EVIROVER: REINFORCING AGENTIC PERCEPTION BEYOND A GLANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual perception is conventionally formulated as a one-shot prediction from a single glance at the image, under the assumption that the image content and the model's parametric knowledge suffice to resolve the query. This assumption often fails in real-world scenarios that hinge on fine-grained visual details or require knowledge-intensive and up-to-date information. We term such cases *perception under insufficient evidence* and formulate perception as an agentic process that can obtain information beyond a single glance. To address the absence of data for this setting, we design two dedicated data generation pipelines, yielding EviRover-SFT-5K and EviRover-RL-12K for training. We further construct EviLens, a human-verified benchmark comprising 688 instances across five perception categories. Building on these data, we present EviRover, to our knowledge the first perception agent explicitly trained to resolve perceptual queries through interaction, using supervised fine-tuning followed by agentic reinforcement learning. Experiments show that the 4B EviRover outperforms its backbone by 30 points on average on EviLens, reaching performance comparable to advanced proprietary models. The gains transfer beyond EviLens to WebEyes, conventional perception benchmarks, and general multimodal benchmarks, including a 15-point improvement on BrowseComp-VL. All code, models, and data are released.

1 INTRODUCTION

Multimodal large language models (MLLMs) have progressed rapidly in reasoning and tool use (Ma et al., 2025; Feng et al., 2026; Du et al., 2026; Yan et al., 2026). These advances have enabled multimodal agents to address increasingly complex visual questions through multi-step interaction (Huang et al., 2026; Fan et al., 2026; Chu et al., 2026; Jiao et al., 2026). Yet in most such systems, agentic interaction serves question answering rather than perceptual prediction. Perception tasks themselves have largely retained their conventional formulation (Wang et al., 2026b; Tang et al., 2026b; Wang et al., 2026a; Deng et al., 2026; Pacini et al., 2026). Grounding, segmentation, and counting are still commonly treated as one-shot predictions from an image-query pair, based on a single *glance* at the input image. This formulation assumes that the image and the model's parametric knowledge are sufficient to resolve the query. In practice, fine-grained visual details may require closer inspection, while identifying the target may depend on knowledge beyond the image. We refer to such cases as *perception under insufficient evidence*. In these settings, a single *glance* is insufficient to determine the required perceptual output, yet the model must still produce a location, a boundary, or a count without the opportunity for further search.

We therefore reformulate perception under insufficient evidence as an evidence-seeking process. Rather than predicting directly from the initial observation, the model can search through finer-grained visual inspection or external knowledge sources to resolve the perceptual task. Whereas existing multimodal agents (Huang et al., 2026; Wu et al., 2026; Zhang et al., 2025) gather evidence to support a textual answer, the proposed formulation treats the perceptual output itself as the objective. This makes perception a stricter test of evidence seeking: in question answering, searched evidence can be converted directly into a textual answer, whereas in perception it must be translated into a location, a boundary, or a count on the input image, which demands fine-grained visual understanding. Moreover, the perceptual output is directly verifiable against spatial or numerical annotations, and is correct only if the agent both acquires the missing evidence and correctly relates it to the visual content. Perception is also of considerable practical value, as accurate perception under-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

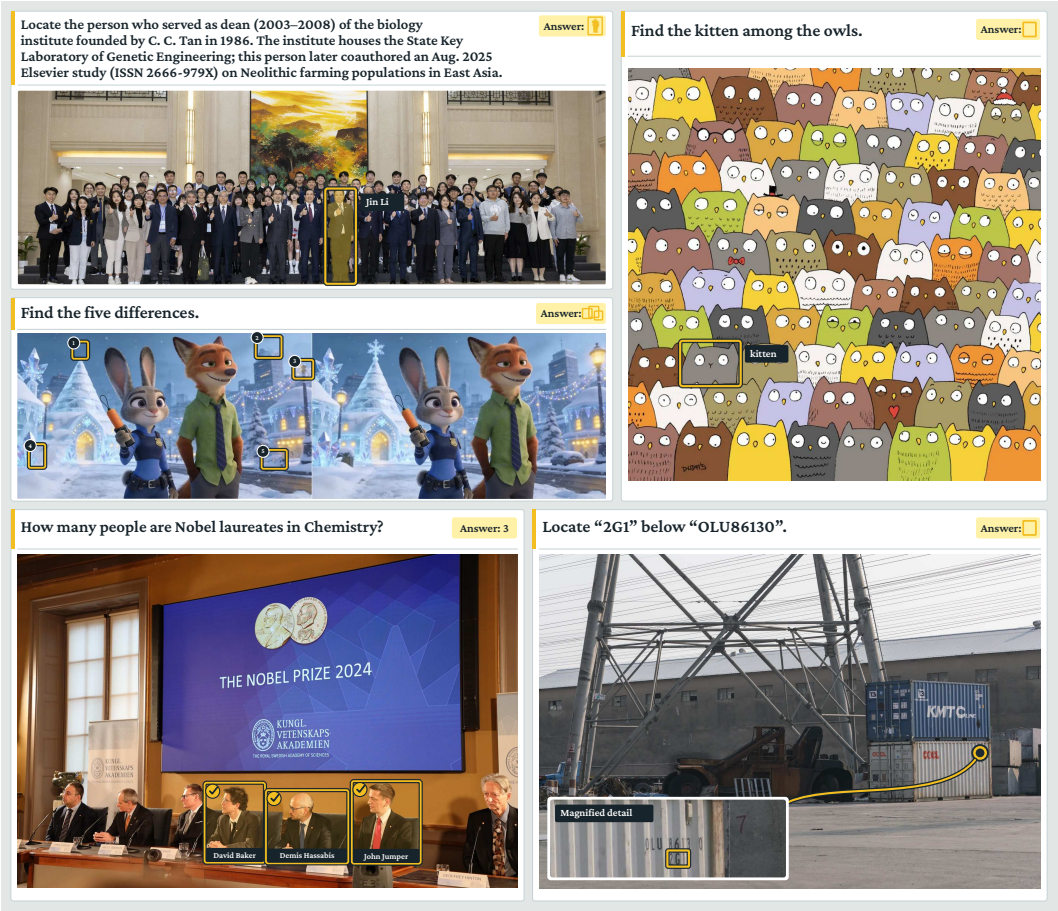


Figure 1: Representative examples from our training data and the EviLens benchmark.

pins downstream applications such as embodied manipulation (Zhang et al., 2024a; Kim et al., 2026) and image editing (Liu et al., 2024a; 2026). Despite these advantages, agentic perception poses a central challenge: the missing evidence varies across queries. A small target calls for closer inspection, whereas an unfamiliar identity calls for external search. Deciding what to acquire and how is therefore part of the task rather than a fixed procedure. Recent prompt-based workflows (Yang et al., 2026; Tang et al., 2026a) have begun to incorporate web search into perception. However, they rely on manually designed prompting strategies at inference time, so their searching behavior is bounded by human-designed heuristics rather than learned from task feedback. Liang et al. (2026) train an agent that interleaves reasoning with web search. But it is limited to segmentation and assumes that missing evidence is always external knowledge.

Inspired by recent developments in agentic reinforcement learning for visual question answering (Zheng et al., 2026; Wu et al., 2026; Hong et al., 2026), we ask: *can we train agents to learn when and how to search to push perception beyond a single glance?*

To answer this question, we introduce EviRover, to our knowledge the first agent trained to search both within and beyond the image to resolve perceptual queries. EviRover moves beyond a single *glance*: at each step, it decides whether further evidence is needed and which action can provide it, before producing a location, a boundary, or a count. As existing perception datasets do not cover this setting, we design two data generation pipelines, each targeting one form of evidence insufficiency. For evidence present in the image but not discernible at a glance, we collect high-resolution images with targets occupying less than 0.1% of the image area, together with scenes requiring multi-step visual exploration, such as I-spy and spot-the-difference puzzles. For evidence beyond the image, we construct queries over group photographs of public figures and anime characters through two procedures: (1) starting from group photographs and verifying the identities of the depicted individuals; and (2) starting from a reference image of a known individual or character and synthesizing a

group photograph containing them with GPT-Image-2 (OpenAI, 2026b), followed by filtering with Seed-2.0-Pro (Seed, 2026b) for visual quality and identity preservation. These pipelines yield two training sets, EviRover-SFT-5K and EviRover-RL-12K, and EviLens, a human-verified benchmark of 688 instances spanning segmentation, counting, and three grounding categories: localization, recognition, and spot-the-difference. Figure 1 presents representative examples.

Extensive experiments demonstrate the effectiveness of EviRover. On EviLens, EviRover improves over its Qwen3-VL-4B-Instruct backbone by 30 points on average across the five categories, bringing a 4B model to a level comparable with advanced proprietary models such as Seed-2.1-turbo and Gemini-3.5-Flash. We also evaluate on WebEyes (Yang et al., 2026), a benchmark for search-based grounding, segmentation, and visual question answering, where EviRover improves over its backbone by 14.4 IoU on grounding and 24.7 gIoU on segmentation. The gains further extend beyond our setting: EviRover consistently improves on conventional perception benchmarks such as ReasonSeg (Lai et al., 2024) and RefCOCOg (Mao et al., 2016; Nagaraja et al., 2016), as well as general multimodal benchmarks, including MMMU (Yue et al., 2024), MMMU-Pro (Yue et al., 2025), and MathVerse (Zhang et al., 2024b), with a 15-point gain on BrowseComp-VL (Geng et al., 2026). These results indicate that learning to search under insufficient evidence not only enhances perception but also yields capabilities that generalize well beyond the training distribution.

