BiVER: Bidirectional Visual Evidence Rewards for Multi-Tool VLM Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Vision Language Models (VLMs) can actively manipulate images through tool calls during reasoning, yet common reinforcement learning objectives provide little direct feedback on the images these tools produce. An irrelevant tool output thus inherits a positive reward whenever the answer is recoverable without it, while an informative one receives no credit when the final answer is wrong. We show that a frozen VLM can assess returned visual evidence along two complementary directions: a *forward* answer-utility score measures support for the reference answer, while a *backward* question-alignment score measures relevance to the question without using the answer. Building on this finding, we propose BiVER (**Bi**directional **V**isual **E**vidence **R**eward) for multi-tool VLM reasoning. It scores intermediate tool outputs in both directions and incorporates their trajectory-level aggregate into GRPO. Because BiVER evaluates the image produced by an executed action rather than its name or arguments, it provides a shared reward criterion for heterogeneous image-producing tools without supervised tool-use trajectories or training a dedicated reward model. Trained from Qwen2.5-VL-7B-Instruct with six tools, BiVER improves its initialization by an average of **11.9** points across high-resolution benchmarks and **5.0** points across document, chart, and table benchmarks, achieving the highest average accuracy among the compared models for each benchmark suite.

1 Introduction

Recent Vision Language Models (VLMs) increasingly treat an image not as a static input but as a workspace that can be manipulated during reasoning (Su et al., 2025). By cropping a region of interest, rotating a skewed page, or drawing auxiliary marks, a model can turn a perception failure into a tractable sub-problem and re-inspect the resulting image. Such behavior is often trained with reinforcement learning (RL) and verifiable answer rewards, using methods such as group relative policy optimization (GRPO) (Shao et al., 2024; Zheng et al., 2026; Su et al., 2026; Lai et al., 2026). These rewards provide limited information about whether a tool call produces useful visual evidence.

This outcome-only reward is blind to what the tools actually produce. Success hinges on intermediate decisions, such as which region to crop or whether to rotate a page before reading it, yet the reward inspects only the final answer. A trajectory whose crop misses the relevant region is still rewarded when the model can answer from the original image or from parametric knowledge, whereas a trajectory whose crop exposes exactly the needed evidence receives nothing if the model subsequently misreads it (Ma et al., 2026). Existing approaches use curated tool-use trajectories or rewards designed for particular actions, such as bounding-box overlap for crops and tool-specific supervision for diverse visual operations (Zhang et al., 2025; Carvalho et al., 2026). These approaches provide useful training signals, but require supervision or reward rules tied to the evaluated action.

We therefore ask whether the usefulness of a tool call can be judged from the image it produces, which we call the *visual tool output*, without access to the tool name, its arguments, or the original image. In a controlled intervention on VizWiz-VQA-Grounding (Chen et al., 2022), we replace each crop issued by the model with a crop of the annotated evidence region, same-sized crops of displaced regions, or a blank image. Given a visual tool output and the question, and the reference answer for the forward score, a frozen VLM evaluator assesses the output along two complementary directions, without access to the tool name, its arguments, or the original image: a *forward answer-*

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.

utility score measures how strongly the tool output supports the reference answer, and a *backward question-alignment* score measures whether it contains the evidence the question asks for. The latter never sees the answer and thus remains informative on trajectories that end in a wrong answer.

Building on these findings, we propose **BiVER**, a **Bi**directional **V**isual **E**vidence **R**eward for multi-tool VLM reasoning. During GRPO training, a frozen evaluator scores every successfully executed tool output in both directions; the scores are averaged within each trajectory, standardized across the trajectories sampled for the same prompt, and added to the outcome reward. BiVER thus evaluates evidence per tool output while assigning credit per trajectory, so that trajectories producing more question-aligned and answer-useful images receive higher advantages even when their final answers are equally correct or wrong. This rewards useful visual evidence beyond final-answer correctness.

We instantiate BiVER on Qwen2.5-VL-7B with six image-producing tools (cropping, rotation, flipping, horizontal and vertical guide lines, and point marking) and train it on a 50K-prompt mixture. Across V^* Bench, HR-Bench 4K, and HR-Bench 8K, BiVER improves its initialization by 11.9 points on average, reaches an average score of 82.5, and surpasses the 4.6$\times$ larger Qwen2.5-VL-32B. The gains extend to DocVQA-RF, InfoVQA-RF, CharXiv, and TableVQA, where BiVER improves the initialization by 5.0 points on average. Reward and tool-set ablations examine the contributions of both evidence scores and tool availability, while tool-use statistics show how tool selection varies across benchmark suites. Our contributions are:

- A controlled intervention study showing that a frozen VLM can judge the value of a visual tool output from the returned image and question via two complementary visual evidence scores.
- BiVER, which aggregates these scores into a group-normalized, trajectory-level reward for GRPO and applies the same evaluation criterion across heterogeneous image-producing tools without annotated tool-use trajectories or training a dedicated reward model.
- Gains over the initialization on all seven benchmarks and the best average accuracy among compared methods, with reward and tool-set ablations and an analysis of tool-use behavior.

2 Related Work

2.1 Reinforcement Learning for Multimodal Reasoning

Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024) has emerged as a data efficient alternative to supervised chain-of-thought distillation and has been widely applied to multimodal reasoning. Visual-RFT (Liu et al., 2025) extends group relative policy optimization to perception and grounding tasks, while Vision-R1 (Huang et al., 2026) combines multimodal cold start data with outcome based RL to elicit longer reasoning traces. Point-RFT (Ni et al., 2026) and ThinkLite-VL (Wang et al., 2026) further refine the training pipeline by grounding reasoning steps to image coordinates and selecting hard examples via MCTS, respectively. Beyond static images, recent work extends multimodal reasoning to spatial and camera motion understanding in videos (Wu et al., 2026; Gu et al., 2026). Other methods explicitly optimize visual search: Chain-of-Focus (Zhang et al., 2025) and CropVLM (Carvalho et al., 2026) learn adaptive or question-conditioned zooming through RL. These approaches demonstrate the effectiveness of outcome supervision, but they either depend on curated reasoning or tool trajectories, or provide only a terminal learning signal that cannot determine whether an intermediate visual action is genuinely useful. Process Reward Models (PRMs) address this granularity gap by scoring individual steps, and VisualPRM (Wang et al., 2025a) extends step-level scoring to multimodal chains of thought; related work also leverages LLM-as-judge (Zheng et al., 2023; Lee et al., 2023) to provide reward signals. ToolsRL (Dong et al., 2026) and RuCL (Chen et al., 2026) improve reasoning by introducing a curriculum learning approach. In contrast, our method derives process supervision directly from standard VQA annotations, rewarding each intermediate visual result for both question alignment and answer utility.

2.2 Thinking with Images

Thinking with images methods (Su et al., 2025) turn vision from a static input into an intermediate reasoning workspace. Early neuro symbolic systems such as VisProg (Gupta & Kembhavi, 2023) and ViperGPT (Surís et al., 2023) prompt language models to compose pretrained vision

modules and image-processing functions into executable programs. Subsequent work more tightly interleaves visual operations with reasoning: Image-of-Thought (Zhou et al., 2024) extracts question conditioned visual rationales step by step, and Visual Sketchpad (Hu et al., 2024) draws marks, boxes, and auxiliary constructions on an image. A parallel line treats high-resolution perception as guided visual search: V^* (Wu & Xie, 2024) formulates search as an LLM guided procedure over image regions, and ZoomEye (Shen et al., 2025) performs tree based exploration of a zoom hierarchy without training. Recent trainable approaches allow models to acquire visual actions through reinforcement learning: DeepEyes (Zheng et al., 2026) incentivizes interleaved zoom-in behavior via outcome based RL, Pixel Reasoner (Su et al., 2026) introduces a curiosity bonus to encourage pixel space operations, Mini-o3 (Lai et al., 2026) scales interaction turns and diversifies reasoning patterns for deep visual search, and Simple-o3 (Wang et al., 2025c) investigates interleaved vision language reasoning with a minimal recipe. Distinctively, our work provides a diverse tool set that the model can compose, and evaluates the result of tool call to provide fine grained process supervision.

