

---



# ACTIVE-O3 : Empowering Multimodal Large Language Models with Active Perception via GRPO\*

---

Muzhi Zhu<sup>1,2</sup>, Hao Zhong<sup>1</sup>, Canyu Zhao<sup>1</sup>, Zongze Du<sup>1</sup>, Zheng Huang<sup>1</sup>, Mingyu Liu<sup>1</sup>, Hao Chen<sup>1</sup>, Cheng Zou<sup>2</sup>, Jingdong Chen<sup>2</sup>, Ming Yang<sup>2</sup>, Chunhua Shen<sup>1</sup>

<sup>1</sup> Zhejiang University, China      <sup>2</sup> Ant Group, China

## Abstract

Active vision, also known as active perception, refers to the process of actively selecting where and how to look in order to gather task-relevant information. It is a critical component of efficient perception and decision-making in humans and advanced embodied agents. Recently, the use of Multimodal Large Language Models (MLLMs) as central planning and decision-making modules in robotic systems has gained extensive attention. However, despite the importance of active perception in embodied intelligence, there is little to no exploration of how MLLMs can be equipped with or learn active perception capabilities. In this paper, we first provide a systematic definition of MLLM-based active perception tasks. We point out that the recently proposed GPT-o3 model’s zoom-in search strategy can be regarded as a special case of active perception; however, it still suffers from low search efficiency and inaccurate region selection. To address these issues, we propose **ACTIVE-O3**, a purely reinforcement learning-based training framework built on top of GRPO, designed to equip MLLMs with active perception capabilities. We further establish a comprehensive benchmark suite to evaluate ACTIVE-O3 across both general open-world tasks—such as small-object and dense-object grounding—and domain-specific scenarios, including small object detection in remote sensing and autonomous driving, as well as fine-grained interactive segmentation.

Experimental results demonstrate that ACTIVE-O3 significantly enhances active perception capabilities compared to Qwen-VL2.5-CoT. For example, Figure 1 shows an example of zero-shot reasoning on the  $V^*$  benchmark, where ACTIVE-O3 successfully identifies the number on the traffic light by zooming in on the relevant region, while Qwen2.5 VL fails to do so. Moreover, across all downstream tasks, ACTIVE-O3 consistently improves performance under fixed computational budgets. We hope that our work here can provide a simple codebase and evaluation protocol to facilitate future research on *active perception MLLM*. Our code is released at: <https://github.com/aim-uofa/Active-o3>.

## 1 Introduction

Among the many components of perception, active perception, the process of selective acquisition of sensory information to achieve specific goals, has proven essential for efficient information gathering and decision making in complex environments [1, 2, 3]. For humans, active perception enables tasks such as focusing on relevant details in a cluttered scene or dynamically adjusting viewpoints to better understand ambiguous objects. Similarly, embodied agents, such as autonomous robots, must also make intelligent choices about where to look and how to look to succeed in real-world tasks [4, 5, 6].

---

\*C. Shen is the corresponding author. Part of the work was done when M. Zhu was an intern at Ant Group.

 Tell me the number on the traffic light.

 Qwen2.5 VL

 ACTIVE-O3

The number on the traffic light is 10. 

**<think>**  
 To answer the question about the number on the traffic light, we need to focus on areas where traffic lights are typically located. In this image, there are two potential regions to consider: \*\*Region 1\*\*: This area is near the center of the image, slightly to the left. It includes a traffic light with a red signal illuminated. This is a strong candidate because it's a common place for traffic lights to be located.  
 ...

**</think>**




The number on the police car is 19.

Figure 1: Zero-shot reasoning on the  $V^*$  benchmark. When asked “Tell me the number on the traffic light”, Qwen2.5 VL incorrectly refers to unrelated text. In contrast, ACTIVE-O3 locates and magnifies the precise area on the traffic light, accurately answering 10 through effective spatial localization.

With the recent surge in the capabilities of multimodal large language models (MLLMs) [7, 8, 9, 10], these models are increasingly being integrated into robotic systems [11, 12, 13, 14, 15, 16] as central modules for planning, reasoning, and decision-making. However, despite their impressive generalization and compositionality, current MLLMs are typically passive consumers of visual inputs, relying on static, fixed views of the environment. This contrasts sharply with the dynamic information-seeking behavior that characterizes active perception.