Our contributions can be summarized as follows:

- We identify *perception under insufficient evidence* and reformulate perception as an evidence-seeking process that looks beyond a single *glance*.
- We present **EviRover**, to our knowledge the first perception agent trained to determine what evidence is missing and how to acquire it.
- To address the absence of data for perception under insufficient evidence, we design two dedicated data generation pipelines, yielding **EviRover-SFT-5K**, **EviRover-RL-12K**, and **EviLens**, a human-verified benchmark of 688 instances.
- Extensive experiments validate EviRover: it improves over its backbone by 30 points on average on EviLens and generalizes to WebEyes, conventional perception benchmarks, and general multimodal benchmarks.

2 Related Work

Visual Perception. Grounding (Kamath et al., 2021), segmentation (Hu et al., 2016), and counting (Amini-Naieni et al., 2023) are conventionally formulated as predicting a location, a boundary, or a count from an image and a query. Beyond specialized detectors (Liu et al., 2024b), segmentation models (Carion et al., 2025), and counting models (Amini-Naieni et al., 2024), MLLM-based methods extend these tasks to free-form queries requiring reasoning (Lai et al., 2024; You & Wu, 2025; Liu et al., 2025). In parallel, a growing body of work has focused specifically on fine-grained perception of small objects in high-resolution images (Goto et al., 2025; Zhang et al., 2026; Jia et al., 2026). However, these advances retain the one-shot formulation, in which the evidence needed to resolve the query is assumed to be present in the input image. Our work lifts this assumption, allowing the perceptual prediction to draw on evidence acquired beyond a single glance at the image.

Evidence-Seeking Multimodal Agents. A broad class of visual questions cannot be resolved from the image and the model's parametric knowledge alone (Chen et al., 2023; Zeng et al., 2026; Choi et al., 2026). One line of work, often referred to as thinking with images (Su et al., 2025), enables models to actively inspect image regions through operations such as zooming and cropping (Zheng et al., 2026; Wu & Xie, 2024; Guo et al., 2026; Zhang et al., 2025). Another line develops multimodal search agents that acquire external knowledge through web search, progressing from short-horizon search to long-horizon deep research (Wu et al., 2026; Geng et al., 2026; Huang et al., 2026; Yao et al., 2026). However, these works acquire evidence to produce textual answers, treating perception as a means rather than an end. Recent attempts extend search to segmentation and grounding (Yang et al., 2026; Tang et al., 2026a; Liang et al., 2026), but either rely on manually designed prompting or restrict evidence seeking to external search for a specific task. In contrast, our work treats perception as the goal of evidence seeking, where the acquired evidence

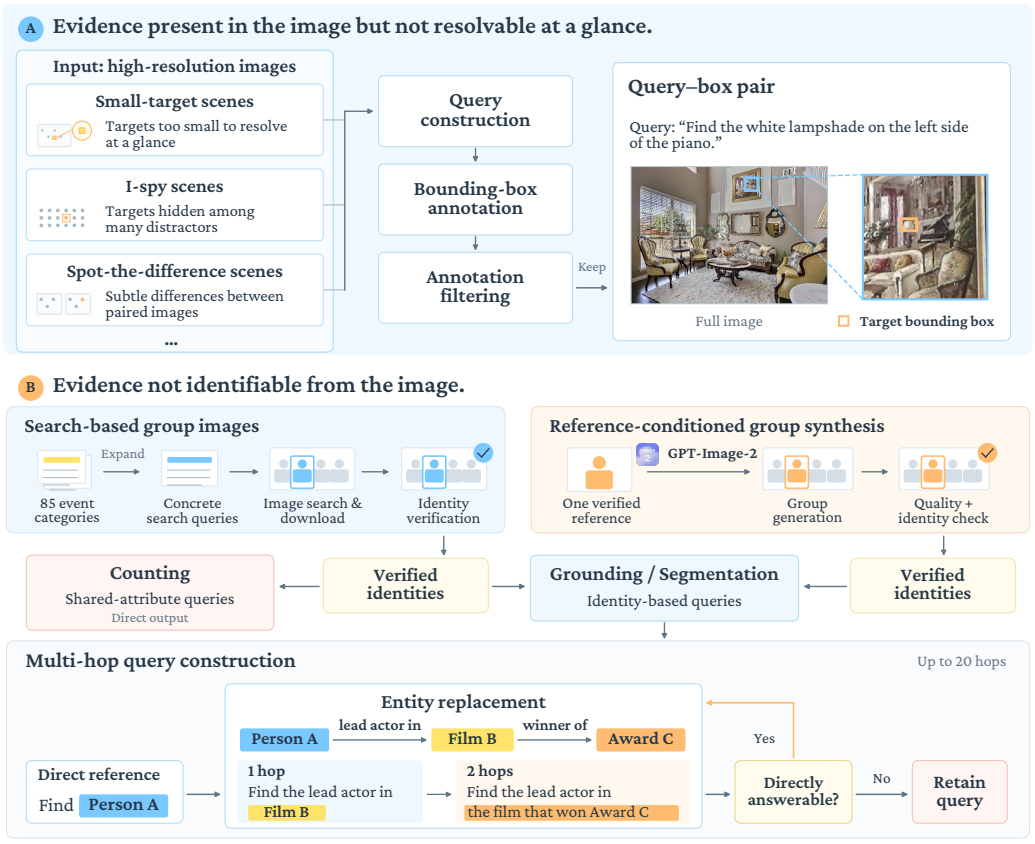


Figure 2: Overview of our data generation pipeline.

must be resolved into a perceptual prediction on the input image. Since the kind of evidence a query lacks varies, deciding what to acquire and how becomes an integral part of the task.

3 METHOD

3.1 DATASET CONSTRUCTION

As illustrated in Figure 2, we design two dedicated data generation pipelines targeting different forms of missing evidence.

Evidence present in the image but not resolvable at a glance. We collect high-resolution images containing targets so small that they cannot be resolved without inspecting the image at a finer scale, in many cases occupying well under 0.1% of the image area, together with images where the target can only be found by comparing or scanning multiple regions, such as I-spy and spot-the-difference scenes. Images are adapted from Mini-o3 (Lai et al., 2026a) and collected from the web through image search.

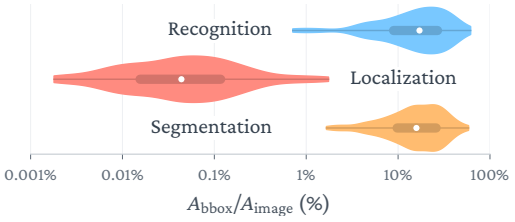


Figure 3: Distribution of target area relative to image area.

Evidence beyond the image. We construct perception queries over group photographs of people, including public figures and anime characters, through two complementary procedures.

The first collects real group photographs and verifies the identities of the people they depict. We begin from a manually written seed list of 85 event categories spanning award ceremonies, music groups, esports competitions, political summits, product launches, academic events, sports competitions, and entertainment programs. We prompt Seed-1.8 to expand each category into concrete search queries, which are then issued to an image search engine to obtain group photographs and their associated captions. However, captions are an unreliable source of identity annotations, as they are absent for most collected images and frequently incomplete or inaccurate when present. We therefore verify identities independently using Seed-2.0-Pro equipped with web search tools.

The second procedure starts from a reference image of a known individual or character and synthesizes a group photograph containing that individual. Since single-person images with verifiable identities are far more abundant than group photographs in which every identity can be verified, this procedure covers a substantially broader range of identities. We initially explored conditioning generation on multiple verified reference images simultaneously. For real individuals, identity preservation degrades sharply as the number of references increases, remaining reliable for only up to approximately three references. Beyond this number, the generated individuals can no longer be reliably matched to their references through reverse image search, a failure further confirmed by manual inspection. Anime characters, in contrast, are defined by distinctive visual traits such as hairstyles and costumes, and remain well preserved under multi-reference conditioning. We therefore condition each generation of real individuals on a single reference identity while leaving the other individuals in the scene unconstrained, and condition anime generation on multiple character references. The images are synthesized with GPT-Image-2 and filtered by Seed-2.0-Pro, which assesses visual quality and verifies identity preservation by searching for the generated individual and confirming that the results match the reference identity.

The two procedures thus yield group photographs with different levels of identity coverage: real photographs may have some or all depicted individuals verified, synthesized photographs of real individuals contain exactly one verified individual, and synthesized anime images may contain multiple verified characters.

Query construction. The verified identities serve as ground truth for query construction. Grounding and segmentation queries require at least one verified identity per image, which serves as the target to be localized or segmented. Counting queries require every depicted individual to be verified, and are formulated over attributes shared among them, such as affiliation with a particular institution or receipt of a particular honor.

In their initial form, these queries name the target directly, for example by asking for the mask of a specific individual. A model that recognizes the named individual can resolve such queries without seeking any additional evidence. To make evidence seeking necessary, we rewrite each query into a multi-hop form through iterative entity replacement. At each step, we select an entity in the current query, obtain its public profile through a search engine, and prompt GPT-5.4-mini to replace the entity with an indirect description that identifies it without naming it. The replacement terminates once Seed-1.8 can no longer resolve the query without tools, with a maximum of 20 hops.

SFT trajectory construction. We generate agent trajectories for the constructed queries, using Gemini-3.0-Pro for localization queries and Seed-2.0-Pro for the remaining tasks, and retain only those that reach the correct answer through a logically consistent sequence of actions. Correctness alone, however, does not ensure that a trajectory exhibits the intended behavior. For example, queries about widely recognized individuals are often resolved by a single search or directly from parametric knowledge, involving little evidence seeking. We therefore downsample such trajectories while retaining a portion of them, so that the model learns both to answer directly when its knowledge suffices and to seek evidence when it does not. Figure 4 shows the resulting composition of EviRover-SFT-5K.

