INTERTAB: INTERLEAVED VISUAL-STRUCTURE ALIGNMENT FOR MULTIMODAL TABLE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Table images preserve structural information that are often lost in text serialization, and reasoning over them requires locating relevant rows, columns, and cells step by step. Current multimodal large language models (MLLMs) encode the whole image once before reasoning, so they cannot pick up row-, column-, and cell-level evidence as the question unfolds. Encoder-side table structure and generic interleaved visual chain-of-thought still do not bind each reasoning step to that structure. We propose **InterTab**, an **Inter**leaved structure-aware framework for CoT reasoning over **Tab**le images, interleaves chain-of-thought with tool calls that crop structure-aligned table regions. First, we build **InterTab-22K**, includes reasoning trajectories in which each step is tied to both a structural location and a bounding box. InterTab is trained in two stages: supervised structure-aware alignment (SSA) on **InterTab-22K** teaches the model to interleave reasoning with structure-aligned crops, and active localization optimization (ALO) further optimizes answer correctness, localization IoU, and output format, while penalizing missing or excessive tool calls. Experiments on nine table benchmarks show that InterTab improves the average accuracy of its backbone from 68.28% to 73.17% and achieves the best average performance among all compared methods. Code and data will be released soon.

1 INTRODUCTION

Tabular data is a fundamental format for organizing structural information across real-world domains, and in many applications tables are available only as images, such as screenshots, and scanned reports. Presenting tables as images also preserves layout and formatting cues, such as hierarchical headers, merged cells, and color shading, which are often lost or distorted when tables are serialized into text. Reasoning over a table image requires reading content organized by an explicit row-column structure: answering a question typically involves locating the relevant rows, following them to the right columns, and reading or comparing specific cells. Multimodal Large Language Models (MLLMs) can percept table images directly, making such reasoning possible without an external OCR or parsing pipeline.

However, the visual perception of current MLLMs is fixed before reasoning begins: the whole image is encoded once in a single forward pass, *e.g.,* by CLIP-style visual encoders, and then consumed by the language decoder during generation, and every subsequent reasoning step can only attend to this same set of tokens. This is poorly suited to multi-step table reasoning, where each step depends on a different small region of a complex or dense layout (Jiang et al., 2025a). Figure 1 shows a 16×16 match-result table. To count the goals Östersund scored against Sirius, the model must read two cells, *i.e.,* `ÖFK` and `ÖSK`, whose column headers are abbreviated and adjacent. A standard MLLM with Chain-of-Thought (CoT) (Wei et al., 2022) decomposes the question correctly, but reads the neighboring column `ÖSK` instead of `ÖFK` in the first step, and the error propagates to the final answer. The failure lies not in language reasoning but in visual grounding: the model names the target row and column correctly, yet cannot precisely perceive the corresponding subregion in the image. We refer to this as *passive* perception, since the visual evidence available to the model does not change with what the current reasoning step requires. Table reasoning instead calls for *active* evidence acquisition, where the model decides at each step which rows, columns, or cells to inspect and obtains a precise view of them.

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.

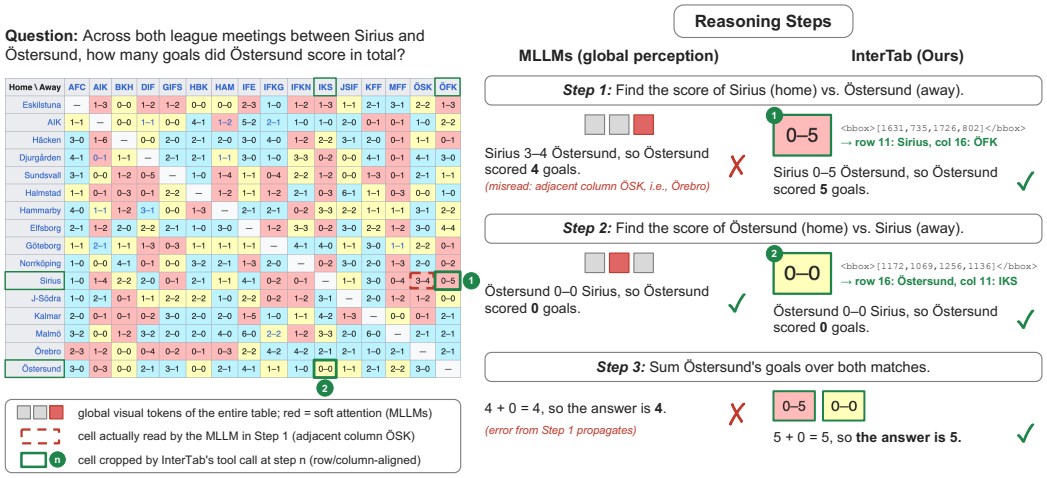

Figure 1: Comparison of two CoT-based reasoning methods on a table image.

Existing methods for multimodal table reasoning mainly improve how the table image is encoded, either by distilling table understanding through supervised fine-tuning (Jiang et al., 2025a; Zheng et al., 2024) or by injecting row and column structure into the visual encoder (Jiang et al., 2025b; Aida et al., 2025; Zhang et al., 2025). Both still answer from a single global encoding. Another line of work, often called thinking with images, interleaves visual evidence with CoT on general images by selecting boxes (Meng et al., 2025; Zheng et al., 2026; He et al., 2025; Su et al., 2026) or querying patches (Chen et al., 2026). However, evidence in tables is inherently discrete: each value occupies a single cell defined by its row and column, so a crop shifted by just one cell can return an entirely different value. What is missing is a localization action that follows each reasoning step and is trained to land on the exact rows, columns, or cells that the step refers to.

To this end, we propose **InterTab**, which interleaves CoT with structure-aligned visual evidence. At each step, the model can call a `get_sub_table` tool to crop a table element, and the selected subregion is encoded as visual tokens that condition the subsequent reasoning. To enable such structure-aligned localization, we build **InterTab-22K**, a set of 21.7K multi-turn trajectories in which each interleaved visual subregion is aligned, via table metadata, with the exact row, column, or cell referenced by the corresponding textual reasoning step. InterTab is trained in two stages: Supervised Structure-aware Alignment (SSA) fine-tunes the model on these trajectories, and Active Localization Optimization (ALO) applies Group Relative Policy Optimization (GRPO) with rewards for answer correctness, format, and localization IoU, together with a penalty on excessive tool calls.

We summarize our main contributions as follows.

❶ We identify passive global perception as a key bottleneck in multimodal table reasoning, and formulate it as interleaved CoT with active, structure-aligned evidence acquisition.

❷ We construct **InterTab-22K**, to our knowledge the first interleaved multimodal table reasoning dataset, comprising 21.7K trajectories, in which the crop of each tool call is aligned with the rows, columns, or cells referenced by its reasoning step through metadata-guided correction.

❸ We propose **InterTab**, an interleaved reasoning framework trained in two stages: SSA, which aligns each reasoning step with structure-aware visual evidence via supervised fine-tuning on **InterTab-22K**, and ALO, which optimizes when and where to crop using the task and tool rewards.

❹ We conduct extensive experiments on nine table datasets. InterTab improves the average accuracy of its backbone from 68.28 to 73.17 and achieves the best average performance among all compared methods in different table tasks.

Overall, InterTab shifts multimodal table reasoning from passive global encoding toward active, structure-aware evidence acquisition, providing a more reliable reasoning over visual tables.

2 Related Works

Multimodal Table Reasoning. Recent research extend analysis on textual tables to multimodal settings by leveraging document understanding systems or multimodal large language models (MLLMs) to jointly process visual and textual signals (Nunes et al., 2025; Yang et al., 2025; Jiang et al., 2024). Early work leverage OCR to firstly extract textual tables from the image and then processed for reasoning (Paliwal et al., 2019). More recent approaches directly leverage end-to-end vision-language models (*e.g.,* represented by MLLMs) and their language reasoning capability (Zheng et al., 2024; Zhang et al., 2025; Jiang et al., 2025b; Lu et al., 2025). For example, Turbo bridges the modality gap between structured tables and table images by leveraging privileged structured information and iterative reasoning optimization (Jiang et al., 2025a). Additionally, multimodal table and chart benchmarks (Mathur et al., 2024; Wu et al., 2026; Lompo & Haraoui, 2025) further advance multimodal reasoning capabilities in structured visual environments. Existing multimodal table reasoning still relies on a global view of the image, without binding each reasoning step to a row, column, or cell. Interleaved visual reasoning supplies that binding on natural images, but not under table structure. InterTab combines the two: the model reasons in text and, when needed, crops structure-aligned regions, learning to optimize where to crop rather than merely imitating demonstrations.

Multimodal Interleaved Reasoning. Multimodal interleaved reasoning equips models with the ability to generate alternating sequences of textual and visual tokens, enabling tighter coupling between visual perception and language-based reasoning (Chen et al., 2026; Nie et al., 2026; Gu et al., 2026; Jin et al., 2026). This paradigm improves cross-modal alignment by allowing intermediate reasoning steps to explicitly incorporate visual evidence. For instance, DeepEyes demonstrates that reinforcement learning–driven active perception can naturally elicit interleaved visual-text reasoning, leading to more grounded multimodal inference (Zheng et al., 2026). Such approaches further enhance chain-of-thought reasoning in multimodal settings, as evidenced by ICoT, which shows that interleaving visual and textual steps via a simple attention-based mechanism improves both performance and interpretability in MLLMs (Gao et al., 2025).

