Learning Where to Look: Observation Policy Optimization for Thinking with Images

Junfeng Wang[1,*] **Jiawei Liu**[1,*] **Yongchao Xu**[1,*] **Tao Jiang**[2]
Jiangbo Ai[2] **Jin Zhang**[2] **Chong Wang**[1,†] **Qixing Zhang**[1]

[1]University of Science and Technology of China
[2]Alibaba Group

{jfwang2002,yongchaoxu}@mail.ustc.edu.cn
{jwliu6,wchong,qixing}@ustc.edu.cn
{cigarette.jt,aijiangbo.ajb,zj146372}@alibaba-inc.com

[*]Equal contribution. [†]Corresponding author.

Abstract

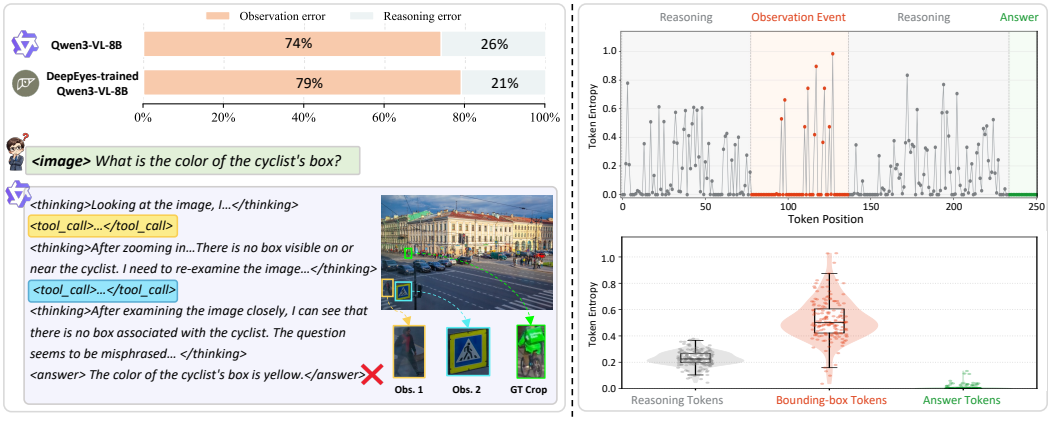
Complex visual question answering often requires vision-language models to *think with images*, i.e., actively acquire local visual evidence by zooming into informative regions of high-resolution images. Our error analysis shows that many failures in this setting originate from incorrect observation actions, where the model fails to localize key regions or obtain crops that contain sufficient evidence for answering. We further observe that bounding-box tokens exhibit substantially higher generation entropy than ordinary language tokens, indicating high uncertainty when the model decides where to look. However, existing reinforcement learning methods based on final-answer rewards provide only trajectory-level feedback, making it difficult to assign precise credit to the observation actions. We propose Observation Policy Optimization (OPO), an RL algorithm for optimizing observation actions in tool-augmented thinking with images. OPO consists of two core modules. First, Uncertainty-Guided Observation Branching (UOB) treats each image zoom-in tool call as an observation event and selectively branches at high-uncertainty, low-redundancy events during rollout, enabling efficient exploration of alternative cropping decisions and their resulting visual evidence. Second, Evidence-Localized Advantage Attribution (ELAA) compares sibling branches using local evidence-sufficiency rewards and assigns credit only to the corresponding observation-action tokens. By decoupling observation-action optimization from trajectory-level outcome supervision, OPO improves visual evidence acquisition and further enhances visual reasoning performance. Experiments show that OPO consistently improves Qwen3-VL models across scales and achieves competitive performance against larger models and tool-augmented baselines. Code is publicly available.[1]

1 Introduction

Vision-language models (VLMs) have made substantial progress in visual question answering, chart understanding, document parsing, and complex scene reasoning [1–5]. However, many high-resolution visual tasks cannot be solved from a single global image view. The evidence required to answer a question may appear only in small text, subtle objects, spatial relations, or local image regions. In such cases, models must actively acquire local visual evidence by selecting informative regions and zooming in, an ability often referred to as *thinking with images* [6–10].

[1]https://github.com/Yann0122/Observation-Policy-Optimization

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Failure analysis on V*

(b) Token Entropy Analysis

Figure 1: Motivation for observation-action optimization. (a) Error-type breakdown of incorrect V* trajectories for Qwen3-VL-8B and DeepEyes-trained Qwen3-VL-8B, together with an example observation-error trajectory. (b) Token-entropy analysis, where the top panel shows an event-aligned entropy trace over a generated response and the bottom panel compares entropy distributions of reasoning tokens, bounding-box tokens, and answer tokens.

In this paper, we study tool-augmented thinking with images, where a VLM invokes an image zoom-in tool during multi-turn reasoning. In this setting, the key challenge is not only to reason over visual evidence, but also to acquire the right evidence in the first place. As shown in Figure 1(a), our error analysis shows that many failures stem from incorrect observation actions: the model either localizes the wrong region or obtains crops that lack sufficient evidence. We further find that bounding-box tokens exhibit substantially higher generation entropy than ordinary language tokens, as shown in Figure 1(b), suggesting that visual localization is a high-uncertainty local decision.

Existing RL-based methods for visual tool use typically optimize tool-augmented reasoning with trajectory-level rewards, such as final-answer correctness, format validity, or tool-use rewards [7–9, 11]. While such rewards can encourage models to use visual tools, they provide imprecise credit assignment for observation actions. A crop that contains sufficient evidence may be penalized if later reasoning fails, while an insufficient crop may be reinforced if the model guesses the answer correctly. This credit mismatch makes it difficult for the model to reliably learn where to look.

To address this problem, we propose Observation Policy Optimization (OPO), a reinforcement learning algorithm that directly optimizes observation actions in tool-augmented thinking with images. OPO consists of two complementary modules. First, Uncertainty-Guided Observation Branching (UOB) treats each zoom-in tool call as an observation event and selectively branches from high-uncertainty, low-redundancy events, allowing the model to explore alternative crops from a shared reasoning prefix. Second, Evidence-Localized Advantage Attribution (ELAA) compares sibling branches using local evidence-sufficiency rewards and assigns the resulting advantage only to the observation-action tokens that produced the corresponding crop. Together, UOB and ELAA separate observation-action optimization from trajectory-level outcome supervision, enabling more precise learning of where to look.

We evaluate OPO using Qwen3-VL backbones [2] on V* [12], HR-Bench 4K, and HR-Bench 8K [13]. OPO consistently improves both model scales, increasing overall accuracy by 10.0, 12.4, and 15.8 percentage points for Qwen3-VL-2B and by 4.7, 4.1, and 4.3 percentage points for Qwen3-VL-8B on the three benchmarks, respectively. OPO-8B also achieves competitive performance against strong open-source, proprietary, and tool-augmented baselines across these benchmarks. Ablation studies further show that uncertainty-guided observation branching and evidence-localized advantage attribution both contribute to these gains.

Our contributions are summarized as follows:

- We identify observation actions as a key bottleneck in thinking with images. Our analysis shows that many failures stem from insufficient visual evidence acquisition, and that observation-action tokens exhibit high local uncertainty.

- We propose OPO, which consists of two complementary modules: Uncertainty-Guided Observation Branching for targeted exploration of alternative cropping decisions, and Evidence-Localized Advantage Attribution for precise credit assignment.
- Experiments show that OPO consistently improves Qwen3-VL models across scales and achieves competitive performance against much larger models and tool-augmented methods.

2 Related Work

Thinking with Images. Thinking with images extends multimodal reasoning by allowing models to acquire and manipulate visual evidence during inference [10]. Within this paradigm, visual exploration methods use crops, zooming, and other image operations to inspect details that may be lost in the global view [14–16]. DeepEyes learns a zoom-based observation policy through reinforcement learning [7]. Broader tool interfaces support more diverse evidence acquisition: Thyme uses executable code for image manipulation and computation [17], while DeepEyesV2 and SenseNova-MARS combine visual operations with search [18, 19]. Alongside these interactive approaches, latent visual reasoning explores internal visual representations [20], and ZwZ distills region-level supervision into the model to enable single-pass perception without inference-time zooming [21].

A related question is how to supervise the visual actions taken during reasoning. CodeVision and CodeV address this through process rewards for code-based tool use. CodeVision rewards tool-use strategy and efficiency, including improvements in crop overlap with a target region [22]. CodeV evaluates the evidence provided by tool outputs and combines these scores with answer correctness to form a trajectory-level advantage [23]. OPO connects evidence supervision to the observation decision itself: it explores alternative crops from a shared reasoning prefix, compares their evidence sufficiency within the resulting sibling group, and assigns the local advantage to the corresponding observation-action tokens. This couples the exploration of candidate observations with credit assignment for the evidence they acquire.