3 Motivating Study

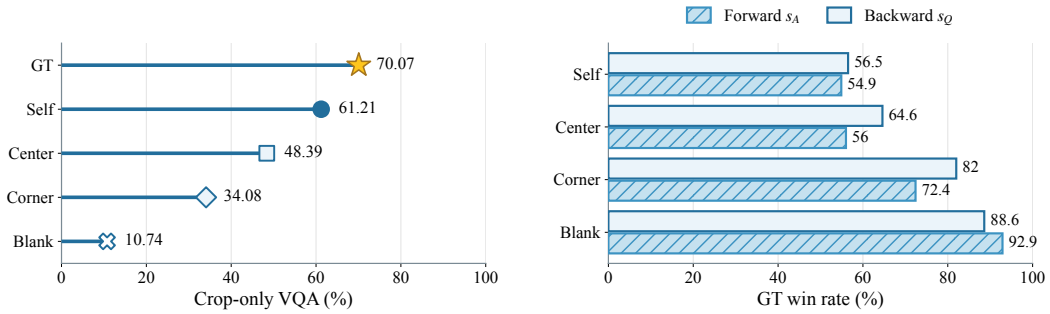


Figure 1: **Visualization of the tool-output intervention.** Left: crop-only VQA from each candidate tool output. Right: GT win rates under the forward and backward scores.

We isolate two properties of an executed tool output V: *(i) forward answer utility*, whether V supports the ground-truth answer, and *(ii) backward question alignment*, whether V contains evidence sought by the question. The first is answer conditioned; the second does not reveal the answer. Both properties must be assessable from the tool output *alone*, without access to the original image. We choose zoom crops for this controlled study because their grounding annotations enable precise interventions. Crucially, neither score uses the tool name or its arguments, so the scoring extends naturally to other image returning tools beyond cropping.

3.1 Experimental Setup

We construct a 2000 example study set from VizWiz VQA Grounding (Chen et al., 2022), which provides ten human answers and a manually annotated answer-grounding polygon for each question. Five examples have empty or degenerate polygons and are excluded, leaving 1,995 examples. The policy and the evaluator are the same Qwen2.5-VL-7B-Instruct model. For each example, the model observes the original image and question and is required to issue exactly one `image_zoom_in_tool` call. We retain only calls that satisfy the XML/JSON and bounding box constraints of the training environment and do not contain a premature final answer; 88.7% (1,769 examples) produce a valid call. From each legal call we construct five candidate tool outputs that share the same question and tool call text; only the returned image differs: GT (bounding box enclosing the human grounding polygon), Self (model predicted bounding box), Center (centered crop of the same size as GT), Corner (same sized corner crop with minimum GT overlap), and Blank (all black image of the same dimensions).

Isolated tool output evaluation. After the intervention we discard the dialogue history and present the frozen evaluator with exactly one tool output V and the question q. For each (q, V) we record three measurements. First, the evaluator answers the question from V alone; we score its

answer with the standard VizWiz consensus score against the ten human answers (*crop-only VQA*). Second, the *forward answer utility* s_A measures the length normalized likelihood that the evaluator produces the reference answer given only the tool output and question. Third, the *backward question alignment* s_Q quantifies whether V contains evidence relevant to q via a binary Yes/No log-odds judgment. Neither score accesses the original image I; formal definitions are given in Section 4.2 (Equations 2–3). All comparisons below are paired on the same legal trajectories.

3.2 Forward Study: Does the Tool Output Support the Answer?

Figure 1 (left) shows a strict ordering of crop-only VQA: GT $>$ Self $>$ Center $>$ Corner $>$ Blank. The human grounded crop reaches 70.07, exceeding the model's own crop by 8.86 points. Relative to the human-grounded crop, the geometry matched center and corner crops reduce crop-only VQA by 21.68 and 35.99 points, respectively, while the blank image reduces it by 59.33 points. Because the evaluator observes only the question and the tool output, these gaps cannot be explained by an extra generation turn, tool call formatting, crop resolution, or visual token count. Since the question remains fixed across paired interventions and only the visual content of the tool output changes, the performance differences reflect the information contained in the returned evidence.

Validating forward utility. We further investigate whether s_A accurately reflects this outcome. Specifically, for each question, we construct all possible pairs of tool outputs in which crop-only VQA differs; s_A ranks the better tool output higher in **90.6%** of pairs. Its GT win rates grow monotonically with the severity of the intervention (54.9% against Self, 92.9% against Blank). The practically decisive test is downstream selection: among Self, Center, Corner, and Blank, choosing the tool output with the highest s_A raises crop-only VQA from 61.21 for Self to 67.72, a gain of 6.51 points, against an oracle ceiling of 70.07 for GT. We evaluate each tool output in isolation because conditioning on the original image can mask differences among tool outputs.

3.3 Backward Study: Can Question Alignment Rank Crops?

Figure 1 (right) shows that s_Q separates GT from spatially irrelevant and blank tool outputs: GT outranks Corner on 82.0% of trajectories and Blank on 88.6%. GT only marginally outranks Self (56.5%), which is expected: a tight model crop that already contains concentrated evidence is as aligned with the question as the human polygon box, so s_Q should be read as evidence usefulness rather than oracle box reconstruction. Note that s_Q uses the binary log-odds of Equation 2, which needs neither distractor candidates nor generation and never sees the answer.

Validating backward alignment. Although s_Q never accesses the answer, it agrees with crop-only VQA on **86.2%** of within-question pairs. Selecting the tool output with the highest s_Q raises crop-only VQA from 61.21 to 64.22, a gain of 3.01 points. Question alignment therefore selects tool outputs that improve the downstream answer without access to the answer.

4 Method

Motivated by the finding that a frozen evaluator can rank tool outputs by usefulness from the tool output alone, we design a visual reward that scores every tool output along both directions and integrates the result into GRPO training, as shown in Figure 2. The central design choice is to score the *realized visual consequence* of an action rather than its name or arguments, giving a common supervision interface for heterogeneous tools, without human annotations for intermediate steps.

4.1 Trajectories with Visual Tool Outputs

Given an image–question pair (I, q), the policy π_θ generates a multi turn trajectory τ_i that interleaves language generation, tool calls, returned images, and a final answer. Its action space contains six deterministic image returning operators: cropping, rotation, flipping, horizontal and vertical guide line drawing, and point marking. Each operator may act on either the original image or a previously returned image, allowing the policy to compose visual transformations over multiple turns.