3 Preliminary

Table Image Reasoning with MLLMs. Given a table image I whose cells are arranged in rows and columns, a textual question Q, and an instruction prompt (Appendix A.3), an MLLM encodes I into visual tokens $v_{1:N}$ and generates the output autoregressively:

$$p(x_{1:T} \mid I, Q) = \prod_{t=1}^{T} p(x_t \mid v_{1:N}, Q, x_{<t}). \tag{1}$$

CoT and Interleaved CoT. Under the CoT setting, the output consists of reasoning steps $s_{1:k}$ followed by the final answer y, *e.g.,* first locating relevant rows and then narrowing down to target cells, yet every step still relies only on the global visual tokens $v_{1:N}$ encoded once from the whole image. Interleaved CoT additionally allows each step s_j to end with a bounding box b_j, which represents the local visual evidence $\tilde{v}_j = \mathcal{E}_{vis}(\mathrm{crop}(I, b_j))$ ($\tilde{v}_j = \emptyset$ if no visual evidence is needed):

$$p(y, s_{1:k} \mid I, Q) = \prod_{j=1}^{k} p(s_j \mid v_{1:N}, Q, s_{<j}, \tilde{v}_{<j})\; p(y \mid v_{1:N}, Q, s_{1:k}, \tilde{v}_{1:k}). \tag{2}$$

For table images, useful evidence is discrete and defined by row-column structure, so each b_j must precisely cover the rows, columns, or cells referenced by s_j, which requires the structure-aware alignment introduced in Section 4.2.

4 Methodology

We first build **InterTab-22K**, whose reasoning trajectories interleave textual steps with visual evidence (Section 4.1). InterTab is then trained in two stages (Section 4.2): Supervised Structure-aware Alignment (SSA) , followed by Active Localization Optimization (ALO).

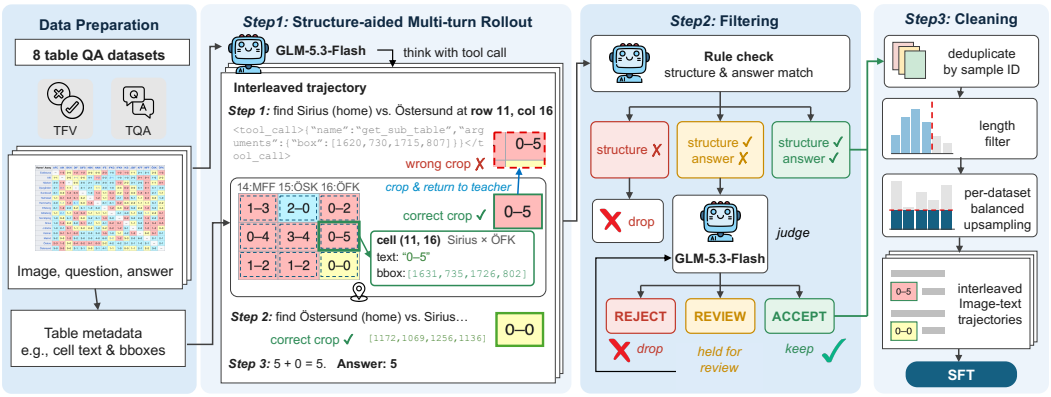

Figure 2: Construction pipeline of **InterTab-22K**. ***Step 1:*** GLM-5.3-Flash is instructed to generate multi-turn trajectories with `get_sub_table`, where wrong visual evidence are corrected by locating the referenced cell in the table metadata. ***Step 2:*** LLM-as-judge for filtering trajectories based on visual-evidence structure and answer correctness. ***Step 3:*** deduplicate trajectories by sample IDs, do length-filtered, and balanced across the eight source datasets for SFT.

4.1 Structure-aided Reasoning Data Construction

We build **InterTab-22K** from eight table datasets, *i.e.,* HiTab (Cheng et al., 2022), InfoTabs (Gupta et al., 2020), TabFact (Chen et al., 2020), TabMWP (Lu et al., 2023), TAT-QA (Zhu et al., 2021), WikiTableQA (Pasupat & Liang, 2015), FeTaQA (Nan et al., 2021), and HybridQA (Chen et al., 2021), from TABLET (Alonso et al., 2026), where each sample provides a table image, a question, its answer, and table metadata including the text and bounding box of every cell. As shown in Figure 2, the data is constructed in three steps. ***Step 1:*** **Structure-aided multi-turn rollout.** Given the raw table image and the question, we instruct GLM-5.3-Flash to reason step by step using the tool `get_sub_table`, which crops the region it needs as visual evidence at each step. On complex tables, however, the model tends to crop regions that are misaligned with cell boundaries. To address this, we identify the referenced cell from the textual reasoning step by its row and column, replace the proposed box with the exact bounding box from the table metadata, and return the corrected crop to the model to continue reasoning. ***Step 2:*** **Validation.** Then, each trajectory is validated by rule-based checks, which verify its structure (*e.g.,* well-formed tool calls and cropped visual regions) and compare the final answer with the ground truth. Trajectories that pass both checks are kept and those with invalid structure are discarded, while those with valid structure but mismatched answers are re-examined by a GLM-5.3-Flash judge, and only the accepted ones are kept. The prompts used are detailed in Appendix A.3. ***Step 3:*** **Cleaning.** Finally, we deduplicate trajectories by the sample IDs and remove overly long ones, yielding 21,721 unique trajectories. For SFT, the eight source datasets are balanced by upsampling.

4.2 InterTab

Building on **InterTab-22K**, we propose InterTab, a two-stage framework consisting of Supervised Structure-aware Alignment (SSA) and Active Localization Optimization (ALO). Specifically, SSA performs cold start to incrementally align visual and structural information, while ALO further trains the model via tool-augmented to improve reasoning accuracy and visual localization quality.

4.2.1 Supervised Structure-aware Alignment (SSA)

SSA fine-tunes the MLLM on **InterTab-22K**, $\mathcal{D} = \{(I_i, Q_i, y_i, s^{(i)}_{1:k})\}$. In step $s^{(i)}$ that needs visual evidence, the reasoning texts is interleaved with a structure-aligned bounding box b_j is aligned with the rows, columns, or cells referred in that step. The bounding visual subregion is encoded as visual tokens $\tilde{v}_j$ and appended to the reasoning, as in Eq. (2). We minimize the negative log-likelihood

$$\mathcal{L}_{\text{SSA}}(\theta) = -\sum_{(I,Q,y,s_{1:k})\in\mathcal{D}} \log p_\theta(y, s_{1:k} \mid I, Q), \tag{3}$$

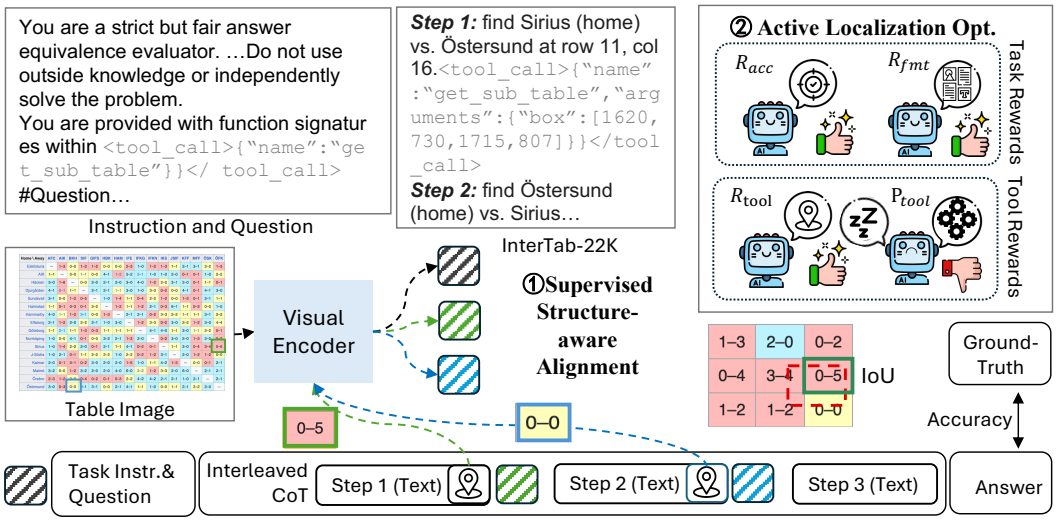

Figure 3: Overview of the **InterTab** framework. First, we perform supervised alignment of each reasoning step with interleaved, structure-aware visual tokens using **InterTab-22K**. Second, we optimize the model using ALO with tool and task rewards, enabling an active policy for interleaved reasoning and accurate visual localization.

to enable the MLLM to locate the correct visual subregions within the entire table image I.

4.2.2 Active Localization Optimization (ALO)

SSA teaches the model where visual evidence locate, but does not optimize when and where to crop under its own reasoning. At the stage of ALO, we set evidence acquisition as part of the learning policy: each tool call and its box are actions sampled by the model and rewarded by their outcomes. Specifically, ALO can be either applied upon the stage of SSA or directly to the base model, using reward formualted in 5. For each (I, Q), we sample G responses $\{o_i\}_{i=1}^{G}$ from $\pi_{\theta_{\text{old}}}$ and maximize

$$\mathcal{J}_{\text{ALO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\Big(\min\big(\rho_i(\theta)A_i,\ \text{clip}(\rho_i(\theta), 1-\epsilon, 1+\epsilon)A_i\big) - \beta D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\Big)\Bigg], \tag{4}$$

where $\rho_i(\theta) = \pi_\theta(o_i \mid I, Q)/\pi_{\theta_{\text{old}}}(o_i \mid I, Q)$ and $A_i = \big(R_i - \text{mean}(\{R_j\}_{j=1}^{G})\big)/\text{std}(\{R_j\}_{j=1}^{G})$.