Agentic Reinforcement Learning. Agentic reinforcement learning formulates large language models as learnable policies embedded in multi-turn interactive environments, enabling them to plan, reason, and invoke tools under partial observations [24]. More broadly, reinforcement learning has also found applications in other domains [25–27]. Early tool-augmented agents mainly relied on prompting or supervised fine-tuning to elicit tool-use behaviors, such as ReAct, Toolformer, FireAct, AgentTuning, and Agent-FLAN [28–32]. Recent studies further apply reinforcement learning to multi-turn tool interaction, using outcome or rule-based rewards to train models to autonomously invoke search engines, browsers, code interpreters, or visual tools for complex reasoning, information seeking, and multimodal tasks [33–40, 7, 41]. ARPO and AEPO further leverage token entropy to improve multi-turn tool-use exploration through branch sampling or entropy-balanced optimization at high-uncertainty tool-call steps [42, 43]. While these methods target general tool-use behavior, OPO specializes agentic RL to visual observation: it uses uncertainty-guided branching to explore alternative crops and evidence-localized attribution to assign feedback according to the visual evidence produced by the observation action.

3 Preliminary Analysis

3.1 Observation Events and Observation-Action Tokens

We follow the image zoom-in tool interface of Qwen3-VL [2], where the model generates tool-call tokens in a predefined format to trigger a zoom-in operation. Each tool call contains a region label, a bounding box, and an image index, which together determine the local visual evidence the model will observe next. We refer to each valid image zoom-in tool call as an *observation event*, and call its region-label, bounding-box and image-index tokens *observation-action tokens*. The concrete tool-call schema, prompt template, and zoom-in tool behavior are provided in Appendix E.

3.2 Failure Analysis on V*

We analyze the error trajectories of Qwen3-VL-8B-Instruct [2] and a Qwen3-VL-8B model trained under the DeepEyes [7] paradigm on the V* [12] dataset, aiming to understand the main sources of failure in tool-augmented visual reasoning. We focus on whether the model's image zoom-in operation acquires local visual evidence sufficient to answer the question. Specifically, we categorize a trajectory as an observation error if the model localizes an irrelevant region that does not contain the key evidence required for answering the question. If the cropped region contains sufficient evidence but the model still fails in subsequent reasoning or answer generation, we categorize it as a reasoning error. As shown in Figure 1(a, top), Qwen3-VL-8B-Instruct achieves an answer accuracy of 87.96% on V*. Among its incorrect answers, approximately 74% of failures are mainly caused by observation errors. For the Qwen3-VL-8B model trained under the DeepEyes paradigm, the answer accuracy improves to 90.05%, yet approximately 79% of its incorrect answers still mainly originate from observation errors. These results indicate that although outcome-level RL can improve final-answer accuracy, failures remain concentrated in the local visual evidence acquisition stage. Therefore, the training process needs to provide more direct optimization signals for the observation actions that determine where the model looks. Localization failures also dominate unsuccessful trajectories in stronger models (Appendix I).

3.3 Token Entropy Analysis

We further examine the token-level generation entropy of different parts of the model response. For decoding position t, let p_t denote the next-token distribution over the full vocabulary, conditioned on the input question x, the tool context $\mathcal{T}$, and the previously generated prefix $\mathcal{R}_{<t}$. We define the token entropy at this position as

$$H_t = -\sum_{j=1}^{V} p_{t,j} \log p_{t,j}, \quad p_t = \pi_\theta(\cdot \mid \mathcal{R}_{<t}, x; \mathcal{T}) = \mathrm{Softmax}\Big(\frac{z_t}{\tau}\Big). \tag{1}$$

Here V is the vocabulary size, $p_{t,j}$ is the probability of the j-th vocabulary token at position t, $z_t \in \mathbb{R}^V$ denotes the pre-softmax logits, and τ is the decoding temperature. This entropy quantifies the uncertainty of the model's next-token prediction at decoding position t. For analysis, we group generated tokens into reasoning tokens, observation-event tokens, and answer tokens. Observation-event tokens correspond to the structured image zoom-in tool call.

As shown in Figure 1(b, bottom), the bounding-box tokens used for localization in observation actions exhibit substantially higher entropy than ordinary reasoning tokens. We conduct this analysis over all samples in the V* [12] dataset. The bounding-box tokens have a mean entropy of 0.522 and a median entropy of 0.501, both more than twice the corresponding mean entropy of 0.229 and median entropy of 0.225 for reasoning tokens. In contrast, answer tokens are nearly deterministic, with a mean entropy of only 0.006. The event-aligned trace in Figure 1(b, top) further shows that entropy spikes concentrate on bounding-box tokens, which directly determine where the model looks. This observation aligns with entropy-guided agentic RL, where high-entropy tool-use steps are treated as promising branching points for adaptive exploration [42]. These results indicate that visual localization is a high-uncertainty observation decision, motivating uncertainty-guided branching around observation events.

4 Method

Observation Policy Optimization (OPO) is a reinforcement learning framework that improves where a model looks during tool-augmented thinking with images. Following the definition above, we denote each candidate observation event by e. OPO introduces two complementary modules for optimizing the model's observation actions: Uncertainty-Guided Observation Branching (UOB, Section 4.1) and Evidence-Localized Advantage Attribution (ELAA, Section 4.2). The full training procedure is summarized in Appendix A.1.

4.1 Uncertainty-Guided Observation Branching

UOB is a rollout-time module for targeted exploration around uncertain observation events. Instead of spending all M samples on independent full-trajectory rollouts, UOB first obtains a small set of

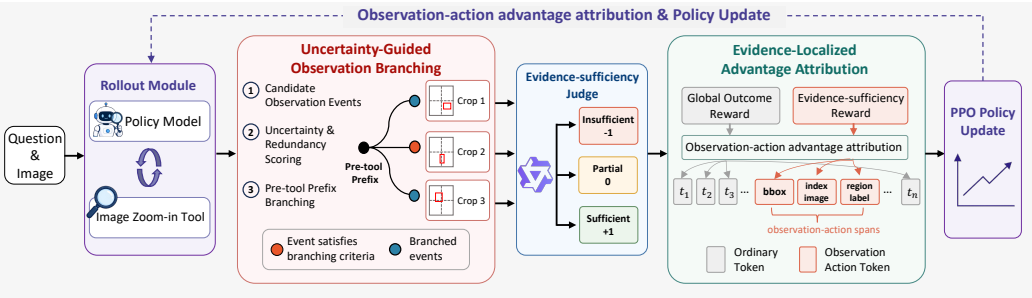


Figure 2: Overview of OPO. UOB concentrates rollout exploration around high-uncertainty and low-redundancy observation decisions by branching from the shared pre-tool prefix. ELAA then compares sibling crops within each branch group using evidence-sufficiency rewards and assigns the resulting local advantage only to the observation-action tokens.

base trajectories and then uses the remaining budget to branch at high-uncertainty observation events while filtering redundant child candidates. This preserves the same reasoning prefix while varying the observation action, making the resulting sibling trajectories directly comparable. As a result, UOB concentrates exploration on observation decisions where the model is most unsure, avoids spending rollout budget on redundant crops, and produces branch groups that provide controlled comparisons for ELAA.

Candidate observation events. For each question, OPO produces M trajectories under a fixed rollout budget. It first samples B base trajectories and reserves the remaining $P = M - B$ rollout slots for observation-event branching and backfill. Each base trajectory is decoded until it reaches the first valid observation event or terminates without invoking the zoom-in tool. For each collected event, we record the shared pre-tool prefix, the generated observation-action spans, and the original candidate crop. These event records provide the basis for UOB to branch from the shared pre-tool prefix and for ELAA to attribute local feedback to the corresponding observation-action tokens.

Event uncertainty. UOB scores each candidate event by the local token entropy around the observation action. We use H_t defined in Eq. (1) as the uncertainty at decoding position t. A larger H_t indicates a more diffuse next-token distribution and thus higher uncertainty.

Motivated by the entropy analysis above, UOB localizes event uncertainty to bounding-box generation. Let $S_{\text{box}}(e)$ denote the bounding-box token span of event e. Since the bounding box specifies the local region to be observed, its generation entropy provides a direct signal of uncertainty in the observation decision.

$$g(e) = \text{mean}_{t \in S_{\text{box}}(e)} H_t + \lambda_{\text{std}} \, \text{std}_{t \in S_{\text{box}}(e)} H_t. \tag{2}$$

The standard-deviation term makes the score sensitive to entropy spikes within the box span, which often indicate unstable coordinate or region decisions; λ_{std} controls the strength of this spike-sensitive signal. A larger $g(e)$ indicates that the event lies in a more uncertain localization decision region.

