DERIVE: DEPENDENCY-AWARE AND EVIDENCE-RESPONSIVE REASONING VIA INTERVENTION AND VERIFIABLE EXECUTION FOR MLLM LONG-DOCUMENT UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) typically tackle long-document reasoning by retrieving sparse evidence from abundant irrelevant pages before reasoning over it. Yet our analysis shows that substantial errors persist even when gold evidence is directly provided, shifting the challenge from locating evidence to reasoning reliably from it. However, existing methods mainly verify final outcomes or individual reasoning states, which does not establish whether downstream reasoning actually depends on the answer-determining evidence. We therefore propose **D**ependency-aware **E**vidence-responsive **R**easoning via **I**ntervention and **V**erifiable **E**xecution (**DERIVE**), which makes evidence-to-reasoning dependencies explicit and executable through Verifiable Execution (VE) chains, and makes them directly testable through controlled Evidence Intervention (EI). VE chains expose evidence provenance and intermediate execution to deterministic verification, while EI modifies answer-determining evidence and re-executes the affected downstream reasoning to produce verifiable dependency responses. DERIVE is trained with VE-chain SFT establishing the grounded executable reasoning format, followed by GRPO with executable verification over original and intervened evidence states to reinforce grounded, executable, and evidence-responsive reasoning. Extensive experiments on six public benchmarks demonstrate that DERIVE consistently outperforms competitive baselines, especially on arithmetic-intensive tasks requiring evidence-dependent reasoning. Code will be released upon acceptance.

1 INTRODUCTION

Document understanding underpins many real-world applications, including information extraction, contract analysis, and report processing Ke et al. (2025). Recent multimodal large language models (MLLMs) have achieved strong performance on single-page document understanding, substantially advancing document perception and question answering Zhu et al. (2025). The frontier is now shifting toward long, multi-page documents, where models must integrate information distributed across pages rather than reason within a single visual context Xiong et al. (2026b). In this setting, answer-relevant evidence is often sparse and scattered across a large amount of irrelevant content, making reliable cross-page reasoning a central challenge Tanaka et al. (2025).

To cope with sparse evidence in long documents, existing approaches have largely focused on improving evidence localization before generation. Early retrieval-augmented methods Radford et al. (2021) rely on dedicated text or multimodal retrievers to identify relevant pages Faysse et al. (2025); Zheng et al. (2026), while subsequent work introduces page selection, pruning, or compression Hu et al. (2025) to reduce the visual context processed by the MLLM. More recent approaches Shi et al. (2026) further integrate retrieval into the MLLM itself, exploiting intermediate representations to score and retain relevant pages during inference. Together, these developments have substantially strengthened evidence localization. As shown in Figure 1(a), lightweight internal retrieval already matches or outperforms dedicated external retrievers across multiple benchmarks.

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.

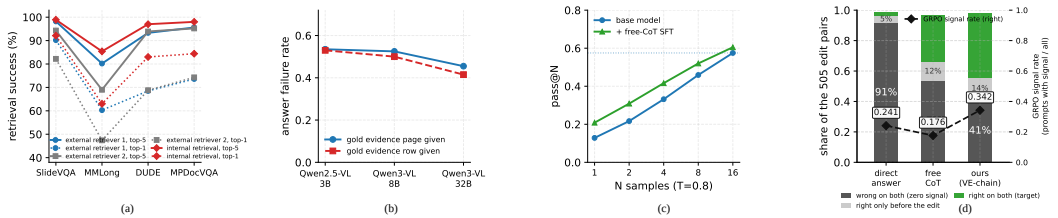


Figure 1: Diagnosing long-document understanding beyond retrieval. (a) Internal and external retrieval performance across four benchmarks. (b) Answer failure rates with gold evidence pages or rows. (c) Pass@N on greedily failed TAT-DQA examples. (d) Outcome distribution on 505 paired evidence edits and GRPO signal rate, defined as the fraction of prompts with non-uniform rollout rewards.

We therefore ask whether retrieval remains the primary bottleneck in long-document reasoning. To probe its upper bound, we directly provide the gold evidence page or even the gold evidence row. As shown in Figure 1(b), roughly half of the questions remain incorrect with the gold page, while more than 40% still fail with row-level evidence. These results show that even oracle evidence localization leaves substantial post-retrieval errors unresolved. One possible explanation is that models simply lack the reasoning capability required after retrieval. However, Figure 1(c) shows a large gap between greedy accuracy and Pass@N, indicating that correct solutions are often reachable through sampling even when the greedy prediction fails.

If correct solutions already appear in the sampling space, a natural strategy is to supervise the reasoning process with Chain-of-Thought (CoT) Wei et al. (2022). Yet Figure 1(d) shows that free-form CoT still leaves outcome-level GRPO supervision highly sparse, since trajectories with the same final outcome receive identical rewards regardless of their intermediate reasoning. Existing structured approaches Xiong et al. (2026b); Yan et al. (2026) provide richer supervision through evidence localization, grounding, or structured intermediate states to relieve this problem.

However, verifying individual states does not establish the dependency between them, where a grounded value or intermediate result can be correct without showing that downstream reasoning actually depends on the cited evidence. The key question is therefore how to verify evidence-to-reasoning dependencies rather than individual states. Our key insight is to make evidence-to-reasoning dependency directly testable through intervention: changing an answer-determining evidence variable should trigger corresponding recomputation in its dependent reasoning states. This idea is illustrated schematically as follows:

$$\underbrace{E,\, X,\, a}_{\text{state verification}} \;\not\Rightarrow\; \underbrace{E \rightsquigarrow X \rightsquigarrow a}_{\text{dependency verified}}, \qquad \underbrace{E \to E'}_{\text{intervention}} \;\Rightarrow\; \underbrace{X \to X' \Rightarrow a \to a'}_{\text{evidence-responsive dependency}} .$$

To this end, we propose **D**ependency-aware **E**vidence-responsive **R**easoning via **I**ntervention and **V**erifiable **E**xecution (**DERIVE**). First, Verifiable Execution (VE) chain represents grounded evidence as explicit intermediate variables and connects them through executable operations, making evidence-to-reasoning dependencies explicit and downstream reasoning deterministically re-executable. Second, Evidence Intervention (EI) modifies answer-determining evidence and re-executes the affected downstream reasoning, making these dependencies directly testable through their induced responses. Together, VE and EI extend process supervision from verifying individual reasoning states to verifying how reasoning depends on and responds to its underlying evidence.

We realize DERIVE through VE-chain SFT followed by GRPO with executable verification over original and intervened evidence states. SFT establishes the grounded executable reasoning format, while the executable verifier provides GRPO with process-level feedback on evidence reading, intermediate execution, and final-answer correctness. Intervened states further expose whether downstream reasoning responds consistently to changes in answer-determining evidence, allowing policy optimization to reinforce reasoning that is both executable and evidence-responsive.

In summary, our main contributions are as follows.

1. We reveal a post-retrieval supervision bottleneck: substantial errors persist with oracle evidence, while outcome-only supervision provides sparse signals and cannot verify evidence-to-reasoning dependencies.

2. We propose DERIVE, combining Verifiable Execution (VE) chains and Evidence Intervention (EI), where VE makes evidence-to-reasoning dependencies explicit and executable, while EI makes them directly testable under controlled evidence changes.
3. We realize DERIVE through VE-chain SFT followed by dependency-aware evidence responsive GRPO, using executable verification over original and intervened evidence states to provide both state- and dependency-level process learning.

2 Related Work

2.1 Retrieval-based MLLMs for Long-Document Understanding

Early approaches to multi-page document understanding largely build on encoder–decoder architectures such as T5 Raffel et al. (2020), addressing long inputs through hierarchical document representations. For example, Hi-VT5 Tito et al. (2023) summarizes each page into a set of special [PAGE] tokens and performs hierarchical decoding over the resulting page-level representations.

With the emergence of MLLMs, document understanding has increasingly shifted toward general multimodal models that directly reason over visual document inputs, benefiting from stronger joint modeling of textual, visual, and layout information Zhu et al. (2022); Wei et al. (2025). However, extending such models to long documents introduces substantial visual-token redundancy and computational overhead, as only a small fraction of pages typically contain answer-relevant evidence. One line of work reduces the input burden through visual-token compression, allowing more document pages to fit within the model context while preserving fine-grained document information. For example, Leopard Jia et al. (2024) adaptively allocates visual token sequences according to image resolution and aspect ratio, enabling more efficient processing of multiple high-resolution text-rich images. mPLUG-DocOwl2 Hu et al. (2025) introduces a high-resolution document compressor that represents each page with a compact set of visual tokens, improving the efficiency of multi-page document understanding. Another line explicitly retrieves relevant pages before generation Li et al. (2026). CREAM Zhang et al. (2024) adopts a coarse-to-fine pipeline that combines embedding-based retrieval with iterative LLM-based ranking, while M3DocRAG Cho et al. (2024) employs a multimodal retriever to identify relevant pages before passing them to the MLLM. SV-RAG Chen et al. (2024), for instance, adapts the MLLM itself as a visual retriever to select evidence pages before answering the query. More recently, URaG Shi et al. (2026) further integrates retrieval into the MLLM inference process, leveraging intermediate hidden representations to select relevant pages and discard irrelevant ones before deeper-layer generation.