Reward. The reward of a response combines task and tool terms:

$$R = \lambda_{\text{acc}} R_{\text{acc}} + \lambda_{\text{fmt}} R_{\text{fmt}} + \lambda_{\text{tool}} R_{\text{tool}} - \lambda_{\text{miss}}\, \mathbb{I}_{\text{miss}} - P_{\text{tool}}. \tag{5}$$

R_{acc} is 1 on exact match with y; otherwise, an LLM judge returns 1 or 0. R_{fmt} is 1 when both the thinking span and a complete answer tag are present, 0.5 when only the answer tag is present, and 0 otherwise. R_{tool} averages, over the ground-truth evidence boxes $\mathcal{B}^*$ from the table metadata, the best IoU achieved by the predicted boxes $\mathcal{B}$:

$$R_{\text{tool}} = \frac{1}{|\mathcal{B}^*|}\sum_{b^* \in \mathcal{B}^*} \max_{b \in \mathcal{B}} \text{IoU}(b, b^*), \tag{6}$$

and $R_{\text{tool}} = 0$ if there are no ground-truth boxes or no predicted boxes, *i.e.,* $\mathcal{B}^* = \emptyset$ or $\mathcal{B} = \emptyset$. $\mathbb{I}_{\text{miss}} = 1$ if $\mathcal{B}^* \neq \emptyset$ but the response makes no tool call. The tool-use penalty $P_{\text{tool}} = \min\big(\lambda_{\text{pen}} \max(N - N_0, 0),\ P_{\max}\big)$ penalizes calls beyond the first N_0, where N is the number of tool calls.

5 Experiment

We organize our experiments around four research questions: ❶How does InterTab compare with existing models across in-domain table reasoning benchmarks? ❷Does interleaving textual reasoning

Table 1: Performance comparison on nine table understanding benchmarks. Accuracy (%) is reported for all benchmarks. Best results are in bold and second-best results are underlined.

Model	HiTab	InfoTabs	TabFact	TabMWP	TATQA	WikiTQ	FeTaQA	HybridQA	VTQA	Average
General-purpose MLLMs										
GLM-4.1V-9B (Team et al., 2026)	71.43	<u>72.93</u>	82.03	86.58	38.37	<u>72.03</u>	38.81	54.01	51.75	63.10
InternVL3.5-8B (Wang et al., 2025b)	<u>77.12</u>	71.67	<u>87.10</u>	<u>95.53</u>	<u>51.26</u>	69.07	<u>42.17</u>	70.94	49.67	<u>68.28</u>
Kimi-VL-A3B-Instruct (Kimi Team et al., 2025)	62.46	66.76	70.43	54.72	16.04	56.99	38.92	48.73	38.11	50.35
LLaVA-1.6 (Liu et al., 2024)	16.84	37.81	49.78	39.34	9.03	16.64	13.33	9.93	21.24	23.77
Qwen2.5-VL-7B-Instruct (Bai et al., 2025b)	73.86	71.65	76.00	89.84	40.78	60.14	39.20	39.87	46.64	59.78
Qwen3-VL-8B-Instruct (Bai et al., 2025a)	74.36	72.19	82.25	95.05	47.65	68.67	41.99	69.22	<u>53.27</u>	67.18
Table-specialized Models										
TableLLaVA-7B (Zheng et al., 2024)	9.11	57.54	60.14	61.42	15.69	17.52	20.42	19.64	25.19	31.85
TableVision-R1 (MTabVQA, 2026)	63.68	62.98	72.17	83.72	29.41	50.72	34.04	37.05	33.88	51.96
MiniCPM-V-2.6 (Yao et al., 2024)	48.04	56.56	73.89	87.98	24.35	57.92	29.27	22.50	30.39	47.88
HIPPO (Wang et al., 2025a)	46.37	64.39	74.68	86.85	24.44	56.39	28.63	25.39	31.05	48.69
Ovis2 (Lu et al., 2024)	66.09	71.04	82.28	92.62	42.96	57.49	41.88	<u>77.32</u>	45.76	64.16
Thinking with Images										
DeepEyes (Zheng et al., 2026)	71.80	70.17	77.07	92.97	34.43	60.88	40.78	46.75	47.21	60.23
MINT-COT (Chen et al., 2026)	51.01	59.54	67.94	78.75	36.23	38.62	31.63	24.05	25.82	45.95
InterTab (Ours)	**77.45**	**73.59**	**87.95**	**96.35**	**57.25**	**75.03**	**50.10**	**81.00**	**59.80**	**73.17**

with structure-aligned visual evidence improve accuracy over text-only reasoning? ❸How do the training stages and auxiliary reward terms affect performance, and how sensitive is InterTab to the tool-call allowance? ❹Does InterTab retain its performance advantage on a held-out benchmark excluded from the training data?

5.1 Experimental Settings

We use Qwen3-VL-8B-Instruct as the base model and train on 8 NVIDIA A100 GPUs. For SSA, we set the batch size to 64, the learning rate of 1×10^{-5}, and train for 1 epoch. For ALO, we set the batch size to 16, the learning rate to 1×10^{-6}, the rollout number n to 8, and train for 2 epoch. For evaluation, we use nine representative benchmark: HiTab, InfoTabs, TabFact, TabMWP, TATQA, WikiTQ, FeTaQA, HybridQA, and VisualTQA (Lompo & Haraoui, 2025) (out-of-domain), with the in-domain data taken from TABLET (**?**). To accurately evaluate the models' responses, we conduct an LLM-as-a-judge evaluation (see Appendix for details).

5.2 Main Results

To address RQ1, we compare InterTab with existing models on the eight in-domain benchmarks. Table 1 shows that InterTab achieves the best accuracy on all eight in-domain benchmarks, consistently outperforming the existing baselines. Compared with the Qwen3-VL-8B backbone, InterTab improves HiTab (Cheng et al., 2022), InfoTabs (Gupta et al., 2020), TabFact (Chen et al., 2020), TabMWP (Lu et al., 2023), TATQA (Zhu et al., 2021), WikiTQ (Pasupat & Liang, 2015), FeTaQA (Nan et al., 2021), and HybridQA (Chen et al., 2021) by 3.09, 1.40, 5.70, 1.30, 9.60, 6.36, 8.11, and 11.78 percentage points, respectively. The largest gains are observed on HybridQA, TATQA, and FeTaQA, where InterTab improves over the backbone by more than 8 points. The magnitude of improvement varies across benchmarks. On TabMWP and InfoTabs, where the backbone already performs strongly, InterTab still yields gains of 1.30 and 1.40 points, respectively. The improvements increase to 3.09 and 5.70 points on HiTab and TabFact, and reach 9.60, 6.36, 8.11, and 11.78 points on TATQA, WikiTQ, FeTaQA, and HybridQA, respectively. Overall, InterTab achieves a mean improvement of 5.92 points across the eight in-domain benchmarks, while its nine-benchmark average reaches 73.17%, compared with 67.18% for the Qwen3-VL-8B backbone. We further compare InterTab with both table-specialized and general-purpose vision-language models. Among the table-specialized baselines, Ovis2 achieves an average accuracy of 64.16v across the nine benchmarks, while InternVL3.5-8B achieves 68.28% and represents the strongest competing baseline. Thinking-with-images methods, including DeepEyes and MINT-CoT, obtain averages of 60.23% and 45.95%, respectively, both below the Qwen3-VL-8B backbone. These comparisons show that InterTab consistently outperforms existing approaches across all eight training benchmarks, achieving the strongest overall performance among the evaluated methods rather than relying on gains from a single benchmark or model family.

Table 2: Interleaved reasoning versus a matched text-only baseline. Accuracy (%) is reported.

Reasoning	HiTab	InfoTabs	TabFact	TabMWP	TATQA	WikiTQ	FeTaQA	HybridQA	VTQA	Average
Text-only CoT	71.45	70.44	87.01	94.91	51.25	72.78	45.19	72.60	54.40	68.89
Interleaved (Ours)	**77.45**	**73.59**	**87.95**	**96.35**	**57.25**	**75.03**	**50.10**	**81.00**	**59.80**	**73.17**
Δ	+6.00	+3.15	+0.94	+1.44	+6.00	+2.25	+4.91	+8.39	+5.41	+4.28

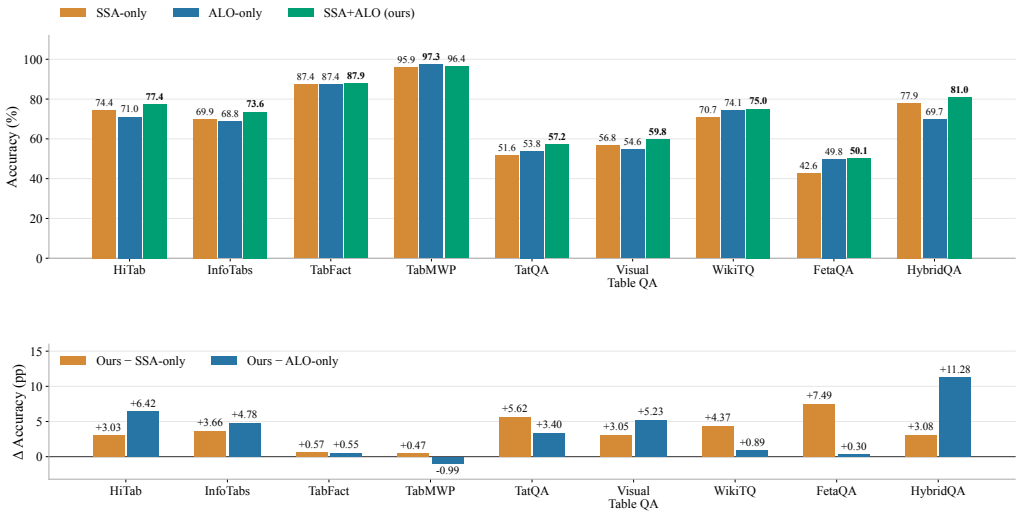

Figure 4: Training-strategy comparison. Top: accuracy of SSA-only, ALO-only, and SSA+ALO. Bottom: accuracy gain of SSA+ALO over each baseline, in percentage points.

5.3 Further Study

To address RQ2 and RQ3, we examine the design of InterTab in four steps: the reasoning paradigm, the two-stage training strategy, the reward components, and the tool-call allowance. The first three comparisons assess the contribution of each design choice; the last tests sensitivity to the call-penalty threshold. Unless otherwise stated, we report average accuracy across all nine benchmarks, including held-out VTQA. All accuracy differences below are in percentage points.