Question-adaptive anomaly and redundancy. Because raw uncertainty varies across questions and prompts, UOB normalizes event uncertainty against the question's own local baseline. Let $b(e)$ be the mean uncertainty in a fixed window of W tokens before the bounding-box span. Over the base candidates $\mathcal{E}_x$ of the same question, we compute

$$z(e) = \frac{g(e) - \text{mean}_{e' \in \mathcal{E}_x} b(e')}{\text{std}_{e' \in \mathcal{E}_x} b(e') + \epsilon}. \tag{3}$$

Here $\mathcal{E}_x$ is the set of base candidate events for question x, $b(e')$ is the pre-action baseline uncertainty of candidate event e', and ϵ is a numerical stability constant. The resulting $z(e)$ is a question-normalized anomaly score: it measures how much the bbox-based event uncertainty $g(e)$ deviates from the typical pre-action uncertainty for the same question. In this form, a large $z(e)$ means that the observation event is unusually uncertain relative to the model's local pre-action uncertainty on the same question.

To avoid repeatedly exploring similar regions within the same branch group, UOB computes a crop redundancy score for each child candidate c sampled from event e:

$$\mathrm{Red}(c \mid e) = \max_{c' \in \mathcal{C}_e} \mathrm{IoU}(b_c, b_{c'}), \tag{4}$$

where b_c is the bounding box of candidate c, and $\mathcal{C}_e$ contains the original candidate and previously accepted child crops on the same image within the branch group. UOB filters out redundant child candidates, allocating the limited rollout budget to diverse observation regions.

Branching from the observation prefix. The branching rule combines uncertainty-guided budget allocation with redundancy-based candidate filtering:

$$Z(e) = \mathcal{A}(z(e), P_{\mathrm{rem}}), \tag{5}$$

where $\mathcal{A}$ is the budget allocator, $Z(e)$ is the maximum number of child rollouts proposed for event e, and P_{rem} is the remaining branch budget. The allocator assigns more proposals to high-uncertainty events and always respects the remaining rollout budget and the number of available trajectory slots. Redundancy filtering determines which proposals are retained.

A child trajectory is not generated from scratch. Instead, OPO preserves the prompt and all tokens up to the pre-tool prefix boundary, then resamples the suffix from that prefix. The newly generated suffix may contain an alternative observation event. Valid child tool calls are accepted if their redundancy scores do not exceed a threshold τ_{red}. Accepted candidates are added to $\mathcal{C}_e$, their tool calls are executed, and the agent continues multi-turn generation until termination. If fewer than M trajectories are obtained after all branch opportunities, ordinary rollouts are used to backfill the group. The trajectories derived from the same pre-tool prefix form a branch group, which serves as the comparison set for ELAA.

4.2 Evidence-Localized Advantage Attribution

ELAA converts the sibling trajectories produced by UOB into local advantages for observation-action tokens. Trajectories in the same branch group share the same pre-tool prefix, but may generate different observation actions and crops. ELAA first evaluates each crop with an evidence-sufficiency reward, then normalizes these local scores within the branch group to obtain a group-local advantage. Finally, this local advantage is applied only to the observation-action token spans that produced the corresponding crop, while the remaining generated tokens keep the global outcome advantage.

Global outcome reward. The global outcome reward is computed from the final answer and task-level formatting and tool-use signals:

$$R_i^{\mathrm{global}} = w_{\mathrm{acc}} r_i^{\mathrm{acc}} + w_{\mathrm{fmt}} r_i^{\mathrm{fmt}} + w_{\mathrm{tool}} r_i^{\mathrm{tool}}. \tag{6}$$

Here R_i^{global} is the trajectory-level outcome reward for trajectory i. The terms r_i^{acc}, r_i^{fmt}, and r_i^{tool} respectively denote answer correctness, format quality, and tool-use quality [7]. The coefficients w_{acc}, w_{fmt}, and w_{tool} control the relative weights of these outcome-level signals.

Evidence-sufficiency reward. For each evidence crop, a frozen visual judge receives the question, the ground-truth answer, and the crop, and is prompted to assess whether the crop contains sufficient visual evidence to support the answer. The judge labels the crop as INSUFFICIENT, PARTIAL, or SUFFICIENT. The reward-judge prompts and model choices are detailed in Appendix F. The default scalar mapping is

$$\rho(\textsc{Insufficient}) = -1, \quad \rho(\textsc{Partial}) = 0, \quad \rho(\textsc{Sufficient}) = 1. \tag{7}$$

Here $\rho(\cdot)$ maps the judge's discrete evidence-sufficiency label to a scalar local reward: insufficient evidence is penalized, partial evidence is neutral, and sufficient evidence is rewarded.

For each sibling trajectory i in branch group g, let $c_{i,g}$ denote the crop produced by its observation action. The group-local evidence reward is

$$R_{i,g}^{\mathrm{evi}} = \rho(c_{i,g}). \tag{8}$$

$R_{i,g}^{\mathrm{evi}}$ measures the quality of visual evidence acquired by the local observation action. This reward is used only for local comparison among sibling trajectories in the same branch group.

Observation-action advantage attribution. After rollout and reward computation, standard GRPO first computes a global group advantage using the shared question identifier:

$$A_i^{\text{global}} = \frac{R_i^{\text{global}} - \text{mean}_{j \in \mathcal{G}(x)} R_j^{\text{global}}}{\text{std}_{j \in \mathcal{G}(x)} R_j^{\text{global}} + \epsilon}. \tag{9}$$

Here $\mathcal{G}(x)$ is the rollout group for question x. The advantage A_i^{global} indicates whether trajectory i achieves a better or worse outcome than its group peers.

Let $\mathcal{S}(g)$ be the set of sibling trajectories in branch group g. ELAA uses only the evidence-sufficiency reward to construct the group-local advantage:

$$A_{i,g}^{\text{local}} = \frac{R_{i,g}^{\text{evi}} - \text{mean}_{j \in \mathcal{S}(g)} R_{j,g}^{\text{evi}}}{\text{std}_{j \in \mathcal{S}(g)} R_{j,g}^{\text{evi}} + \epsilon}. \tag{10}$$

Because normalization is performed only within the same branch group, $A_{i,g}^{\text{local}}$ reflects relative evidence-quality differences among alternative observation actions sampled from the same prefix.

Let $m_{i,t}(g)$ be a binary mask that is one only for the image-index, region-label, and bounding-box spans generated by trajectory i in branch group g. ELAA applies the evidence-local advantage only to these observation-action tokens:

$$A_{i,t} = (1 - m_{i,t}(g))\, A_i^{\text{global}} + m_{i,t}(g) \lambda_{\text{local}} A_{i,g}^{\text{local}}. \tag{11}$$

Thus, tokens outside the observation-action spans are trained with the global outcome advantage, while observation-action tokens receive the evidence-local advantage computed within their corresponding branch group. This yields a clean credit assignment between downstream reasoning and visual evidence acquisition.

After obtaining the token-level advantage $A_{i,t}$, OPO updates the policy with the standard clipped PPO objective. OPO does not change the form of the PPO update; its effect enters through the advantage $A_{i,t}$ defined above. The full policy objective is provided in Appendix A.2.

5 Experiments

5.1 Experimental Setup

Baselines. To comprehensively assess the effectiveness of OPO, we compare it against three categories of baselines. First, we include advanced proprietary models, including GPT-5 mini and GPT-5.2 [44, 45], Claude Opus 4.6 [46], and Gemini-3 Pro [47], which represent strong closed-source general-purpose vision-language systems. Second, we compare with state-of-the-art open-source vision-language models, including LLaVA-OneVision [48], Qwen2.5-VL [1], Qwen3-VL [2], Qwen3.5 [49], and Kimi K2.5 [50]. Third, we consider methods explicitly designed with workflows or tool-use procedures, including SEAL [12], DyFo [51], ZoomEye [52], DeepEyes [7], and DeepEyesV2 [18]. These workflow-based systems are particularly relevant because they also aim to improve fine-grained perception by changing how visual evidence is acquired or processed.

Benchmarks. Since fine-grained high-resolution perception tasks naturally test a model's ability to observe visual details, we evaluate models on three popular benchmarks: V* [12], HR-Bench 4K, and HR-Bench 8K [13]. Details of these benchmarks are provided in Appendix B, and the evaluation protocol is described in Appendix C.

Training Details. We train Qwen3-VL-Instruct [2] with OPO for 200 iterations on 8 NVIDIA H20 GPUs. Each training batch contains 64 prompts, and each prompt uses a rollout budget of $M = 8$, consisting of $B = 2$ initial rollouts and $P = 6$ remaining trajectory slots for observation-event branching and backfill. For each sample, the model can generate up to five assistant turns and invoke the image zoom-in tool during reasoning. We follow the tool-call format of Qwen3-VL; the detailed prompt template is provided in Appendix E. For fair comparison, we use the same training data as DeepEyes [7], with data details provided in Appendix D. Additional rollout and optimization details are provided in Appendix H.