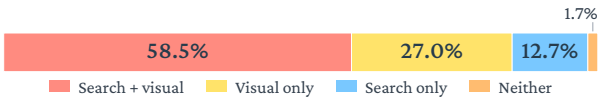


Figure 4: Tool-use composition of EviRover-SFT-5K.

Together, these procedures yield two training sets, EviRover-SFT-5K and EviRover-RL-12K.

Table 1: Comparison of EviLens with existing perception benchmarks.

Benchmark	Task			Required ability		
	Grounding	Segmentation	Counting	Visual acquisition	Visual comparison	External knowledge
RefCOCOg (Mao et al., 2016)	✓	✓	✗	✗	✗	✗
ReasonSeg (Lai et al., 2024)	✗	✓	✗	✗	✗	✓
Ref-Adv (Dong et al., 2026)	✓	✗	✗	✗	✓	✗
SOREC (Goto et al., 2025)	✓	✗	✗	✓	✗	✗
OK-VOS (Liang et al., 2026)	✗	✓	✗	✗	✗	✓
WebEyes (Yang et al., 2026)	✓	✓	✗	✗	✗	✓
EviLens (Ours)	✓	✓	✓	✓	✓	✓

3.2 EviLens

We construct EviLens with the pipelines above and manually verify every instance, yielding a benchmark of 688 instances for evaluating perception under insufficient evidence. It covers segmentation, counting, and three grounding categories that differ in the type of missing evidence. *Localization* targets are explicitly named but tiny or hidden among clutter, as in high-resolution scenes and I-spy puzzles. *Recognition* targets are visually salient but referred to indirectly, requiring external information to identify. *Spot-the-difference* targets are defined relative to a second panel and require cross-panel comparison. *Segmentation* and *counting* share the recognition setting but output masks and counts, respectively.

Table 1 compares EviLens with existing benchmarks in task coverage and required abilities. EviLens contains 140 localization, 182 recognition, 15 spot-the-difference, 195 segmentation, and 156 counting instances, with the spot-the-difference instances comprising 79 annotated differences in total.

Metrics. Recognition and localization are evaluated by mean box IoU and R@0.5. Segmentation is evaluated by gIoU (mean per-instance IoU) and cIoU (cumulative intersection over cumulative union). Counting is evaluated by exact-match accuracy. Spot-the-difference is evaluated by micro- and macro-F1 under greedy one-to-one matching at IoU 0.5, where micro-F1 pools matches across images and macro-F1 averages per-image scores.

3.3 Model Training

We train EviRover in two stages. SFT establishes the interaction format and basic tool-use skills, while RL teaches the model to identify what evidence each query lacks and how to acquire it, using the correctness of the final perceptual output as feedback.

Agent framework. EviRover operates with a set of ten tools, shared across trajectory labeling, training, and evaluation. Four tools acquire external evidence: text search, text-to-image search, image search, and web browsing. Three tools support fine-grained visual inspection, such as cropping a region for closer examination and rendering a candidate region on the original image for verification. The remaining three support task-specific prediction: SAM3-based mask generation, side-by-side region comparison for spot-the-difference queries, and a Python interpreter. Table 6 summarizes the available tools; detailed specifications and complete tool schemas are provided in Appendix A.5.2.

Supervised fine-tuning. We fine-tune Qwen3-VL-4B-Instruct on EviRover-SFT-5K (Section 3.1) to initialize tool use and task-specific prediction. The loss is computed only over model-generated tokens, with tool observations masked.

Reinforcement learning. We further optimize the SFT model on EviRover-RL-12K. Since the training data span heterogeneous perception tasks with substantially different reward distributions, we adopt EMA-GRPO (Feng et al., 2026), which maintains task-specific running reward statistics to normalize the learning signal across tasks.

We use task-specific outcome rewards in $[0, 1]$. Let $\hat{p}$ denote a predicted box and g its ground-truth counterpart, and let $\hat{c}$ and c denote the predicted and ground-truth counts. For segmentation, let $\hat{M}$ be the mask returned by SAM3 from the predicted box and point prompts, and M the ground-truth mask. The reward is

$$R = \begin{cases} \mathrm{IoU}(\hat{p}, g), & \text{grounding}, \\ \mathrm{IoU}(\hat{M}, M), & \text{segmentation}, \\ \mathbb{1}[\hat{c} = c], & \text{counting}, \\ F_1^{\mathrm{soft}}(\hat{P}, G), & \text{spot-the-difference}, \end{cases} \tag{1}$$

where segmentation predictions are required to provide a bounding box together with at least one positive and one negative point.

For spot-the-difference, a prediction may recover only a subset of the annotated differences, so we use a soft set-level reward. Let $\hat{P} = \{\hat{p}_1, \ldots, \hat{p}_n\}$ be the predicted boxes in the left panel and $G = \{g_1, \ldots, g_m\}$ the annotated differences. We form the candidate set

$$\mathcal{C} = \{(i, j) : \mathrm{IoU}(\hat{p}_i, g_j) \geq \tau\}, \tag{2}$$

sort candidate pairs by IoU in descending order, and greedily construct a one-to-one matching $\mathcal{M}$. Each matched pair is weighted by its overlap:

$$\mathrm{STP} = \sum_{(i,j)\in\mathcal{M}} \mathrm{IoU}(\hat{p}_i, g_j)^{\alpha}, \qquad F_1^{\mathrm{soft}} = \frac{2\,\mathrm{STP}}{n + m}, \tag{3}$$

with the reward taken as zero when $n + m = 0$. The threshold τ admits partially correct boxes, while α downweights low-quality matches; implementation details are provided in Appendix A.2.1.

For policy-loss aggregation, we use sequence-mean-token-mean reduction instead of the token-mean reduction used in EMA-GRPO. This avoids the length bias of token-mean, which linearly amplifies the influence of long rollouts and can cause a small number of unusually long trajectories to disproportionately dominate the policy update. We discuss this choice in detail in Appendix A.4.

4 Experiments

4.1 Training and Evaluation Settings

Training details. We initialize from Qwen3-VL-4B-Instruct and train on a single node of 8 NVIDIA A800-80GB GPUs with FSDP. Both stages use AdamW ($\beta = (0.9, 0.999)$, weight decay 0.01) with gradient clipping at 1.0. Supervised fine-tuning uses a learning rate of 5×10^{-6} under a cosine schedule with warmup ratio 0.03, a global batch size of 64, and 2 epochs. Reinforcement learning uses a constant learning rate without warmup, and samples 8 rollouts for each of 8 prompts per step, giving 64 rollouts per step divided into 4 mini-batches for the policy update.

Evaluation details. We evaluate primarily on EviLens, and additionally on WebEyes, a search-based grounding, segmentation and VQA benchmark. All methods receive the same input prompt (Appendix A.5.1) and use the same task-specific output interfaces, with coordinates normalized to $[0, 1000]$ and boxes in `xyxy` format. Direct baselines predict a box from the image and query in a single pass; segmentation converts the predicted box into a mask with SAM3, and counting returns an integer. Agentic baselines and EviRover additionally have access to the tool framework of Section 3.3, with a limit of 35 tool calls per trajectory. Further details are given in Appendix A.3.

4.2 Performance Analysis on EviLens

As shown in Table 2, EviLens is challenging for all model families. Among proprietary models, Gemini-3.5-Flash, Seed-2.1-Turbo, and GPT-5.6-Sol perform best overall. Gemini-3.5-Flash and Seed-2.1-Turbo lead on recognition and counting, and Seed-2.1-Turbo achieves the highest segmentation score of any model (0.779 gIoU). GPT-5.6-Sol performs notably poorly on spot-the-difference, achieving a micro-F1 of only 0.112, as it consistently predicts a single bounding box even when the prompt indicates multiple differences.

Table 2: Evaluation results on EviLens.

Model	Grounding						Segmentation		Counting
	Localization		Recognition		Spot Diff				
	IoU	R@.5	IoU	R@.5	$F1_{mi}$	$F1_{ma}$	gIoU	cIoU	Acc
Closed-source Models									
GPT-5.6-Sol (OpenAI, 2026a)	0.358	0.407	0.615	0.660	0.112	0.087	0.627	0.699	51.3
GPT-5.6-Luna (OpenAI, 2026a)	0.264	0.250	0.533	0.571	0.192	0.224	0.544	0.564	35.5
GPT-5.6-Terra (OpenAI, 2026a)	0.304	0.321	0.528	0.548	0.231	0.313	0.539	0.567	43.4
Seed-2.1-Turbo (ByteDance Seed, 2026)	0.331	0.326	0.636	0.674	0.241	0.273	0.779	0.714	60.5
Seed-1.8 (Seed, 2026a)	0.360	0.393	0.542	0.560	0.190	0.274	0.613	0.610	48.0
Qwen3.7-Plus (Qwen Team, 2026)	0.351	0.414	0.569	0.606	0.137	0.187	0.577	0.576	46.7
Gemini-3.5-Flash (Google DeepMind, 2026)	0.373	0.421	0.636	0.668	0.274	0.280	0.579	0.582	61.8
Open-Source Segmentation Models									
Seg-R1-7B (You & Wu, 2025)	–	–	–	–	–	–	0.266	0.382	–
Seg-Zero-7B (Liu et al., 2025)	–	–	–	–	–	–	0.343	0.407	–
SAM3-Agent (Carion et al., 2025)	–	–	–	–	–	–	0.213	0.287	–
Open-Source Grounding Models									
Perception-R1 (Yu et al., 2026)	0.007	0.000	0.106	0.044	0.000	0.000	–	–	17.3
UniVG-R1 (Bai et al., 2025b)	0.051	0.036	0.283	0.264	0.014	0.016	–	–	–
Open-Source General Models									
Minimax-M3 (Lai et al., 2026b)	0.018	0.000	0.286	0.329	0.010	0.013	0.329	0.469	37.8
OneThinker-8B (Feng et al., 2026)	0.110	0.086	0.353	0.357	0.065	0.106	0.409	0.387	25.6
InternVL-3.5-8B (Wang et al., 2025)	0.020	0.007	0.093	0.033	0.000	0.000	0.081	0.081	20.5
Our Models									
Qwen3VL-4B-Inst (Bai et al., 2025a)	0.276	0.251	0.265	0.231	0.005	0.008	0.203	0.279	19.2
+ SFT	0.369	0.371	0.595	0.626	0.114	0.179	0.584	0.633	42.8
EviRover(Ours)	0.444	0.500	0.648	0.687	0.171	0.266	0.686	0.708	50.0

Open-source models perform substantially worse. Despite strong results on conventional perception benchmarks, segmentation specialists reach at most 0.343 gIoU, and grounding specialists achieve near-zero IoU on localization. General-purpose open-source MLLMs exhibit similarly limited performance. The difficulty of EviLens therefore lies not in perceptual precision but in acquiring the evidence each query lacks, a capability that existing perception benchmarks do not evaluate.