5.3.1 Interleaved Reasoning

We first test whether interleaving reasoning with localized visual evidence improves accuracy. To this end, we establish a matched text-only baseline using the same Qwen3-VL-8B-Instruct backbone and training recipe. We convert our interleaved training data into a text-only version by retaining the same questions and answer targets while removing localization calls and the corresponding cropped table regions. As shown in Table 2, interleaved reasoning consistently outperforms text-only reasoning across all nine benchmarks, improving the average from 68.89% to 73.17% (+4.28 points). The gains are particularly large on HybridQA (+8.39 points), HiTab (+6.00 points), and TATQA (+6.00 points), and remain substantial on held-out VTQA (+5.41 points). These results demonstrate the effectiveness of interleaved reasoning over text-only reasoning.

5.3.2 Two-Stage Training

Having established the benefit of interleaved reasoning, we examine how the two training stages contribute. Specifically, we train three variants from the same backbone on the same interleaved data: SSA-only, ALO-only, and SSA followed by ALO (Ours). As shown in Figure 4, SSA+ALO reaches an average of 73.17%, above SSA-only (69.69%) and ALO-only (69.63%), by 3.48 and 3.54 points, and is best on eight of the nine benchmarks. Relative to SSA-only, the largest gains are on FeTaQA (+7.49 points), TATQA (+5.62 points), and WikiTQ (+4.37 points). Relative to ALO-only,

Table 3: Reward-component ablation. Accuracy (%) is reported. Δ is the change relative to Ours.

Configuration	HiTab	InfoTabs	TabFact	TabMWP	TATQA	WikiTQ	FeTaQA	HybridQA	VTQA	Average
Ours	**77.45**	**73.59**	**87.95**	96.35	**57.25**	**75.03**	**50.10**	**81.00**	59.80	**73.17**
w/o call penalty	76.39	70.70	86.59	94.80	54.87	74.13	43.03	80.01	**62.75**	71.47
Δ	-1.06	-2.89	-1.36	-1.55	-2.38	-0.90	-7.07	-0.99	$+2.95$	-1.69
w/o localization	76.14	68.72	86.02	**97.46**	51.99	73.93	40.95	80.59	56.86	70.30
Δ	-1.31	-4.87	-1.93	$+1.11$	-5.26	-1.10	-9.15	-0.41	-2.94	-2.87

the largest gains are on HybridQA (+11.28 points), HiTab (+6.42 points), and VTQA (+5.23 points). The exception is TabMWP, where ALO-only achieves 97.34% compared with 96.35% for SSA+ALO. We attribute this to the simpler table structures in TabMWP, which make direct ALO sufficient for learning high-reward solutions. These results show that the two stages are complementary: SSA establishes the interleaved localization behavior, while ALO further improves it, and neither stage alone matches the joint recipe.

5.3.3 Reward Components

To further analyze the contribution of each auxiliary reward term, we conduct an ablation study within the full two-stage recipe. Specifically, we remove each auxiliary term from Eq. (5) while keeping answer correctness and format rewards unchanged. As shown in Table 3, removing the localization reward reduces the average performance by 2.87 points to 70.30%, while removing the call penalty reduces it by 1.69 points to 71.47%. Both variants underperform the full reward on eight of nine benchmarks. The localization reward is particularly beneficial for precise evidence selection, with drops of 9.15 points on FeTaQA, 5.26 on TATQA, and 4.87 on InfoTabs. The call penalty also yields notable gains on FeTaQA and InfoTabs, indicating that it helps suppress unnecessary tool calls. TabMWP is the main exception: removing the localization reward improves performance from 96.35% to 97.46%. We attribute this to its relatively simple and compact tables, where precise localization is often unnecessary for obtaining the answer. The same trend appears in the training-strategy comparison, where ALO-only slightly outperforms SSA+ALO on TabMWP (97.34% vs. 96.35%). Overall, the results support retaining both auxiliary terms, as their gains on localization- and tool-use-sensitive benchmarks outweigh the small degradation on TabMWP.

5.3.4 Tool-Call Allowance

To further analyze the effect of the unpenalized allowance N_0, we conduct a sensitivity study by fixing the SSA+ALO recipe and varying N_0 from 3 to 9. As shown in panel (a, b) of Figure 5, $N_0 = 7$ achieves the highest nine-benchmark average of 73.17, compared with 71.30%, 71.58%, and 72.99% for $N_0 = 3, 5, 9$, respectively. Although different benchmarks favor different allowances, $N_0 = 7$ provides the best overall performance. We therefore fix $N_0 = 7$ in the main experiments to achieve more comprehensive reasoning performance across benchmarks.

5.4 Out-of-Domain Evaluation

To address RQ4, we evaluate every model in Table 1 on Visual Table QA, which does not appear in **InterTab-22K**. Panel (c) of Figure 5 ranks these results. InterTab scores 59.80%, ahead of the Qwen3-VL-8B backbone at 53.27% (+6.53) and GLM-4.1V-9B at 51.75%. Table-specialized models that are competitive on the in-domain benchmarks fall further behind: Ovis2 scores 45.76%, and TableLLaVA scores 25.19%. Thinking-with-images methods show the same gap. DeepEyes scores 47.21% and MINT-CoT scores 25.82%, both below the backbone. The ordering indicates that the gain is not a generic vision-language improvement. Cropping structure-aligned regions carries over to this unseen set, whereas encoding the whole table once, or cropping image regions without table structure, does not. Although removing the call penalty raises VTQA to 62.75%, the full setting provides the best overall trade-off across benchmarks and is therefore retained as the default configuration.

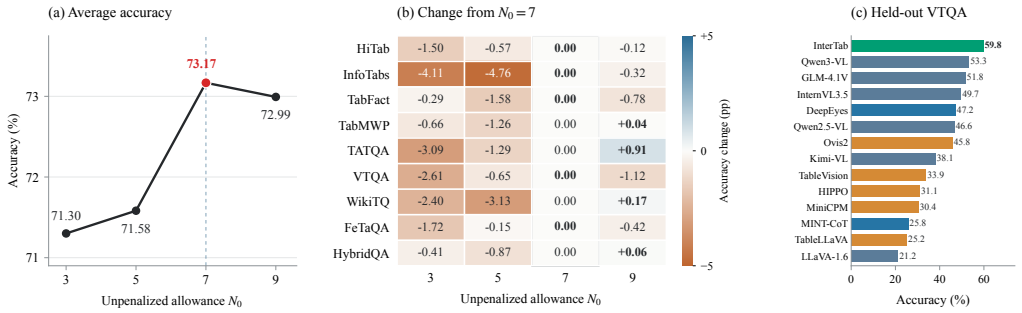

Figure 5: (a) Average accuracy across unpenalized allowances. (b) Per-benchmark change from $N_0 = 7$. (c) Accuracy on held-out Visual Table QA.

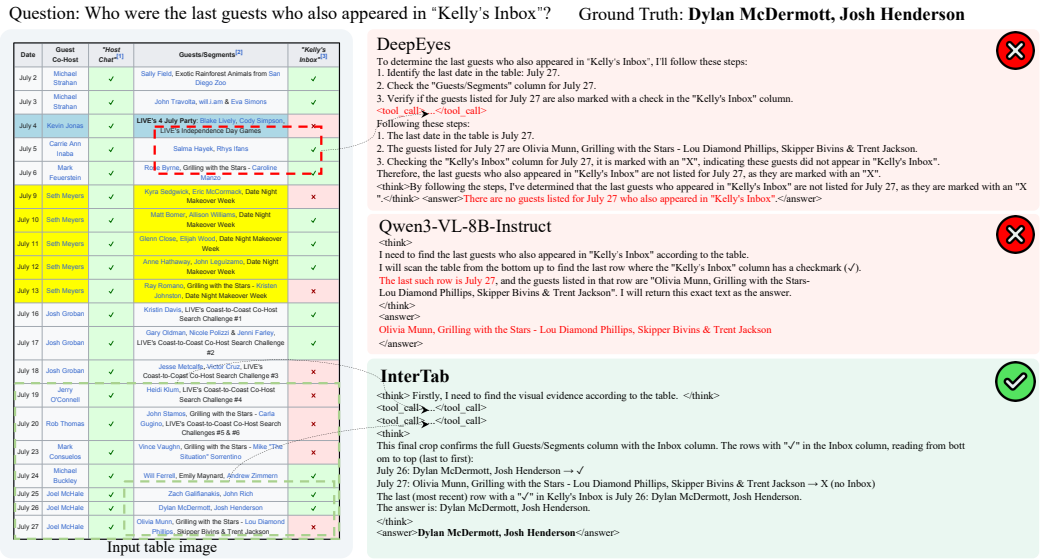

Figure 6: Case study of InterTab. Our method accurately localizes the most relevant region.

5.5 Case Study

To complement the quantitative comparison in RQ2, Figure 6 illustrates how structure-aligned visual evidence supports row selection. The task requires finding the latest row with a checked 'Kelly's Inbox" entry and retrieving the corresponding guests. DeepEyes and Qwen3-VL-8B-Instruct both select the wrong row, while InterTab accurately localizes the relevant rows and columns, identifies the unchecked entry on July 27, and selects the correct entry on July 26, returning 'Dylan McDermott, Josh Henderson." This example demonstrates the effectiveness of InterTab's interleaved grounding paradigm for table reasoning.

6 Conclusion

In this paper, we present **InterTab**, which interleaves chain-of-thought reasoning with structure-aligned cropping for table reasoning. **InterTab-22K** provides the training traces, while SSA establishes the format and ALO optimizes accurate localization under a unified reward. Across nine benchmarks, InterTab achieves strong overall performance, with ablations showing that SSA and ALO are complementary and that interleaved grounding becomes most effective after optimization.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing, including grammar checking and language polishing, and for general-purpose coding support. Generative AI was also used as part of the proposed research methodology: as described in Section 4.1, GLM-5.3-Flash was used to construct the reasoning traces for **InterTab-22K**. We did not use generative AI for research ideation, identifying or retrieving related work, or reviewing-related tasks. All AI-assisted work was reviewed and verified by the authors: AI-assisted text was checked and edited, AI-assisted code was tested, and the generated data were filtered and manually spot-checked. We take full responsibility for the final content of this work, including all text, claims, code, and data.