Table 1: Main results on high-resolution perception benchmarks. LLaVA-OV abbreviates LLaVA-OneVision. Tool denotes the inference-time mechanism: Zoom for image zoom-in crop tools, WF for manually designed visual-search workflows, Code for code-execution tools, General for code-execution, image search and text search tools, C+S for image cropping and search, $\times$ for no tool. Best and second-best results are bolded and underlined. † denotes results reproduced by us, and – indicates that the corresponding entry is not reported or not applicable.

Model	Backbone	Tool	Param.	V* Bench			HR-Bench 4K			HR-Bench 8K		
				Attr.	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	Overall
Advanced proprietary models												
GPT-5 mini [44]	–	–	–	–	–	78.6	–	–	78.6	–	–	74.4
GPT-5.2 [45]	–	–	–	–	–	75.9	–	–	–	–	–	–
Claude Opus 4.6 [46]	–	–	–	–	–	66.5	–	–	–	–	–	–
Gemini-3 Pro [47]	–	–	–	–	–	88.0	–	–	–	–	–	–
State-of-the-art open-source models												
LLaVA-OV [48]	–	$\times$	7B	75.7	75.0	75.4	72.0	54.0	63.0	67.3	52.3	59.8
Qwen2.5-VL [1]	–	$\times$	7B	73.9	67.1	71.2	85.2	52.2	68.8	78.8	51.8	65.3
Qwen2.5-VL [1]	–	$\times$	32B	87.8	88.1	87.9	89.8	58.0	73.9	84.5	56.3	70.4
Qwen3-VL [2]	–	Zoom	235B-A22B	–	–	<u>93.7</u>	–	–	<u>85.4</u>	–	–	**82.4**
Qwen3-VL [2]	–	Zoom	30B-A3B	–	–	89.5	–	–	82.5	–	–	79.3
Qwen3-VL [2]	–	Zoom	32B	–	–	91.1	–	–	84.6	–	–	81.6
Qwen3.5 [49]	–	–	397B-A17B	–	–	91.1	–	–	–	–	–	–
Qwen3.5 [49]	–	Code	397B-A17B	–	–	**95.8**	–	–	–	–	–	–
Kimi K2.5 [50]	–	–	1T-A32B	–	–	77.0	–	–	–	–	–	–
Specialized visual reasoning methods												
SEAL [12]	LLaVA	WF	7B	74.8	76.3	75.4	–	–	–	–	–	–
DyFo [51]	Qwen2-VL	WF	7B	80.0	82.9	81.2	–	–	–	–	–	–
ZoomEye [52]	LLaVA-OV	WF	7B	**93.9**	85.5	90.6	84.3	55.0	69.6	<u>88.5</u>	50.0	69.3
DeepEyes [7]	Qwen2.5-VL	Zoom	7B	<u>91.3</u>	<u>88.2</u>	90.1	91.3	59.0	75.1	86.8	58.5	72.6
DeepEyesV2 [18]	Qwen2.5-VL	General	7B	–	–	81.8	–	–	77.9	–	–	73.8
Thyme-RL [17]	Qwen2.5-VL	Code	7B	83.5	80.3	82.2	91.0	63.0	77.0	86.5	57.5	72.0
ZwZ-8B [21]	Qwen3-VL	$\times$	8B	–	–	91.1	–	–	84.4	–	–	<u>82.0</u>
SenseNova-MARS [19]	Qwen3-VL	C+S	8B	–	–	92.2	–	–	83.1	–	–	78.4
Qwen3-VL-2B† [2]	–	Zoom	2B	82.6	65.8	75.9	79.3	52.8	66.0	68.0	53.3	60.6
+ OPO	Qwen3-VL	Zoom	2B	89.6	80.3	85.9	90.5	66.3	78.4	85.3	<u>67.5</u>	76.4
Δ (vs Qwen3-VL-2B)	–	–	–	+7.0	+14.5	+10.0	+11.2	+13.5	+12.4	+17.3	+14.2	+15.8
Qwen3-VL-8B† [2]	–	Zoom	8B	<u>91.3</u>	82.9	88.0	<u>92.5</u>	<u>70.3</u>	81.4	87.8	65.3	76.5
+ OPO	Qwen3-VL	Zoom	8B	**93.9**	**90.8**	92.7	**95.8**	**75.3**	**85.5**	**93.0**	**68.5**	80.8
Δ (vs Qwen3-VL-8B)	–	–	–	+2.6	+7.9	+4.7	+3.3	+5.0	+4.1	+5.3	+3.3	+4.3

5.2 Main Results

For readability, we denote Qwen3-VL-2B/8B trained with OPO as OPO-2B/8B.

Overall performance. Table 1 presents the main results of OPO on high-resolution perception benchmarks. Overall, OPO consistently improves performance across model scales and datasets. Based on Qwen3-VL-2B, OPO-2B improves the overall accuracy on V*, HR-Bench 4K, and HR-Bench 8K by 10.0, 12.4, and 15.8 points, respectively. Based on Qwen3-VL-8B, OPO-8B also brings gains of 4.7, 4.1, and 4.3 points. The gains suggest that observation-action optimization is particularly effective for acquiring fine-grained visual evidence in high-resolution settings. These results show that OPO substantially improves smaller models while remaining effective at larger scales.

Comparison with strong baselines. Compared with strong open-source and proprietary models, OPO-8B also achieves competitive performance. On V*, OPO-8B obtains 92.7% overall accuracy, outperforming several strong baselines, including Gemini-3 Pro, Qwen3-VL-32B, and Qwen3.5, while remaining below the much larger Qwen3-VL-235B-A22B and the Code-enabled Qwen3.5. On HR-Bench 4K, OPO-8B reaches 85.5% overall accuracy, achieving the best result in the table and slightly surpassing Qwen3-VL-235B-A22B at 85.4%. On HR-Bench 8K, OPO-8B improves Qwen3-VL-8B from 76.5% to 80.8%, approaching the performance of much larger models. These results suggest that directly optimizing observation actions can effectively improve high-resolution visual evidence acquisition, rather than relying solely on scaling model size.

Comparison with specialized visual reasoning methods. OPO-8B also compares favorably with other specialized visual reasoning methods. SEAL, DyFo, ZoomEye, DeepEyes, DeepEyesV2,

Table 2: Controlled comparison between DeepEyes-style outcome-level GRPO and OPO under the same backbone and zoom-tool training setup. All results are reproduced by us.

Model	V* Bench			HR-Bench 4K			HR-Bench 8K		
	Attr.	Spatial	Overall	FSP	FCP	Overall	FSP	FCP	Overall
Qwen2.5-VL-7B [1]	53.0	55.3	53.9	68.5	55.3	61.9	60.8	48.0	54.4
+ DeepEyes-style GRPO [7]	87.0	81.6	84.8	88.0	55.0	71.5	76.3	59.8	68.0
+ OPO	93.9	80.3	88.5	87.8	60.3	74.0	78.8	57.5	68.1
Δ (vs GRPO)	+6.9	-1.3	+3.7	-0.2	+5.3	+2.5	+2.5	-2.3	+0.1
Qwen3-VL-2B [2]	82.6	65.8	75.9	79.3	52.8	66.0	68.0	53.3	60.6
+ DeepEyes-style GRPO [7]	85.2	79.0	82.7	88.0	55.0	71.5	85.3	57.0	71.1
+ OPO	89.6	80.3	85.9	90.5	66.3	78.4	85.3	67.5	76.4
Δ (vs GRPO)	+4.4	+1.3	+3.2	+2.5	+11.3	+6.9	+0.0	+10.5	+5.3
Qwen3-VL-8B [2]	91.3	82.9	88.0	92.5	70.3	81.4	87.8	65.3	76.5
+ DeepEyes-style GRPO [7]	87.8	93.4	90.1	91.5	73.3	82.4	86.3	69.3	77.8
+ OPO	93.9	90.8	92.7	95.8	75.3	85.5	93.0	68.5	80.8
Δ (vs GRPO)	+6.1	-2.6	+2.6	+4.3	+2.0	+3.1	+6.7	-0.8	+3.0

Thyme-RL, and SenseNova-MARS improve fine-grained visual perception through visual search, dynamic focusing, or tool use, while ZwZ distills region-level supervision into the model for single-pass perception without tools. OPO-8B achieves 92.7% and 85.5% overall accuracy on V* and HR-Bench 4K, respectively, ranking first among these methods on both benchmarks; on HR-Bench 8K, it achieves 80.8%, second only to ZwZ-8B. Notably, OPO-2B achieves 78.4% and 76.4% on HR-Bench 4K and 8K, outperforming several 7B baselines. These results show that targeted observation exploration and localized credit attribution enable OPO to improve the acquisition and use of fine-grained visual evidence.