A recent attempt to move towards active perception in MLLMs is the *zoom-in search strategy proposed in GPT-o3*. Although this strategy offers a first step, it remains limited by inefficient region proposals and low target localization accuracy(see Figures 16 in Appendix), especially in dense or fine-grained scenarios. Crucially, there is still a lack of systematic frameworks and evaluation protocols to study and develop active perception capabilities within MLLMs.

In this paper, we proffer **ACTIVE-O3**, a novel reinforcement learning-based training framework built on Group Relative Policy Optimization (GRPO) [17], specifically designed to equip MLLM with active perception skills. We provide a formal task definition for MLLM-based active perception, and construct a comprehensive benchmark suite to evaluate performance across a wide range of tasks—from open-world grounding of small and dense objects, to domain-specific applications such as remote sensing, autonomous driving, and fine-grained segmentation. Our extensive experiments show that ACTIVE-O3 substantially improves search efficiency, accuracy, and downstream task performance under fixed computational budgets, compared to strong baselines such as Qwen-VL2.5-CoT. Furthermore, we observe that, despite not being explicitly trained on reasoning or question answering data, ACTIVE-O3 demonstrates remarkable zero-shot generalization and reasoning capabilities on challenging fine-grained understanding tasks such as  $V^*$  [18] benchmark (see Figure 1).

Our primary contributions are summarized as follows:

- We propose ACTIVE-O3, the first reinforcement learning framework for active perception with MLLMs, formalized via a unified two-stage policy that separates region proposal (sensing) and task execution. Our method combines structured instruction prompts with a dual-form reward design—integrating both task-aware and heuristic feedback—to guide the model toward producing diverse, interpretable, and task-effective region proposals.
- We target two representative yet challenging applications—namely, small/dense object detection and interactive segmentation—and demonstrate that ACTIVE-O3 significantly improves perception quality and task performance across both general-purpose and domain-specific visual tasks.
- We establish a comprehensive benchmark and release all code, prompts, and evaluation protocols to facilitate reproducible research and future exploration in MLLM-based active perception.

## 2 Related Work

### 2.1 Reinforcement Learning for Multimodal Large Language Models

Large Language Models (LLMs) and their multimodal extensions (MLLMs) have achieved impressive progress in language and visual understanding tasks [7, 19, 20, 21, 9, 22, 23, 24, 10, 25]. While supervised learning and instruction tuning remain the dominant approaches for training MLLMs, several limitations persist—such as aligning model behavior with human preferences and handling complex reasoning tasks. Reinforcement Learning (RL) has been introduced as a promising approach to address these challenges. An early and influential example is Reinforcement Learning from Human Feedback (RLHF) [26], which was primarily developed to align model behavior with human preferences and played a central role in the success of ChatGPT [7]. A recent advancement in this direction is Group Relative Policy Optimization (GRPO), proposed in DeepSeek-R1 [17] and DeepSeek-Math [27]. GRPO introduces a novel way to estimate the advantage function using the mean and variance of rewards across a group of responses, guided by verifiable reward signals. This approach eliminates the need for a separate critic model and significantly enhances reasoning capabilities on complex problems. Concurrently, several works [28, 29, 30, 31, 32] have explored applying GRPO to MLLMs. However, these efforts mainly focus on text-centric reasoning or simple visual grounding tasks. In contrast, our work investigates how GRPO can empower MLLMs with active perception abilities, targeting visually grounded reasoning tasks that require spatial understanding and goal-directed attention. Moreover, due to the difficulty of collecting high-quality trajectories for active perception scenarios, reinforcement learning becomes even more essential in this context.