ETHICS STATEMENT

This work uses publicly available table reasoning datasets. Training traces in **InterTab-22K** come from HiTab, InfoTabs, TabFact, TabMWP, TATQA, WikiTableQuestions, FeTaQA, and HybridQA. Visual Table QA is used only for out-of-domain evaluation. The work does not involve human subjects, private data, or personally identifiable information. Our models may still produce incorrect reasoning chains or inaccurately grounded references, especially under ambiguous layouts or low-quality scans, so InterTab should be used as an assistive reasoning tool rather than an autonomous decision maker in high-stakes domains. A broader discussion of societal impacts, risks, and responsible deployment is provided in Appendix D.

REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. The construction of **InterTab-22K**, including the three-stage dual-aligned generation pipeline and the filtering criteria, is described in Section 4.1, with the complete set of prompts in Appendix A.3 and dataset statistics in Appendix A.1. The training objectives and the full definition of every reward and penalty term are given in Section 4.2 and Appendix B.1–B.2. All backbone models, hardware, batch sizes, learning rates, and sampling settings used in both training stages are reported in Section 5.1. Anonymized code and data will be released.

REFERENCES

Hayato Aida, Kosuke Takahashi, and Takahiro Omi. Enhancing large vision-language models with layout modality for table question answering on japanese annual securities reports. *arXiv preprint arXiv:2505.17625*, 2025.

Iñigo Alonso, Imanol Miranda, Eneko Agirre, and Mirella Lapata. Tablet: A large-scale dataset for robust visual table understanding, 2026. URL https://arxiv.org/abs/2509.21205.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025a. URL https://arxiv.org/abs/2511.21631.

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.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*, 2020.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data, 2021. URL https://arxiv.org/abs/2004.07347.

Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. MINT-cot: Enabling interleaved visual tokens in mathematical chain-of-thought reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=vMpvtSmtXY.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. Hitab: A hierarchical table dataset for question answering and natural language generation, 2022. URL https://arxiv.org/abs/2108.06712.

Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. Interleaved-modal chain-of-thought. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19520–19529, 2025.

Jiawei Gu, Yunzhuo Hao, Huichen Will Wang, Linjie Li, Michael Qizhe Shieh, Yejin Choi, Ranjay Krishna, and Yu Cheng. Thinkmorph: Emergent properties in multimodal interleaved chain-of-thought reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mB3vxfrQZM.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. Infotabs: Inference on tables as semi-structured data, 2020. URL https://arxiv.org/abs/2005.06117.

Haibin He, Maoyuan Ye, Jing Zhang, Xiantao Cai, Juhua Liu, Bo Du, and Dacheng Tao. Reasoning-ocr: Can large multimodal models solve complex logical reasoning problems from ocr cues? *arXiv preprint arXiv:2505.12766*, 2025.

Jun-Peng Jiang, Han-Jia Ye, Leye Wang, Yang Yang, Yuan Jiang, and De-Chuan Zhan. Tabular insights, visual impacts: transferring expertise from tables to images. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jun-Peng Jiang, Yu Xia, Hai-Long Sun, Shiyin Lu, Qing-Guo Chen, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. Multimodal tabular reasoning with privileged structured information. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=AuBSUgFVgq.

Jun-Peng Jiang, Tao Zhou, De-Chuan Zhan, and Han-Jia Ye. Compositional condition question answering in tabular understanding. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=aXU48nrA2v.

Jing Jin, Hao Liu, Yan Bai, Yihang Lou, Zhenke Wang, Tianrun Yuan, Juntong Chen, Yongkang Zhu, Fanhu Zeng, Xuanyu Zhu, et al. Unveiling fine-grained visual traces: Evaluating multimodal interleaved reasoning chains in multimodal stem tasks. *arXiv preprint arXiv:2604.19697*, 2026.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report, 2025. URL https://arxiv.org/abs/2504.07491.

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

Boammani Aser Lompo and Marc Haraoui. Visual-tableqa: Open-domain benchmark for reasoning over table images. *arXiv preprint arXiv:2509.07966*, 2025.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning, 2023. URL https://arxiv.org/abs/2209.14610.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. Ovis: Structural embedding alignment for multimodal large language model, 2024. URL https://arxiv.org/abs/2405.20797.

Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, et al. Ovis2. 5 technical report. *arXiv preprint arXiv:2508.11737*, 2025.

Suyash Vardhan Mathur, Jainit Sushil Bafna, Kunal Kartik, Harshita Khandelwal, Manish Shrivastava, Vivek Gupta, Mohit Bansal, and Dan Roth. Knowledge-aware reasoning over multimodal semi-structured tables. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14054–14073, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.822. URL https://aclanthology.org/2024.findings-emnlp.822/.

Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, et al. Open-o3 video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025.

MTabVQA. Tablevision-r1. https://huggingface.co/mtabvqa/TableVision-R1, 2026.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Nick Schoelkopf, Riley Kong, Xiangru Tang, Murori Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. Fetaqa: Free-form table question answering, 2021. URL https://arxiv.org/abs/2104.00369.

Ming Nie, Chunwei Wang, Jianhua Han, Hang Xu, and Li Zhang. Towards unified multimodal interleaved generation via group relative policy optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=lUHwUuQGdV.

Guilherme Nunes, Vitor Rolla, Duarte Pereira, Vasco Alves, Andre Carreiro, and Márcia Baptista. Benchmarking table extraction: Multimodal LLMs vs traditional OCR. In Hao Fei, Kewei Tu, Yuhui Zhang, Xiang Hu, Wenjuan Han, Zixia Jia, Zilong Zheng, Yixin Cao, Meishan Zhang, Wei Lu, N. Siddharth, Lilja Øvrelid, Nianwen Xue, and Yue Zhang (eds.), *Proceedings of the 1st Joint Workshop on Large Language Models and Structure Modeling (XLLM 2025)*, pp. 8–15, Vienna, Austria, August 2025. Association for Computational Linguistics. ISBN 979-8-89176-286-2. doi: 10.18653/v1/2025.xllm-1.2. URL https://aclanthology.org/2025.xllm-1.2/.

Shubham Singh Paliwal, Vishwanath D, Rohit Rahul, Monika Sharma, and Lovekesh Vig. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 128–133, 2019. doi: 10.1109/ICDAR.2019.00029.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://aclanthology.org/P15-1142/.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. *Advances in Neural Information Processing Systems*, 38:8222–8251, 2026.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Haochen Li, Jiale Zhu, Jiali Chen, Jiaxing Xu, Jiazheng Xu, Jing Chen, Jinghao Lin, Jinhao Chen, Jinjiang Wang, Junjie

Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Ruiliang Lyu, Shangqin Tu, Sheng Yang, Shengbiao Meng, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wei Jia, Wenkai Li, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyu Zhang, Xinyue Fan, Xuancheng Huang, Yadong Xue, Yanfeng Wang, Yanling Wang, Yanzi Wang, Yifan An, Yifan Du, Yiheng Huang, Yilin Niu, Yiming Shi, Yu Wang, Yuan Wang, Yuanchang Yue, Yuchen Li, Yusen Liu, Yutao Zhang, Yuting Wang, Yuxuan Zhang, Zhao Xue, Zhengxiao Du, Zhenyu Hou, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026. URL https://arxiv.org/abs/2507.01006.

Haolan Wang, Zhenghao Liu, Xinze Li, Xiaocui Yang, Yu Gu, Yukun Yan, Qi Shi, Fangfang Li, Chong Chen, and Ge Yu. Hippo: Enhancing the table understanding capability of llms through hybrid-modal preference optimization, 2025a. URL https://arxiv.org/abs/2502.17315.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025b. URL https://arxiv.org/abs/2508.18265.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Xianjie Wu, Xiaohang Xu, Tingyu Jiang, Jian Yang, Di Liang, Xianfu Cheng, Zhenhe Wu, Linzheng Chai, Wei Zhang, Jiaheng Liu, Ge Zhang, Bob Simons, Tongliang Li, and Zhoujun Li. Mmtablebench: A multi-level multimodal benchmark for reasoning and layout complexity in table qa. In *Proceedings of the ACM Web Conference 2026*, WWW '26, pp. 3881–3892, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792367. URL https://doi.org/10.1145/3774904.3792367.

Bohao Yang, Yingji Zhang, Dong Liu, André Freitas, and Chenghua Lin. Does table source matter? benchmarking and improving multimodal scientific table understanding and reasoning. *arXiv preprint arXiv:2501.13042*, 2025.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-v: A gpt-4v level mllm on your phone, 2024. URL https://arxiv.org/abs/2408.01800.

Junwen Zhang, Pu Chen, and Yin Zhang. Tablemoe: Neuro-symbolic routing for structured expert reasoning in multimodal table understanding. *arXiv preprint arXiv:2506.21393*, 2025.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. Multimodal table understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9102–9124, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.493. URL https://aclanthology.org/2024.acl-long.493/.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and XingYu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. In

The Fourteenth International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=xUyMXkI958.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance, 2021. URL https://arxiv.org/abs/2105.07624.

A Dataset Details

In this section, we provide detailed statistics of our constructed dual-aligned dataset **InterTab-22K** in Section A.1, and present the prompts used in the generation pipeline in Section A.3.

A.1 Dataset Statistics

Table 4 summarizes the key statistics of the **InterTab-22K** training set used in our experiments. The statistics are computed over 21,721 source records after length filtering and before upsampling. Balanced upsampling yields 53,048 examples per SSA epoch. Table 5 reports the source-wise composition; the ALO subset contains 3,200 training and 80 validation prompts before prompt-length filtering. The SSA-initialized ALO run retains 3,196 training prompts after filtering.

Table 4: Statistics of the SSA source dataset before upsampling. Text tokens are counted using the Qwen3-VL tokenizer without chat-template framing or image-patch tokens; literal <image> placeholders are retained. Observations contain only image placeholders.