This progression suggests that evidence localization is becoming increasingly intrinsic to the MLLM itself rather than relying solely on a separate retrieval system. These methods substantially improve evidence localization and computational efficiency, but primarily address which evidence should be retained for downstream reasoning. Our work instead focuses on what happens after relevant evidence has been identified, asking whether subsequent reasoning reliably depends on that evidence.

2.2 Structured Reasoning for Long-Document Understanding

Recent work has moved beyond direct answer prediction toward explicit evidence-guided reasoning for long-document understanding Han et al. (2025); Zhu et al. (2026). DocR1 Xiong et al. (2026b) introduces Evidence Page-Guided GRPO to encourage a coarse-to-fine process that first localizes relevant pages before answering. DocSeeker Yan et al. (2026) further structures reasoning into an Analysis-Localization-Reasoning workflow and combines CoT distillation with evidence-aware GRPO to jointly optimize evidence localization and answer accuracy. Moving to finer-grained grounding, LAT Liu et al. (2026) aligns reasoning references with page- and region-level evidence and rewards attribution consistency, while DocScope Feng et al. (2026) evaluates long-document reasoning through page localization, region grounding, fact extraction, and answer verification. Together, these works demonstrate the value of exposing and verifying intermediate reasoning states rather than supervising final answers alone.

However, viewed through our lens, such supervision remains largely *state-based*: it verifies whether evidence is localized or grounded, whether an intermediate fact is supported, or whether the final answer is correct under a fixed document state Xiong et al. (2026a); Zheng et al. (2026). Correct-

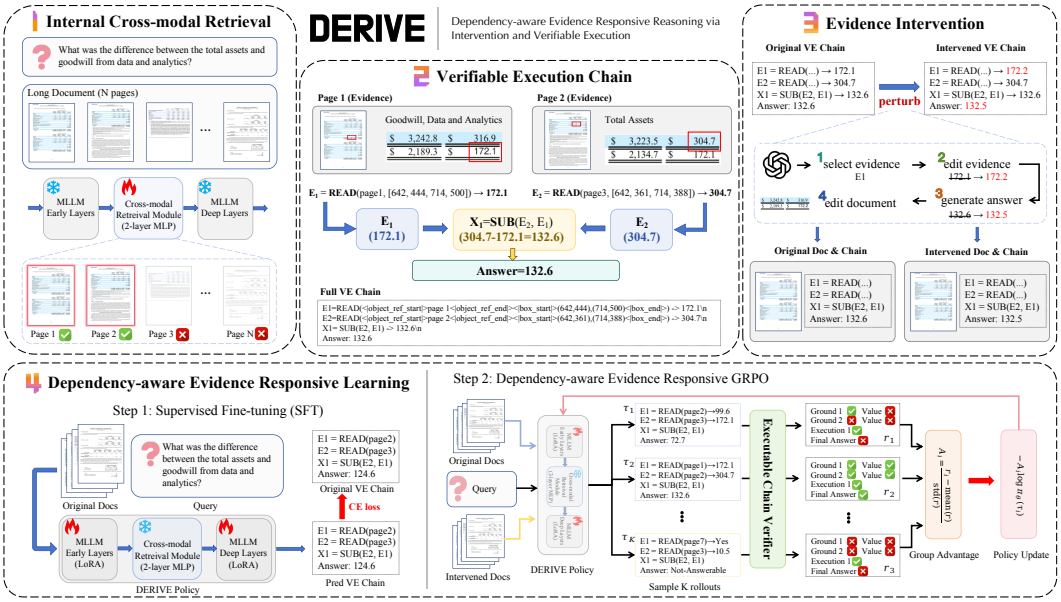


Figure 2: Overview of our proposed DERIVE. (1) Internal cross-modal retrieval localizes relevant evidence pages. (2) Verifiable Execution (VE) chains make evidence-to-reasoning dependencies explicit and executable through grounded reads and deterministic operations. (3) Evidence Intervention (EI) makes these dependencies directly testable by modifying answer-determining evidence and re-executing the affected downstream reasoning. (4) Dependency-aware Evidence-Responsive Learning combines VE-chain SFT and GRPO with executable verification to turn state and dependency verification into evidence-responsive supervision.

ness of these individual states does not establish the dependency between them. In particular, it does not test whether downstream reasoning would update correctly when the answer-determining evidence changes. DERIVE targets this missing dimension by representing grounded evidence and intermediate reasoning through executable dependencies and making those dependencies directly testable through controlled evidence interventions. This shifts verification from asking whether individual reasoning states are correct to whether reasoning actually depends on, and responds to, its underlying evidence.

3 METHOD

DERIVE equips an MLLM with explicit and verifiable evidence-to-reasoning dependencies, making downstream reasoning executable and directly testable under controlled evidence interventions. As illustrated in Figure 2, DERIVE consists of four stages. First, an internal cross-modal retrieval module prunes a long document to a small set of relevant pages within the model forward pass (§3.1). Second, DERIVE represents grounded evidence and downstream computation as Verifiable Execution (VE) chains, making evidence provenance and intermediate operations explicit and deterministically executable (§3.2). Third, Evidence Intervention modifies answer-determining evidence and re-executes the affected downstream computation through the same dependency, yielding aligned original and intervened VE chains with verifiable responses (§3.3). Finally, DERIVE adopts a two-stage learning scheme in which VE-chain SFT establishes the grounded executable reasoning format, followed by GRPO with executable verification over original and intervened evidence states (§3.4).

3.1 INTERNAL CROSS-MODAL RETRIEVAL

We adopt the internal retrieval mechanism of URaG Shi et al. (2026) to localize relevant pages before performing document reasoning. Given a long document $D = \{p_i\}_{i=1}^{N}$ and a question q, let $\mathbf{H}^{(L)}$

denote the multimodal hidden states after an early MLLM layer L. A lightweight retrieval head g_{ret}, implemented with two linear projections with a GELU Hendrycks & Gimpel (2016) activation, maps these hidden states into a lower-dimensional retrieval space, followed by ℓ_2 normalization. We denote the resulting question-token representations as $\mathbf{Q} = \{\mathbf{q}_t\}_{t=1}^{T_q}$ and the visual-token representations of page p_i as $\mathbf{V}_i = \{\mathbf{v}_{i,u}\}_{u=1}^{T_i}$, where T_q and T_i denote the numbers of question and visual tokens, respectively. Following contextualized late interaction, the relevance score of page p_i is defined as:

$$s_i = \sum_{t=1}^{T_q} \max_{1 \le u \le T_i} \mathbf{q}_t^\top \mathbf{v}_{i,u}. \tag{1}$$

Let $\mathcal{I}_m$ denote the indices of the m highest-scoring pages and $D_{\text{ret}} = \{p_i \mid i \in \mathcal{I}_m\}$ the retained document context. Visual tokens from all remaining pages are removed before the subsequent Transformer layers, so downstream reasoning is performed only over D_{ret} and the question.

We train g_{ret} while keeping the MLLM fixed, following the retrieval objective of URaG. After this stage, the retrieval module is frozen throughout VE-chain SFT and GRPO. The retrieval training objective and hard-negative construction are provided in Appendix A.5.

3.2 Verifiable Execution Chain

Given the retrieved document context D_{ret}, DERIVE represents reasoning as a Verifiable Execution (VE) chain, which connects grounded evidence reads to downstream computation through explicit executable dependencies. Formally, a VE chain $\mathcal{C}$ consists of evidence nodes, computation nodes, and a final answer, which are defined as

$$E_k = \textsc{Read}(p_{i_k}, b_k) \to e_k, \qquad X_j = \textsc{Op}_j(\mathbf{z}_j) \to x_j, \qquad \textsc{Answer} \to a, \tag{2}$$

where $p_{i_k} \in D_{\text{ret}}$ denotes the source page indexed by $i_k \in I_m$, b_k is the grounded region, and e_k is the value read from that region by evidence node E_k. Each computation node X_j applies a deterministic operator $\textsc{Op}_j$ to arguments $\mathbf{z}_j$ that reference preceding evidence values, computation results, or literal constants. These references induce an explicit dependency structure from grounded evidence to intermediate computation and the final answer.

Once the READ values are grounded, all downstream operations can be deterministically re-executed without invoking the MLLM. VE chains therefore expose both evidence provenance and execution semantics to direct verification, rather than relying on the linguistic plausibility of a free-form rationale. We retain only chains whose grounded values pass local verification and whose execution recovers the gold answer; construction and serialization details are provided in Appendix A.2. Because the resulting dependencies are explicit and executable, modifying a grounded evidence node yields a well-defined set of downstream states that must be recomputed, which enables the evidence intervention introduced next.