Suppose that τ_i contains K_i attempted tool calls $\{(c_{i,t}, e_{i,t}, V_{i,t})\}_{t=1}^{K_i}$, where $c_{i,t}$ is the call, $e_{i,t} \in \{0, 1\}$ denotes successful execution, and $V_{i,t}$ is defined as the visual tool output when $e_{i,t} = 1$ and

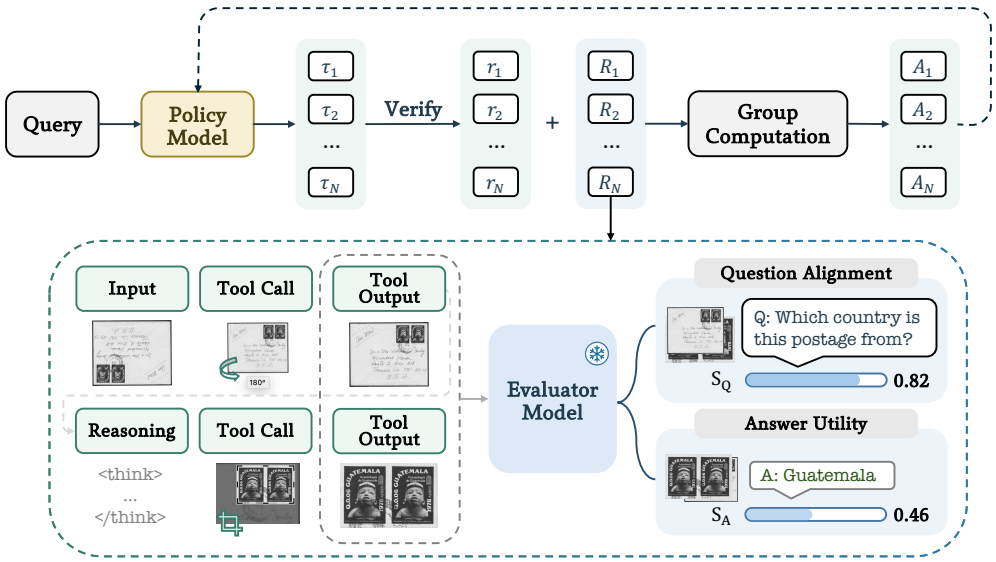


Figure 2: **Method overview.** The policy iteratively selects from six image returning tools and observes their outputs. The evaluator scores every successfully executed, non duplicate tool output independently in two directions: backward question alignment and forward answer utility. Per-output scores are averaged into S_Q and S_A (bars rescaled to $[0, 1]$ for illustration), normalized within each GRPO prompt group, and combined with the outcome reward.

as $\varnothing$ otherwise. We remove failed calls and exact pixel duplicates, preserve the first occurrence of each image, and re-index the retained tool outputs in chronological order:

$$\mathcal{V}(\tau_i) = (V_{i,1}, \ldots, V_{i,T_i}), \qquad 0 \le T_i \le T_{\max}. \tag{1}$$

The policy observes every successful result during rollout, including outputs later removed as duplicates. Reward computation, however, operates only on $\mathcal{V}(\tau_i)$. Each retained image is evaluated independently, without the corresponding tool name, coordinates, or policy reasoning. Consequently, different actions are compared through the evidence they expose rather than through tool-specific heuristics, which provides a unified supervision interface for all tools.

4.2 Bidirectional Visual Evidence Scores

Let p_ϕ be a frozen multimodal evaluator whose parameters are held fixed throughout policy optimization. For every retained tool output $V_{i,t}$, the evaluator scores it along two complementary directions, validated in the motivating study (Section 3). The prompts $\mathcal{P}_Q(q, V_{i,t})$ and $\mathcal{P}_A(q, V_{i,t})$ contain the image returned by the tool and the question; the reference answer is only scored as a continuation of $\mathcal{P}_A$. Neither prompt contains the original image I, the preceding dialogue, or tool metadata, so within-prompt comparisons reflect differences in the content of the tool outputs alone.

Question-alignment score. The backward alignment measures whether an output contains visual evidence relevant to the information requested by q. Rather than assigning likelihood to the entire question, which can be dominated by linguistic regularities, we use the binary log-odds

$$s_Q(q, V_{i,t}) = \log \frac{p_\phi\big(\text{Yes} \mid \mathcal{P}_Q(q, V_{i,t})\big)}{p_\phi\big(\text{No} \mid \mathcal{P}_Q(q, V_{i,t})\big)}. \tag{2}$$

This score is independent of the answer and therefore rewards question directed evidence even when the sampled trajectory ultimately answers incorrectly, preventing answer leakage during supervision.

Answer-utility score. The forward utility measures whether the same output supports the reference answer a. We use its teacher forced, token length normalized log likelihood:

$$s_A(q, a, V_{i,t}) = \frac{1}{|a|} \log p_\phi\big(a \mid \mathcal{P}_A(q, V_{i,t})\big), \tag{3}$$

where $|a|$ is the number of answer tokens. In contrast to s_Q, this term is answer conditioned and tests whether question relevant content is sufficiently informative about the desired answer. The two scores address distinct failure modes: alignment alone may reward relevant but insufficient evidence, whereas utility alone need not distinguish evidence selected for the question from incidental support.

Trajectory aggregation. To obtain one score per trajectory, we average the per tool output scores within each direction: $S_{Q,i} = \frac{1}{T_i}\sum_t s_Q(q, V_{i,t})$ and $S_{A,i} = \frac{1}{T_i}\sum_t s_A(q, a, V_{i,t})$. Unlike a sum, the mean gives no unconditional reward for more calls ; exact pixel duplicates are dropped beforehand.

4.3 Group Normalized Visual Evidence Reward

Before the two trajectory level scores can be combined into a single reward term, their heterogeneous scales must be reconciled. The raw values of S_Q and S_A are not directly commensurate: they arise from different likelihood contrasts, and their scales vary substantially across questions. GRPO, in contrast, estimates relative advantages among trajectories sampled for the same prompt. Let $\mathcal{I}_g = \{i : T_i > 0\}$ index trajectories with a valid process score in prompt group g. For any scalar score x_i, define $z_g(x_i) = (x_i - \mu_g(x))/\sigma_g(x)$, where $\mu_g(x)$ and $\sigma_g(x)$ are computed over $i \in \mathcal{I}_g$. This within-prompt normalization removes question-level offsets and retains only variation among alternative trajectories. If $|\mathcal{I}_g| < 2$ or $\sigma_g(x) = 0$, we set the corresponding process shaping term to zero, avoiding noisy gradient updates when trajectory variation is insufficient.

As shown in the motivating study, the two scores capture complementary aspects of evidence quality: s_Q identifies question relevant evidence without access to the answer, while s_A verifies whether that evidence actually supports the correct answer. Combining them rewards tool outputs that are simultaneously question-aligned and answer-useful, mitigating the failure modes of either signal alone. We first normalize the two directions separately,

$$\widetilde{S}_{Q,i} = z_g(S_{Q,i}), \qquad \widetilde{S}_{A,i} = z_g(S_{A,i}), \tag{4}$$

so neither direction dominates merely because of its numerical scale. We then form their equal-weight combination $J_i = \frac{1}{2}\big(\widetilde{S}_{Q,i} + \widetilde{S}_{A,i}\big)$ and normalize again to obtain $Z_i = z_g(J_i)$. The second normalization controls the variance of the combined signal, which otherwise depends on the within-group correlation between the two directions. For a single-direction variant, J_i is the surviving normalized score. We set $Z_i = 0$ for trajectories outside $\mathcal{I}_g$ and for groups left unshaped by the conditions above. Appendix A provides derivations of the aggregation and normalization properties.

Training objective. Let R_i^{base} denote the original DeepEyes (Zheng et al., 2026) trajectory reward, which evaluates answer correctness, output format, and valid tool use. We add the bidirectional process term to obtain

$$R_i = R_i^{\text{base}} + \lambda Z_i. \tag{5}$$

Although its evidence is collected at intermediate states, Z_i is a trajectory-level shaping signal rather than a stepwise reward. GRPO therefore assigns relative advantage to complete trajectories whose tools produce more informative tool outputs; evidence is evaluated per output, but credit is assigned per trajectory. This preserves the original optimization procedure while making its return sensitive to intermediate tool consequences, including those preceding an incorrect final answer. Because Z_i is standardized within each prompt group, λ controls the strength of process shaping relative to the base reward. We attach R_i to the final response and use the standard GRPO advantage estimator.