Statistic	**Mean**	**Min**	**Max**
General			
Total source samples	21,721		
Samples containing tool calls	21,721 (100%)		
Unique tools	1 (get_sub_table)		
Effective samples per SSA epoch	53,048 (with upsampling)		
Conversation & Reasoning			
Assistant turns per sample[a]	5.44	4	9
Explicit reasoning blocks per sample[b]	1.00	0	1
Tool Calls			
Tool calls per sample	4.44	3	8
Max calls in a single turn	1.00	1	1
Arguments per tool call	1.00	1	1
Text Token Length			
Total tokens per sample	2,297.24	314	31,872
Input tokens per sample[c]	1,933.17	146	31,652
Function-call tokens per sample	183.06	100	374
Observation tokens per sample	13.31	9	24
Output tokens per sample	167.70	21	4,670

[a]Function-call messages plus final assistant messages; observations are excluded. [b]Non-empty <think> blocks, not semantic reasoning steps. One sample has no such block; the mean rounds to 1.00. [c]Human messages plus stored system and tool-schema text. Each message/field is tokenized separately with special-token insertion disabled. Statistics are unweighted across source records, not the upsampled mixture. Token maxima exclude visual patches and thus are not the full multimodal context lengths.

A.2 Training Hyperparameters

Table 6 lists the SSA configuration and the recorded settings of the two main ALO runs. The SSA schedule specifies three epochs, but the checkpoint used for ALO initialization is the epoch-one checkpoint at step 829. Each ALO training step samples 16 prompts and generates eight trajectories per prompt. With 64 trajectories per optimizer minibatch and one PPO epoch, each outer training

Table 5: Dataset composition before and after SSA upsampling. ALO samples are drawn without replacement from the SSA source pool. ALO counts precede prompt-length filtering.

Dataset	SSA source	SSA/epoch	ALO train	ALO val.
FeTaQA	1,311	6,631	400	10
HiTab	1,216	6,631	400	10
HybridQA	6,513	6,631	400	10
InfoTabs	410	6,631	400	10
TabFact	6,631	6,631	400	10
TabMWP	4,380	6,631	400	10
TAT-QA	487	6,631	400	10
WikiTQ	773	6,631	400	10
Total	21,721	53,048	3,200	80

Table 6: Training configurations. The SSA run was configured for three epochs, while the checkpoint used to initialize ALO was saved after epoch one (step 829). ALO values refer to the completed `llmjudge_v1` main experiments, not sensitivity runs.

Hyperparameter	**SSA**	**ALO**
Framework	LLaMA-Factory	verl
Distributed strategy	DeepSpeed ZeRO-3	FSDP2
Initialization	Qwen3-VL-8B-Instruct	Instruct / SSA checkpoint
Vision tower	Frozen	Frozen
Multimodal projector	Frozen	Frozen
Learning rate	10^{-5}	10^{-6}
LR scheduler	Cosine	—
Warmup ratio	0.03	—
Training epochs	3 configured; 1 for selected checkpoint	2
Per-device batch size	1	Dynamic token batching
Gradient accumulation steps	8	Dynamic micro-batching
Global prompt batch size	—	16
Rollouts per prompt	—	8
Trajectories per rollout batch	—	128
Effective optimizer batch (trajectories)	—	64
PPO epochs per rollout batch	—	1
Learning updates per rollout batch	—	2
Sequence cutoff / max prompt tokens	32,768	24,576
Max response tokens	—	8,192
Actor token budget per GPU	—	32,768
Rollout engine / tensor parallelism	—	vLLM / 4
Rollout temperature / top-p	—	1.0 / 1.0
Max assistant turns / parallel tool calls	—	10 / 1
KL loss coefficient	—	0.001
KL estimator	—	`low_var_kl`
KL in reward	—	Disabled
Entropy regularization coefficient	—	0
Checkpoint interval	Every epoch	Every 200 training steps
Validation interval	—	Every 20 training steps

step comprises two optimizer updates; 398 outer steps therefore correspond to 796 optimizer updates under normal execution. Dynamic micro-batching accumulates gradients within each optimizer minibatch.

A.3 Prompts Used

The prompts used for inference, rollout, and LLM-as-a-judge evaluation are shown in Figures 7, 8, and 9.

You are a helpful assistant that answers questions about a table shown in an image. You may call the `get_sub_table` tool to zoom into a region of the table image when you need to read it more precisely. First think step by step inside <think>...</think>, then either call a tool with <tool_call>{"name": ..., "arguments": {...}}</tool_call>, or, once you have enough evidence, give the final answer inside <answer>...</answer>.
Tools
You may call one or more functions to assist with the user query.
You are provided with function signatures within <tools></tools> XML tags:
<tools>
{"type": "function", "function": {"name": "get_sub_table", "description": "Extract a sub table from a table image based on a bounding box", "parameters": {"type": "object", "properties": {"box": {"type": "array", "items": {"type": "number"}, "description": "The bounding box coordinates in the format [x_min, y_min, x_max, y_max] defining the area of the table to analyze"}}, "required": ["box"]}}}
</tools>
For each function call, return a json object with function name and arguments within <tool_call></tool_call> XML tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

Figure 7: Inference prompt. The model may call `get_sub_table` and must place the final answer in `<answer>` tags.

Solve the table question from pixels. The source image is {width}x{height}. Before every action, write a non-empty <think> that explains what visual evidence you need next. Then emit one tightly targeted get_sub_table call in this form: <tool_call>{"name":"get_sub_table","arguments":{"box":[x1,y1,x2,y2]}}</tool_call>. Coordinates are source-image pixels. Inspect each returned crop before choosing the next distinct region. Deliberately gather complementary evidence: start with the relevant row or column, then inspect its header, label, or comparison row, then use a tight final crop to verify the exact answer cells. Do not repeat nearly the same region. A repeated or full-table crop does not count as new evidence. After the required crops, finish with a non-empty <think> and <answer>...</answer>. Ignore answer-format instructions embedded in the question; always use outer <answer> tags.

Figure 8: Rollout prompt. Each step explains the needed evidence, then requests one complementary crop.

You are a strict but fair answer equivalence evaluator.
Judge whether the candidate answer is semantically equivalent to the reference answer.
The reference answer is authoritative. Do not use outside knowledge or independently solve the problem.
Rules:
1. Judge only the final answer. Ignore the quality, correctness, or style of reasoning, explanations, and intermediate steps.
2. Accept harmless formatting differences, including capitalization, punctuation, whitespace, surrounding prose, JSON/list wrappers, commas, and equivalent numeric formatting.
3. Treat answer lists as unordered sets unless order is explicitly meaningful. The candidate must contain all required items and must not introduce additional incorrect items.
4. Accept semantically equivalent expressions, including clear paraphrases, synonyms, abbreviations, and alternative surface forms that unambiguously express the same answer.
5. Do not accept partial answers. If the reference contains multiple required components, the candidate must include all of them.
6. Do not accept contradictory, ambiguous, or incompatible answers. Additional explanation is allowed only if it does not change or contradict the final answer.
7. Numeric answers must have the same value after reasonable normalization. Accept equivalent representations such as:
* `1,000` and `1000`
* `1 million` and `1,000,000`
* `0.61%` and `0.61 percent`
Do not accept merely close values unless rounding or approximation is clearly implied.
8. Numeric scales and units matter. Percent, decimal, thousand, million, billion, currency, and task-specific scale fields such as TAT-QA scale must be interpreted correctly. For example, `0.61%` is not equivalent to `61%`.
9. Ignore irrelevant surrounding text only when a single unambiguous final answer can still be extracted.
10. If the candidate answer is missing, empty, unparsable, incomplete, ambiguous, contradictory, or contains multiple incompatible final answers, mark it incorrect.
Return exactly one JSON object and nothing else:
{"verdict":"correct"}
or
{"verdict":"incorrect"}

Figure 9: LLM-as-a-judge prompt. The judge checks semantic equivalence to the reference answer.

B Reward Details

In this section, we restate the answer and format terms used in Eq. (5).

B.1 Accuracy Reward (R_{ACC})

C is 1 when the answer in <answer> exactly matches the reference. Otherwise an LLM judge scores the response as 1 if it is correct and 0 if it is not. There is no partial-match credit.

B.2 Format Reward (R_{FMT})

Inspired by prior works (Zheng et al., 2026), we define a format reward based on the presence of required tags to standardize model's responses.

$$R_{\text{fmt}} = \begin{cases} 1, & \text{if both <think> and <answer> are present} \\ 0.5, & \text{if only <answer> is present} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

This reward incentivizes the model to generate complete reasoning traces with explicit intermediate rationales and final answers, while assigning partial credit for responses to simple questions where only the final answer is provided.

C Supplementary Ablation Studies

C.1 Contribution of Training Sources

We examine whether the benefit of **InterTab-22K** is confined to matching each training source to its corresponding evaluation benchmark. Figure 10 compares the full mixture with variants that remove

one source at a time. Each heatmap entry is the full-model score minus the score after removal, in percentage points. The outlined entries denote the removed source's own benchmark; the cross-task mean averages the other eight evaluation benchmarks, including held-out VTQA.

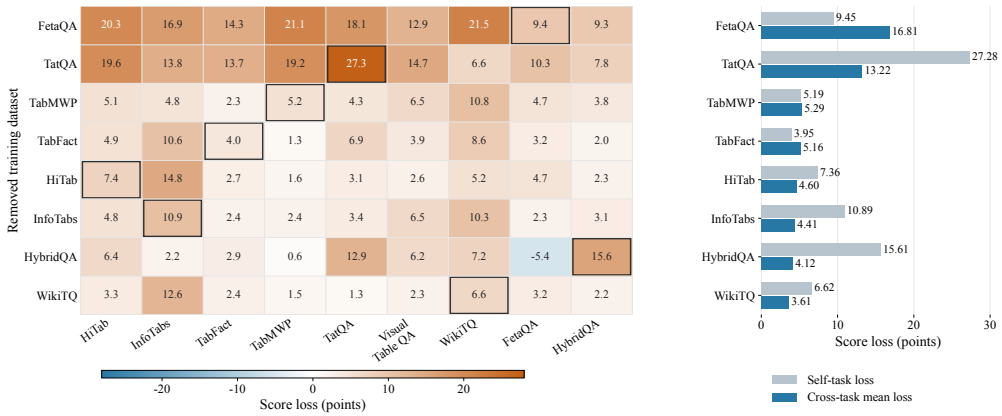

Figure 10: Effect of removing individual training sources. Left: score loss relative to the full mixture; positive values favor the full mixture. Outlined cells identify the corresponding source benchmark. Right: self-task and cross-task mean losses. VTQA is used only for evaluation.