3.3 Evidence Intervention

VE chains make evidence-to-reasoning dependencies explicit, but an explicit dependency is not necessarily a verified one. DERIVE therefore makes these dependencies directly testable through Evidence Intervention (EI). Given an original VE chain $\mathcal{C}$, we select an answer-determining evidence node:

$$E_r = \textsc{Read}(p_{i_r}, b_r) \to e_r \tag{3}$$

and then replace its grounded value with an edited value e'_r. Only the corresponding document region is modified while the question and all remaining content are kept unchanged, producing an intervened document:

$$D' = \mathcal{I}(D; p_{i_r}, b_r, e'_r). \tag{4}$$

where $\mathcal{I}$ denotes the intervention operator. The intervened reasoning target is not generated independently. Instead, DERIVE preserves the execution structure of $\mathcal{C}$ and re-executes only the states that depend on E_r. Let $\mathcal{A}_r$ denote the downstream computation nodes reachable from E_r in $\mathcal{C}$. Nodes outside $\mathcal{A}_r$ remain unchanged, while nodes in $\mathcal{A}_r$ are deterministically recomputed after substituting e_r with e'_r. This produces an intervened VE chain $\mathcal{C}'$ and answer a':

$$e_r \to e'_r \quad \Longrightarrow \quad \mathcal{C} \to \mathcal{C}' \quad \Longrightarrow \quad a \to a'. \tag{5}$$

EI therefore turns a latent evidence dependency into an observable target response. The aligned original and intervened states $(D, \mathcal{C})$ and $(D', \mathcal{C}')$ provide the dependency-aware supervision used in Sec. 3.4.

3.4 Dependency-aware Evidence-Responsive Learning

DERIVE converts executable reasoning states and intervention-induced dependency responses into training signals through a two-stage learning scheme. We first use VE-chain SFT to establish the grounded executable reasoning format, and then optimize the policy with GRPO over both original and intervened evidence states using executable verification as process-level feedback.

VE-chain SFT. For each training instance (D, q), let $\mathbf{c} = \mathrm{Ser}(\mathcal{C}) = (c_1, \ldots, c_T)$ denote the serialized verified original VE chain. We optimize the standard autoregressive objective, which is defined as:

$$\mathcal{L}_{\mathrm{SFT}} = -\frac{1}{T} \sum_{t=1}^{T} \log \pi_\theta \left(c_t \mid D, q, c_{<t} \right). \tag{6}$$

This stage initializes the policy to produce grounded READ nodes and executable operations in the VE format before policy optimization.

Dependency-aware Evidence-Responsive GRPO. We further optimize the SFT-initialized policy over both original and intervened evidence states, which can be denoted as:

$$(\widetilde{D}, \widetilde{\mathcal{C}}) \in \{(D, \mathcal{C}), (D', \mathcal{C}')\} \tag{7}$$

For each state, we sample K trajectories $\{\tau_i\}_{i=1}^{K}$ from $\pi_\theta(\cdot \mid \widetilde{D}, q)$ and parse each trajectory into a predicted VE chain $\hat{\mathcal{C}}_i$. An executable verifier $\mathcal{V}$ evaluates the predicted chain against its corresponding verified target:

$$r_i = \mathcal{V}\left(\hat{\mathcal{C}}_i, \widetilde{\mathcal{C}}\right), \qquad A_i = \frac{r_i - \bar{r}}{s_r + \epsilon}, \qquad \bar{r} = \frac{1}{K} \sum_{j=1}^{K} r_j, \tag{8}$$

where s_r is the within-group reward standard deviation and $\epsilon > 0$ is a small constant for numerical stability. The verifier checks answer correctness and executable-chain consistency for all states. For intervened states, it additionally verifies whether the edited evidence is correctly read and whether its affected downstream computation is re-executed consistently. Because $\mathcal{C}'$ is deterministically derived from $\mathcal{C}$ under the intervention $e_r \to e'_r$, this feedback evaluates not only individual reasoning states but also the model's response to the underlying evidence dependency. The complete verifier reward is provided in Appendix A.4 and requires no learned reward model.

The group-relative policy objective is defined as:

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{K} \sum_{i=1}^{K} A_i \log \pi_\theta \left(\tau_i \mid \widetilde{D}, q \right). \tag{9}$$

Original and intervened states share the same policy, verifier, and optimization rule. Intervention changes the evidence state and its deterministically induced target response, allowing executable verification to reinforce reasoning that is both correct under each state and responsive to the dependency between them.

Overall, DERIVE is trained in three stages. We first train the lightweight retrieval module with the MLLM frozen, then fix the retriever and perform VE-chain SFT, followed by dependency-aware GRPO over original and intervened evidence states. These stages establish evidence localization, executable reasoning, and evidence-responsive policy optimization, respectively. The complete procedure is summarized in Algorithm 1 in Appendix A.5.

4 Experiments

4.1 Experimental Settings

Datasets & Benchmarks. We evaluate DERIVE on six document understanding benchmarks, including MPDocVQA Tito et al. (2023), DUDE Van Landeghem et al. (2023), SlideVQA Tanaka

Table 1: Comparison with state-of-the-art methods on six long-document understanding benchmarks. DERIVE is evaluated with Qwen2.5-VL-3B and 7B backbones and compared withgeneral-purpose MLLMs and document-specific methods. Results marked with † are taken from previously published results Xiong et al. (2026b); Shi et al. (2026), while the remaining results are reproduced under our evaluation setup. Higher is better for all metrics, and "–" denotes unavailable results.

Methods	#Param	Venue	MPDocVQA	DUDE	SlideVQA	TAT-DQA	LongDocURL
Llama-3.2 Meta AI (2024)	11B	-	57.6	20.8	-	-	9.2
CREAM † Zhang et al. (2024)	14B	ACM MM'24	65.3	52.5	-	-	-
InternVL3 Zhu et al. (2025)	8B	-	79.4	47.6	51.7	28.8	36.8
Qwen2.5-VL Bai et al. (2025)	3B	-	84.4	50.6	59.1	39.7	40.0
Qwen2.5-VL Bai et al. (2025)	7B	-	87.2	55.0	66.4	42.1	<u>51.1</u>
SV-RAG † Chen et al. (2024)	4B	ICLR'25	71.0	45.0	-	-	-
mPLUG-DocOwl2 † Hu et al. (2025)	8B	ACL'25	67.9	31.2	29.6	22.2	5.3
VDocRAG † Tanaka et al. (2025)	8B	CVPR'25	62.6	44.0	42.0	-	39.8
URaG Shi et al. (2026)	3B	AAAI'26	86.1	53.9	62.8	42.3	40.6
URaG Shi et al. (2026)	7B	AAAI'26	**87.8**	55.1	68.2	46.7	49.7
DocR1 † Xiong et al. (2026b)	7B	AAAI'26	87.4	54.4	**71.9**	64.8	-
DERIVE (Qwen2.5-VL base)	3B	-	86.2	<u>55.2</u>	65.3	<u>64.8</u>	45.9
DERIVE (Qwen2.5-VL base)	7B	-	<u>87.7</u>	**58.5**	<u>69.4</u>	**68.5**	**53.1**

Table 2: Results on MMLongBench-Doc across evidence modalities and locations. TXT, LAY, CHA, TAB, and IMG denote text, layout, chart, table, and image evidence, respectively; SIN, MUL, and UNA denote single-page, multi-page, and unanswerable questions. Results marked with † are taken from previously published results Xiong et al. (2026b); Shi et al. (2026), while the remaining results are reproduced under our evaluation setup. Overall ACC reports the benchmark-level accuracy.

Methods	#Param	Evidence Modalities					Evidence Locations			Overall
		TXT	LAY	CHA	TAB	IMG	SIN	MUL	UNA	ACC
InternVL3 Zhu et al. (2025)	8B	28.4	26.7	13.1	20.7	24.9	29.0	16.3	26.2	24.1
Qwen2.5-VL Bai et al. (2025)	3B	29.0	26.8	18.6	17.4	22.4	31.8	15.7	27.8	25.5
Qwen2.5-VL Bai et al. (2025)	7B	30.0	26.9	22.4	20.6	22.9	33.4	<u>17.5</u>	24.7	26.2
SV-RAG † Chen et al. (2024)	4B	26.3	22.1	25.0	20.7	25.2	34.0	10.6	15.7	23.0
mPLUG-DocOwl1.5 † Hu et al. (2024)	8B	8.2	8.4	2.0	3.4	9.9	7.4	6.4	6.2	6.9
URaG Shi et al. (2026)	3B	28.1	24.8	23.2	25.4	21.0	36.8	13.9	<u>46.7</u>	31.2
URaG Shi et al. (2026)	7B	**31.9**	28.1	<u>29.4</u>	<u>27.1</u>	27.7	**40.7**	16.9	45.1	<u>33.7</u>
DERIVE (Qwen2.5-VL base)	3B	29.1	<u>28.4</u>	27.3	25.1	<u>28.0</u>	38.2	16.7	39.5	33.5
DERIVE (Qwen2.5-VL base)	7B	<u>30.5</u>	**29.2**	**29.5**	**27.8**	**29.4**	<u>39.5</u>	**18.4**	**66.0**	**35.6**

et al. (2023), MMLongBench-Doc Ma et al. (2024), LongDocURL Deng et al. (2025), and TAT-DQA Zhu et al. (2022), covering multi-page question answering, slide understanding, long-document comprehension, and arithmetic-intensive document reasoning. Following the standard evaluation protocols of each benchmark, we report Average Normalized Levenshtein Similarity (ANLS) for MPDocVQA and DUDE, Exact Match (EM) for SlideVQA and TAT-DQA, Generalized Accuracy and F1-score for MMLongBench-Doc, and Generalized Accuracy for LongDocURL. We follow the official data splits for all benchmarks. DERIVE is trained on the training sets of MP-DocVQA, DUDE, SlideVQA, and TAT-DQA. MMLongBench-Doc and LongDocURL provide no training split and are therefore held out entirely from training and used only for evaluation. All results are reported on the official test splits.