5 Experiments

5.1 Experimental Setup

Training datasets. We build on the training dataset of DeepEyes (Zheng et al., 2026), which pairs tool oriented data fine-grained visual search on natural images and ArxivQA scientific figures with the general purpose ThinkLite-VL question answering set. To create situations in which tools other than zooming are useful, we further add a subset of DocVQA (Mathew et al., 2021) pages, 70% of which are rotated or flipped. The full mixture contains 50,052 prompts. Every image tool row exposes the same six tools during training, so no tool is tied to a particular data source. The composition of each part and the augmentation procedure are detailed in Appendix C.

Table 1: **Results on high-resolution benchmarks.** The "Avg." column reports the unweighted mean of the three benchmark overall scores. The mark * indicates reproduced results. The best results are highlighted in **bold**, while the second-best are underlined.

Model	Param Size	V^* Bench			HR-Bench 4K			HR-Bench 8K			Avg.
		Attr	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	Overall	
Proprietary Models											
GPT-4o (Hurst et al., 2024)	–	–	–	66.0	70.0	48.0	59.0	62.0	49.0	55.5	60.2
o3 (OpenAI, 2025)	–	–	–	95.7	–	–	–	–	–	–	–
Open-Source General-Purpose Models											
Qwen2.5-VL (Bai et al., 2025b)	7B	75.7	77.6	76.4	82.8	59.5	71.1	74.5	54.2	64.4	70.6
Qwen2.5-VL (Bai et al., 2025b)	32B	87.8	88.1	87.9	89.8	58.0	73.9	84.5	56.3	70.4	77.4
Thinking with Images											
ZoomEye (Shen et al., 2025)	7B	93.9	85.5	90.6	84.3	55.0	69.7	88.5	50.0	69.3	76.5
ToolsRL (Dong et al., 2026)	7B	**95.6**	89.4	**92.5**	91.2	60.6	75.9	88.1	58.3	73.2	80.5
Pixel-Reasoner (Su et al., 2026)	7B	83.5	76.3	80.6	86.0	60.3	72.9	80.0	54.3	66.9	73.5
Mini-o3* (Lai et al., 2026)	7B	86.1	86.8	86.4	86.0	**66.0**	76.0	81.0	59.8	70.4	77.6
DeepEyes* (Zheng et al., 2026)	7B	91.3	85.5	89.0	92.2	59.5	75.9	86.2	60.0	73.1	79.3
BiVER	7B	92.2	**89.5**	91.1	**95.0**	63.2	**79.1**	**91.0**	**63.8**	**77.4**	**82.5**
Δ(Qwen2.5-VL-7B)		*↑ 16.5*	*↑ 11.9*	*↑ 14.7*	*↑ 12.2*	*↑ 3.7*	*↑ 8.0*	*↑ 16.5*	*↑ 9.6*	*↑ 13.0*	*↑ 11.9*

Baselines and benchmarks. We compare with proprietary GPT-4o (Hurst et al., 2024) and o3 (OpenAI, 2025), open-source Qwen2.5-VL (Bai et al., 2025b), and thinking with images methods including ZoomEye (Shen et al., 2025), Pixel-Reasoner (Su et al., 2026), Mini-o3 (Lai et al., 2026), DeepEyes (Zheng et al., 2026), and ToolsRL (Dong et al., 2026). Evaluations span three task families: (i) *high-resolution visual reasoning* (V^* Bench (Wu & Xie, 2024), HR-Bench 4K/8K (Wang et al., 2025b)), (ii) *document understanding* (DocVQA-RF (Mathew et al., 2021), InfoVQA-RF (Mathew et al., 2022)), and (iii) *chart and table understanding* (CharXiv (Wang et al., 2024), TableVQA-Bench (Kim et al., 2024)). Detailed descriptions are provided in Appendix B.

Implementation details. The main policy is initialized from Qwen2.5-VL-7B-Instruct, and a frozen copy of the initial checkpoint serves as the process evaluator p_ϕ, keeping the reward self-contained without a stronger external model. For each training prompt, we sample $G = 16$ trajectories, allow at most $T_{\max} = 5$ tool calls, and set the process-reward coefficient to $\lambda = 0.3$. Training is implemented with the `verl` framework (Sheng et al., 2024) on NVIDIA H200 GPUs. The complete training hyperparameters, and evaluator prompts are provided in Appendix C and Appendix E.2.

5.2 Main Results

High-resolution fine-grained visual reasoning. As shown in Table 1, BiVER outperforms the baseline Qwen2.5-VL-7B across all high-resolution benchmarks, improving it by 14.7, 8.0, and 13.0 points on V^* Bench, HR-Bench 4K, and HR-Bench 8K, respectively (11.9 points on average). It also exceeds the 4.6× larger Qwen2.5-VL-32B in average score (82.5 vs. 77.4). RL on our data mixture accounts for part of these gains (Section 5.3); adding tool use with the visual evidence reward yields further gains of 4.7, 5.1, and 4.8 points, respectively. On HR-Bench 8K, BiVER exceeds the strongest listed baseline by 4.2 points and achieves the highest scores on both FSP and FCP. These results are consistent with our motivation: for high-resolution inputs, evaluating the visual evidence returned by a tool call provides a direct signal about whether the resulting view supports the question and its answer.

Table 2: **Document, chart, and table results.** RF: rotated/flipped; *: reproduced results. Best in **bold**, second-best underlined.

Model	Document		Chart/Table		Avg.
	DocVQA-RF	InfoVQA-RF	CharXiv	Table VQA	
Qwen2.5-VL-7B	65.2	53.3	41.4	66.4	56.6
Mini-o3*	57.7	55.2	42.3	63.5	54.7
DeepEyes*	67.7	**59.7**	43.5	67.7	59.7
BiVER	**72.7**	58.1	**44.8**	**70.9**	**61.6**
Δ(Base)	*↑ 7.5*	*↑ 4.8*	*↑ 3.4*	*↑ 4.5*	*↑ 5.0*

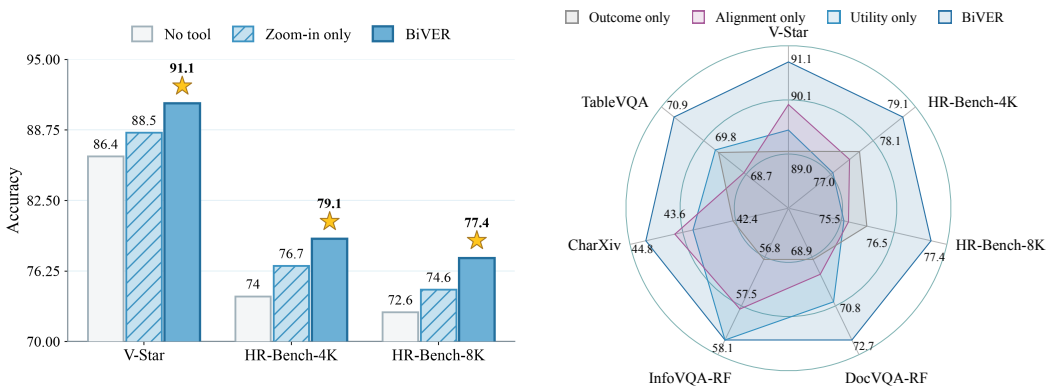


Figure 4: **Ablation studies.** Left: tool-set ablation on V^* Bench, HR-Bench 4K, and HR-Bench 8K. Right: reward ablation on seven benchmarks.

Evaluation on document, chart, and table benchmarks. As shown in Table 2, BiVER generalizes to tasks whose difficulty stems from orientation, dense text, and layout rather than spatial scale. On the rotated/flipped DocVQA-RF, BiVER improves the initialization by 7.5 points (from 65.2 to 72.7), with further gains on InfoVQA-RF (4.8), CharXiv (3.4), and TableVQA (4.5), yielding the highest average accuracy of 61.6 among compared methods. The gains on the RF benchmarks are relevant to our multi-tool setting because these inputs can require changes in orientation as well as closer inspection of visual details. Unlike a reward tied to a particular tool action, our visual evidence reward evaluates the returned image after the tool executes. It therefore applies to crops, orientation changes, and annotations through the same scoring formulation. We further evaluate our method on Qwen3-VL-4B and Qwen3-VL-8B backbones, with results provided in Appendix D.