### 2.2 Active Perception

Active perception refers to the paradigm in which an agent intelligently and dynamically controls its own sensors or actions to achieve a specific task or goal. Early foundational work [1, 2, 3]—often termed “active vision” when focusing on visual sensors—demonstrated that by actively controlling parameters such as camera pose or sensor configuration, agents can transform otherwise ill-posed perception problems into well-posed ones. This enables more efficient and effective information gathering for tasks like object recognition, scene understanding, navigation, and manipulation. With the advent of deep reinforcement learning, agents are now able to learn sophisticated sensorimotor policies end-to-end from raw sensory inputs and reward signals [33, 34, 35, 36, 37, 6], without

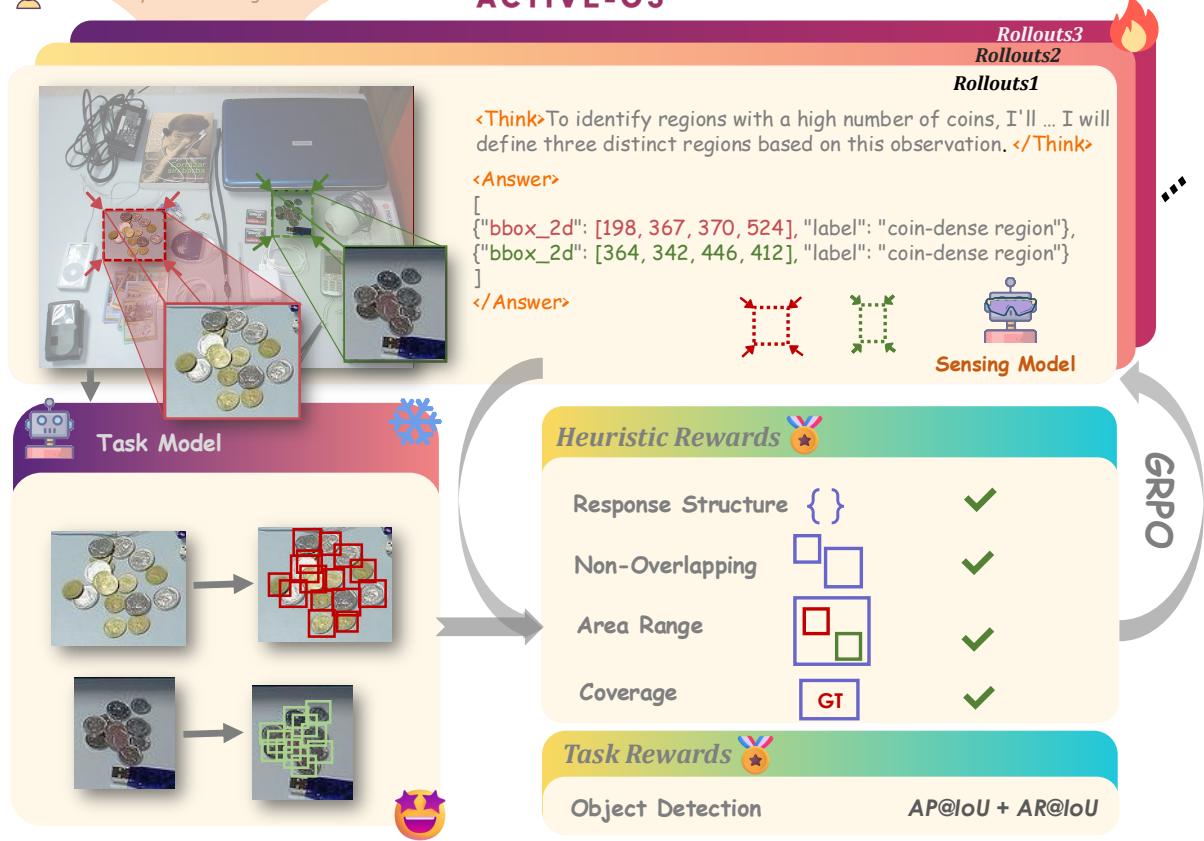


Figure 2: Overview of the proposed Active-O3 framework. Given a multimodal query (e.g., “find all coins”), traditional task models often miss or misidentify target objects due to limited perceptual coverage. Active-O3 enhances perception by allowing the model to actively propose informative subregions (zoom-in regions) based on a learnable sensing policy.

the need for explicit models of environmental uncertainty or information gain. More recently, the principles of active perception have been widely embraced in the field of embodied AI [4, 5, 6, 35], where agents must not only perceive but also interact purposefully with their environments to accomplish complex goals. Meanwhile, there is a clear trend toward integrating Multimodal Large Language Models (MLLMs) as the central reasoning modules—or “brains”—of embodied AI systems [15, 14, 38]. In this context, enabling MLLMs with active perception capabilities is of critical importance for advancing the autonomy and intelligence of such systems. However, despite

rapid progress in MLLM research, active perception remains largely underexplored. Our work aims to bridge this gap, leveraging the strong generalization and reasoning capabilities of MLLMs to tackle challenges in active perception.