Implementation Details. We instantiate DERIVE with Qwen2.5-VL-3B and Qwen2.5-VL-7B Bai et al. (2025) as backbones. Following URaG, internal page retrieval is performed at layer 6 for the 3B model and layer 5 for the 7B model, retaining the top-5 pages for subsequent generation. The vision tower and retrieval module are frozen throughout training. All experiments are conducted in bfloat16 on 8 $\times$ NVIDIA H200 GPUs.

For SFT, we apply LoRA Hu et al. (2021) to the seven language-model projections with rank 32, scaling factor 64, and dropout 0.1, and train for one epoch with AdamW Loshchilov & Hutter (2017) at a learning rate of 5×10^{-5}. Starting from the SFT checkpoint, we perform GRPO Shao et al. (2024) with $K = 8$, temperature 0.9, top-p 0.95. GRPO updates only the language-model LoRA parameters using AdamW with a fixed learning rate of 10^{-5} and group-normalized rewards, while

keeping the visual encoder and retrieval module frozen. For the 3B model, this corresponds to 59.87M trainable parameters, or 1.57% of the full model.

4.2 Main Results

Table 1 compares DERIVE with general-purpose MLLMs and recent document-specific methods across long-document understanding benchmarks. DERIVE-7B achieves the best results on DUDE, TAT-DQA, and LongDocURL with 58.5, 68.5, and 53.1, respectively, while remaining competitive on MPDocVQA and SlideVQA. Under controlled backbone comparisons, DERIVE improves Qwen2.5-VL-3B by 4.6, 6.2, 25.1, and 5.9 points on DUDE, SlideVQA, TAT-DQA, and LongDocURL, and improves the 7B backbone by 3.5, 3.0, 26.4, and 2.0 points, respectively, while maintaining comparable MPDocVQA performance. The particularly large gains on TAT-DQA highlight the benefit of DERIVE for evidence-dependent computation, while improvements on the remaining benchmarks show that the gains extend beyond arithmetic reasoning.

Table 2 evaluates DERIVE across evidence modalities and locations on MMLongBench-Doc. DERIVE achieves the strongest overall accuracy, reaching 33.5 and 35.6 with the 3B and 7B backbones, improving the corresponding Qwen2.5-VL baselines by 8.0 and 9.4 points. Compared with URaG under the same model scale and internal retrieval mechanism, DERIVE further improves overall accuracy by 2.3 points at 3B and 2.1 points at 7B. The 7B model also performs strongly across layout, chart, table, image, multi-page, and unanswerable subsets, with a particularly large gain on unanswerable questions from 45.1 to 66.0. These results show that DERIVE's gains extend across diverse evidence types and beyond answerable single-page cases.

Table 3: Evidence responsiveness on 505 paired TAT-DQA evidence edits. "Orig." and "Interv." report EM; "Both ✓" requires both states to be correct; the final column reports intervention accuracy conditioned on a correct original prediction. EI denotes Evidence Intervention.

Methods	Orig.	Interv.	Both ✓	P(interv.✓|Orig.✓)
Qwen2.5-VL 3B	9.1	4.4	2.8	30.4
free-CoT	30.3	25.7	21.6	71.2
free-CoT+GRPO	51.2	46.1	41.8	79.6
ours				
+ VE-chain SFT	53.5	45.6	41.0	76.7
+ GRPO (w/o EI)	<u>56.2</u>	<u>49.1</u>	<u>45.7</u>	<u>81.3</u>
+ GRPO (w/ EI)	**58.2**	**51.9**	**48.1**	**82.7**

4.3 Analysis of VE Chain & Evidence Intervention

Table 4: TAT-DQA breakdown by answer type. EM is reported for arithmetic, span, multi-span, and count questions. EI denotes Evidence Intervention.

Methods	Arithmetic	Span	Multi-span	Count
Qwen2.5-VL 3B	10.0	66.5	69.8	0.0
free-CoT	34.7	45.7	26.4	25.0
free-CoT+GRPO	47.1	67.1	**81.1**	25.0
ours				
+ VE-chain SFT	<u>50.0</u>	68.2	69.8	**50.0**
+ GRPO (w/o EI)	49.4	**68.2**	**81.1**	<u>25.0</u>
+ GRPO (w/ EI)	**54.7**	**69.4**	<u>77.4</u>	**50.0**

Table 3 evaluates behavior under controlled changes to answer-determining evidence. VE-chain SFT reaches 41.0% joint accuracy across both states and 76.7% conditional intervention accuracy, which executable-verification GRPO improves to 45.7% and 81.3%. Adding Evidence Intervention further raises performance to 58.2 on original instances, 51.9 on intervened instances, 48.1 joint accuracy, and 82.7% conditional intervention accuracy. These results show that executable verification and intervention provide complementary supervision for evidence-responsive reasoning.

Table 4 reveals distinct effects across TAT-DQA answer types. VE-chain SFT yields the largest immediate gains on computation-heavy questions, improving arithmetic EM from 10.0 to 50.0 and count EM from 0.0 to 50.0, while changing span and multi-span performance only marginally. This pattern is consistent with VE supervision explicitly exposing grounded values and executable operations. GRPO provides complementary gains beyond supervised execution. Without EI, the largest improvement appears on multi-span questions, where EM rises from 69.8 to 81.1. Adding EI yields the best arithmetic and span results at 54.7 and 69.4, while restoring count accuracy to 50.0. Although multi-span drops to 77.4, it remains well above the base model. Overall, VE-chain supervision is most effective for explicit computation, while GRPO and EI provide complementary gains across reasoning types.

4.4 Analysis of Dependency-aware Evidence Responsive Learning

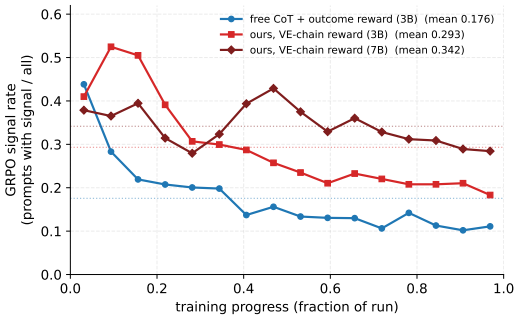


Figure 3: GRPO signal rate throughout training. VE-chain verification yields more rollout groups with non-uniform rewards than outcome-only supervision. Dashed lines denote training means.

Figure 3 examines how often GRPO receives an effective group-relative learning signal throughout training. With free-form CoT and outcome-only rewards, the signal rate quickly decreases as training progresses, averaging only 0.176 over the full run. In contrast, VE-chain verification maintains substantially more distinguishable rollout groups, increasing the average signal rate to 0.293 for the 3B model and 0.342 for the 7B model.

This difference follows directly from the granularity of the verifier. Outcome-only rewards collapse trajectories with the same final correctness into identical feedback, even when their intermediate reasoning differs. VE-chain verification can additionally distinguish trajectories through grounded evidence reading and executable intermediate steps, preserving reward variation even when the final answer alone provides no distinction. The consistently higher signal rate therefore provides a mechanistic explanation for the gains of executable process supervision in GRPO.

Figure 4 examines whether DERIVE's gains can be explained simply by additional intervention data. Adding the intervention data to outcome-level GRPO improves most benchmarks, showing that exposure to controlled evidence changes is itself useful. However, applying VE-chain verification to the same data yields further gains across all five settings, including +2.05 on TAT-DQA and +4.2 on the intervened evaluation set. Notably, on the latter, chain supervision contributes twice the gain of additional data alone (+4.2 vs. +2.1). These results separate the benefit of intervention data from that of dependency-aware supervision: exposing evidence changes provides useful examples, while VE-chain verification teaches the model how downstream reasoning should respond to those changes.

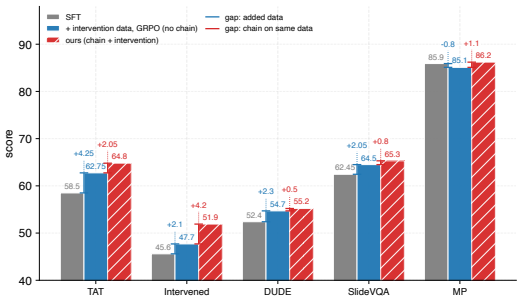


Figure 4: Effect of intervention data and VE-chain supervision. Blue annotations measure the gain from adding intervention data to GRPO, while red annotations measure the additional gain from VE-chain verification on the same data.