Tool-call behavior analysis. We examine tool-call frequency during training and tool selection across benchmark families at inference time. Figure 3 shows that BiVER maintains frequent tool interaction over training. Its tool-call trajectory differs from that of DeepEyes, which provides only a zoom-in tool; because the available action spaces also differ, this comparison describes policy behavior rather than isolating the effect of the reward on call frequency. Table 3 shows how the policy uses the available tools across tasks. Zoom-in accounts for most calls on high-resolution benchmarks, whereas rotation and flipping together account for 46.0% of calls on document benchmarks. This variation is compatible with the central design of BiVER: the reward assesses the visual state produced by an executed tool call, rather than assigning a separate score to each tool type or its arguments. The same evidence-quality criterion can thus guide the policy when different tasks call for different visual operations, without requiring tool-specific reward rules.

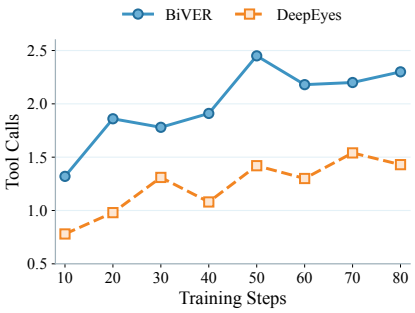


Figure 3: **Tool-call frequency.**

5.3 Ablation Studies

Tool set. We compare three configurations: **No tool**, which uses outcome-only reward without tool access; **Zoom-in only**, which uses the crop tool and the visual evidence reward; and **BiVER**, which uses all six tools with the same visual evidence reward. Figure 4 (left) and Table 4 show consistent improvements across these configurations. Relative to no-tool RL, the six-tool model gains 6.5 points on DocVQA-RF and 4.4 points on TableVQA. Because the no-tool configuration also changes the reward, this comparison measures their combined effect. The comparison between zoom-in only and six tools holds the reward design fixed and shows additional gains of 3.3 and 2.1 points on the two benchmarks, respectively. The fixed six-tool comparison in Figure 4 (right) isolates the contribution of the reward more directly: on DocVQA-RF, outcome-only training gains 2.7 points over no-tool training, and adding the visual evidence reward gains another 3.8 points.

Table 3: **Tool-call distribution** across benchmarks at inference. Invalid: malformed calls.

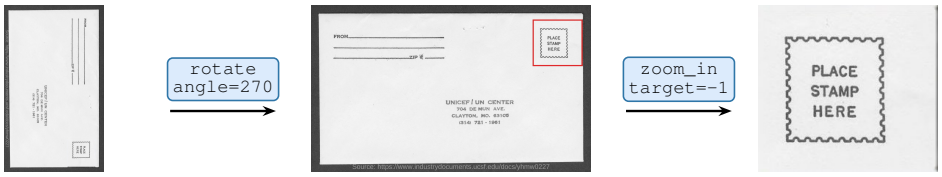
Benchmark	Zoom-in	Rotate	Flip	Line	Point	Invalid
High-Resolution	**89.9%**	0.1%	0.1%	5.0%	2.1%	2.9%
Document	**33.9%**	28.0%	18.0%	8.5%	10.6%	1.1%
Chart/Table	**80.7%**	0.9%	0.9%	10.3%	6.1%	1.2%

Table 4: **Tool-set ablation** on document and table benchmarks.

Tools	DocVQA-RF	TableVQA
No tool	66.2	66.5
Zoom-in only	69.4	68.8
Six tools (BiVER)	**72.7**	**70.9**

Case study: rotate, then zoom into the rotated view (DocVQA-RF)

Q: What is written within the square? **Answer:** PLACE STAMP HERE ✓



(a) Input page

(b) Output of call 1

(c) Output of call 2

"Text runs vertically; rotate the page upright first."

"The square is the stamp box at the top right; too small to read, so zoom in."

"The square reads 'PLACE STAMP HERE'."

Figure 5: **Composed tool use by BiVER.** Quotes are abridged; the red box marks the crop of call 2.

These results show the value of scoring what a tool call produces when the policy has multiple visual operations available, using a shared criterion across tool types.

Reward. We fix the six-tool action space and the outcome reward, and vary the visual evidence reward (Figure 4, right). We compare four configurations: **Outcome only** uses no visual evidence reward; **Alignment only** adds the backward question alignment score S_Q, which assesses whether the visual tool output contains evidence relevant to the question; **Utility only** adds the forward answer utility score S_A, which assesses whether the visual tool output supports the reference answer; and **BiVER** combines S_Q and S_A. Averaged over the seven benchmarks, the four configurations reach 68.6, 69.0, 69.2, and 70.6, respectively. The two directions are complementary: alignment helps most where the challenge is locating a small relevant region (V^* Bench +1.1, CharXiv +1.6), and utility helps most where the evidence must contain the answer text (DocVQA-RF +2.0, InfoVQA-RF +1.3). On HR-Bench 4K and 8K, neither direction alone beats outcome-only training, yet their combination gains 1.3 and 1.4 points. BiVER is best or tied-best on all seven benchmarks, confirming that question alignment and answer utility capture distinct aspects of evidence quality.

6 Case Study

Figure 5 shows the policy composing two tools on a rotated DocVQA-RF page. Neither suffices alone: rotation leaves the stamp box too small to read, and a crop of the original page returns sideways text. The upright page exposes the queried region and the crop contains the answer, matching the two directions of BiVER and illustrating effective tool composition (more cases in Appendix F).

7 Conclusion

We present BiVER, which scores every visual tool output for forward answer utility and backward question alignment and folds the scores into GRPO as trajectory-level shaping. It provides a unified training signal for heterogeneous image-producing tools without supervised trajectories. Experiments on Qwen2.5-VL-7B show that BiVER consistently improves performance on high-resolution, document, chart, and table benchmarks. Limitations are discussed in Appendix G.

AI USE STATEMENT

We used generative AI tools to assist with portions of code construction and with drafting, revising, and polishing the manuscript. These tools were used as assistants rather than as end-to-end authors or experimenters. We did not use generative AI to autonomously design or execute the research, nor did we use it to generate human-subject data or replace expert judgment; human-subject data collection is not applicable to this work. We have reviewed all AI-assisted work. AI-assisted code was checked and tested by the authors before use, and AI-drafted or edited text was revised by the authors, who verified the scientific claims and citations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

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

Miguel Carvalho, Hélder Dias, and Bruno Martins. Cropvlm: learning to zoom for fine-grained vision-language perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11170–11179, 2026.

Chongyan Chen, Samreen Anjum, and Danna Gurari. Grounding answers for visual questions asked by visually impaired people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19098–19107, 2022.

Yukun Chen, Jiaming Li, Longze Chen, Ze Gong, Jingpeng Li, Zhen Qin, Hengyu Chang, Ancheng Xu, Zhihao Yang, Hamid Alinejad-Rokny, et al. Rucl: Stratified rubric-based curriculum learning for multimodal large language model reasoning. *arXiv preprint arXiv:2602.21628*, 2026.

Qihua Dong, Gozde Sahin, Pei Wang, Zhaowei Cai, Robik Shrestha, Hao Yang, and Davide Modolo. Visual reasoning through tool-supervised reinforcement learning. *arXiv preprint arXiv:2604.19945*, 2026.