### 3 MLLM-based Active Perception: Definition and Analysis

In this section, we provide a formal definition of active perception tasks based on multi-modal large language models (MLLMs) (see Figure 2 for our framework.)

**Modular View of Active Perception.** Consider an embodied agent that receives a human instruction  $\mathcal{I}$  and is required to perform a complex physical-world task. At each time step  $t$ , the agent state is defined as  $s_t = (s_t^{\text{env}}, s_t^{\text{cam}})$ , where  $s_t^{\text{env}}$  describes the environment (e.g., objects and their properties), and  $s_t^{\text{cam}}$  denotes the sensor's pose and viewpoint. A deterministic observation function  $g$  maps the current system state to a visual observation:

$$o_t = g(s_t) + \epsilon_t,$$

where  $\epsilon_t$  is a stochastic noise term.<sup>2</sup> The action space is similarly factorized as  $a_t = (a_t^{\text{env}}, a_t^{\text{cam}}) \in \mathcal{A}$ , where  $a_t^{\text{env}}$  denotes the task-oriented interaction action (e.g., grasping, pointing), and  $a_t^{\text{cam}}$  controls the sensing parameters (e.g., moving or rotating the camera). In order to effectively interact with the environment, the agent must continuously adjust its visual perspective based on current observations to acquire more informative inputs that guide subsequent actions. Active perception can thus be modeled as a coordination between two modules:

- **Task Model  $\mathcal{M}_A$ :** decides how to act on the environment to accomplish external tasks. It takes the current observation  $o_t$  and the task instruction  $\mathcal{I}$  as input, and outputs a task-level action:

$$a_t^{\text{env}} = \mathcal{M}_A(o_t, \mathcal{I})$$

- **Sensing Model  $\mathcal{M}_O$ :** decides how to control perception parameters to improve observation quality. It also takes the current observation and task instruction as input, and outputs a perception action:

$$a_t^{\text{cam}} = \mathcal{M}_O(o_t, \mathcal{I})$$

In our formulation, each action component primarily affects a specific part of the system state:  $a_t^{\text{cam}}$  updates  $s_t^{\text{cam}}$ , and  $a_t^{\text{env}}$  updates  $s_t^{\text{env}}$ , formalized as

$$s_{t+1}^{\text{cam}} = f^{\text{cam}}(s_t^{\text{cam}}, a_t^{\text{cam}}), \quad s_{t+1}^{\text{env}} = f^{\text{env}}(s_t^{\text{env}}, a_t^{\text{env}})$$

where  $f^{\text{cam}}$  and  $f^{\text{env}}$  are deterministic transition functions.

**System Dynamics.** At each time step, the system operates in a closed loop as follows: 1) the sensing model selects a perception action  $a_t^{\text{cam}} = \mathcal{M}_O(o_t^{\text{prev}}, \mathcal{I})$ , which updates the sensor state via  $s_t^{\text{cam}} \leftarrow f^{\text{cam}}(s_t^{\text{cam}}, a_t^{\text{cam}})$ ; 2) the system receives a new observation  $o_t = g(s_t) + \epsilon_t$ ; 3) based on  $o_t$  and  $\mathcal{I}$ , the action model selects an interaction action  $a_t^{\text{env}} = \mathcal{M}_A(o_t, \mathcal{I})$ , which updates the environment state as  $s_{t+1}^{\text{env}} = f^{\text{env}}(s_t^{\text{env}}, a_t^{\text{env}})$ .

**Objective Function** We jointly optimize the action model  $\mathcal{M}_A$  and the sensing model  $\mathcal{M}_O$  to maximize task success while minimizing perceptual cost:

$$\max_{\mathcal{M}_A, \mathcal{M}_O} \mathbb{E} \left[ \sum_{t=1}^T R(s_t, a_t^{\text{env}}) - \lambda \cdot C(a_t^{\text{cam}}) \right]$$

where  $R(s_t, a_t^{\text{env}})$  denotes the task-level reward (e.g., success or progress),  $C(a_t^{\text{cam}})$  is the cost of the sensing action (e.g., viewpoint shift or latency), and  $\lambda$  is a balancing factor.