5 Conclusion

We presented DERIVE, a framework for improving post-retrieval reasoning in long-document understanding. Our analysis shows that strong evidence localization alone does not ensure reliable reasoning, and that outcome-only supervision provides limited signal for exploiting intermediate evidence-to-reasoning processes. DERIVE addresses this limitation by representing grounded evidence and downstream reasoning as Verifiable Execution chains, enabling deterministic process verification and controlled evidence interventions. Through VE-chain SFT and evidence-responsive GRPO, DERIVE learns reasoning that is grounded, executable, and responsive to changes in answer-determining evidence. Experiments across six public benchmarks demonstrate consistent improvements over competitive baselines, with particularly strong gains on arithmetic-intensive reasoning tasks. Overall, our results suggest that advancing long-document understanding requires not only better evidence localization, but also explicit and verifiable supervision over how retrieved evidence drives downstream reasoning.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript drafting and editing, literature organization, and limited code development and debugging. Generative AI was also used during training-data construction: a teacher model assisted in generating VE-chain decompositions for selected datasets, and an external language model proposed non-numeric replacement values for Evidence Intervention. All generated training targets and interventions were subsequently filtered and verified using the deterministic grounding, execution, and consistency checks described in the Appendix. Generative AI was not used to generate experimental results or to formulate or prove mathematical claims. All AI-assisted text, code, data construction, technical claims, and references were reviewed and verified by the authors, who made all final decisions regarding methodology, experimental design, analysis, and presentation.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of DERIVE and its training procedure in Sec. 3.1–3.4, including internal retrieval, VE-chain construction, evidence intervention, and SFT/GRPO optimization. The Appendix further documents the retrieval objective, VE-chain construction and serialization, evidence-intervention pipeline, executable reward design, and complete training algorithm (Appendix A.2, A.3, A.4, and A.5). We also report the datasets, model configurations, optimization hyperparameters, trainable parameter counts, and computational costs used in our experiments.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Jian Chen, Ruiyi Zhang, Yufan Zhou, Tong Yu, Franck Dernoncourt, Jiuxiang Gu, Ryan A Rossi, Changyou Chen, and Tong Sun. Sv-rag: Lora-contextualizing adaptation of mllms for long document understanding. *arXiv preprint arXiv:2411.01106*, 2024.

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

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.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *International Conference on Learning Representations*, volume 2025, pp. 61424–61449, 2025.

Xiang Feng, Jiawei Zhou, Zhangfeng Huang, Kewei Wang, Shanshan Ye, Jinxin Hu, Zulong Chen, Yong Luo, and Jing Zhang. Docscope: Benchmarking verifiable reasoning for trustworthy long-document understanding. *arXiv preprint arXiv:2605.08888*, 2026.

Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3096–3120, 2024.

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. *arXiv preprint arXiv:2106.09685*, 2021.

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. *arXiv preprint arXiv:2410.01744*, 2024.

Wenjun Ke, Yifan Zheng, Yining Li, Hengyuan Xu, Dong Nie, Peng Wang, and Yao He. Large language models in document intelligence: A comprehensive survey, recent advances, challenges, and future trends. *ACM Transactions on Information Systems*, 44(1):1–64, 2025.

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

Shuochen Liu, Pengfei Luo, Chao Zhang, Yuhao Chen, Haotian Zhang, Qi Liu, Xin Kou, Tong Xu, and Enhong Chen. Look as you think: Unifying reasoning and visual evidence attribution for verifiable document rag via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32159–32167, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. `https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/`, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 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 AAAI Conference on Artificial Intelligence*, volume 40, pp. 25357–25365, 2026.

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 AAAI Conference on Artificial Intelligence*, volume 37, pp. 13636–13645, 2023.

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.

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.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Junyu Xiong, Yonghui Wang, Rongjian Gu, Chenyu Liu, Bing Yin, Wengang Zhou, and Houqiang Li. Hierarchical evidence-driven reasoning for long document understanding. *arXiv preprint arXiv:2607.04625*, 2026a.

Junyu Xiong, Yonghui Wang, Weichao Zhao, Chenyu Liu, Bing Yin, Wengang Zhou, and Houqiang Li. Docr1: Evidence page-guided grpo for multi-page document understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11178–11186, 2026b.

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. *arXiv preprint arXiv:2604.12812*, 2026.

Jinxu Zhang, Yongqi Yu, and Yu Zhang. Cream: coarse-to-fine retrieval and multi-modal efficient tuning for document vqa. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 925–934, 2024.

Yuanlei Zheng, Pei Fu, Hang Li, Ziyang Wang, Yuyi Zhang, Wenyu Ruan, Xiaojin Zhang, Zhongyu Wei, Zhenbo Luo, Jian Luan, et al. Doc-v*: Coarse-to-fine interactive visual reasoning for multi-page document vqa. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45901–45923, 2026.

Dawei Zhu, Rui Meng, Jiefeng Chen, Sujian Li, Tomas Pfister, and Jinsung Yoon. Doclens: A tool-augmented multi-agent framework for long visual document understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26804–26829, 2026.

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4857–4866, 2022.

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.

A APPENDIX

A.1 DETAILS OF DATASETS

We evaluate DERIVE on six multimodal document understanding benchmarks. MP-DocVQA, DUDE, SlideVQA, and TAT-DQA are used for training and evaluation, while MMLongBench-Doc and LongDocURL are held out entirely from training.

- **MP-DocVQA** Tito et al. (2023). MP-DocVQA contains 46,436 question-answer pairs over 5,929 multi-page documents and 47,952 page images, with up to 20 pages per document. The official split contains 36,230/5,187/5,019 train/validation/test questions. We train on the official training split and report results on the test split.
- **DUDE** Van Landeghem et al. (2023). DUDE contains approximately 41.5K questions over 5,019 visually rich multi-page documents, averaging 5.72 pages per document. It covers diverse question and answer types, including layout reasoning, arithmetic, multi-hop reasoning, and unanswerable cases. We use the official training split for training and the test split for evaluation.
- **SlideVQA** Tanaka et al. (2023). SlideVQA contains 14,484 question-answer pairs over 2,619 slide decks, each with 20 slides, and provides evidence-page annotations and arithmetic expressions for numerical questions. The official split contains 10,617/1,652/2,215 train/validation/test questions. We train on the training split and evaluate on the test split.
- **TAT-DQA** Zhu et al. (2022). TAT-DQA contains 16,558 questions over financial reports with both textual and tabular content. The train/development/test splits contain 13,251/1,645/1,662 questions, covering single-span, multi-span, counting, and arithmetic answers. We use the training split for training and the official test split for evaluation.
- **MMLongBench-Doc** Ma et al. (2024). MMLongBench-Doc contains 1,091 questions over 135 long PDF documents, averaging 47.5 pages and 21,214 textual tokens per document. It covers multiple evidence modalities and locations, including cross-page and unanswerable questions. We use the complete benchmark only for held-out evaluation.
- **LongDocURL** Deng et al. (2025). LongDocURL contains 2,325 question-answer pairs over 396 PDF documents and more than 33,000 pages. It covers document understanding, numerical reasoning, and cross-element localization across text, layout, figures, and tables. We use LongDocURL exclusively for held-out evaluation.

A.2 DETAILS OF VE-CHAIN CONSTRUCTION

Serialized format. A VE chain is represented as a newline-separated sequence of typed assignments terminated by an `Answer:` line. A `READ` step binds an evidence variable `E`i to a value and its source page, while an operator step binds `X`j to the result of applying an operator to previously defined variables. Page indices are 1-based with respect to the original document.

Construction pipeline. VE-chain construction separates reasoning decomposition from evidence localization. We use a teacher model (GPT-5.6) to identify the evidence values needed for the answer and express their dependencies with structured operators. The proposed values are then localized on their corresponding pages using page-level text matching. The gold answer is provided to the teacher during offline target construction, but a chain is retained only if it passes the verification checks described below.

Per-source construction. **TAT-DQA** provides facts, their layout mappings, and derivation expressions for arithmetic and counting questions. We parse each derivation into executable operators and bind its numeric leaves to the corresponding facts.

MPDocVQA and **SlideVQA** use the teacher model together with the annotated evidence pages and their extracted text. The teacher identifies the required `READ` values and expresses the subsequent reasoning using the predefined operator vocabulary.

DUDE is processed similarly when a reliable grounded chain can be constructed. Examples for which no reliable chain is available remain direct-answer targets, while answers attributable to a single evidence page are represented with a single `READ`.

Operators and execution. We use a closed operator vocabulary covering arithmetic (`ADD`, `SUB`, `MUL`, `DIV`), selection (`ARGMAX`, `ARGMIN`, `COMPARE`), aggregation (`COUNT`, `LIST`), textual composition (`CONCAT`), and structural relations (`RELATE`). Structured operators make dependencies explicit and allow deterministic execution. All executable operators are recomputed locally, and chains with inconsistent outputs are discarded. For multi-read arithmetic chains, `READ` steps follow formula-argument order so that evidence variables align directly with subsequent operator arguments.

Verification. Teacher-generated chains are retained only if their final answer matches the dataset gold, every `READ` value can be matched to its cited page, and every executable operator reproduces its declared output. We additionally check the validity of page and variable references and the consistency of the serialized final answer. These checks ensure that retained VE chains are grounded in document evidence and executable from their declared dependencies.

A.3 Details of Evidence Intervention

Figure 6 shows representative interventions from the four training sources. For each example, we select one answer-determining grounded evidence value e_r and replace it with e'_r, while keeping the question and unrelated document content unchanged. The edited document D' and its corresponding VE chain $\mathcal{C}'$ are then constructed as described below.

Evidence editing. We modify only the document region corresponding to the selected evidence. For text-layer PDFs, the original value is removed and the replacement is inserted at the same location. For image-based documents, the original glyphs are removed from the rendered page and the replacement is written into the same region. To avoid introducing edit-specific visual cues, both the original and intervened states undergo the same rewrite procedure: the original state writes back e_r, while the intervened state writes e'_r using the same rendering configuration.