Comparison with outcome-level GRPO. Table 2 further provides a controlled comparison between OPO and DeepEyes-style outcome-level GRPO. The DeepEyes results in Table 1 are taken from the original paper, whereas the DeepEyes-style GRPO baselines in Table 2 are retrained by us under the same backbone, training budget, and zoom-tool setup as OPO. DeepEyes-style GRPO relies only on outcome-level rewards such as final-answer correctness, format quality, and tool-use signals. In contrast, OPO further introduces uncertainty-guided local branching and evidence-localized attribution. Although DeepEyes-style GRPO performs better on a few subtasks, OPO is more stable in terms of overall accuracy, especially on HR-Bench. This suggests that outcome-level rewards can encourage the model to use the zoom-in tool, but they still struggle to assign local evidence-quality feedback to the specific observation actions. By exploring alternative crops around high-uncertainty observation events and applying evidence-sufficiency feedback to the corresponding observation-action tokens, OPO more effectively optimizes where the model looks.

Qualitative examples and analysis are provided in Appendix G and Figures 5–6.

5.3 Ablation Study

Effect of core modules. Table 3 presents the ablation results based on Qwen3-VL-2B. Unless otherwise specified, all OPO variants use the default rollout setting $M = 8, B = 2$, where M is the total rollout budget, B is the number of base trajectories, and the remaining $P = M - B$ slots are used for event-local branching and backfill. Under the default rollout budget, the full OPO achieves the best performance among all module ablations, showing that both uncertainty-guided observation exploration and evidence-localized attribution are important for improving high-resolution visual evidence acquisition.

Since ELAA relies on sibling trajectories produced by UOB for local comparison, we do not remove UOB alone; instead, we jointly remove UOB and ELAA. Without UOB and ELAA, the model degenerates into DeepEyes-style outcome-level GRPO training [7]. This variant performs substantially worse than the full OPO on all three benchmarks, suggesting that ordinary full-trajectory sampling is insufficient for exploring alternative observation actions, especially in high-resolution tasks where an incorrect crop can directly affect the final answer.

Removing ELAA keeps the UOB branching mechanism but removes localized evidence attribution, so the evidence-sufficiency feedback is no longer applied specifically to observation-action tokens and instead becomes a trajectory-level signal. This variant still shows consistent degradation compared with the full method, indicating that outcome-level or trajectory-level supervision alone can cause credit mismatch between observation actions and downstream reasoning. Meanwhile, w/o ELAA remains better than w/o UOB & ELAA, suggesting that UOB's local branching itself is beneficial, but ELAA is needed to more effectively convert local evidence quality into observation-action learning signals.

We further examine the design choices inside UOB. Replacing uncertainty-guided event selection with random branching reduces performance, indicating that high-entropy observation events are more informative branching points than randomly selected zoom events. Removing redundancy filtering also hurts performance, suggesting that without penalizing crops that heavily overlap with previous observations, the limited branching budget can be wasted on redundant visual regions.

Table 3: Ablation study on Qwen3-VL-2B. We report overall accuracy. The default setting is $M = 8, B = 2$, where M is the rollout budget and B is the number of base trajectories. The default full OPO setting is highlighted in bold.

Variant	V*	HR-Bench 4K	HR-Bench 8K
Qwen3-VL-2B	75.9	66.0	60.6
OPO full	**85.9**	**78.4**	**76.4**
w/o UOB & ELAA	82.7	71.5	71.1
w/o ELAA	83.8	76.5	73.8
w/o uncert. sel.	83.3	76.9	73.0
w/o redund. filt.	84.8	76.5	75.1
$M = 4, B = 1$	82.4	74.2	72.5
$M = 8, B = 1$	84.3	76.6	74.8
$M = 8, B = 2$	**85.9**	**78.4**	**76.4**
$M = 8, B = 4$	84.8	77.1	75.3
$M = 16, B = 2$	86.2	78.8	76.9

Effect of rollout allocation. We also analyze the effect of rollout budget and base-trajectory allocation. The smaller setting $(M = 4, B = 1)$ performs worse, showing that insufficient rollout budget limits local observation exploration. When fixing $M = 8$, using only $B = 1$ provides too few base candidates and may miss useful observation events, while increasing to $B = 4$ reduces the budget available for local branching and limits the exploration of alternative crops. The default setting $M = 8, B = 2$ achieves a better balance between base-candidate coverage and event-local branching. Increasing the budget to $M = 16, B = 2$ brings only marginal gains, suggesting diminishing returns from larger rollout budgets.

6 Conclusion

In this work, we study a key challenge in tool-augmented thinking with images: models must learn not only how to reason, but also where to look during reasoning. We show that many failures in high-resolution fine-grained visual tasks originate from the observation stage rather than downstream language reasoning, and that the observation-action tokens determining local crops exhibit substantially higher uncertainty than ordinary language tokens. These findings suggest that visual evidence acquisition is a critical decision-making process that should be optimized directly, rather than treated as an ordinary part of trajectory-level generation. Based on these observations, we propose Observation Policy Optimization (OPO), which improves visual evidence acquisition through uncertainty-guided observation branching and evidence-localized advantage attribution. OPO performs local branching at high-uncertainty and low-redundancy observation events, allowing the model to explore alternative crops from shared reasoning prefixes. It then assigns evidence-sufficiency feedback only to the observation-action tokens that produce the corresponding crops, thereby reducing the credit mismatch between final-answer rewards and local evidence quality. Experiments across high-resolution visual reasoning benchmarks show that OPO consistently improves Qwen3-VL-2B and Qwen3-VL-8B, and OPO-8B achieves competitive performance against much larger models and tool-augmented baselines. These results highlight observation policy optimization as an important direction for improving active visual reasoning.

Acknowledgments and Disclosure of Funding

This work was supported by the Ministry of Industry and Information Technology under Grant No. BB2320000141; the National Key Research and Development Plan under Grant No. 2024YFC3014700; the Central Government-Guided Local Science and Technology Development Funds of Anhui Province under Grant No. 2023CSJGG1100; the National Natural Science Foundation of China (NSFC) under Grant No. 62476260; the Fundamental Research Funds for the Central Universities under Grant No. WK2100000057; and the CCF-1688 Yuanbao Cooperation Fund under Grant No. 2025006. The numerical computations were performed on the supercomputing system at the Supercomputing Center of the University of Science and Technology of China.

References

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

[2] 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*, 2025.

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

[4] 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. *arXiv preprint arXiv:2504.07491*, 2025.

[5] Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

[6] Xu Chu, Xinrong Chen, Guanyu Wang, Zhijie Tan, Kui Huang, Wenyu Lv, Tong Mo, and Weiping Li. Qwen look again: Guiding vision-language reasoning models to re-attention visual information. *arXiv preprint arXiv:2505.23558*, 2025.

[7] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

[8] Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*, 2025.

[9] Meng Cao, Haoze Zhao, Can Zhang, Xiaojun Chang, Ian Reid, and Xiaodan Liang. Ground-r1: Incentivizing grounded visual reasoning via reinforcement learning. *arXiv preprint arXiv:2505.20272*, 2025.

[10] Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, Linjie Li, Yu Cheng, Heng Ji, Junxian He, and Yi R. Fung. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers, 2025. URL `https://arxiv.org/abs/2506.23918`.

[11] 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.

[12] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.

[13] 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, pages 7907–7915, 2025.

[14] Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Chain-of-Focus: Adaptive visual search and zooming for multimodal reasoning via RL. *arXiv preprint arXiv:2505.15436*, 2025. URL https://arxiv.org/abs/2505.15436v1.

[15] Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Zheng Huang, Mingyu Liu, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, et al. Active-o3: Empowering multimodal large language models with active perception via grpo. *arXiv preprint arXiv:2505.21457*, 2025.

[16] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. OpenThinkIMG: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025. URL https://arxiv.org/abs/2505.08617.

[17] Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, Haonan Fan, Kaibing Chen, Jiankang Chen, Haojie Ding, Kaiyu Tang, Zhang Zhang, Liang Wang, Fan Yang, Tingting Gao, and Guorui Zhou. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*, 2025. URL https://arxiv.org/abs/2508.11630.

[18] Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

[19] Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, Gao Huang, Dahua Lin, and Lewei Lu. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025. URL https://arxiv.org/abs/2512.24330.

[20] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*, 2025.

[21] Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, Zhuosheng Zhang, and Weiran Huang. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*, 2026. URL https://arxiv.org/abs/2602.11858.

[22] Zirun Guo, Minjie Hong, Feng Zhang, Kai Jia, and Tao Jin. Thinking with programming vision: Towards a unified view for thinking with images. *arXiv preprint arXiv:2512.03746*, 2025. URL https://arxiv.org/abs/2512.03746.

[23] Xinhai Hou, Shaoyuan Xu, Manan Biyani, Moyan Li, Jia Liu, Todd C. Hollon, and Bryan Wang. Codev: Code with images for faithful visual reasoning via tool-aware policy optimization. *arXiv preprint arXiv:2511.19661*, 2025. URL https://arxiv.org/abs/2511.19661.