---

<sup>2</sup>In this paper, we focus on the deterministic mapping  $g(s_t)$  and do not explicitly model observation noise.

## Prompt for ACTIVE-O3 Detection

- “Find up to three different regions in the image that likely contain a high number of ‘**{object}**’.”
- “Even if the ‘**{object}**’ are not clearly visible, infer where they are most likely to appear.”
- “Each region should cover multiple ‘**{object}**’ and include some visual context.”
- “The selected regions should be as distinct as possible, with minimal or no overlap between them.”
- “Return the coordinates in JSON format as: {“bbox\\_2d”: [x1, y1, x2, y2], “label”: “**{object}**-dense region”}.”
- “Explain your reasoning in `<think>...</think>` and output the final result in `<answer>...</answer>`.”
- “Example: `<think>` thinking process here `</think> <answer>` JSON format here `</answer>`”

Figure 3: Prompt for ACTIVE-O3-DET.

**Specialization to 2D Visual Scenes** While our general formulation applies to embodied agents in complex physical environments, such settings are often difficult to deploy and evaluate in a reproducible manner. To facilitate more controlled and fair comparisons, we specialize the problem to a simplified yet representative 2D scenario: active perception over static images.

In this setting, the environment state  $s_t^{\text{env}}$  is a high-resolution static image  $I \in \mathbb{R}^{H \times W \times 3}$ . The sensing action  $a_t^{\text{cam}}$  specifies a rectangular region within  $I$ , parameterized as a bounding box  $(x, y, w, h)$ <sup>3</sup>. The observation  $o_t$  is obtained by cropping the region defined by  $a_t^{\text{cam}}$  from  $I$  and resizing it to a fixed resolution :

$$o_t = \text{ResizeCrop}(I, a_t^{\text{cam}})$$

The task model  $\mathcal{M}_A$  then operates on the selected region to perform downstream functions such as classification, detection, or answering visual questions. This setting preserves the core challenge of active perception—selecting informative views—while simplifying execution and enabling systematic evaluation.

**Objective in 2D Active Perception** A key property of the 2D visual scenario is that the environment state  $s_t^{\text{env}}$  remains static across time (since the interaction action  $a_t^{\text{env}}$  does not change the image). In the 2D setting, we assume a fixed task model  $\mathcal{M}_A$  and focus on learning a sensing policy  $\mathcal{M}_O$  that selects  $K$  informative regions from a static image  $I$  based on an initial observation  $o_{\text{init}}$  and instruction  $\mathcal{I}$ . Here,  $o_{\text{init}}$  represents a low-resolution global view of the image (e.g., a thumbnail), which serves as a coarse prior for guiding the selection of detailed regions. The goal is to maximize overall task performance under a fixed sensing budget. Formally, the optimization objective is:

$$\max_{\mathcal{M}_O} \mathbb{E}_{I, \mathcal{I}} \left[ \sum_{k=1}^K R(\mathcal{M}_A(o_k), \mathcal{I}) \right], \quad \text{where } \begin{cases} \{a_k^{\text{cam}}\}_{k=1}^K = \mathcal{M}_O(o_{\text{init}}, \mathcal{I}) \\ o_k = \text{ResizeCrop}(I, a_k^{\text{cam}}) \end{cases} \quad (1)$$

We treat active perception in this static 2D setting as a single-step decision problem ( $T = 1$ ). As a result, the index  $k$  refers to parallel candidate sensing actions rather than time steps.

## 4 ACTIVE-O3

Building on the formulation in the previous section, we now present ACTIVE-O3, a unified framework for MLLM-driven active perception in vision-language tasks. We target two representative and challenging applications: (1) small object detection/grounding and (2) interactive segmentation. Both tasks require selecting multiple informative regions from an image before performing task-specific action.

---

<sup>3</sup>We focus on axis-aligned rectangular regions and omit rotation for simplicity, although it can be incorporated into the action space.