Replacement construction. Replacement values are chosen to preserve the surface form and semantic type of the original evidence. Numeric replacements preserve formatting such as decimal precision, separators, and percentage notation. For non-numeric evidence, we use an external language model to propose a replacement of the same semantic type and language form. Candidates are retained only when they differ from the original value and can be rendered naturally within the original evidence region.

Dependency-consistent targets. For chains in which the selected evidence feeds downstream computation, we preserve the original dependency structure and replace e_r with e'_r. All affected downstream operations are then deterministically re-executed to produce the intervened chain $\mathcal{C}'$ and answer a'. Interventions that lead to invalid execution or leave the final answer unchanged are discarded.

For examples whose answer is determined directly by a single grounded `READ`, the replacement value directly defines the intervened answer and no downstream computation is required.

Pair verification. Each intervention pair is checked before training. We verify that the edit is confined to the intended evidence region, that the replacement remains visually valid, and that the intervened answer is consistent with the edited evidence and the corresponding VE execution. These checks ensure that the original and intervened states differ only in the selected answer-determining evidence while preserving the surrounding document context.

A.4 Details of Executable Chain Verification & Reward Calculation

For each sampled trajectory τ_i, the executable verifier parses the generated VE chain and assigns a deterministic scalar reward. Let $\hat{a}_i$ denote the predicted final answer, $\mathcal{Y}$ the set of valid gold-answer variants under the current document state, and refuse_i indicate that $\hat{a}_i$ is a refusal. The reward is

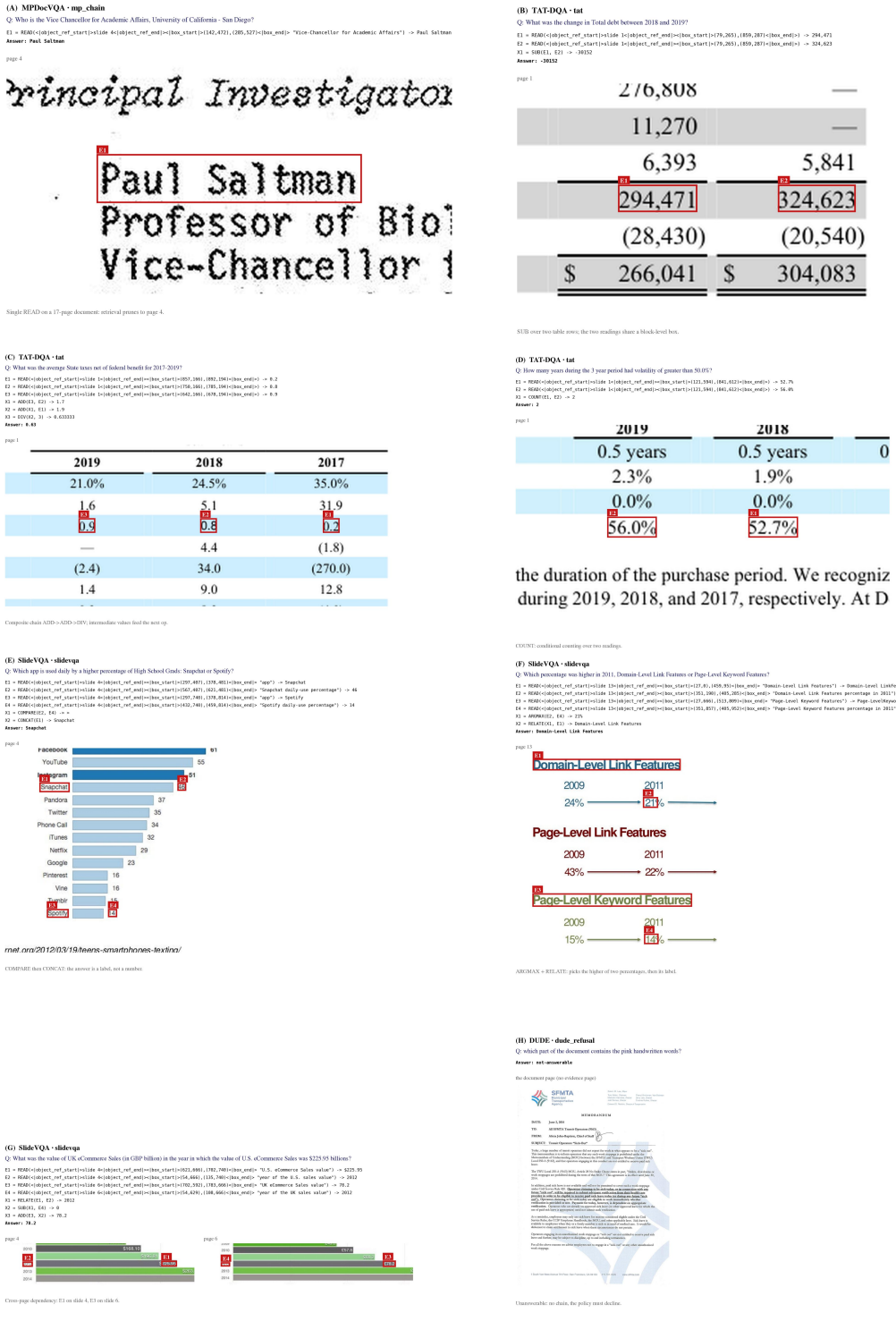


Figure 5: **Examples of Verifiable Execution (VE) chains from the training corpus.** Cases A–H cover grounded `READ`, arithmetic and compositional operators, `COUNT`, `COMPARE`/`CONCAT`, `ARGMAX`/`RELATE`, cross-page dependencies, and unanswerable questions. Chains are shown in the exact serialized format used for SFT.

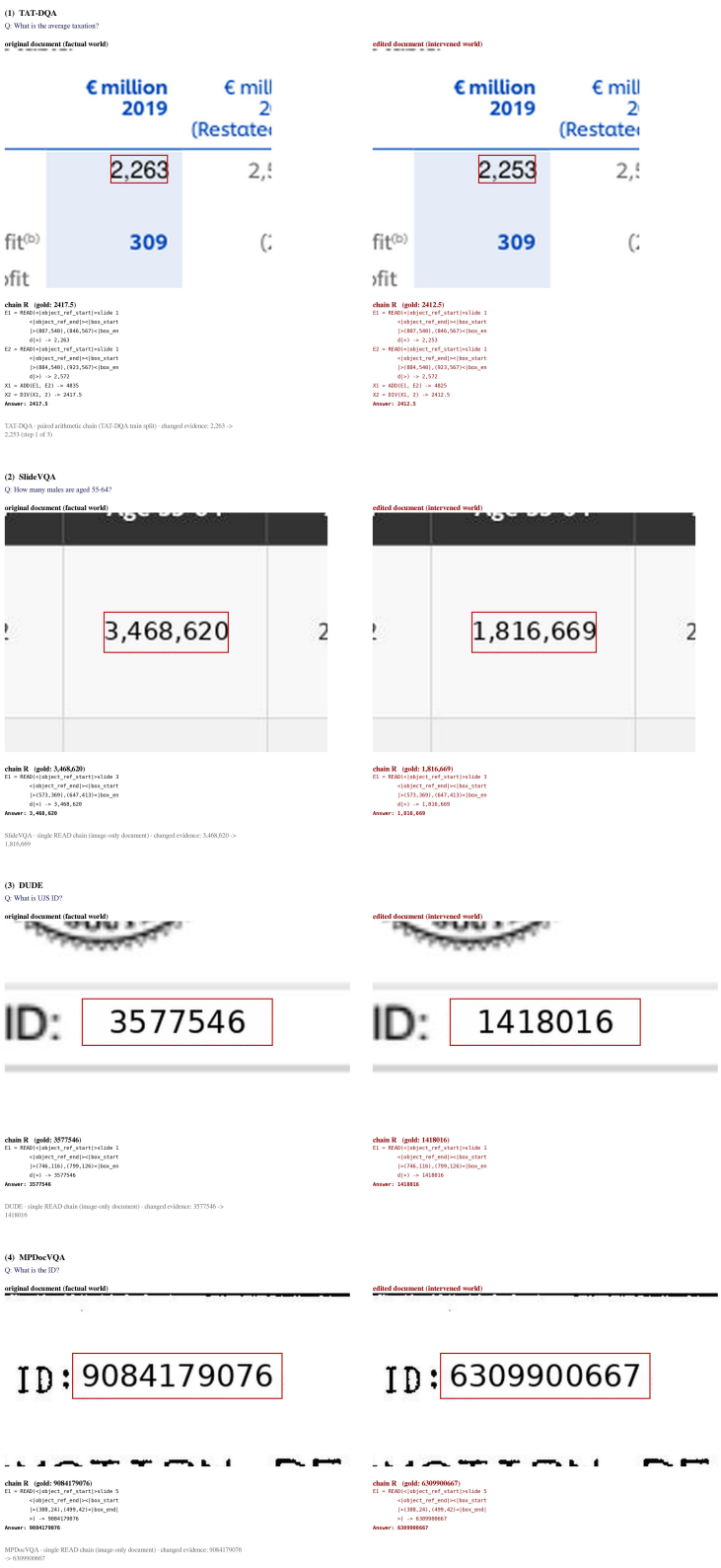


Figure 6: Examples of evidence interventions across the four training sources. Each case modifies a single answer-determining evidence cell while keeping the question and remaining document content unchanged. For TAT-DQA, the edit is propagated through the multi-step VE dependency; for SlideVQA, DUDE, and MPDocVQA, the edited `READ` value directly determines the new answer.