Table 3: Evaluation results on the WebEyes benchmark.

Model	Grounding		Segmentation		VQA
	IoU	R@.5	gIoU	cIoU	
Doubao-Seed-2.0-Pro	35.7	44.4	61.2	43.3	65.4
Gemini-3.1-Pro	30.5	35.1	54.6	38.8	63.8
Qwen3VL-4B-Inst	25.6	29.9	24.8	21.7	36.0
Qwen3VL-8B-Inst	26.8	32.6	35.8	25.9	36.3
Pixel-Searcher-8B	34.1	41.3	39.1	32.4	**42.2**
EviRover(Ours)	**40.0**	**48.5**	**49.5**	**38.0**	40.3

Against this backdrop, EviRover achieves the best localization and recognition results of all evaluated models, with a clear margin on localization (0.444 IoU versus 0.373 for Gemini-3.5-Flash), and ranks second on segmentation behind Seed-2.1-Turbo. On counting and spot-the-difference, where its backbone scores 19.2 and 0.005 respectively, EviRover surpasses most proprietary models, coming within 1.3 points of GPT-5.6-Sol on counting. Relative to its backbone, EviRover improves the primary metric of every category, by 30.2 points on average, bringing a 4B model to a level comparable with the strongest proprietary systems.

4.3 Performance on WebEyes

Table 3 reports results on WebEyes, where baseline results are taken from the original paper and bold marks the best result among open-source models. EviRover achieves the best grounding performance among all evaluated models, including the proprietary systems, with 40.0 IoU and 48.5

R@0.5. On segmentation, it outperforms Pixel-Searcher-8B, a prompt-based workflow built on a larger backbone, by 10.4 gIoU. On VQA, a task absent from our training data, it performs comparably to Pixel-Searcher-8B. These results suggest that the evidence-seeking behavior learned by EviRover transfers beyond EviLens.

4.4 Generalization beyond EviLens

Conventional perception. To examine whether evidence-seeking training affects standard perception, we evaluate EviRover on grounding and segmentation in ReasonSeg and RefCOCOg (Table 4). EviRover improves over its backbone on every split and metric, and outperforms task-specific models on most metrics, such as Seg-Zero-7B on segmentation and Perception-R1 on grounding. The capabilities acquired through evidence-seeking training thus carry over to conventional perception.

Table 4: Evaluation results on traditional perception benchmarks.

Model	ReasonSeg				RefCOCOg Grounding						RefCOCOg Segmentation			
	Val		Test		Val			Test			Val		Test	
	gIoU	cIoU	gIoU	cIoU	@50	@75	@95	@50	@75	@95	gIoU	cIoU	gIoU	cIoU
Seg-Zero-7B	62.6	62.0	57.5	52.0	-	-	-	-	-	-	68.3	65.3	68.8	66.8
Perception-R1	-	-	-	-	85.6	75.6	**31.7**	85.3	75.9	**32.7**	-	-	-	-
Qwen3VL-4B-Inst	63.9	58.9	60.9	53.9	85.7	74.9	28.6	85.3	75.4	29.4	71.6	68.8	72.3	69.9
EviRover (Ours)	**68.7**	**62.3**	**62.6**	**56.5**	**86.1**	**76.4**	31.5	**85.4**	**76.7**	32.5	**73.7**	**71.9**	**73.7**	**72.3**

General multimodal benchmarks. Beyond perception, we assess whether the learned behavior benefits broader multimodal capabilities, spanning multidisciplinary reasoning (MMMU and MMMU-Pro), mathematical reasoning (MathVerse), and agentic visual search (BrowseComp-VL), as shown in Table 5. EviRover improves over its backbone on all four benchmarks. On the reasoning benchmarks the gains reach up to 4 points, while on BrowseComp-VL accuracy more than doubles, rising from 9.5 to 24.8. The consistent gains on reasoning show that specializing in evidence-seeking perception also improves the model's general capabilities, and the substantially larger gain on BrowseComp-VL shows that the learned behavior transfers to agentic visual question answering, which requires acquiring external evidence.

Table 5: Generalization to unseen benchmarks.

Model	MMMU	MMMU-Pro		MathVerse	BrowseComp-VL		
	Acc.	Std.	Vision	Overall	L1	L2	Overall
Qwen3-VL-4B-Inst	63.1	44.5	46.9	62.4	10.6	8.5	9.5
EviRover (Ours)	**66.2**	**48.5**	**48.4**	**64.5**	**33.7**	**16.0**	**24.8**

5 Conclusion

In this work, we study perception under insufficient evidence, where a single glance at the image does not suffice to resolve the query. We reformulate perception in this setting as an evidence-seeking process. To support this formulation, we develop two dedicated data generation pipelines and construct EviLens, a human-verified benchmark for evaluating perception under insufficient evidence. Building on these data, we train EviRover to actively seek the missing evidence required by each query through supervised fine-tuning followed by agentic reinforcement learning. Experiments show that EviRover substantially improves over its backbone on EviLens and WebEyes, and that these gains extend to conventional perception and general multimodal benchmarks. These results suggest that perception can move beyond a single glance: learning to identify and acquire missing evidence improves perceptual accuracy and yields evidence-seeking capabilities that generalize to broader multimodal reasoning.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and improving the clarity and presentation of the manuscript. During experimentation, they were also used to facilitate the setup, adaptation, and debugging of baseline implementations and to assist with reproducing existing results. All AI-assisted code and experimental procedures were reviewed and verified by the authors, and all reported experimental results were obtained through experiments conducted by the authors. The authors take full responsibility for the content, claims, and results presented in this work.

REPRODUCIBILITY STATEMENT

We provide implementation and training details, including model configurations, hyperparameters, prompts, tool specifications, and reward formulations in Section 3.3, Section 4.1 and Appendix A.2. Details of the data construction and evaluation protocols are also documented to facilitate reproduction of our experiments. All code, models, and data are released.

REFERENCES

Niki Amini-Naieni, Kiana Amini-Naieni, Tengda Han, and Andrew Zisserman. Open-world text-specified object counting. *arXiv preprint arXiv:2306.01851*, 2023.

Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *Advances in Neural Information Processing Systems*, 37:48810–48837, 2024.

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

Sule Bai, Mingxing Li, Yong Liu, Jing Tang, Haoji Zhang, Lei Sun, Xiangxiang Chu, and Yansong Tang. Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. *arXiv preprint arXiv:2505.14231*, 2025b.

ByteDance Seed. Seed2.1 Officially Released: Advancing AI Productivity. https://seed.bytedance.com/en/blog/seed2-1-officially-released-advancing-ai-productivity, June 2026.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14948–14968, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.925. URL https://aclanthology.org/2023.emnlp-main.925/.

Changin Choi, Wonseok Lee, Jungmin Ko, and Wonjong Rhee. Progressive multimodal search and reasoning for knowledge-intensive visual question answering. In Maria Liakata, Viviane P.

Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19291–19315, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.881. URL https://aclanthology.org/2026.acl-long.881/.

Zheng Chu, Xiao Wang, Jack Hong, Huiming Fan, Yuqi Huang, Yue Yang, Guohai Xu, Chenxiao Zhao, Cheng Xiang, Shengchao Hu, et al. Redsearcher: A scalable and cost-efficient framework for long-horizon search agents. *arXiv preprint arXiv:2602.14234*, 2026.

Ken Deng, Yunhan Yang, Jingxiang Sun, Xihui Liu, Yebin Liu, Ding Liang, and Yan-Pei Cao. Geosam2: Unleashing the power of sam2 for 3d part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6367–6376, 2026.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-adv: Exploring mllm visual reasoning in referring expression tasks. *arXiv preprint arXiv:2602.23898*, 2026.

Yifan Du, Zikang Liu, Jinbiao Peng, Jie Wu, Junyi Li, Jinyang Li, Wayne Xin Zhao, and Ji-Rong Wen. Towards long-horizon agentic multimodal search. *arXiv preprint arXiv:2604.12890*, 2026.

Kaixuan Fan, Kaituo Feng, Haoming Lyu, Dongzhan Zhou, and Xiangyu Yue. Sophiavl-r1: Reinforcing mllms reasoning with thinking reward. In *International Conference on Learning Representations*, volume 2026, pp. 9290–9310, 2026.

Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, et al. Onethinker: All-in-one reasoning model for image and video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5432–5443, 2026.

Xinyu Geng, Peng Xia, Zhen Zhang, Xinyu Wang, Qiuchen Wang, Ruixue Ding, Chenxi Wang, Jialong Wu, Kuan Li, Yida Zhao, et al. Webwatcher: Breaking new frontiers of vision-language deep research agent. In *International Conference on Learning Representations*, volume 2026, pp. 61240–61271, 2026.