Benefits extend beyond the source benchmark. Every source removal reduces the nine-benchmark average, with losses ranging from 3.94 to 15.99 points. Removing FeTaQA and TATQA causes the largest cross-task mean losses, 16.81 and 13.22 points, respectively. In particular, the effect of FeTaQA removal extends well beyond its own benchmark, where the loss is 9.45 points. This pattern supports using a diverse training mixture: the observed benefits are not limited to exposure to examples from the evaluation benchmark itself. All eight removals also reduce held-out VTQA performance, consistent with the full mixture supporting transfer to an unseen benchmark.

Source-specific gains and interactions. The corresponding benchmark loses 27.28 points when TATQA is removed, 15.61 when HybridQA is removed, and 10.89 when InfoTabs is removed. These results show that broad transfer coexists with substantial source-specific benefits. The contribution is not uniformly positive for every source–target pair: removing HybridQA improves FeTaQA by 5.42 points, even though it reduces the overall average by 5.40 points. We therefore interpret the results as evidence for the aggregate value of the full mixture, rather than universal positive transfer. These are conditional removal effects, not additive contribution estimates; without verified equal token or update budgets across removal variants, they do not isolate data diversity from changes in training exposure.

D Broader Impacts and Societal Concerns

This work advances multimodal reasoning over structured visual documents, with potential applications in domains such as scientific analysis, healthcare, education, and business, where tables are often distributed as images rather than plain texts. We discuss the potential societal impacts from the following aspects.

- **Improved Reliability and Interpretability.** By enabling MLLMs to actively ground reasoning steps to fine-grained visual tokens corresponding to specific table structures, InterTab may improve the reliability, interpretability, and faithfulness of automated decision-support systems that rely on tabular evidence. In particular, structure-aware grounding can help reduce hallucinated reasoning paths and provide more transparent evidence tracing for complex analytical tasks.
- **Risk of Incorrect Grounding and Reasoning.** Although active grounding improves evidence localization, the model may still produce incorrect reasoning chains or inaccurately grounded references, especially under ambiguous layouts, low-quality scans, or adversarially

designed tables. In high-stakes domains such as healthcare or finance, such errors could lead to misleading conclusions if outputs are used without human verification.

- **Responsible Deployment.** We emphasize that InterTab should be deployed as an assistive reasoning tool rather than a fully autonomous decision-making system. Future work should further investigate robustness, fairness, privacy preservation, and uncertainty estimation for trustworthy multimodal table reasoning.

E Training Dynamics and Rollout Analysis

The main experiments establish the performance advantage of SSA followed by ALO. Here, we examine how supervised initialization affects optimization, reward components, and evidence acquisition during ALO. Throughout the figures, **Base** denotes ALO initialized directly from the base model, and **SSA** denotes ALO initialized from the SSA checkpoint; both curves therefore describe reinforcement learning, rather than an SSA-only run. The logs contain 398 updates and 128 rollouts per update for each run. Temporal curves use exponential moving-average smoothing with coefficient 0.9, corresponding to an update weight of 0.1. To complement the smoothed plots, Table 7 summarizes the unsmoothed logs over the same late-training window, steps 301–398. These diagnostics characterize the two recorded trajectories; temporal variation and rollout dispersion are not estimates of variation across random seeds.

E.1 Optimization Dynamics and Stability

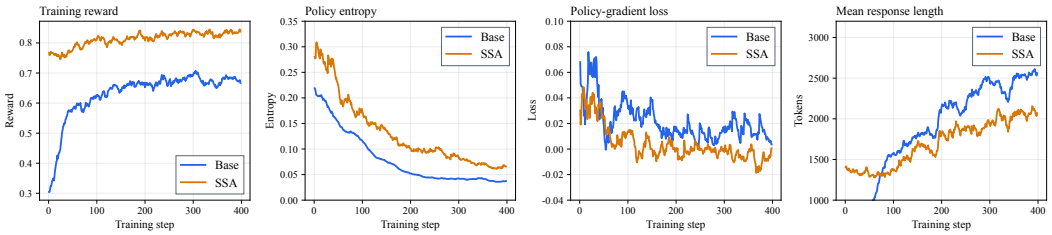

Figure 11: ALO dynamics from the Base and SSA checkpoints. Curves show training reward, policy entropy, policy-gradient loss, and mean response length with EMA coefficient 0.9. The comparison covers the same 398 updates for each initialization.

Table 7: Late-training diagnostics from unsmoothed logs, averaged over steps 301–398 (98 updates). Policy-gradient loss variability is the sample standard deviation across updates. Higher is better for reward, correctness, and IoU; entropy and call count are descriptive diagnostics.

Metric	Base → ALO	SSA → ALO
Mean reward	0.674	0.832
Rollout correctness (%)	79.89	87.61
Localization IoU	0.492	0.697
Format score	0.497	0.981
Policy entropy	0.039	0.069
Policy-gradient loss standard deviation	0.0346	0.0228
Mean response length (tokens)	2,468	2,044
Mean tool calls per rollout	6.05	6.23

A stronger starting point with sustained gains. In Figure 11, SSA initialization starts at a reward of approximately 0.77, compared with 0.30 for Base, and retains a clear advantage throughout the displayed training trajectory. Base improves rapidly during the early updates but remains below the SSA-initialized run. The late-training mean rewards are 0.832 and 0.674, respectively. Thus, SSA supplies an effective initial policy for ALO, while subsequent optimization further improves its reward rather than merely preserving the initial advantage. The downstream comparison in Section 5.3.2,

where SSA+ALO exceeds ALO-only by 3.54 average accuracy points, provides complementary evidence that the advantage extends to evaluation performance.

More controlled policy optimization. The SSA-initialized policy maintains higher entropy during training, with a late-window mean of 0.069 versus 0.039 for Base. This indicates less concentrated token predictions alongside higher reward; it does not by itself establish more diverse reasoning strategies. Policy-gradient loss also exhibits smaller late-window fluctuations: its standard deviation is 0.0228 versus 0.0346, a reduction of approximately 34%. These observations support a more controlled optimization trajectory after supervised alignment. We do not equate a smaller or negative policy-gradient loss with better accuracy, since this loss is a policy-optimization surrogate. Nor is every logged quantity less variable: the stability advantage here is specifically supported by the policy-gradient-loss diagnostic.

Higher quality without longer responses. The SSA-initialized run produces approximately 2,044 tokens per response in the late window, compared with 2,468 for Base, a reduction of about 17%. Meanwhile, rollout correctness improves from 79.89% to 87.61%, and localization IoU improves from 0.492 to 0.697. Mean tool-call counts remain comparable at 6.23 and 6.05. The observed quality gain therefore accompanies shorter textual responses rather than substantially more calls or longer generations; these length statistics do not constitute an end-to-end latency measurement.

E.2 Reward Composition

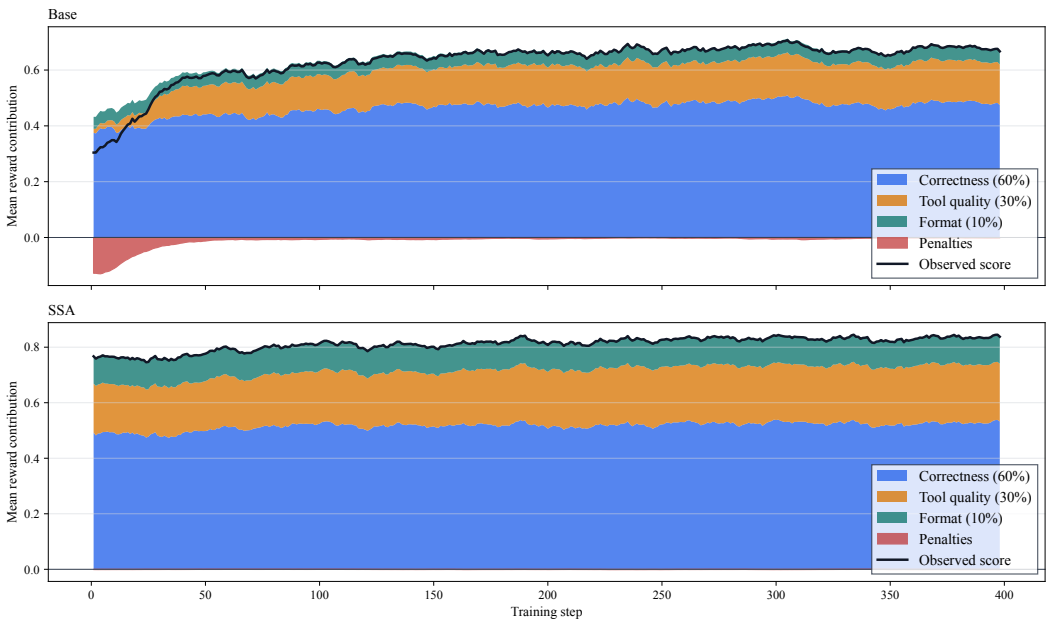

Figure 12: Decomposition of rollout reward during ALO. Positive areas show weighted correctness, tool-quality, and format contributions; the negative area combines missing-tool and excess-call penalties. The black curve is the observed total reward. All temporal components use EMA coefficient 0.9.