Given an image  $I$  and instruction  $\mathcal{I}$ , we first generate a global observation  $o_{\text{init}}$  by resizing  $I$ . A shared multi-modal large language model (MLLM) is treated as a unified policy  $\pi$  that generates a textual response  $y$ —containing both intermediate reasoning and action outputs—conditioned on the visual input and instruction, i.e.,  $\pi(y | o, \mathcal{I})$ .

The MLLM is then guided by two prompts:  $\mathcal{I}_O$  for proposing regions, and  $\mathcal{I}_A$  for performing task-specific operations. We extract actionable components from  $y$  via task-specific parsers tailored to each subtask. In this setup:

- **Sensing module:**

$$\mathcal{M}_O(o_{\text{init}}, \mathcal{I}_O) := \text{Parse}_{\text{cam}}(\pi(y | o_{\text{init}}, \mathcal{I}_O))$$

which produces  $K$  candidate perception actions  $\{a_k^{\text{cam}}\}_{k=1}^K$  parsed from the full response.

- **Task module:**

$$\mathcal{M}_A(o_k, \mathcal{I}_A) := \text{Parse}_{\text{env}}(\pi(y | o_k, \mathcal{I}_A))$$

which operates on the  $k$ -th region crop and produces the final task-level output  $a_k^{\text{env}}$ .

In detection-style tasks,  $a_k^{\text{env}}$  shares the same state as  $a_k^{\text{cam}}$ —a bounding box list; the distinction lies in their roles:  $a^{\text{cam}}$  selects candidate regions for further inspection, while  $a^{\text{env}}$  expresses the final localization prediction. We evaluate the alignment between  $a_{1:K}^{\text{env}}$  and the ground truth boxes  $\text{GT}_{\text{box}} = \{(x_1, y_1, x_2, y_2)\}$  using standard detection metrics such as Average Precision (AP) and Average Recall (AR).

#### 4.1 Sensing Policy via MLLM

To enable active perception without additional supervised fine-tuning (SFT), we leverage the instruction-following and reasoning capabilities of MLLMs to implement the sensing policy  $\mathcal{M}_O$  via prompting. This zero-shot setup serves as the necessary starting point for our subsequent reinforcement learning (RL) optimization, which assumes the initial model has non-trivial performance.

We design a task-specific instruction prompt  $\mathcal{I}_O$  (Figure 3) to guide  $\mathcal{M}_O$  in producing meaningful and diverse region proposals  $a_{1:K}^{\text{cam}}$ . The prompt serves three key purposes:

- **Format regularization:** The prompt enforces a structured output format and encourages step-by-step reasoning using tags such as `<think>` and `<answer>`.
- **Task guidance:** It introduces domain-specific priors, such as encouraging the model to:
  - infer likely object locations even when objects are not clearly visible,
  - select spatially diverse and minimally overlapping regions,
  - prefer regions with sufficient surrounding context to support downstream decisions.

These constraints help  $\mathcal{M}_O$  generate interpretable and effective sensing actions that form the basis for active region selection.

#### 4.2 Policy Improvement with GRPO

While the prompt-based sensing policy  $\mathcal{M}_O$  provides a strong initialization, it lacks adaptability to task-specific feedback. A central challenge is that the utility of a sensing action  $a^{\text{cam}}$  cannot be evaluated in isolation—it must be judged by its downstream effect on task performance via  $\mathcal{M}_A$ . This indirect supervision makes it difficult to provide ground-truth labels or optimal targets for training  $\mathcal{M}_O$ . Moreover, we desire  $\pi(y | o_{\text{init}}, \mathcal{I}_O)$  to produce not only candidate regions but also intermediate reasoning traces, which are inherently difficult to supervise through standard imitation learning or SFT. To overcome these challenges, we adopt a reinforcement learning-based approach that enables  $\mathcal{M}_O$ —or more precisely, the underlying language policy  $\pi$ —to improve itself based on task-level reward signals. We apply **GRPO**, a lightweight method that avoids the need for training a separate critic model. Let  $\pi_\theta$  denote the current policy and  $\pi_{\theta_{\text{old}}}$  the behavior policy used to sample  $N$  responses  $\{y_n\}_{n=1}^N$ . Each response contains reasoning and candidate region proposals parsed as  $a_{1:K}^{\text{cam}} = \text{Parse}_{\text{cam}}(y_n)$ . The training objective is:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{I, \mathcal{I}} \left[ \frac{1}{N} \sum_{n=1}^N \min(w_n(\theta) A_n, \text{clip}(w_n(\theta), 1 - \epsilon, 1 + \epsilon) A_n) - \beta \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right] \quad (2)$$