Google DeepMind. Gemini 3.5 Flash: Model Card. https://deepmind.google/models/model-cards/gemini-3-5-flash/, May 2026.

Kanoko Goto, Takumi Hirose, Mahiro Ukai, Shuhei Kurita, and Nakamasa Inoue. Referring expression comprehension for small objects. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21231–21242. IEEE, 2025.

Zirun Guo, Minjie Hong, Feng Zhang, Kai Jia, and Tao Jin. Thinking with programming vision: Towards a unified view for thinking with images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33467–33476, 2026.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, and Guohai Xu. Deepeyesv2: Toward agentic multimodal model. In *International Conference on Learning Representations*, volume 2026, pp. 114851–114872, 2026.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from Natural Language Expressions. In *European Conference on Computer Vision*, pp. 108–124, 2016. doi: 10.1007/978-3-319-46448-0_7. URL https://mlanthology.org/eccv/2016/hu2016eccv-segmentation/.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, et al. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*, 2026.

Wenqi Jia, Ruifan Li, Pengyue Lin, Fangxiang Feng, Zhanyu Ma, and Xiaojie Wang. Small object, great challenge: A benchmark for small object visual grounding. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026. URL https://openreview.net/forum?id=rWFSj0L615.

Zhengbo Jiao, Yiming Cheng, Yilei Jiang, Kaituo Feng, Rui Huang, Tianyi Jiang, Juanxi Tian, Qunzhong Wang, Tailai Chen, Qianshan Wei, et al. Searcheyes: Towards frontier multimodal deep search intelligence via search world simulation. *arXiv preprint arXiv:2607.05943*, 2026.

Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 1760–1770. IEEE, 2021.

Seonho Kim, Junhyeong Hong, Kyungjae Lee, and Yoonseon Oh. Insight bench: Towards grounded in-situ guidance for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35070–35079, June 2026.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9579–9589. IEEE, 2024.

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

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, et al. Minimax sparse attention. *arXiv preprint arXiv:2606.13392*, 2026b.

Tianming Liang, Qirui Du, Jian-Fang Hu, Haichao Jiang, Zicheng Lin, and Wei-Shi Zheng. Seg-research: Segmentation with interleaved reasoning and external search. *arXiv preprint arXiv:2602.04454*, 2026.

Chang Liu, Xiangtai Li, and Henghui Ding. Referring image editing: Object-level image editing via referring expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13128–13138, June 2024a.

Delong Liu, Haotian Hou, Zhaohui Hou, Zhiyuan Huang, Shihao Han, Mingjie Zhan, Zhicheng Zhao, and Fei Su. Inter-edit: First benchmark for interactive instruction-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37290–37300, June 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024b.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025.

Yan Ma, Linge Du, Xuyang Shen, Shaoxiang Chen, Pengfei Li, Qibing Ren, Lizhuang Ma, Yuchao Dai, Pengfei Liu, and Junjie Yan. One rl to see them all: Visual triple unified reinforcement learning. *arXiv preprint arXiv:2505.18129*, 2025.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *European conference on computer vision*, pp. 792–807. Springer, 2016.

OpenAI. GPT-5.6 System Card. `https://deploymentsafety.openai.com/gpt-5-6`, July 2026a.

OpenAI. GPT-Image-2. `https://developers.openai.com/api/docs/models/gpt-image-2`, 2026b. OpenAI API model documentation.

Giacomo Pacini, Lorenzo Bianchi, Luca Ciampi, Nicola Messina, Giuseppe Amato, and Fabrizio Falchi. Countingdino a training-free pipeline for class-agnostic counting using unsupervised backbones. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 806–815. IEEE, 2026.

Qwen Team. Qwen3.7-Plus: Multimodal agent intelligence, May 2026. URL https://qwen.ai/blog?id=qwen3.7-plus.

Bytedance Seed. Seed1. 8 model card: Towards generalized real-world agency. *arXiv preprint arXiv:2603.20633*, 2026a.

Bytedance Seed. Seed2. 0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026b.

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

Song Tang, Guangquan Jie, Henghui Ding, and Yu-Gang Jiang. Rose: Retrieval-oriented segmentation enhancement. *arXiv preprint arXiv:2604.14147*, 2026a.

Wei Tang, Yanpeng Sun, Qinying Gu, and Zechao Li. Visual position prompt for mllm based visual grounding. *IEEE Transactions on Multimedia*, 2026b.

Hao Wang, Limeng Qiao, Zequn Jie, Zhijian Huang, Chengjian Feng, Qingfang Zheng, Lin Ma, Xiangyuan Lan, and Xiaodan Liang. X-sam: From segment anything to any segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26187–26196, 2026a.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, et al. Locateanything: Fast and high-quality vision-language grounding with parallel box decoding. In *European Conference on Computer Vision*, pp. 336–357. Springer, 2026b.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun MA, and Ziwei Liu. MMSearch-r1: Incentivizing LMMs to search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2456–2487, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.114. URL https://aclanthology.org/2026.acl-long.114/.

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

Wentao Yan, Shengqin Wang, Huichi Zhou, Yihang Chen, Kun Shao, Yuan Xie, and Zhizhong Zhang. Prommsearchagent: A generalizable multimodal search agent trained with process-oriented rewards. *arXiv preprint arXiv:2604.20486*, 2026.

Bokang Yang, Xinyi Sun, Kaituo Feng, Xingping Dong, Dongming Wu, and Xiangyu Yue. From web to pixels: Bringing agentic search into visual perception. *arXiv preprint arXiv:2605.12497*, 2026.

Huanjin Yao, Qixiang Yin, Min Yang, Ziwang Zhao, Yibo Wang, Haotian Luo, Jingyi Zhang, and Jiaxing Huang. Mm-deepresearch: A simple and effective multimodal agentic search baseline. *arXiv preprint arXiv:2603.01050*, 2026.

Zuyao You and Zuxuan Wu. Seg-r1: Segmentation can be surprisingly simple with reinforcement learning. *arXiv preprint arXiv:2506.22624*, 2025.

En Yu, Kangheng Lin, Liang Zhao, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, Xiangyu Zhang, et al. Perception-r1: Pioneering perception policy with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:94827–94853, 2026.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, 2025.

Yu Zeng, Wenxuan Huang, Zhen Fang, Shuang Chen, Yufan Shen, Yishuo Cai, Xiaoman Wang, Zhenfei Yin, Lin Chen, Zehui Chen, et al. Vision-deepresearch benchmark: Rethinking visual and textual search for multimodal large language models. *arXiv preprint arXiv:2602.02185*, 2026.

Junjie Zhang, Chenjia Bai, Haoran He, Wenke Xia, Zhigang Wang, Bin Zhao, Xiu Li, and Xuelong Li. Sam-e: leveraging visual foundation model with sequence imitation for embodied manipulation. *arXiv preprint arXiv:2405.19586*, 2024a.

Lu Zhang, Jiazuo Yu, Haomiao Xiong, Ping Hu, Yunzhi Zhuge, Huchuan Lu, and You He. Finers: Fine-grained reasoning and segmentation of small objects with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:67403–67426, 2026.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024b.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, et al. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*, 2025.

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

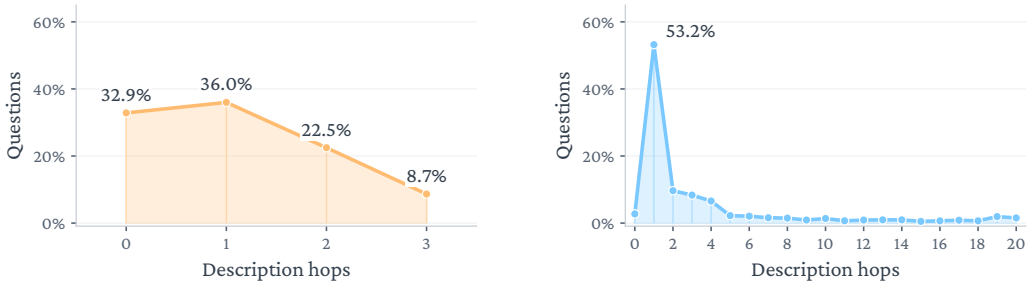
A Appendix

A.1 Dataset Construction

We use Gemini-3.0-Pro as the annotation model for localization because, at the time of data annotation, it was the only model that could successfully complete the task.

Beyond correctness, we also filter SFT trajectories by their behavior patterns. Trajectories in which the annotation model answers directly, or verifies its answer with only a single tool call, typically reflect knowledge that the annotation model already possesses. Since Qwen3VL-4B-Instruct may lack this knowledge, imitating such trajectories would teach it to answer without supporting evidence, encouraging hallucination. We therefore downsample these trajectories, retaining a portion so that the model still learns to answer directly when its own knowledge suffices.

Figure 5 shows the distribution of description hops constructed for anime and real-person queries in the RL data. Anime queries contain substantially fewer hops than real-person queries. Since entity replacement terminates once Seed-1.8 can no longer resolve the query without tools, this difference reflects the limited coverage of anime characters in the model's parametric knowledge. For example, a single hop describing a character as released in a specific game at a particular time with a particular skill is often sufficient to terminate the construction.



(a) Anime questions

(b) Real-person questions

Figure 5: Distribution of description hops in the RL data for anime and real-person questions.