[24] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.

[25] Bowen Ye, Yongchao Xu, Zhijian Li, Xiang Yin, Junkai Ma, and Wenzhao Li. CoEvo-Mem: Co-evolving retrieval policy and memory bank for LLM agents. *arXiv preprint arXiv:2608.01739*, 2026. URL https://arxiv.org/abs/2608.01739.

[26] Yongchao Xu, Bowen Ye, Jiefeng Gan, Junkai Ma, Wenzhao Li, Sen Tao, Yi Wei, and Jiawei Liu. VideoEvolve: Co-evolving memory and retrieval for long video understanding. *arXiv preprint arXiv:2610.10183*, 2026. URL https://arxiv.org/abs/2610.10183.

[27] Yongchao Xu, Jiawei Liu, Junfeng Wang, Sen Tao, Na Jiang, and Zheng-Jun Zha. Learning to diversify and focus: A reinforcement framework for open-vocabulary HOI detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 34673–34682, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xu_Learning_to_Diversify_and_Focus_A_Reinforcement_Framework_for_Open-Vocabulary_CVPR_2026_paper.html.

[28] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36: 68539–68551, 2023.

[30] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023.

[31] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. AgentTuning: Enabling generalized agent abilities for LLMs. *arXiv preprint arXiv:2310.12823*, 2023. URL https://arxiv.org/abs/2310.12823.

[32] Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. Agent-flan: Designing data and methods of effective agent tuning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9354–9366, 2024.

[33] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs, 2025. URL https://arxiv.org/abs/2504.13958.

[34] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

[35] Yifan Wei, Xiaoyan Yu, Yixuan Weng, Tengfei Pan, Angsheng Li, and Li Du. AutoTIR: Autonomous tools integrated reasoning via reinforcement learning. *arXiv preprint arXiv:2507.21836*, 2025. URL https://arxiv.org/abs/2507.21836.

[36] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[37] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025.

[38] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025.

[39] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

[40] Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use. *arXiv preprint arXiv:2505.19255*, 2025.

[41] Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025.

[42] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. *CoRR*, abs/2507.19849, 2025. doi: 10.48550/ARXIV.2507.19849. URL https://doi.org/10.48550/arXiv.2507.19849.

[43] Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic entropy-balanced policy optimization. *CoRR*, abs/2510.14545, 2025. doi: 10.48550/ARXIV.2510.14545. URL https://doi.org/10.48550/arXiv.2510.14545.

[44] OpenAI. GPT-5 System Card. https://cdn.openai.com/gpt-5-system-card.pdf, August 2025. Accessed: 2026-05-07.

[45] OpenAI. Update to GPT-5 System Card: GPT-5.2. https://cdn.openai.com/pdf/3a4153c8-c748-4b71-8e31-aecbde944f8d/oai_5_2_system-card.pdf, December 2025. Accessed: 2026-05-07.

[46] Anthropic. Claude Opus 4.6 System Card. https://www-cdn.anthropic.com/0dd865075ad3132672ee0ab40b05a53f14cf5288.pdf, 2026. Accessed: 2026-05-07.

[47] Google DeepMind. Gemini 3 Pro Model Card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf, December 2025. Accessed: 2026-05-07.

[48] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024. URL https://arxiv.org/abs/2408.03326.

[49] Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

[50] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

[51] Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9098–9108, 2025.

[52] 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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6602–6618, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.335. URL https://aclanthology.org/2025.emnlp-main.335/.

A Algorithm

A.1 Algorithm Procedure

Algorithm 1 summarizes the full OPO training procedure, combining the rollout-time observation branching of UOB (Section 4.1) with the evidence-localized advantage assignment of ELAA (Section 4.2).

Algorithm 1 OPO with UOB and ELAA

Require: Policy π_θ, input sample x, total rollout budget M, base rollout budget B, branch budget $P = M - B$

UOB: rollout-time observation branching

Generate B base rollouts until the first valid image zoom-in call or termination.
For each candidate observation event e, record its pre-tool prefix, observation-action spans, and the original candidate crop.
Estimate bounding-box uncertainty $g(e)$ from the bounding-box token span.
Compute the question-normalized anomaly score $z(e)$.
for each candidate event e while the branch budget remains do **do**
    Allocate at most $Z(e) = \mathcal{A}(z(e), P_{\text{rem}})$ child proposals.
    Initialize $\mathcal{C}_e$ with the original candidate crop.
    Preserve the prompt and pre-tool prefix, then resample $Z(e)$ suffixes from that state.
    Sequentially accept valid child candidates with $\text{Red}(c \mid e) \leq \tau_{\text{red}}$ and add them to $\mathcal{C}_e$.
    Execute accepted child tool calls, continue each retained trajectory to termination, and form a branch group g from trajectories sharing the same pre-tool prefix.
end for
Backfill with ordinary rollouts until M complete trajectories are obtained.

ELAA: evidence-localized advantage attribution

Compute global outcome rewards R_i^{global} and standard GRPO advantages A_i^{global} by question id.
For each branch group g, judge each crop $c_{i,g}$ and compute group-local evidence rewards $R_{i,g}^{\text{evi}} = \rho(c_{i,g})$.
Normalize $R_{i,g}^{\text{evi}}$ within $\mathcal{S}(g)$ to obtain $A_{i,g}^{\text{local}}$.
Apply $A_{i,g}^{\text{local}}$ only to image-index, region-label, and bounding-box spans; use A_i^{global} for other generated tokens.
Update π_θ with the clipped PPO policy objective.

A.2 Policy Update

After obtaining the token-level advantage $A_{i,t}$, OPO updates the policy with the standard clipped PPO objective. OPO does not change the form of the PPO update; its effect enters through the advantage $A_{i,t}$ defined in the main text, which assigns global outcome advantages to ordinary tokens and evidence-local advantages to observation-action tokens.

For generated token $y_{i,t}$, let

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x_i, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x_i, y_{i,<t})}. \tag{12}$$

The policy objective is

$$\mathcal{L}_{\text{policy}} = -\sum_{i,t} m_{i,t}^{\text{resp}} \min\left(r_{i,t}(\theta)A_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A_{i,t}\right), \tag{13}$$

where $m_{i,t}^{\text{resp}}$ is the response mask. Tool-observation tokens returned by the environment are masked out and do not contribute to the policy loss.

B Benchmark Details

We evaluate OPO on three high-resolution perception benchmarks: V* [12], HR-Bench 4K, and HR-Bench 8K [13]. Following the high-resolution evaluation setting used in DeepEyes [7], these

benchmarks focus on the ability of models to acquire and interpret fine-grained visual evidence from high-resolution images. Many examples cannot be answered reliably from a global image view or a heavily downsampled image alone; instead, they require models to locate local evidence such as small objects, subtle attributes, local text, or spatial relations.

V* [12] focuses on guided visual search over natural images and evaluates whether a model can locate and use critical visual details in high-resolution images. The V* split used in our evaluation contains 191 examples in total, including 115 examples from `direct_attributes` and 76 examples from `relative_position`. For V*, we report the benchmark-defined attribute, spatial, and overall accuracies.

HR-Bench [13] contains two high-resolution evaluation versions, HR-Bench 4K and HR-Bench 8K, which measure whether models can resolve local visual evidence under increasingly demanding image resolutions. Each version contains 200 original questions, including 100 Fine-grained Single-instance Perception (FSP) questions and 100 Fine-grained Cross-instance Perception (FCP) questions. FSP evaluates fine-grained perception over a single instance, such as attribute recognition, OCR, and visual prompting, while FCP evaluates fine-grained perception across multiple instances, such as map analysis, chart analysis, and spatial relationship assessment.

In our evaluation, each original HR-Bench question is expanded into four option-rotation instances. As a result, `hr_bench_4k` contains 800 evaluation rows and `hr_bench_8k` contains 800 evaluation rows, yielding 1,600 rows in total. When reporting results, we aggregate performance according to the original benchmark definitions and report FSP, FCP, and overall accuracy following the official scoring protocol.

These benchmarks are particularly aligned with our setting because successful prediction often requires an explicit observation action: the model must first decide which local region should be inspected and then answer based on the acquired local evidence. Therefore, for both OPO and our reproduced Qwen3-VL tool-use baselines, we keep the image zoom-in tool enabled during inference. The model is allowed to perform multi-turn tool-augmented reasoning, and we evaluate only the final answer after the trajectory terminates.

C Evaluation Protocol

We evaluate saved checkpoints with deterministic multi-turn generation. For each benchmark example, we use the same system instructions, response format, and tool schema as in training, and generate one assistant turn at a time using greedy decoding. Each sample allows up to 10 tool-use turns. When the generated response contains a valid `<tool_call>...</tool_call>` block, the evaluator parses the JSON arguments, converts the bounding box to image coordinates, crops the selected image from the visible image pool, appends the crop as a tool message, and continues generation. The trajectory terminates when an `<answer>...</answer>` block is produced, an invalid tool call is encountered, or the maximum number of turns is reached.