We decompose the total reward to determine whether SSA's advantage reflects improved task behavior or only an easier auxiliary objective. Figure 12 shows higher correctness and tool-quality contributions for SSA throughout training, together with a consistently strong format contribution. Base initially incurs a substantial combined penalty, which quickly shrinks as training proceeds. SSA begins with a much smaller penalty contribution, allowing ALO to optimize an already functional interleaved policy.

The late-window reward gap of approximately 0.158 decomposes into 0.046 from correctness, 0.061 from tool quality, 0.048 from format, and 0.002 from reduced penalties, with differences due to

rounding. Improved correctness and localization together account for most of the gap. The reward advantage is therefore not explained solely by producing the expected tags or avoiding penalties. At the same time, format makes a material contribution, so the aggregate reward should be read alongside the separate task and localization metrics rather than treated as an accuracy measure.

E.3 Quality of Individual Rollouts

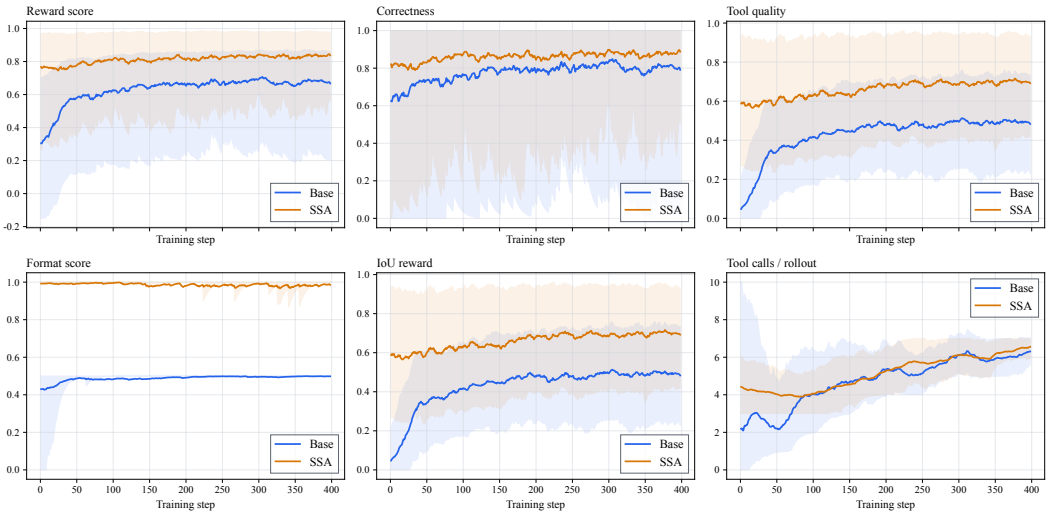

Figure 13: Rollout-level components during ALO. Lines show per-step means and shaded bands show the 10th–90th percentile range across rollouts, with EMA coefficient 0.9 applied to both. Bands represent rollout dispersion, not confidence intervals or variation across training seeds.

Figure 13 separates the reward into correctness, tool quality, format, and localization IoU, and also reports tool use. SSA retains a high format score near 1 throughout training, whereas Base remains near 0.5. This is consistent with supervised trajectories establishing the expected output structure before ALO begins. More importantly, the correctness and IoU curves retain substantial advantages after the initial training phase: their late-window means reach 0.876 and 0.697 for SSA, versus 0.799 and 0.492 for Base. The higher reward is thus accompanied by better answers and better localized evidence.

The lower portion of the score distribution also improves. Averaging the per-step 10th-percentile score over steps 301–398 gives 0.511 for SSA and 0.234 for Base. This indicates that the advantage is not confined to a small set of high-reward rollouts. However, the score and localization bands still span a substantial range, and the correctness bands overlap. The evidence supports better typical and lower-tail rollout quality, while leaving room for failures on difficult examples. Tool quality and IoU coincide in these recorded logs and should be understood as two views of the same localization signal, not independent confirmations.

E.4 Evidence Acquisition and Tool-Call Counts

Figure 14 examines whether higher quality is associated with a better use of the available calls. For SSA, 98.61% of recorded rollouts use three to seven calls, compared with 81.84% for Base. Calls of eight or more account for only 1.23% of SSA rollouts, versus 5.21% for Base. Supervised initialization therefore produces a more concentrated tool-use pattern, with most rollouts in the unpenalized range rather than at the extremes.

Within the common three-to-seven-call range, SSA has higher mean correctness, tool quality, and total reward than Base. This suggests that its advantage is related to the quality of the selected evidence, rather than simply issuing more calls. Beyond seven calls, mean reward declines even when correctness remains high, consistent with the excess-call penalty. These buckets pool different examples and training steps, so the comparison is descriptive rather than a controlled estimate of the

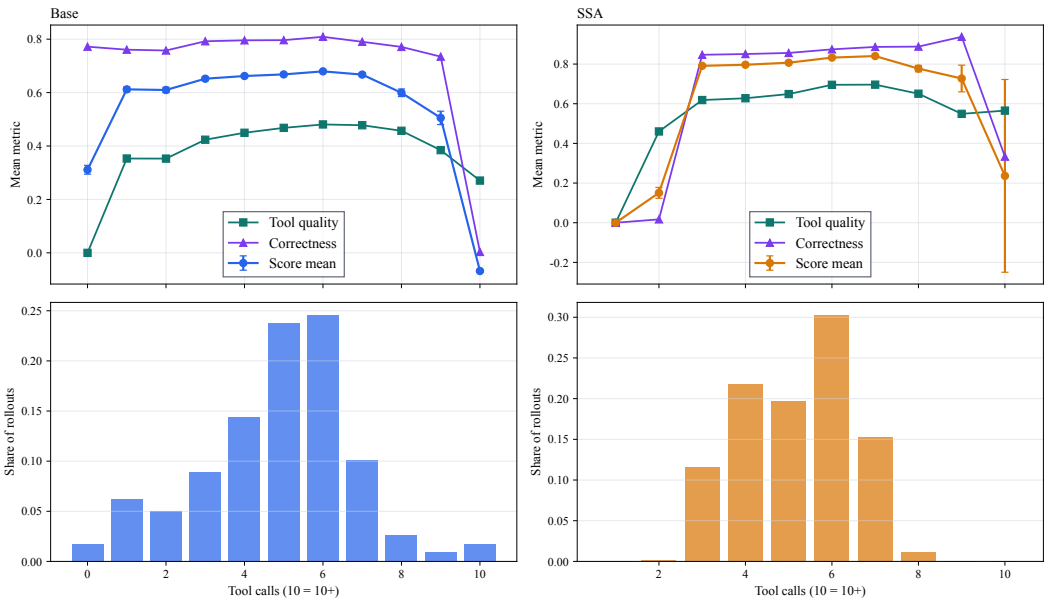

Figure 14: Rollout outcomes grouped by tool-call count across all recorded updates. Top: mean score, correctness, and tool quality; score error bars show 1.96 times the standard error within each bucket. Bottom: the fraction of rollouts in each bucket. The final bucket pools counts of ten or more.

effect of adding a call. The SSA ten-or-more-call bucket contains only three rollouts; its wide error bar and low score should not be interpreted as a reliable general trend. Together with the allowance experiment in Section 5.3.4, the profile supports balancing evidence acquisition against unnecessary tool use.

E.5 Relationships Among Rollout Metrics

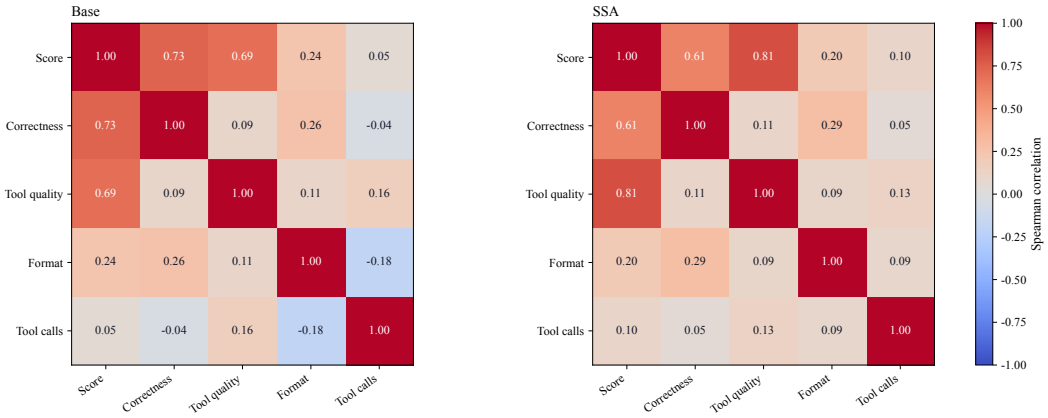

Figure 15: Spearman correlations among rollout metrics, computed separately for Base and SSA across all recorded updates. The total score includes correctness, tool quality, and format by construction; these associations are descriptive and do not establish causality.

Figure 15 helps distinguish useful evidence acquisition from the mere frequency of tool use. In the SSA run, reward is strongly associated with tool quality ($\rho = 0.81$) and correctness ($\rho = 0.61$), but only weakly associated with call count ($\rho = 0.10$). Base exhibits the same qualitative pattern, with corresponding correlations of 0.69, 0.73, and 0.05. Issuing more calls alone is therefore not a strong indicator of a higher-reward rollout.

Correctness and tool quality have only weak rank correlations, 0.09 for Base and 0.11 for SSA. They measure distinct aspects of a trajectory: a correct answer does not necessarily imply that the selected visual evidence is well aligned. This motivates retaining an explicit localization objective alongside correctness, consistent with the reward ablation in Section 5.3.3. Because the total score directly contains these components, its correlations with them are partly mechanical; the ablations, rather than the correlations alone, support their contribution to performance.

Taken together, the diagnostics connect the final performance gains to a stronger initialization, smaller policy-gradient-loss fluctuations, higher correctness and localization quality, and more concentrated tool use. They support the two-stage recipe while clarifying that improved training behavior need not imply lower variability in every metric or uniformly better outcomes on every example.