Table 5: DUDE breakdown by answerability and document length. "Ans." and "Not-Ans." denote performance on answerable and unanswerable questions, respectively. Page-number columns group examples by document length. "Int. Ret." denotes internal retrieval, and EI denotes Evidence Intervention.

Methods	Ans.	Not-Ans.	Page Numbers				
			1	2-5	6-10	11-20	>20
base + Int. Ret.	<u>44.7</u>	0.4	39.7	39.4	<u>39.4</u>	36.3	35.3
+ VE-chain SFT	42.4	**85.1**	<u>44.3</u>	<u>42.9</u>	38.9	<u>36.9</u>	<u>35.3</u>
+ GRPO (w/ EI)	**45.4**	<u>71.5</u>	**47.9**	**48.2**	**43.6**	**44.3**	**41.2**

defined as

$$r_i = \begin{cases} \mathbb{1}[\text{refuse}_i], & \mathcal{Y} = \{\texttt{not-answerable}\}, \\ \mathbb{1}[\hat{a}_i \in \mathcal{Y}] + w_G V_G(\tau_i) + w_E V_E(\tau_i) & \text{otherwise,} \\ \quad + \mathbb{1}[\text{pair}]\,(w_r V_{\text{read}}(\tau_i) + w_x V_{\text{reexec}}(\tau_i)) - w_{\text{ref}}\,\mathbb{1}[\text{refuse}_i], & \end{cases} \tag{10}$$

where $w_G = 0.2$, $w_E = 0.3$, $w_r = w_x = 0.2$, and $w_{\text{ref}} = 0.35$. Answer correctness is evaluated against all valid gold variants to avoid assigning negative group-relative advantages to semantically correct predictions. For unanswerable examples, process terms are disabled: a clean refusal receives reward 1, while an unsupported chain followed by a refusal receives no additional process credit.

V_G: evidence grounding. V_G measures the fraction of `READ` steps whose emitted values can be matched to their cited pages using the same page-level matching procedure as in VE-chain construction. It therefore verifies that the evidence values used by the generated chain are supported by the document, rather than only being internally consistent.

V_E: executable consistency. V_E re-executes each valid operator from its preceding variables and measures the fraction of operator outputs that are reproduced correctly. Operators that depend on an ungrounded `READ` are not considered valid, so execution credit is given only to computation supported by grounded evidence. No reference VE chain is required.

V_{read} and V_{reexec}: intervention verification. For intervention examples, V_{read} verifies that the generated chain reads the modified evidence value from the corresponding document state. V_{reexec} then substitutes the paired evidence value and re-executes the affected downstream operations; it equals 1 when the result matches the gold answer of the paired state. Together, these terms verify whether the generated reasoning responds consistently to the controlled evidence change.

Components that are not applicable to a trajectory are omitted from the reward. All terms are computed deterministically from the generated VE -hain, document evidence, gold answer, and intervention metadata, without a learned reward model or reference VE chain.

A.5 Details of Training Pipeline

Internal Cross-modal Retrieval. For completeness, the retrieval head is pretrained with a hard-negative objective. Let $\mathcal{P}$ denote the set of ground-truth evidence pages and $\mathcal{N}$ the remaining pages. The scores of positive and negative samples can be calculated as:

$$S_{\text{pos}} = \sum_{i \in \mathcal{P}} s_i, \qquad S_{\text{neg}} = \sum_{j \in \text{Top}_{|\mathcal{P}|}(\mathcal{N})} s_j, \tag{11}$$

where the highest-scoring $|\mathcal{P}|$ negatives are used when sufficient negatives are available, the retrieval loss can be defined as:

$$\mathcal{L}_{\text{ret}} = \text{softplus}\,(S_{\text{neg}} - S_{\text{pos}})\,. \tag{12}$$

Overall Training Pipeline Algorithm 1 summarizes the complete training pipeline of DERIVE. The three stages optimize different components sequentially, while the retrieval module remains fixed during both VE-chain SFT and GRPO.

Table 6: Parameter overhead and trainable parameters of DERIVE."Retr." reports the parameters introduced by the internal retrieval module, while "Trainable LoRA" reports the parameters updated during SFT and GRPO. Percentages are relative to the corresponding model size excluding output embeddings.

Model	Total (B)	w/o out-emb (B)	Retr. (M)	%	Trainable LoRA (M)	%
Qwen2.5-VL-3B	3.755	3.443	0.00	0.000	—	—
Qwen2.5-VL-7B	8.292	7.747	0.00	0.000	—	—
URaG-3B / ours	3.757	3.446	2.62	0.070	59.87	1.593
URaG-7B / ours	8.296	7.751	4.20	0.051	80.74	0.973

Table 7: **Prefill computation with and without top-5 page pruning.** We report costs for 20, 60, and 100 pages at 1024 visual tokens per page. "Linear only" follows the $2NT$ accounting used by URaG, while "With attention" additionally includes quadratic attention. Green values denote relative FLOP reductions from top-5 pruning. All costs are reported in tera-FLOPs (T).

Scale	Pages	Linear only		With attention	
		all (T)	top-5 (T)	all (T)	top-5 (T)
3B	20	231.8	151.3 (-34.7%)	356.1	178.6 (-49.8%)
3B	60	694.6	399.5 (-42.5%)	1809.7	591.9 (-67.3%)
3B	100	1157.5	647.7 (-44.0%)	4252.9	1170.1 (-72.5%)
7B	20	395.7	221.6 (-44.0%)	564.9	266.3 (-52.9%)
7B	60	1185.7	547.2 (-53.9%)	2703.4	880.9 (-67.4%)
7B	100	1975.6	872.9 (-55.8%)	6188.8	1784.1 (-71.2%)

A.6 More Experimental Results

More Results. Table 5 shows that the different training stages affect answerability and document length in complementary ways. VE-chain SFT dramatically improves performance on unanswerable questions from 0.4 to 85.1, while answerable accuracy decreases slightly from 44.7 to 42.4. This suggests that structured VE supervision strongly improves the model's ability to recognize when the available evidence does not support an answer, but can shift the balance away from answering answerable cases.

GRPO with Evidence Intervention restores this balance, increasing answerable performance to 45.4 while retaining strong unanswerable performance at 71.5. It also improves performance across every document-length bucket relative to the retrieval-only baseline, reaching 47.9, 48.2, 43.6, 44.3, and 41.2 from one-page to $>$ 20-page documents. Notably, the gains persist for longer documents, including improvements of 8.0 points on 11–20 pages and 5.9 points on documents longer than 20 pages. These results indicate that the benefit of dependency-aware optimization is not restricted to short-context cases, but extends to long-document reasoning where evidence must be used reliably after retrieval.

Case Study. Figure 8 illustrates representative post-retrieval failure modes. In Figure 8(a) and (c), the relevant evidence is already localized, yet small reading errors in grounded values propagate directly to incorrect answers. Figure 8(b) shows a different failure: the required values are available, but the baseline applies an incorrect arithmetic dependency, while VE-chain SFT terminates before completing the required computation. DERIVE resolves both types of errors by combining precise grounded reads with executable downstream reasoning, recovering the correct answers in all three cases. These examples illustrate that the remaining errors after retrieval arise from both individual reasoning states and the dependencies that connect them, motivating the joint state- and dependency-level supervision used by DERIVE.

Parameter Efficiency. DERIVE introduces little additional parameter overhead, as shown in Table 6. The internal retrieval module contains only 2.62M parameters for the 3B model and 4.20M for the 7B model, corresponding to 0.070% and 0.051% of the respective backbones. During SFT and GRPO, only 59.87M and 80.74M LoRA parameters are optimized, accounting for 1.59% and

Table 8: Sensitivity to process-reward weights. We jointly scale the four process-reward weights by s while keeping the answer-correctness reward fixed at 1.0. Results are reported on TAT-DQA development examples and the 505-pair evidence-intervention set. "Orig." and "Interv." denote accuracy on the original and intervened states, while "Both" requires both states to be correct. The $1\times$ setting is used throughout the paper.

s	(w_g, w_e, w_r, w_x)	TAT-DQA	$\mathrm{em}_{\mathrm{orig.}}$	$\mathrm{em}_{\mathrm{interv.}}$	both
$0.5\times$	$(0.10, 0.15, 0.10, 0.10)$	64.8	57.4	51.1	47.3
$1\times$	$(0.20, 0.30, 0.20, 0.20)$	64.8	58.2	51.9	48.1
$2\times$	$(0.40, 0.60, 0.40, 0.40)$	64.1	56.8	50.9	47.1

0.97% of the 3B and 7B models, respectively. Thus, the gains of DERIVE are obtained without full-model fine-tuning or substantial additional model capacity.

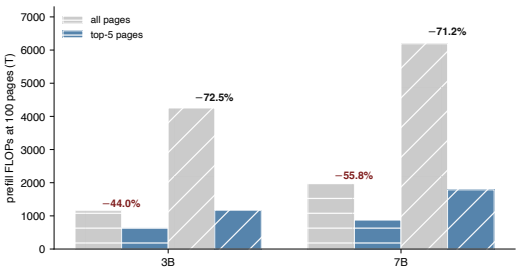


Figure 7: Prefill FLOPs for 100-page documents with and without top-5 page pruning. Horizontal and diagonal hatching denote linear-only and attention-aware accounting, respectively. Percentages indicate the FLOP reduction from page pruning.