For V*, we evaluate the direct attributes and relative position splits. Each example is treated as a multiple-choice visual question, with the first option as the benchmark-provided correct answer. The final answer is extracted from the answer tags. We apply an LLM judge to compare the model answer with the standard answer using few-shot consistency examples and return a binary judgement. We report accuracy for each split and compute the V* overall score as the mean over all evaluated examples.

For HR-Bench, we evaluate the HR-Bench 4K and HR-Bench 8K splits. Images are decoded from the benchmark TSV files and resized according to the Qwen-VL visual grid before inference. The same multi-turn tool loop and answer extraction procedure are used. The judging script reports accuracy separately for the benchmark's Fine-grained Single-instance Perception (FSP) and Fine-grained Cross-instance Perception (FCP) categories and computes each split's overall score as the unweighted average of these two category accuracies. These scores correspond to the HR-Bench columns reported in Table 1.

D Training Data Details

For fair comparison with DeepEyes, we use the same tool-oriented training data recipe rather than constructing a new training corpus. As described in DeepEyes [7], the training mixture contains about 47K samples from three complementary sources: Visual Search, ArxivQA, and ThinkLite-VL. Visual Search contributes about 22K samples, roughly 47% of the mixture, and emphasizes natural-image grounding and fine-grained perception. ArxivQA contributes about 14K samples, roughly 30%, covering scientific plots, diagrams, and chart-style visual inputs. ThinkLite-VL contributes about 11K samples, roughly 23%, and provides multimodal reasoning examples involving arithmetic, commonsense inference, and problem solving.

The DeepEyes data curation process follows a tool-use-oriented filtering pipeline. First, it removes questions that are either too easy or too difficult by estimating model accuracy from multiple responses. Second, it converts questions into an open-ended format and filters unreliable conversions. Third, it removes unverifiable samples, including examples with incorrect answers or unreadable questions. Finally, it prioritizes samples where tool use can improve visual evidence acquisition, especially cases where the model fails from the original image but can answer correctly when given the relevant cropped region. Chart data and reasoning data are retained according to their own preprocessing assumptions because they already provide complementary structured-visual and reasoning supervision.

E Prompt and Tool Configuration

We use the Qwen3-VL tool-calling format in both training and evaluation. Each example is converted into a conversation containing tool instructions and the user's visual question. The tool instructions provide the `image_zoom_in_tool` function signature inside `<tools>...</tools>` XML tags and require each function call to be returned as a JSON object inside `<tool_call>...</tool_call>` tags. The user message contains the image placeholder, the original question, candidate options, and the instruction to think first, call tools when needed, and finally provide an answer enclosed by answer tags. The concrete prompt and tool-call schema are summarized in Figure 3.

The image zoom-in tool crops a region from the selected image and returns the cropped image as the next tool observation. The tool maintains a visible image pool consisting of the original input image followed by all tool-returned crops from previous turns. Thus, `img_idx=0` refers to the original image, while later indices can refer to earlier crops. For Qwen3-VL models, bounding boxes are interpreted in the normalized $[0, 1000]$ coordinate system and converted to image-space pixels before cropping.

F Reward and Evidence Judge

The reward function has two parts: a trajectory-level final-answer reward and an event-local evidence reward. The final-answer reward first extracts the model's answer from `<answer>...</answer>` after the thinking segment. A judge then compares the extracted answer with the ground-truth answer given the original question and returns a binary correctness label. We denote this answer reward by $r_{\mathrm{acc}} \in \{0, 1\}$. The implementation also includes a tool-use reward r_{tool}, which is one only when the trajectory uses the zoom-in tool and the final answer is correct, and a format reward r_{fmt}, which is -1 when the response violates the required thinking/answer format and 0 otherwise. The global trajectory reward is

$$R^{\mathrm{global}} = 0.8 r_{\mathrm{acc}} + 1.2 r_{\mathrm{tool}} + 0.2 r_{\mathrm{fmt}}. \tag{14}$$

This global reward is used for the ordinary trajectory-level GRPO signal.

For OPO's local observation-action supervision, we additionally use an evidence judge. For each branch event, the training pipeline keeps the crops associated with that event and sends each crop, together with the question and ground-truth answer, to a frozen visual judge. The judge must return exactly one of three labels: `insufficient`, `partial`, or `sufficient`. These labels are mapped to local evidence rewards -1, 0, and 1, respectively.

The exact prompts for the final-answer judge and the local evidence judge are shown in Figure 4. In our implementation, the final-answer reward judge is instantiated with Qwen3-Plus, while the local evidence reward judge is instantiated with Qwen3-VL-Max.

```
SYSTEM:
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.

# 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": "image_zoom_in_tool",
    "description": "Zoom in on a specific region of an image by cropping it based on a bounding
    box (bbox) and an optional object label.",
    "parameters": {
      "type": "object",
      "properties": {
        "bbox_2d": {
          "type": "array",
          "items": {"type": "number"},
          "minItems": 4,
          "maxItems": 4,
          "description": "The bounding box of the region to zoom in, as [x1, y1, x2, y2], where
          (x1, y1) is the top-left corner and (x2, y2) is the bottom-right corner."
        },
        "label": {
          "type": "string",
          "description": "The name or label of the object in the specified bounding box."
        },
        "img_idx": {
          "type": "integer",
          "description": "0-based index into the currently visible image list, including the
          original input images followed by tool-returned crops from earlier turns."
        }
      },
      "required": ["bbox_2d", "label", "img_idx"]
    }
  }
}
</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>

USER:
<image>
Your role is that of a research assistant specializing in visual information. Answer questions
about images by looking at them closely and then using research tools. Please follow this
structured thinking process and show your work.

Start an iterative loop for each question:
- First, look closely: Begin with a detailed description of the image, paying attention to the
user's question. List what you can tell just by looking, and what you'll need to look up.
- Next, find information: Use a tool to research the things you need to find out.
- Then, review the findings: Carefully analyze what the tool tells you and decide on your next
action.

Continue this loop until your research is complete.

To finish, bring everything together in a clear, synthesized answer that fully responds to the
user's question.

Question: {question}
Options:
A. {option_1}
B. {option_2}
...

First think in <thinking>...</thinking>. If needed, call tools using
<tool_call>...</tool_call>. Final answer must be enclosed in <answer>...</answer>.
```

Figure 3: Prompt and tool-call schema used for tool-augmented visual reasoning, including the tool signature for `image_zoom_in_tool` and the required `<tool_call>` return format.

```
LLM ANSWER JUDGE:
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.

VLM EVIDENCE JUDGE:
You are a visual evidence judge.

Your task is NOT to answer the question.
Your task is to decide whether the current cropped image contains enough visible
evidence to support the given ground-truth answer.

Inputs:
1. Question
2. Ground-truth answer
3. Current cropped image

Judging rule:
Decide whether the cropped image provides sufficient visual evidence for the
ground-truth answer.

Labels:
- insufficient: the crop clearly does not contain enough visual evidence to
  support the answer
- partial: the crop contains some relevant evidence, but it is still incomplete,
  ambiguous, or lacks enough context/detail to confidently support the answer
- sufficient: the crop contains enough clear visual evidence to support the answer

Important constraints:
- Judge only from the cropped image.
- Do NOT rely on commonsense, outside knowledge, or guess from the answer itself.
- Even if the ground-truth answer looks plausible, if the crop does not visibly
  support it, do NOT output sufficient.
- For spatial or relation questions, check whether enough context is preserved.
- For fine-grained attribute or OCR questions, check whether the key detail is
  visible and clear enough.

Output exactly one label from:
insufficient
partial
sufficient
```

Figure 4: Prompts for final-answer judging and local visual-evidence judging. The LLM judge produces the trajectory-level correctness signal, while the VLM judge produces the event-local evidence reward for ELAA.

G Qualitative Results

Figure 5 and Figure 6 show qualitative comparisons between direct Qwen3-VL-8B inference and OPO-8B. In the direct-attribute example (Figure 5), direct Qwen3-VL-8B inference is distracted by salient but irrelevant yellow regions, such as the pedestrian-crossing sign, and mistakenly treats them as evidence for the color of the cyclist's box. In contrast, OPO-8B focuses its observation action on the cyclist and the attached box, correctly identifying the box as green from localized evidence. In the spatial-relation example (Figure 6, top), Qwen3-VL-8B observes both the cone and the baby carriage but fails to reliably resolve their left-right relation; OPO-8B instead crops a local region containing both objects and correctly determines that the carriage is to the right of the cone. In the HR-Bench example (Figure 6, bottom), Qwen3-VL-8B relies on a coarse building-level observation and mistakes the signboard for an oval shape, whereas OPO-8B accurately localizes the small signboard and zooms into it, allowing it to recognize its rectangular shape from the clear edges and background. These cases show that OPO's gains do not merely come from invoking the zoom-in tool, but from more reliably selecting question-relevant and evidence-sufficient observation regions, thereby grounding the final answer in localized visual evidence.