where  $w_n(\theta) = \frac{\pi_\theta(y_n|o_{\text{init}}, \mathcal{I}_O)}{\pi_{\theta_{\text{old}}}(y_n|o_{\text{init}}, \mathcal{I}_O)}$  is the importance ratio between current and behavior policies<sup>4</sup>,  $A_n$  is a normalized reward-based advantage for sample  $n$ , and  $\pi_{\text{ref}}$  is a frozen reference policy (e.g., the base MLLM) used to regularize the update.

$$A_n = \frac{r_n - \text{mean}(\{r_1, \dots, r_N\})}{\text{std}(\{r_1, \dots, r_N\})} \quad (3)$$

### 4.3 Dual-Form Reward Design

The reward function  $r_n$  in Eq. 2 is a crucial component of the GRPO objective. It provides feedback on the quality of the selected regions and the reasoning traces generated by the MLLM. To effectively guide learning under different supervision regimes, we design two types of reward functions: a **task-aware reward** that is coupled with the task model and reflects the success of the final task, and a **heuristic reward** that is decoupled from the task model and based on intrinsic properties of the proposed regions.

**Heuristic Reward.** This reward evaluates a single MLLM response based on task-independent criteria that promote interpretable and spatially meaningful region proposals. It is composed of four components:

- **Format Validity.** The response must conform to a valid structured format. We reward responses that are parseable as JSON with bounding boxes under the `bbox_2d` field and that include both reasoning and answer segments marked by `<think>` and `<answer>` tags.
- **Non-overlapping Proposals.** To encourage spatial diversity, we reward proposals whose pairwise Intersection-over-Union (IoU) falls below a threshold. Responses with any overlapping regions are penalized.
- **Area Range Constraint.** Each bounding box is required to fall within a reasonable size range relative to the image (e.g., 1% to 50%). This avoids overly small or overly large boxes that may be either noisy or uninformative.
- **Coverage-Based Reward.** When ground truth masks or boxes are available, we assess how well the predicted regions align with task-relevant areas. This can include: (i) the proportion of ground-truth mask pixels covered by a region, (ii) the percentage of ground-truth boxes matched by at least one proposal, or (iii) the Dice/IoU between predicted and reference masks.

The final heuristic reward  $\mathcal{R}_{\text{heuristic}}(y)$  is computed as a weighted sum of the above components.

**Task-Aware Reward.** The task-aware reward evaluates the quality of the selected regions based on their downstream utility as measured by task-specific performance metrics. To compute this reward, we execute the task model  $\mathcal{M}_A$  on each selected region  $o_k$ , generating outputs  $a_k^{\text{env}}$ . This requires additional forward passes of  $\mathcal{M}_A$  during training, for which we implement an efficient batched inference system to support parallel evaluation.

The form of the reward depends on the specific task:

- **Detection:**  $\mathcal{M}_A$  returns a set of predicted bounding boxes  $\{b_i\}_{i=1}^K$ , which are compared against ground-truth boxes  $\{b_j\}_{j=1}^J$  using standard metrics such as Average Precision (AP) and Average Recall (AR), based on IoU matching.
- **Interactive Segmentation:**  $\mathcal{M}_A$  predicts interaction points (positive/negative) based on each region, which are fed to a local instance of Segment Anything (SAM) via an internal API. The resulting segmentation mask is compared against ground-truth masks using mean Intersection over Union (mIoU).

This reward provides precise task-aligned feedback and is critical for fine-tuning the sensing policy toward optimal end-task performance. Formal definitions and implementation details are provided in Appendix Sections B and C.

---

<sup>4</sup>In our implementation, we adopt a single-update variant of GRPO where  $\pi_{\theta_{\text{old}}} = \pi_\theta$  during training.