A.2 Training Details

A.2.1 Reward Details

Segmentation fallback. The segmentation reward requires SAM3 to return a mask for the predicted box and point prompts. When SAM3 fails to produce a mask, we do not assign zero reward. Under group-relative advantage estimation, a zero reward is not a neutral outcome: it enters the group mean and shifts the advantages of every other rollout for the same prompt, so a tool failure unrelated to prediction quality would corrupt the learning signal for the rest of the group. We instead treat the predicted box as a rectangular mask and compute the IoU against the ground truth, which preserves the ordering of rollouts by localization quality while remaining a strict lower bound on what SAM3 would have returned. Trajectories scored this way are flagged in the training metadata so that the fallback rate can be monitored.

Spot-the-difference matching. We set $\tau = 0.1$ and $\alpha = 2$.

The candidate threshold τ is deliberately set below the evaluation threshold of 0.5. Using the evaluation threshold for reward computation would assign zero credit to predictions that overlap a ground-truth difference but have not yet reached the evaluation criterion, yielding no intermediate reward signal for approximately localized predictions. Setting $\tau = 0.1$ instead allows such predictions to receive partial credit and provides a denser learning signal during training.

A low matching threshold alone, however, can make weak overlaps overly rewarding. In particular, if every match above τ were counted as a unit true positive, a loosely localized box with $\mathrm{IoU} = 0.1$ would receive the same positive credit as a tightly localized one with $\mathrm{IoU} = 0.9$. This can encourage

overprediction, since emitting additional coarse boxes increases the chance of obtaining low-IoU matches. We therefore weight each matched pair by IoU^{α}. With $\alpha = 2$, for example, matches at IoU 0.15 and 0.9 contribute 0.0225 and 0.81, respectively. Thus, weak matches remain informative but contribute substantially less than well-localized predictions.

This weighting also interacts naturally with the set-level precision–recall trade-off. Let the current soft true-positive mass be S and the current reward be

$$F_1^{\mathrm{soft}} = \frac{2S}{n+m}.$$

Adding one prediction that matches a previously unmatched ground-truth difference with IoU q changes the reward to

$$\frac{2(S+q^{\alpha})}{n+m+1}.$$

The additional prediction improves the reward if and only if

$$q^{\alpha} > \frac{S}{n+m} = \frac{F_1^{\mathrm{soft}}}{2}.$$

Hence, an additional box is beneficial only when the quality of the newly recovered match is sufficiently high relative to the current set-level performance. Unmatched or duplicate predictions increase n without increasing S and therefore strictly decrease the reward. Together, the low threshold τ and the exponent α provide a dense signal for approximate localization while discouraging indiscriminate or overly coarse box proposals.

Matching is greedy rather than optimal. Sorting candidate pairs by IoU and accepting each pair whose predicted box and ground-truth region are both unmatched costs $O(nm \log nm)$ and is stable in practice, whereas optimal bipartite matching would change the reward only in rare configurations where a lower-IoU pairing yields a higher total.

A.3 Evaluation Details

Coordinate formats. Gemini predicts boxes in `yxyx` format, as $(y_{\min}, x_{\min}, y_{\max}, x_{\max})$, whereas the remaining models use `xyxy`, as $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$. All predictions are converted to a common format before scoring.

Decoding and tools. All models are decoded at temperature 0.7. Text search, image search, and text-to-image search are served by Serper; web page content is retrieved with Jina.

Spot-the-difference. Predicted boxes are first restricted to those lying entirely within the left panel, since the task requires differences to be marked there. The remaining boxes are matched to the annotated differences by greedy one-to-one assignment: candidate pairs with IoU at least 0.5 are sorted by IoU in descending order, and each pair is accepted if neither its predicted box nor its annotated region has already been matched. Each accepted pair counts as a true positive. Precision, recall, and F1 are then computed from the resulting counts. We report micro F1, which pools true positives, predictions, and annotated regions across all images before computing F1, and macro F1, which computes F1 per image and averages over images. Micro F1 is treated as the primary metric: with 15 images containing 79 annotated differences, it is the more stable of the two. Each instance is evaluated over eight rollouts, and both metrics are averaged across rollouts.

A.4 Why Token-Mean Reduction Is Ill-Suited for Agentic RL

We consider a group of N trajectories sampled for the same input. Let trajectory i contain L_i valid response tokens and receive a trajectory-level advantage A_i. In GRPO, the advantages are centered

within each group, such that

$$\sum_{i=1}^{N} A_i = 0. \tag{4}$$

For clarity, we first omit PPO clipping and importance ratios, since the issue considered here arises purely from the reduction over tokens and trajectories. Let

$$g_{i,t} = \nabla_\theta \log \pi_\theta(a_{i,t} \mid s_{i,t}) \tag{5}$$

denote the score function of token t in trajectory i, and define its trajectory-average score as

$$\bar{g}_i = \frac{1}{L_i} \sum_{t=1}^{L_i} g_{i,t}. \tag{6}$$

The same argument applies to the clipped PPO surrogate by replacing $g_{i,t}$ with the corresponding token-level surrogate gradient.

Token-mean induces length-biased trajectory weighting. Under token-mean reduction, the policy loss is

$$\mathcal{L}_{\text{tok}} = -\frac{1}{\sum_j L_j} \sum_{i=1}^{N} \sum_{t=1}^{L_i} A_i \log \pi_\theta(a_{i,t} \mid s_{i,t}). \tag{7}$$

Its gradient can be rewritten as

$$\nabla_\theta \mathcal{L}_{\text{tok}} = -\sum_{i=1}^{N} \frac{L_i}{\sum_j L_j} A_i \bar{g}_i. \tag{8}$$

Thus, trajectory i receives effective weight

$$q_i^{\text{tok}} = \frac{L_i}{\sum_j L_j}. \tag{9}$$

In contrast, sequence-mean-token-mean gives

$$\mathcal{L}_{\text{seq}} = -\frac{1}{N} \sum_{i=1}^{N} \frac{1}{L_i} \sum_{t=1}^{L_i} A_i \log \pi_\theta(a_{i,t} \mid s_{i,t}), \tag{10}$$

and hence

$$\nabla_\theta \mathcal{L}_{\text{seq}} = -\frac{1}{N} \sum_i A_i \bar{g}_i. \tag{11}$$

Therefore, token-mean does not merely change the normalization constant. It changes the sampling measure over trajectories from

$$q_i^{\text{seq}} = \frac{1}{N} \tag{12}$$

to the length-biased distribution in Eq. equation 9.

Equivalently, for any trajectory-level quantity f_i,

$$\mathbb{E}_{\text{tok}}[f] = \frac{\sum_i L_i f_i}{\sum_i L_i}, \tag{13}$$

which is the empirical analogue of size-biased sampling,

$$p_{\text{tok}}(\tau) = \frac{L(\tau) p(\tau)}{\mathbb{E}_p[L]}. \tag{14}$$

Thus, long trajectories are systematically overrepresented in the policy update. This distinction is particularly important in agentic RL, where trajectory length is policy-dependent: additional search, repeated inspection, unsuccessful attempts, or redundant interactions all increase L_i. There is generally no reason for such a trajectory to receive a proportionally larger policy update merely because more tokens were generated.

Token-mean breaks the balance induced by centered advantages. Equation equation 4 implies that the total positive and negative advantage masses are exactly balanced:

$$\sum_{A_i>0} A_i = \sum_{A_i<0} |A_i|. \tag{15}$$

Sequence-level averaging preserves this balance at the level of trajectory coefficients. Token-mean instead produces positive and negative coefficient masses

$$M_+^{\text{tok}} = \sum_{A_i>0} L_i A_i, \qquad M_-^{\text{tok}} = \sum_{A_i<0} L_i |A_i|. \tag{16}$$

Their difference is

$$M_+^{\text{tok}} - M_-^{\text{tok}} = \sum_i L_i A_i. \tag{17}$$

Since $\bar{A} = 0$,

$$\begin{aligned} \text{Cov}(L, A) &= \frac{1}{N} \sum_i (L_i - \bar{L})(A_i - \bar{A}) \\ &= \frac{1}{N} \sum_i L_i A_i. \end{aligned} \tag{18}$$

Therefore,

$$\boxed{M_+^{\text{tok}} - M_-^{\text{tok}} = N \ \text{Cov}(L, A)}. \tag{19}$$

Consequently,

$$\text{Cov}(L, A) < 0 \quad \Longrightarrow \quad M_-^{\text{tok}} > M_+^{\text{tok}}. \tag{20}$$

Hence, although GRPO explicitly centers trajectory-level advantages, token-mean reintroduces an imbalance whenever trajectory length correlates with advantage. This effect does not require long trajectories to be negative more frequently. It is sufficient for negative advantages to have larger magnitude at large L, since Eq. equation 19 depends on the joint magnitude of $L_i A_i$, rather than only on the sign probability $P(A_i < 0 \mid L_i)$.

Heavy-tailed trajectory lengths induce gradient concentration. Token-mean also reduces the effective number of trajectories contributing to an update. Using the trajectory weights in Eq. equation 9, define the effective sample size as

$$N_{\text{eff}} = \frac{1}{\sum_i \left(q_i^{\text{tok}}\right)^2}. \tag{21}$$

Substituting Eq. equation 9 gives

$$N_{\text{eff}} = \frac{\left(\sum_i L_i\right)^2}{\sum_i L_i^2}. \tag{22}$$

Let

$$\bar{L} = \frac{1}{N} \sum_i L_i \tag{23}$$

and define the squared coefficient of variation

$$\text{CV}_L^2 = \frac{\frac{1}{N} \sum_i (L_i - \bar{L})^2}{\bar{L}^2}. \tag{24}$$

Since

$$\frac{1}{N} \sum_i L_i^2 = \bar{L}^2 \left(1 + \text{CV}_L^2\right), \tag{25}$$

Eq. equation 22 becomes

$$\boxed{N_{\text{eff}} = \frac{N}{1 + \text{CV}_L^2}}. \tag{26}$$

Thus,

$$\mathrm{CV}_L^2 \uparrow \implies N_{\text{eff}} \downarrow . \tag{27}$$

For heterogeneous or heavy-tailed agentic trajectories, a small number of unusually long rollouts can therefore account for a disproportionately large fraction of the optimization step.