H Implementation Details

In the main implementation, the rollout budget is $M = 8$, with $B = 2$ base rollouts and $P = 6$ remaining trajectories for event-local branching and backfill. The pre-action baseline window is

$W = 16$, and event uncertainty is computed only from the bounding-box token span, without using image-index or post-crop tokens. Branches preserve the pre-event prefix and resample the suffix from the prompt-plus-prefix state. If the total number of trajectories remains below M, ordinary rollouts are used as backfill. For local attribution, the local token spans cover image-index, region-label, and bounding-box tokens, and the local advantage is computed only from the event-local evidence reward.

I Localization Failures in Stronger Models

To examine whether the observation bottleneck persists in stronger models, we evaluate Qwen3-VL-32B-Instruct, Qwen3.7-Max, and Kimi K3 on all 191 examples in V*. We analyze their inference trajectories without applying OPO training. We manually inspect each unsuccessful trajectory, including incorrect responses and cases without a valid final answer. A trajectory is classified as a localization failure when failure to acquire the visual evidence needed to answer the question is the primary cause of the unsuccessful response.

Table 4: Localization failures in stronger models on V*. Percentages in the last column are computed over unsuccessful trajectories.

Model	Accuracy	Unsuccessful	Localization failures
Qwen3-VL-32B-Instruct	169/191 (88.5%)	22	17/22 (77.3%)
Kimi K3	172/191 (90.1%)	19	15/19 (78.9%)
Qwen3.7-Max	182/191 (95.3%)	9	9/9 (100.0%)
Overall	523/573 (91.3%)	50	41/50 (82.0%)

As shown in Table 4, the three models achieve an aggregate accuracy of 91.3%, yet localization failures account for 41 of their 50 unsuccessful trajectories (82.0%). This pattern appears in all three models: localization failures account for 17/22 unsuccessful trajectories for Qwen3-VL-32B-Instruct, 15/19 for Kimi K3, and 9/9 for Qwen3.7-Max. Under the evaluated observation interface, acquiring fine-grained visual evidence thus remains a major source of failure even for stronger models. These findings motivate extending observation-policy optimization to larger backbones and additional model families, with controlled OPO training as a direction for future work.

J Limitations and Future Work

Our work focuses on optimizing observation actions under an image zoom-in tool. This setting directly targets a key bottleneck in high-resolution fine-grained perception, where the model must decide where to look in order to acquire useful local evidence. However, tool-augmented visual reasoning in more general scenarios may involve a broader set of tools, such as OCR, detection, segmentation, search, code execution, or more complex image manipulation tools. Therefore, our current experiments do not fully characterize how OPO generalizes to a more diverse tool ecosystem.

Second, due to computational and time constraints, we train and evaluate OPO mainly on Qwen3-VL-2B and Qwen3-VL-8B, rather than further applying it to newer or stronger native multimodal backbones such as Qwen3.5. As a result, although our experiments show that OPO remains effective when scaling from 2B to 8B models, its behavior on larger and more capable backbone models remains to be further validated.

Future work can extend OPO in two main directions. First, OPO can be generalized to broader visual and multimodal tools, enabling models to learn not only where to look, but also when to invoke different tools and how to combine them for evidence acquisition. Second, applying OPO to stronger native multimodal models would further test its scalability and clarify whether observation-action optimization can complement stronger visual encoders, longer-context reasoning, and more general tool-use capabilities.

K Societal Impacts

OPO aims to improve the ability of vision-language models to acquire fine-grained visual evidence from high-resolution images. This can have positive impacts in applications that require careful

inspection of visual details, such as document understanding, chart analysis, scientific image interpretation, accessibility tools, and assistance with complex visual question answering. By making observation actions more reliable, the method may also reduce failures where a model answers from insufficient or irrelevant visual evidence.

At the same time, stronger tool-augmented visual reasoning systems may introduce risks if deployed without appropriate oversight. More reliable zooming and local evidence acquisition could be misused for privacy-invasive visual analysis, automated surveillance, or high-stakes decision making where model errors are difficult for users to detect. The method may also increase user trust in generated answers even when the acquired evidence is incomplete or the evidence judge is miscalibrated.

<image> What is the color of the cyclist's box?

<thinking>Looking at the image, I can see a cyclist on the left side of the street, near the crosswalk. The cyclist is wearing a bright green jacket and is riding a bicycle. There is no visible "box" associated with the cyclist in the image. The question seems to be asking about the color of the cyclist's box, but there is no box visible. I will need to look more closely at the cyclist and their immediate surroundings to see if there is a box that I missed.</thinking>

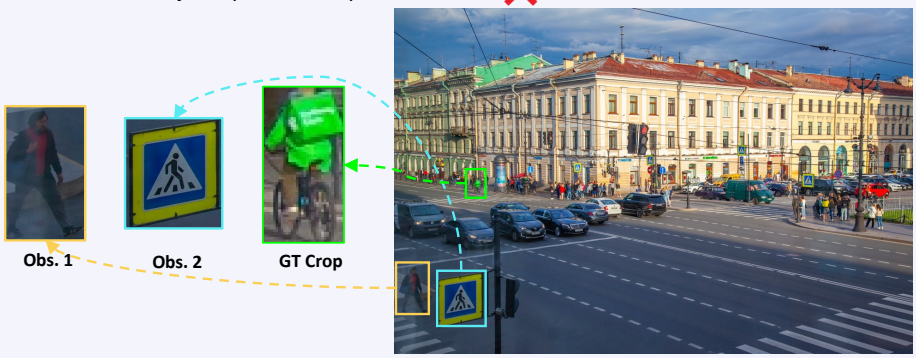
<tool_call>{"name": "image_zoom_in_tool", "arguments": {"label": "a cyclist on the left side of the street", "bbox_2d": [14, 745, 70, 888], "img_idx": 0}} </tool_call>

<thinking>After zooming in on the cyclist, I can see that the person is wearing a dark jacket with a red scarf or inner layer visible. There is no box visible on or near the cyclist. The question asks about the color of the cyclist's box, but there is no box in the image. I need to re-examine the image to see if I missed something or if the question is misphrased.</thinking>

<tool_call>{"name": "image_zoom_in_tool", "arguments": {"label": "a closer view of the pedestrian crossing sign", "bbox_2d": [80, 765, 175, 925], "img_idx": 0}}</tool_call>

<thinking>After examining the image closely, I can see that there is no box associated with the cyclist. The question seems to be misphrased or refers to something else. However, there is a pedestrian crossing sign in the image, which has a yellow border. It's possible that the question is referring to the color of this sign. Since the sign is yellow, and the options provided are A. green, B. black, C. yellow, and D. red, the correct answer would be C. yellow. I will now provide the answer.</thinking>

<answer>C. The color of the cyclist's box is yellow.</answer> ✗



<thinking>I can see a cyclist in the image, and I need to identify the color of their box. The cyclist is wearing a green outfit, and there is a box or container near them. I will zoom in on the cyclist and their box to get a closer look and confirm the color.</thinking>

<tool_call>{"name": "image_zoom_in_tool", "arguments": {"label": "a cyclist wearing a green outfit", "bbox_2d": [148, 493, 168, 539], "img_idx": 0}}</tool_call>

<thinking>Based on the close-up image, I can clearly see that the cyclist is wearing a green outfit and has a large box attached to the back of their bicycle. The box is also green, matching the cyclist's outfit. This confirms that the color of the cyclist's box is green.</thinking>

<answer>A. The color of the cyclist's box is green.</answer> ✓



Figure 5: Qualitative comparison on a V* direct-attribute example.



Figure 6: Qualitative comparisons on V* and HR-Bench. The top example is a V* spatial-relation question, while the bottom example is from HR-Bench.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction make claims that accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper discusses the limitations of the work in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: There is no theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This work provides all the information needed to reproduce the main experimental results of the paper in the Implementation Details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is publicly available at `https://github.com/Yann0122/Observation-Policy-Optimization`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setups, including data splits, training, and testing details, are provided in the Method and Experiments sections.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the default evaluations in the vision-language models field, which do not require error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide them in the implementation details of the main paper and the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impacts are provided in the Appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The proposed method uses pre-trained models. This proposed method is safe under the safeguards of adopted pre-trained models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the original paper that produced the code package or dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and accompanying documentation have been publicly released at `https://github.com/Yann0122/Observation-Policy-Optimization`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: There is no research with human subjects in this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.