Prefill Efficiency. Table 7 shows that internal page pruning substantially reduces prefill computation as document length increases. Under the linear $2NT$ accounting used by URaG, retaining only the top-5 pages reduces the 100-page cost by 44.0% for the 3B model and 55.8% for the 7B model. When the quadratic attention term is included, the reductions increase to 72.5% and 71.2%, respectively. The larger savings under full attention accounting arise because pruning shortens the sequence processed by the deeper Transformer layers from all document pages to only the retained top-5 pages. Thus, the linear accounting provides a conservative estimate of the actual prefill savings obtained from internal retrieval. For the 3B model, the linear-accounting reductions closely reproduce those reported by URaG, confirming that our implementation preserves the expected retrieval efficiency.

Figure 7 highlights the computational benefit of internal retrieval on 100-page documents. Top-5 pruning reduces prefill FLOPs by 44.0% and 55.8% for the 3B and 7B models under the linear accounting, with substantially larger reductions of 72.5% and 71.2% when attention is included. The larger attention-aware savings arise because retrieval removes most visual tokens before the deeper Transformer layers, where computation grows with sequence length. Thus, internal retrieval not only localizes evidence but also substantially reduces the cost of long-document reasoning.

Reward-weight sensitivity. Table 8 studies the sensitivity of DERIVE to the relative strength of process supervision. Halving the four process-reward weights leaves TAT-DQA unchanged at 64.8, but slightly reduces performance on the intervention set, with joint accuracy decreasing from 48.1 to 47.3. Doubling the weights degrades both TAT-DQA and intervention performance, yielding 64.1 TAT-DQA EM and 47.1 joint accuracy. The default $1\times$ setting therefore provides the strongest overall performance across the original, intervened, and joint metrics. Importantly, the differences are modest: none of the paired comparisons remains significant after Bonferroni correction, indicating that performance is relatively stable around the default reward scale rather than being highly sensitive to precise weight tuning.

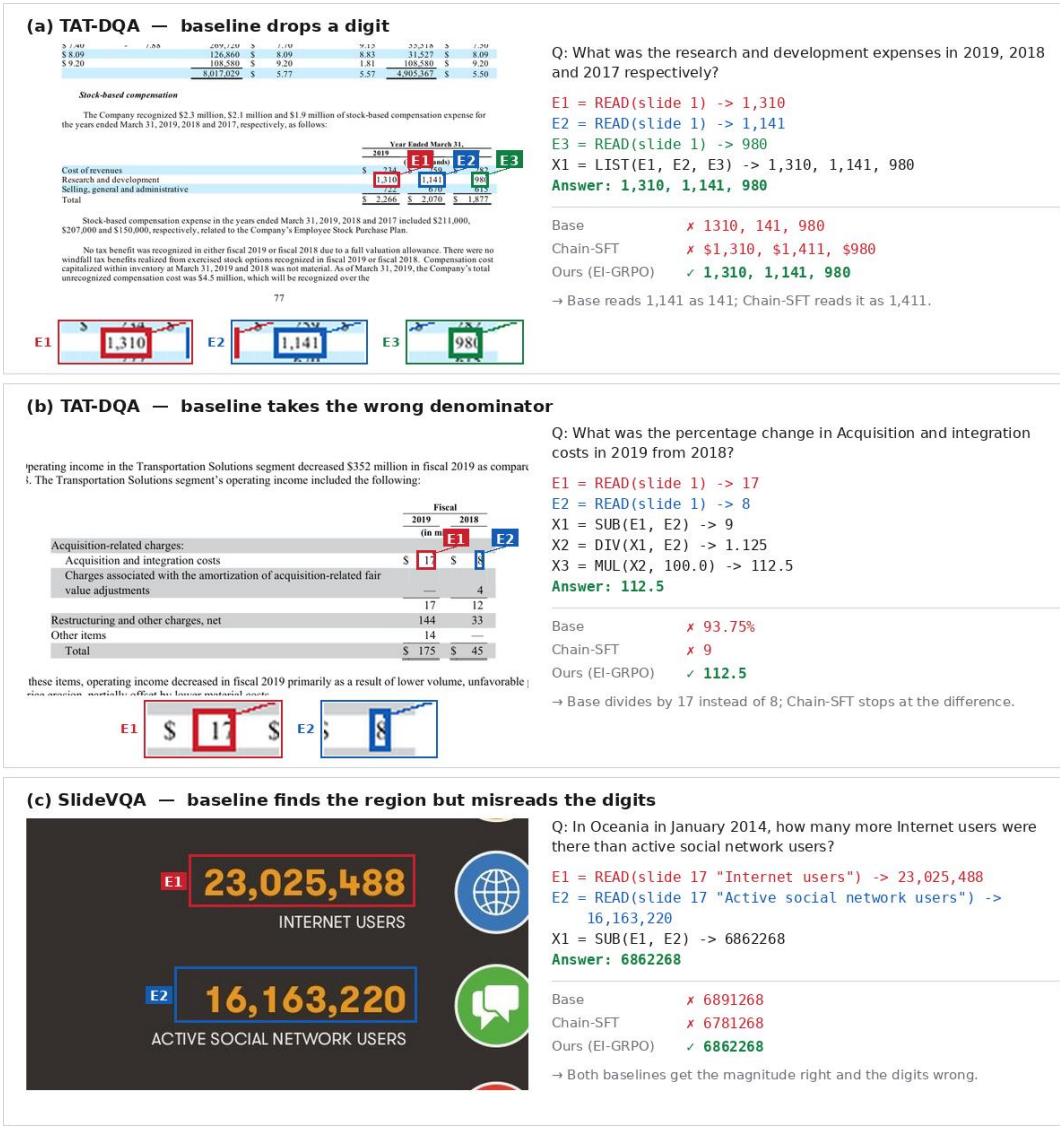


Figure 8: Qualitative case studies of post-retrieval reasoning. (a) On TAT-DQA, the baselines misread a grounded value, while DERIVE recovers all evidence values correctly. (b) The baseline follows an incorrect arithmetic dependency and VE-chain SFT stops at an intermediate result, whereas DERIVE executes the complete SUB→DIV→MUL chain. (c) On SlideVQA, both baselines localize the relevant regions but misread individual digits, while DERIVE produces the correct grounded reads and subtraction result. Colored boxes indicate the evidence regions referenced by the VE chain.

Algorithm 1 Training Pipeline of DERIVE

Require: Training set $\mathcal{D}_{\text{train}}$, pretrained MLLM policy π_θ, retrieval head g_ϕ, rollout size K
Ensure: Trained policy π_θ and retrieval head g_ϕ

Stage 1: Internal Cross-modal Retrieval
Freeze the MLLM parameters θ
for each (D, q) in $\mathcal{D}_{\text{train}}$ **do**
    Extract early-layer multimodal representations
    Compute page scores $\{s_i\}_{i=1}^{N}$ using g_ϕ
    Update ϕ using $\mathcal{L}_{\text{ret}}$, using Eq. 12
end for
Freeze the trained retrieval head g_ϕ

Prepare Verified VE Training Data
Construct original VE chains from the training annotations
Retain verified instances $\mathcal{D}_{\text{VE}} = \{(D, q, \mathcal{C})\}$
For intervention-eligible instances, edit an answer-determining evidence node $e_r \to e'_r$
Re-execute its affected downstream computation to obtain $(D', \mathcal{C}')$

Stage 2: VE-chain Supervised Fine-tuning
Enable the trainable parameters of π_θ
for each $(D, q, \mathcal{C})$ in $\mathcal{D}_{\text{VE}}$ **do**
    Run the fixed internal retriever g_ϕ
    Serialize $\mathbf{c} \leftarrow \text{Ser}(\mathcal{C})$
    Update θ using $\mathcal{L}_{\text{SFT}}$, using Eq. 6
end for

Stage 3: Dependency-aware Evidence-Responsive GRPO
for each $(D, q, \mathcal{C})$ in $\mathcal{D}_{\text{VE}}$ **do**
    $\mathcal{S} \leftarrow \{(D, \mathcal{C})\}$
    if an intervention pair $(D', \mathcal{C}')$ is available **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{(D', \mathcal{C}')\}$
    end if
    for each $(\widetilde{D}, \widetilde{\mathcal{C}})$ in $\mathcal{S}$ **do**
        Run the fixed internal retriever g_ϕ
        Sample $\{\tau_i\}_{i=1}^{K} \sim \pi_\theta(\cdot \mid \widetilde{D}, q)$
        for $i = 1, \ldots, K$ **do**
            Parse τ_i into $\hat{\mathcal{C}}_i$
            Compute executable reward $r_i \leftarrow \mathcal{V}(\hat{\mathcal{C}}_i, \widetilde{\mathcal{C}})$
            if $\widetilde{D} = D'$ **then**
                Include edited-evidence and downstream re-execution verification
            end if
        end for
        Compute $\{A_i\}_{i=1}^{K}$ using Eq. 8
        Update θ using $\mathcal{L}_{\text{GRPO}}$
    end for
end for
return π_θ, g_ϕ