Tianjun Gu, Tianyu Xin, Kuan Zhang, Bowen Yang, Kok-Chung Chua, Peize Li, Xinran Zhang, Yupeng Chen, Qiyue Zhao, Qinlei Xie, Jianhang Liu, Yucheng Lu, Yinan Han, Marco Pavone, and Yiming Li. Towards spatial supersensing in the wild. In *The Nineteenth European Conference on Computer Vision*, 2026.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14953–14962. IEEE, 2023.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Zhe Xu, Yao Hu, Shaohui Lin, et al. Vision-r1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pp. 63794–63812, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*, 2024.

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In *International Conference on Learning Representations*, volume 2026, pp. 76722–76746, 2026.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14369–14387, 2024.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2034–2044. IEEE, 2025.

Yan Ma, Weiyu Zhang, Tianle Li, Linge Du, Xuyang Shen, and Pengfei Liu. What does vision tool-use reinforcement learning really learn? disentangling tool-induced and intrinsic effects for crop-and-zoom. *arXiv preprint arXiv:2602.01334*, 2026.

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

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. *Advances in Neural Information Processing Systems*, 38:20538–20559, 2026.

OpenAI. OpenAI o3 and o4-mini system card. `https://openai.com/index/o3-o4-mini-system-card/`, 2025.

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.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6613–6629, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

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.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, et al. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11854–11864. IEEE, 2023.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*, 2025a.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025b.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *Advances in Neural Information Processing Systems*, 38: 118818–118850, 2026.

Ye Wang, Qianglong Chen, Zejun Li, Siyuan Wang, Shijie Guo, Zhirui Zhang, and Zhongyu Wei. Simple o3: Towards interleaved vision-language reasoning. *arXiv preprint arXiv:2508.12109*, 2025c.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024.

Hang Wu, Yujun Cai, Zehao Li, Haonan Ge, Bowen Sun, Junsong Yuan, and Yiwei Wang. Camreasoner: Reinforcing camera movement understanding via structured spatial reasoning. *arXiv preprint arXiv:2602.00181*, 2026.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13084–13094. IEEE, 2024.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv preprint arXiv:2505.15436*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, and Chao Shen. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 126775–126798, 2026.

Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. Image-of-thought prompting for visual reasoning refinement in multimodal large language models. *arXiv preprint arXiv:2405.13872*, 2024.

A Analytical Properties of the Process Reward

This section records the algebraic properties underlying the aggregation and normalization choices in Section 4. These results characterize the reward transformation after a rollout group has been sampled; they do not constitute a guarantee of policy improvement or convergence.

Mean aggregation does not reward call count by itself. Fix one trajectory and one scoring direction $D \in \{Q, A\}$. For $T \geq 1$, write its aggregate as

$$S_{D,T} = \frac{1}{T} \sum_{t=1}^{T} s_D(V_t). \tag{6}$$

After appending a new unique output V_{T+1}, direct substitution gives

$$\begin{aligned} S_{D,T+1} - S_{D,T} &= \frac{T S_{D,T} + s_D(V_{T+1})}{T+1} - S_{D,T} \\ &= \frac{s_D(V_{T+1}) - S_{D,T}}{T+1}. \end{aligned} \tag{7}$$

Consequently, an additional output increases the trajectory score if and only if its score exceeds the current mean; a below-mean output decreases it. Moreover, an exact pixel duplicate is removed before aggregation and therefore leaves the score unchanged. Thus, unlike a sum, the mean supplies no unconditional incentive to increase the number of tool calls.

Within-group standardization is invariant to affine score scales. Fix a prompt group g with a nonzero empirical standard deviation, and consider a positive affine transformation $x'_i = ax_i + b$ with $a > 0$. Its group statistics satisfy

$$\mu_g(x') = a\mu_g(x) + b, \qquad \sigma_g(x') = a\sigma_g(x). \tag{8}$$

It follows that

$$z_g(x'_i) = \frac{ax_i + b - a\mu_g(x) - b}{a\sigma_g(x)} = z_g(x_i). \tag{9}$$

Hence the normalized ordering is unaffected by the offset or units of the raw score. This property is important here because S_Q is a binary log-odds whereas S_A is a length-normalized sequence log-likelihood. Standardizing the two directions separately prevents their arbitrary numerical scales from determining their relative weights.

The final normalization removes correlation-dependent scale. Let

$$U_i = z_g(S_{Q,i}), \qquad V_i = z_g(S_{A,i}) \tag{10}$$

for valid trajectories in group g. By construction, both variables have empirical mean zero and variance one. Denote their empirical correlation by $\rho_g = \mathrm{Cov}_g(U, V)$. For $J_i = (U_i + V_i)/2$, we obtain

$$\mu_g(J) = \frac{1}{2}\big(\mu_g(U) + \mu_g(V)\big) = 0, \tag{11}$$

$$\begin{aligned}\mathrm{Var}_g(J) &= \frac{1}{4}\big(\mathrm{Var}_g(U) + \mathrm{Var}_g(V) + 2\ \mathrm{Cov}_g(U, V)\big) \\ &= \frac{1}{4}\big(1 + 1 + 2\rho_g\big) = \frac{1 + \rho_g}{2}.\end{aligned} \tag{12}$$

Although each direction has unit variance, their average therefore does not: its scale changes with the correlation between the two signals. Applying $Z_i = z_g(J_i)$ restores zero mean and unit variance whenever $\mathrm{Var}_g(J) > 0$, so a fixed λ has a consistent within-group scale. In the degenerate case $\mathrm{Var}_g(J) = 0$, the method sets $Z_i = 0$ as specified in the main text.

B Baselines and Benchmarks

Baselines. We organize the baselines into three groups.

General-purpose VLMs. GPT-4o (Hurst et al., 2024) and o3 (OpenAI, 2025) are proprietary multimodal models that serve as upper-bound references. Qwen2.5-VL (Bai et al., 2025b) is the open-source backbone from which our policy and several baselines are initialized; we report both the 7B and 32B variants to isolate the effect of scale.

Visual search methods. ZoomEye (Shen et al., 2025) performs tree-based zoom exploration guided by an LLM without any additional training, providing a training-free reference for visual search.

Thinking-with-images methods. Pixel-Reasoner (Su et al., 2026) augments RL with a curiosity bonus to encourage pixel-space operations. Mini-o3 (Lai et al., 2026) scales interaction turns and diversifies reasoning patterns for deep visual search. DeepEyes (Zheng et al., 2026) trains a zoom-in policy via outcome-based RL on the same base model as ours, making it the most direct single-tool comparison. ToolsRL (Dong et al., 2026) extends DeepEyes with curriculum learning and multiple tools.

Benchmarks. We evaluate on seven benchmarks spanning three complementary task families.

High-resolution fine-grained visual reasoning. V^* Bench (Wu & Xie, 2024) tests attribute recognition and spatial reasoning over high-resolution natural images. We report Attribute (Attr), Spatial,

and Overall accuracy. HR-Bench (Wang et al., 2025b) provides 4K and 8K image sets and evaluates fine-grained single-instance perception (FSP) and cross-instance perception (FCP). This family is designed to stress-test whether a model can locate and resolve small visual details.

Document understanding. DocVQA-RF and InfoVQA-RF are rotation/flip variants of DocVQA (Mathew et al., 2021) and InfographicVQA (Mathew et al., 2022), respectively. In each variant, 70% of pages are randomly rotated or flipped so that text cannot be read without first applying a geometric correction, testing layout understanding, text recognition, and visual–textual reasoning under orientation perturbation. DocVQA-RF is built from the DocVQA validation split and is disjoint from our DocVQA training subset (Appendix C).

Chart and table understanding. CharXiv (Wang et al., 2024) contains scientific chart questions that require reading axes, legends, and trend lines. TableVQA-Bench (Kim et al., 2024) evaluates comprehension of rendered tables, including cell localization and arithmetic over table entries. Together, these two benchmarks assess structured-content reasoning that is complementary to the spatial and document tasks above.