By contrast, sequence-level averaging uses $q_i = 1/N$ and yields

$$N_{\text{eff}} = N. \tag{28}$$

Token-mean therefore converts heavy-tailed trajectory length directly into concentrated optimization influence.

Interaction with negative-advantage updates. Consider a sampled token a with softmax logits z_j and the unclipped policy-gradient loss

$$\ell = -A \log \pi(a). \tag{29}$$

Its gradient with respect to the logits is

$$\frac{\partial \ell}{\partial z_j} = -A \left(\mathbb{1}[j = a] - \pi(j) \right). \tag{30}$$

For $A < 0$, gradient descent decreases the sampled token logit relative to competing tokens. For any $j \neq a$,

$$\Delta(z_a - z_j) = \eta A \left(1 - \pi(a) + \pi(j)\right) < 0. \tag{31}$$

Hence, negative-advantage updates suppress the sampled action and redistribute probability mass toward alternatives. When the sampled action already carries relatively high probability, such updates tend to flatten the local policy distribution.

Combined with Eq. equation 20, token-mean disproportionately magnifies such suppressive updates whenever long trajectories carry more negative advantage mass:

$$L_i \text{ large}, \quad A_i < 0 \implies L_i |A_i| \text{ large}. \tag{32}$$

Thus, even a moderate dependence between trajectory length and advantage can be substantially amplified by linear length weighting.

Summary. With trajectory-level rewards, each trajectory receives a single semantic learning signal A_i. Token-mean nevertheless makes its total optimization influence proportional to the number of generated tokens:

$$\boxed{\text{trajectory influence} \propto L_i |A_i|}. \tag{33}$$

This introduces three undesirable properties:

$$\text{size bias:} \qquad q_i \propto L_i, \tag{34}$$

$$\text{sign imbalance:} \qquad \sum_i L_i A_i = N \,\mathrm{Cov}(L, A), \tag{35}$$

$$\text{gradient concentration:} \qquad N_{\text{eff}} = \frac{N}{1 + \mathrm{CV}_L^2}. \tag{36}$$

All three arise solely from the reduction rule and become more pronounced as trajectory lengths become more heterogeneous. This is particularly undesirable when length is an endogenous consequence of agent behavior rather than a measure of sample importance.

We therefore replace the fixed linear dependence with a tempered family

$$q_i^{(\beta)} = \frac{L_i^\beta}{\sum_j L_j^\beta}, \qquad 0 \leq \beta < 1. \tag{37}$$

The two endpoints correspond to

$$\beta = 0 \quad \Longrightarrow \quad q_i = \frac{1}{N}, \tag{38}$$

and

$$\beta = 1 \quad \Longrightarrow \quad q_i = \frac{L_i}{\sum_j L_j}. \tag{39}$$

For $0 < \beta < 1$, trajectory length remains represented in the optimization objective, but its influence grows sublinearly. The purpose of this tempering is not to assume that long trajectories are intrinsically undesirable, but to remove the unjustified linear coupling between trajectory length and optimization influence introduced by token-mean reduction.

A.5 Prompt and Tool Specifications

We provide the system prompt and tool specifications used by EviRover during training and evaluation. Unless otherwise stated, the same prompt and tool interface are used across all perception tasks.

A.5.1 System Prompt

The following system prompt is used during training and evaluation.

System prompt

```
You are a multimodal agent. Use the provided tools to gather
  information and analyze images, then answer the question.

# Output Format (strict)

Each assistant turn must be exactly one of:

(1) Reasoning then tool call:
<think>
...
</think>
<tool_call>
{"name": "tool_name", "arguments": {...}}
</tool_call>

(2) Reasoning then final answer:
<think>
...
</think>
<answer>
...
</answer>

(3) Tool call without reasoning (when no thinking is needed):
<tool_call>
{"name": "tool_name", "arguments": {...}}
</tool_call>

Rules:
- One tool call per turn.
- The final answer must be inside <answer>...</answer>.
- Coordinates are normalized to 0-1000.
- Bounding boxes use xyxy format: [x_min, y_min, x_max, y_max].
- Points use [x, y].
- For grounding tasks, the answer is a bounding box: <answer>[x_min,
  y_min, x_max, y_max]</answer>.
```

System prompt (continued)

```
- For segmentation tasks, the answer is a JSON object: <answer>{"
  boxes":[x_min,y_min,x_max,y_max],"positive_points":[[x,y],[x,y],[x,
  y]],"negative_points":[[x,y],[x,y],[x,y]]}</answer>.
- For counting tasks, the answer is an integer: <answer>N</answer>.
- For spot-the-difference tasks, the answer is a list of bounding
  boxes on the LEFT panel: <answer>[[x1,y1,x2,y2], [x1,y1,x2,y2],
  ...]</answer>.

# Important Notes About the Target

- The target is often SMALL and easy to miss. Pay attention to fine
  details, small text, subtle attributes, relative position, precise
  boundaries, and local context.
- Use tools strategically to reduce uncertainty.
- Use `crop` to inspect small or unclear regions.
- Use `verify` to check global placement on the full image: whether
  the candidate box is on the correct object instance, in the correct
   relative location, and at the right overall scale.
- Use `verify_part` to check local content and tightness: whether the
   box really contains the target, includes all of it, and is not too
   loose.
- Your bounding box must tightly fit the target | avoid overly large
  boxes. A good bbox should contain the target and little else.
- For spot-the-difference tasks: differences are often SMALL and
  subtle (colors, shapes, missing/added elements). All bounding boxes
   must be on the LEFT panel only (x coordinates in range 0-500).
  Confirm each candidate with `compare_lr` before reporting it, and
  drop the ones where both panels look the same.
- For counting tasks: use crop to inspect specific persons or regions
   more closely when faces are small or occluded.

# Tool Use Policy

First, observe the given images carefully. Then, based on your
  observation and reasoning, use the tools based on what is uncertain
  :
- Use `crop` when the target is hard to see, small, densely
  surrounded by other objects, or requires reading local details.
- Use `compare_lr` for spot-the-difference tasks: when you have a
  candidate difference on the LEFT panel, call `compare_lr` on it to
  see that region side by side with the same region of the RIGHT
  panel. If the two sides look the same, that candidate is NOT a
  difference | drop it and do not include it in your answer.
- Use `verify` when you have a candidate box and want to check
  whether it is in the correct place on the full image and whether
  its scale looks reasonable.
- Use `verify_part` when you want to inspect the exact content inside
   a candidate box and check whether the target is fully included and
   tightly framed.
- Use `python` when you need to perform precise coordinate
  calculations, analyze pixel values, or do numerical computation.
- Use `verify_mask` when you need to verify segmentation quality by
  overlaying a SAM3-generated mask on the image.
- Use `text_search` when you need to search the web for factual
  information to identify the target.
- Use `text_search_image` when you need to find reference images by
  text description.
- Use `image_search` when you need to identify objects or people by
  reverse image search.
- Use `browse` to extract specific details from a webpage when search
   results are insufficient.
```

A.5.2 TOOL SPECIFICATIONS

Table 6 summarizes the available tools and their interfaces. The complete tool schemas are listed below.

Conventions. All spatial coordinates are normalized to $[0, 1000]$ independently of image resolution. Boxes are given in `xyxy` format and points as $[x, y]$. The query image is referenced as `ORIGINAL`; images returned by tools are numbered `IMG_001`, `IMG_002`, ... in the order they appear in the trajectory, and may be passed as inputs to subsequent tool calls.

Table 6: Tools available to the agent. *Visual* indicates whether the tool returns an image to the model.

Tool	Function	Parameters	Visual
`text_search`	Web text search returning titles, snippets, and links	`query`, `top_k` (default 6)	
`text_search_image`	Image search from a text query	`query`, `top_k` (default 5, range 1–10)	✓
`image_search`	Reverse image search over the full image or a specified region	`image_id`, `boxes` (optional), `top_k` (default 3)	
`browse`	Opens a URL and extracts content relevant to a query with a summary model	`url`, `query`	
`crop`	Crops and enlarges a region for closer inspection	`image_id`, `boxes`	✓
`verify`	Renders proposed boxes on the original image	`image_id`, `boxes`	✓
`verify_part`	Crops a proposed region for final confirmation	`image_id`, `boxes`	✓
`verify_mask`	Runs SAM3 on a proposed box with optional point prompts and returns the mask overlay	`image_id`, `boxes`, `positive_points`, `negative_points`	✓
`compare_lr`	Returns corresponding left and right regions side by side	`image_id`, `boxes`	✓
`python`	Executes Python for computation, coordinate transformation, and image analysis	`code`	*

Evidence acquisition. `image_search` accepts an optional box, allowing the agent to isolate a region before searching rather than querying with the full image. We use Qwen3.5-27B as the summary model in `browse`.

Visual inspection and confirmation. `crop` and `verify_part` both return an enlarged region, but differ in intent: the former is used to examine detail, the latter to confirm a candidate answer. `verify` draws proposed boxes on the original image, allowing the agent to assess and refine a localization against its surrounding context. All three may be called repeatedly within a trajectory.

Task-specific prediction. `verify_mask` runs SAM3 locally on the proposed box together with any point prompts and returns the resulting mask as an overlay, so that the agent can inspect and revise its prompts before committing to a final answer. `compare_lr` applies only to spot-the-difference queries: given a box in the left panel, it returns the corresponding regions of both panels side by side, separated by a colored divider. Boxes must lie within the left panel. The `python` tool is provided with the original image and all tool-returned images as preloaded variables, along with `numpy`, `PIL`, `cv2`, `scipy`, `math`, `json`, and `copy`. It returns an image only when the executed code produces one.

