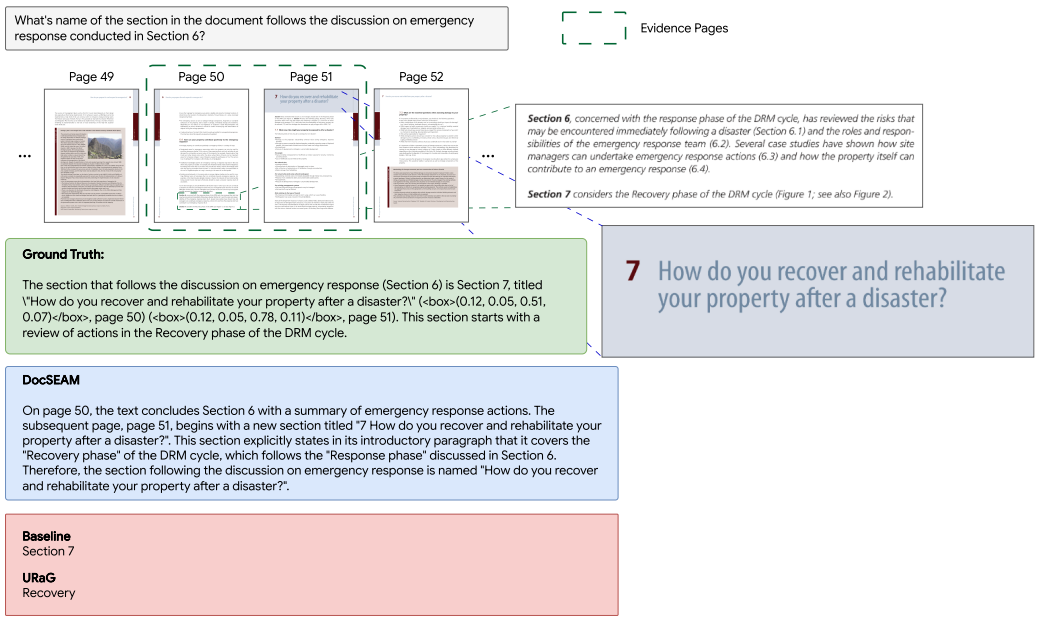


Figure 9: **Qualitative comparison on section tracking across page breaks.** When identifying a section title continuing over a page boundary (Pages 50-51), DocSEAM extracts the complete section title and context, while baseline models fail to retrieve the full title.