C Training Configuration

Data mixture. The training set consists of four parts. The first three are the training corpus released by DeepEyes (Zheng et al., 2026). (i) *Visual search* (22,362 samples): fine-grained perception questions built on the V^* data (Wu & Xie, 2024) derived from COCO2017, where answering requires locating small objects and subtle visual cues in natural images. (ii) *ArxivQA* (13,659 samples) (Li et al., 2024): scientific plots, diagrams, and schematic charts that introduce structured, symbolic visual content beyond natural scenes. (iii) *ThinkLite-VL* (11,031 samples) (Wang et al., 2026): multimodal reasoning questions (arithmetic, commonsense, problem solving) that broaden reasoning variety and reduce modality-specific overfitting. (iv) *DocVQA subset* (3,000 samples): questions randomly sampled from the DocVQA (Mathew et al., 2021) training split, 70% of whose pages are rotated or flipped. This subset does not overlap with the DocVQA-RF evaluation set.

The resulting mixture has 50,052 prompts. All image tool rows expose the same six tools, and the tool set is never restricted per data source.

Environment. Training is implemented with the `verl` framework (Sheng et al., 2024) on 32 NVIDIA H200 GPUs (4 nodes $\times$ 8 GPUs). Every visual row receives the system prompt in Appendix E; the six image-returning tools in Table 7 are advertised through the standard function-calling template. Each assistant turn may contain at most one tool call, a trajectory may contain at most five tool calls, and generation terminates as soon as an `<answer>` block is produced. Successful calls return exactly one image inside a `<tool_response>` block with no accompanying text. Failed calls (bad arguments, out-of-range coordinates, crops smaller than 28×28 pixels) return a short text-only failure message and the trajectory continues, so the policy can correct its call in the next turn.

Rewards. The base reward follows DeepEyes (Zheng et al., 2026). Answer correctness is judged by Qwen2.5-72B-Instruct with the prompt in Appendix E.3. The process term of Equation 5 is computed by a frozen policy model. Table 5 lists the remaining hyperparameters.

Table 5: Training hyperparameters of the multi-tool policy. Unless stated otherwise, the single-tool baselines in Section 5.3 share every value and differ only in the advertised tool set.

Hyperparameter	Value
Policy initialization	Qwen2.5-VL-7B-Instruct
Process evaluator	Frozen Qwen2.5-VL-7B-Instruct
PPO mini-batch	256 prompts
Learning rate	1×10^{-6}
Rollout sampling	temperature 1.0, top-p 1.0
Max prompt / response length	8,192 / 20,480 tokens
GRPO group size G	16
Maximum tool calls $T_{\max}$	5
Tools	6 image-returning tools (Table 7)
Outcome reward weights	$0.8\,R^{\text{acc}} + 0.2\,R^{\text{format}} + 1.2\,R^{\text{tool}}$
Process reward	$\lambda = 0.3$
Hardware	4×8 NVIDIA H200 GPUs

D Backbone Generalization

To examine how BiVER performs across backbone generations and model scales, we additionally train Qwen3-VL-4B and Qwen3-VL-8B (Bai et al., 2025a). We keep the training data, tool set, evaluator construction (a frozen copy of the respective initial checkpoint), and hyperparameters unchanged, altering only the base model.

As shown in Table 6, both backbones improve across all seven benchmarks. The 4B and 8B models improve by 11.5 and 13.5 points, respectively, on HR-Bench 8K. Notably, the improvements do not diminish with scale: the 8B model gains are overall comparable to those of the 4B model and larger on HR-Bench 4K, HR-Bench 8K, and CharXiv, indicating that the process reward supplies information that a stronger base model does not already possess.

Table 6: **Backbone generalization across Qwen3-VL model sizes.** Each trained policy is compared with its own base model.

			High-Resolution			Document		Chart/Table	
Backbone	Param Size	Variant	V^*	HR-4K	HR-8K	DocVQA-RF	InfoVQA-RF	CharXiv	TableVQA
Qwen3-VL	4B	Base	81.7	73.8	67.0	68.7	53.2	39.6	63.8
		BiVER	92.7	80.0	78.5	74.8	58.7	43.6	71.5
		Δ	*↑11.0*	*↑6.2*	*↑11.5*	*↑6.1*	*↑5.5*	*↑4.0*	*↑7.7*
	8B	Base	84.3	74.8	68.1	76.4	62.5	46.4	67.0
		BiVER	93.2	81.4	81.6	81.6	66.9	51.6	72.2
		Δ	*↑8.9*	*↑6.6*	*↑13.5*	*↑5.2*	*↑4.4*	*↑5.2*	*↑5.2*

E Prompts

E.1 Policy Prompts

System prompt. Every visual training example and every evaluation query uses the following system prompt. The chat template then appends the JSON schemas of the advertised tools in a `<tools>` block and the standard instruction to emit calls inside `<tool_call>` tags.

```
You are a helpful assistant. You can call functions to assist with the
    ↪ user query. Important: You must call only one function at a
```

```
    ↪ time. After each function call, wait for the execution result
    ↪ before making the next function call if needed.
```

User prompt.

```
{question}
Think first, use any provided visual tool when it helps, then answer.
    ↪ Format strictly as: <think>...</think> <tool_call>...</tool_call
    ↪ > (if tools are needed) <answer>...</answer>
```

Tool schemas. Table 7 lists the six tools exactly as advertised to the policy. All tools share the optional `target_image` argument (`0`: original image; positive n: the n-th tool output; -1: the latest image, default) and return a single image.

Table 7: Function schemas of the six image-returning tools. Required arguments are marked with *.

Tool	Description	Arguments
`image_zoom_in_tool`	Crop a rectangular region from a selected image. Coordinates are pixels in that exact selected image.	`bbox_2d`* (array): pixel-space box `[x1, y1, x2, y2]`
`image_rotate_tool`	Rotate a selected image clockwise without interpolation. Pilot angles are 90, 180, or 270 degrees.	`angle`* (number): clockwise angle: 90, 180, or 270
`image_flip_tool`	Flip a selected image horizontally or vertically.	`direction`* (string): `horizontal` or `vertical`
`image_draw_horizontal_line_tool`	Draw a horizontal guide line across a selected image.	`height_location`* (number): Y coordinate in the selected image; `color` (string, default red); `thickness` (number, default 3); `style` (`solid` or `dashed`, default solid)
`image_draw_vertical_line_tool`	Draw a vertical guide line across a selected image.	`width_location`* (number): X coordinate in the selected image; `color`, `thickness`, `style` as above
`image_mark_points_tool`	Mark one or more pixel coordinates on a selected image.	`points`* (array): list of pixel-space `[x, y]` points; `color` (default red); `size` (marker radius, default 8); `thickness` (default 3); `shape` (`circle` or `cross`, default circle)

Tool responses. A successful call is returned to the policy as a user turn containing only the output image:

```
<tool_response>
[returned image]
</tool_response>
```

A failed call returns a text-only `<tool_response>` that reports the validation error (for example, a non-integer `width_location` or a box smaller than 28×28 pixels).

E.2 FROZEN-EVALUATOR PROMPTS

Both process scores present the evaluator with a single returned image followed by one of the prompts below. The original image, tool call, dialogue history, and policy reasoning are never included.