Tool schemas. The following schemas form the tool-specification block of the system prompt. Line wrapping and indentation are adjusted for readability.

Tool schemas

```
<tools>
{
  "type": "function",
  "function": {
    "name": "text_search",
    "description": "Search the web for one text query. Use one query
  per call.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {"type": "string", "description": "One concise
  search query."},
        "top_k": {"type": "integer", "description": "Number of
  results, default 10."}
      },
      "required": ["query"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "text_search_image",
    "description": "Search web images by text query and return image
  references with titles, URLs, page URLs, and local paths.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {"type": "string", "description": "One concise image
   search query."},
        "top_k": {"type": "integer", "description": "Number of image
  results, default 5."}
      },
      "required": ["query"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "image_search",
    "description": "Reverse image search: upload the given image (
  optionally cropped to a bbox) and return visually similar images
  with evidence summaries.",
    "parameters": {
      "type": "object",
      "properties": {
        "image_id": {"type": "string", "description": "Image id to
  search, usually ORIGINAL."},
        "image_path": {"type": "string", "description": "Local image
  file path (optional if image_id is given)."},
        "boxes": {"type": "array", "items": {"type": "number"}, "
  minItems": 4, "maxItems": 4, "description": "Optional normalized
  0-1000 bbox to crop before searching."},
        "top_k": {"type": "integer", "description": "Number of
  results, default 3."}
```

Tool schemas (continued)

```
      },
      "required": ["image_id"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "browse",
    "description": "Browse a webpage URL and extract information
  relevant to the query.",
    "parameters": {
      "type": "object",
      "properties": {
        "url": {"type": "string", "description": "The webpage URL."},
        "query": {"type": "string", "description": "The information
  to extract from the page."}
      },
      "required": ["url", "query"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "crop",
    "description": "Crop a region from an image so you can inspect
  details. Returns the cropped and enlarged image.",
    "parameters": {
      "type": "object",
      "properties": {
        "image_id": {"type": "string", "description": "Image id,
  usually ORIGINAL."},
        "boxes": {"type": "array", "items": {"type": "number"}, "
  minItems": 4, "maxItems": 4, "description": "Bbox [x_min, y_min,
  x_max, y_max] in normalized coordinates (0-1000)."}
      },
      "required": ["image_id", "boxes"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "verify",
    "description": "Draw an unlabeled rectangle on the image for bbox
   verification. You can call this multiple times to iteratively
  refine your bbox.",
    "parameters": {
      "type": "object",
      "properties": {
        "image_id": {"type": "string", "description": "Image id,
  usually ORIGINAL."},
        "boxes": {"type": "array", "items": {"type": "number"}, "
  minItems": 4, "maxItems": 4, "description": "Bbox [x_min, y_min,
  x_max, y_max] in normalized coordinates (0-1000)."}
      },
      "required": ["image_id", "boxes"]
```

Tool schemas (continued)

```
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "verify_part",
    "description": "Crop the bbox region for final verification. You
  can call this multiple times to iteratively refine your bbox.",
    "parameters": {
      "type": "object",
      "properties": {
        "image_id": {"type": "string", "description": "Image id,
  usually ORIGINAL."},
        "boxes": {"type": "array", "items": {"type": "number"}, "
  minItems": 4, "maxItems": 4, "description": "Bbox [x_min, y_min,
  x_max, y_max] in normalized coordinates (0-1000)."}
      },
      "required": ["image_id", "boxes"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "verify_mask",
    "description": "Run local SAM3 from the proposed bbox plus
  optional positive/negative points and return a visual overlay of
  the generated mask.",
    "parameters": {
      "type": "object",
      "properties": {
        "image_id": {"type": "string", "description": "Image id,
  usually ORIGINAL."},
        "boxes": {"type": "array", "items": {"type": "number"}, "
  minItems": 4, "maxItems": 4, "description": "Normalized 0-1000 bbox
  ."},
        "positive_points": {"type": "array", "items": {"type": "array
  ", "items": {"type": "number"}, "minItems": 2, "maxItems": 2}},
        "negative_points": {"type": "array", "items": {"type": "array
  ", "items": {"type": "number"}, "minItems": 2, "maxItems": 2}}
      },
      "required": ["image_id", "boxes"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "compare_lr",
    "description": "For spot-the-difference images only. Given ONE
  box on the LEFT panel, return the left region and the SAME region
  of the right panel stitched side by side (separated by a magenta
  divider) at identical size, so the two can be compared directly.
  This is the only way to see both versions of a location up close: a
   location at (x,y) on the left corresponds to (x+500,y) on the
  right, so a single crop containing both would have to span more
  than half the image.",
```

Tool schemas (continued)

```
    "parameters": {
      "type": "object",
      "properties": {
        "image_id": {"type": "string", "description": "Image id,
  usually ORIGINAL."},
        "boxes": {"type": "array", "items": {"type": "number"}, "
  minItems": 4, "maxItems": 4, "description": "Bbox [x_min, y_min,
  x_max, y_max] normalized 0-1000 on the LEFT panel; x_max must be <=
   500."}
      },
      "required": ["image_id", "boxes"]
    }
  }
}

{
  "type": "function",
  "function": {
    "name": "python",
    "description": "Execute Python code to perform calculations,
  coordinate transformations, or image analysis. Use print() for text
   output. Assign a PIL Image to `_result_image` to return a visual
  result. Code with no output (no print, no _result_image) is not
  allowed. Pre-injected variables: `original_image` (PIL Image), `
  IMG_001`..`IMG_00N` (PIL Images from previous tool results), `
  num_images` (int). Libraries available: numpy (as np), PIL.Image,
  PIL.ImageDraw, math, json, cv2, scipy, copy.",
    "parameters": {
      "type": "object",
      "properties": {
        "code": {"type": "string", "description": "Python code to
  execute."}
      },
      "required": ["code"]
    }
  }
}
</tools>
```

A.6 AGENT TRAJECTORIES ON TRAINING AND BENCHMARK EXAMPLES

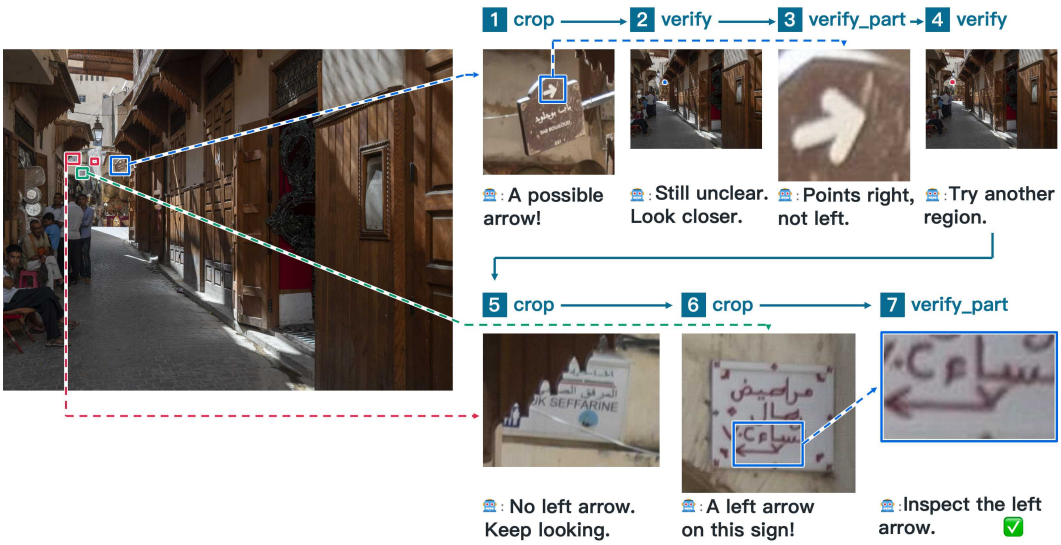


Figure 6: Example localization trajectory generated by EviRover on the training set.

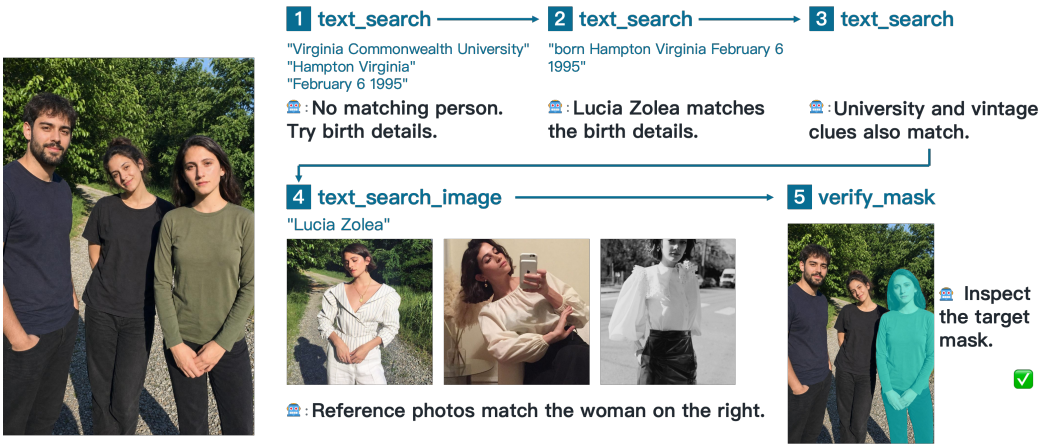


Figure 7: Example segmentation trajectory generated by EviRover on the training set.