Backward question alignment (s_Q). The score is the log-odds of the single tokens `Yes` and `No`.

```
[returned image]
The image above is the actual image returned by a visual tool.

Question: {question}

Does this tool-output image itself contain visual evidence that is
    ↪ useful for answering the question? Judge only the tool output
    ↪ shown above. Answer with only Yes or No.
```

Forward answer utility (s_A). The score is the length-normalized log-probability of the ground-truth answer as a continuation of the prompt. The conditioning prompt ends at `Final answer:`; the answer in braces is the teacher-forced target and is not part of the input.

```
[returned image]
Based on the image(s) above, answer the following question.
Question: {question}
Final answer: {ground-truth answer}
```

E.3 OUTCOME JUDGE PROMPT

Answer correctness during training (and for open-ended benchmarks at test time) is decided by Qwen2.5-72B-Instruct with the following prompt. The extracted content of the final `<answer>` block is inserted as the model's answer.

```
# System
You are an expert evaluator. Your task is to determine if a model's
    ↪ answer is semantically equivalent to a provided standard answer,
    ↪  given a specific question.
Your evaluation must be strict. The model's answer is only correct if
    ↪ it fully matches the meaning of the standard answer.
You must provide your final judgement as a single word: either "
    ↪ CORRECT" or "INCORRECT". Do not provide any explanation or other
    ↪  text.

# User
I will provide a question, a standard answer, and a model's answer.
    ↪ You must evaluate if the model's answer is correct.

---
**Example 1:**
[Question]: Is the countertop tan or blue?
[Standard Answer]: The countertop is tan.
[Model's Answer]: tan
[Your Judgement]: CORRECT
---
**Example 2:**
[Question]: Is the man phone both blue and closed?
[Standard Answer]: Yes, the man phone is both blue and closed.
[Model's Answer]: No.
[Your Judgement]: INCORRECT
---
**Task:**
[Question]: {question}
```

```
[Standard Answer]: {ground truth}
[Model's Answer]: {model answer}
[Your Judgement]:
```

F CASE STUDIES

We show three trajectories of our multi-tool policy on the evaluation benchmarks of Section 5. All returned images are produced by executing the listed tool calls with the training tool implementation; red boxes on input images are added for illustration only and were not shown to the model.

Case 1: attribute recognition with a crop (V* Bench)



(a) Input image (2250 × 1500); red box = requested crop.



(b) Returned image (340 × 528).

The queried flag occupies a tiny fraction of the image; the policy zooms into it once and answers from the returned crop.

```
[User]  Question: Is the flag red or white?
        Options:
        A. The color of the flag is white.
        B. The color of the flag is red.
        Think first, use any provided visual tool when it helps, then
    ↪ answer. Format strictly as: ...
[Assistant] <think>I need to zoom in on the flag to determine its
    ↪ color. </think>
        <tool_call>
        {"name": "image_zoom_in_tool", "arguments": {"bbox_2d":
    ↪ [1550,650,1890,1178]}}
        </tool_call>
[Tool]  <tool_response> [image, panel (b)] </tool_response>
[Assistant] <think>Now I can see the flag clearly. The color is white.
    ↪  </think>
        <answer><think>The color of the flag is white.</think> The
    ↪ color of the flag is white. </answer>
[Ground truth]  A. The color of the flag is white.   (correct)
```

Case 2: relative position with point marking (V* Bench)



(a) Returned image (2250 × 1500) with the two requested markers; red box = region enlarged in (b).

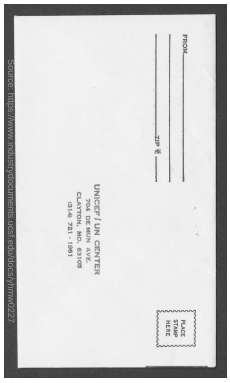


(b) Enlarged view: markers on the cyclist (left) and the woman's handbag (right).

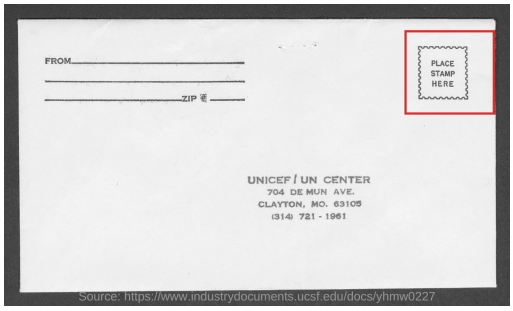
Instead of cropping, the policy marks both referents on the full image and compares their marked positions. Markers are redrawn with a larger stroke for legibility; the policy received the default radius-8 markers.

```
[User]  Question: Is the cyclist on the left or right side of the
    ↪ woman's handbag?
        Options:
        A. The cyclist is on the left side of the woman's handbag.
        B. The cyclist is on the right side of the woman's handbag.
        Think first, use any provided visual tool when it helps, then
    ↪ answer. Format strictly as: ...
[Assistant] <think>I need to use the image_mark_points_tool to mark
    ↪ the cyclist and the woman's handbag to determine their positions
    ↪ .</think>
        <tool_call>
        {"name": "image_mark_points_tool", "arguments": {"points":
    ↪ [[1046,985], [1968,1004]]}}
        </tool_call>
[Tool]  <tool_response> [image, panel (b)] </tool_response>
[Assistant] <think>Now I can see that the cyclist is on the left side
    ↪ of the woman's handbag. So, the answer is A. The cyclist is on
    ↪ the left side of the woman's handbag.</think>
        <answer>The cyclist is on the left side of the woman's handbag
    ↪ . So, the answer is A.</answer>
[Ground truth]  A. The cyclist is on the left side of the woman's
    ↪ handbag.   (correct)
```

Case 3: rotation followed by a crop on a rotated page (DocVQA-RF)



(a) Input page (819 × 1376), rotated by 90°.



(b) Output of call 1 (270° clockwise rotation, 1376 × 819); red box = crop requested by call 2.



(c) Output of call 2: crop of (b), 245 × 228, about 5% of the page.

The queried text sits in a small stamp box on an envelope that is rotated by 90°. The policy first restores the page orientation with a rotation, then crops the stamp box from the rotated output (`target_image=-1`). The returned crop is upright and legible, and the policy answers with the exact span.

```
[User]  What is written within the square?
        Answer with the exact minimal text span from the document, no
    ↪ explanation. Think first, use any provided visual tool when it
    ↪ helps, then answer. Format strictly as: ...
[Assistant] <think>The page is rotated: the text runs vertically and
    ↪ reads top-to-bottom, so the page has been turned 90 degrees
    ↪ clockwise. I will rotate it 270 degrees clockwise to bring it
    ↪ upright before looking for the square.</think>
        <tool_call>
        {"name": "image_rotate_tool", "arguments": {"angle": 270}}
        </tool_call>
[Tool]  <tool_response> [image, panel (b)] </tool_response>
[Assistant] <think>Now the envelope is upright. The square is the
    ↪ small dashed stamp box in the top-right corner of the rotated
    ↪ image; its text is too small to read reliably, so I will zoom in
    ↪  on it.</think>
        <tool_call>
        {"name": "image_zoom_in_tool", "arguments": {"bbox_2d":
    ↪ [1089,71,1334,299], "target_image": -1}}
        </tool_call>
[Tool]  <tool_response> [image, panel (c)] </tool_response>
[Assistant] <think>Now I can see the text clearly. The square reads '
    ↪ PLACE STAMP HERE'.</think>
        <answer>The text within the square is 'PLACE STAMP HERE'.</
    ↪ answer>
[Ground truth]  PLACE STAMP HERE   (correct)
```

G LIMITATIONS

Our current formulation aggregates per-output scores into a trajectory-level advantage, following the standard GRPO setup in which all tokens in a trajectory share one advantage. A more fine-grained extension could estimate the evidence gain of each output over its input image and assign it locally to the corresponding tool-call tokens. We use a frozen copy of the initial policy as the evaluator, so that the reward requires no external model and the gains cannot be attributed to distillation from a stronger teacher. While this evaluator reliably separates tool outputs of clearly different quality, its discrimination is bounded by the capacity of the initial model; a larger evaluator is orthogonal to our formulation and may further sharpen the signal. Finally, the forward score uses a reference answer,

so the full bidirectional reward is designed for training data with verifiable answers; the backward score itself remains answer-free.