## 5 Experiments



Figure 4: Visualization details of our proposed method on three datasets.

### 5.1 Compared Methods

In this section, we introduce three baseline methods and a variant of ACTIVE-O3 to conduct a comparison. (see Figure 4 for visualization result and more ablation results can be found in Appendix.)

**Grounding DINO (GDINO)** [39]. Grounding DINO is one of the strongest open-world object detection and grounding models available, and it has been widely adopted in the research community [40]. We use it as a non-MLLM-based task model  $\mathcal{M}_A$ , which performs grounding directly on images without requiring additional instruction modules such as  $\mathcal{M}_O$ . Despite its simplicity, it can handle a variety of grounding tasks effectively.

**Qwen2.5-VL-7B** [10]. We adopt Qwen2.5-VL-7B as an MLLM-based task model  $\mathcal{M}_A$ , allowing us to evaluate the performance of a pure MLLM on small object detection and grounding tasks, without any auxiliary guidance from  $\mathcal{M}_O$ .

**Qwen2.5-VL-CoT.** As introduced in Section 4.1, we can formulate a sensing policy  $\mathcal{M}_O$  by prompting an MLLM with a crafted instruction  $\mathcal{I}_O$ . In this baseline, we reuse Qwen2.5-VL-7B both as the policy model  $\mathcal{M}_O$  (to generate action proposals) and as the task model  $\mathcal{M}_A$  (to execute the proposed actions  $a_k^{\text{env}}$ ). This setup tests the effectiveness of using a single MLLM for both sensing and acting.

**ACTIVE-O3 + GDINO.** Although ACTIVE-O3 uses a unified MLLM model  $\pi$  to instantiate both  $\mathcal{M}_A$  and  $\mathcal{M}_O$  during RL training, it allows decoupling at test time. In this variant, we replace the action model  $\mathcal{M}_A$  with Grounding DINO, while retaining the original  $\mathcal{M}_O$  from ACTIVE-O3. This configuration tests whether ACTIVE-O3’s sensing policy can generalize when paired with a stronger, specialized task model.

### 5.2 Open-World Small/Dense Object Grounding

**Dataset.** We build our benchmark on the LVIS dataset [41], known for its rich long-tail vocabulary and abundance of small, densely packed objects. For small object grounding, we use instances under 100 pixels; for dense grounding, we select images with over 15 annotated instances. In both cases,

Table 1: Comparison of grounding and detection performance on **LVIS<sub>small</sub>** and **LVIS<sub>dense</sub>**. Numbers in parentheses denote improvements over the corresponding baseline.

| Method                 | LVIS <sub>small</sub> |                   |                   |                   | LVIS <sub>dense</sub> |                    |                    |                     |
|------------------------|-----------------------|-------------------|-------------------|-------------------|-----------------------|--------------------|--------------------|---------------------|
|                        | AP <sub>s</sub>       | AR <sub>s</sub>   | AP <sub>s</sub>   | AR <sub>s</sub>   | AP <sub>m</sub>       | AR <sub>m</sub>    | AP <sub>l</sub>    | AR <sub>l</sub>     |
| <b>Qwen2.5-VL</b>      | 1.2                   | 1.8               | 1.6               | 2.0               | 9.7                   | 11.0               | 15.0               | 18.7                |
| <b>GDINO</b>           | 0.5                   | 1.2               | 5.7               | 6.3               | 20.2                  | 22.5               | 40.2               | 44.9                |
| <b>Qwen2.5-VL-CoT</b>  | 1.2                   | 2.2               | 2.5               | 3.5               | 11.2                  | 14.4               | 20.3               | 25.8                |
| <b>ACTIVE-O3</b>       | <b>2.2 (+1.0)</b>     | <b>4.6 (+2.8)</b> | <b>4.3 (+2.7)</b> | <b>5.5 (+3.5)</b> | <b>14.3 (+4.6)</b>    | <b>19.7 (+8.7)</b> | <b>20.9 (+5.9)</b> | <b>33.3 (+14.6)</b> |
| <b>ACTIVE-O3+GDINO</b> | <b>1.2 (+0.7)</b>     | <b>2.5 (+1.3)</b> | <b>7.0 (+1.3)</b> | <b>7.9 (+1.6)</b> | <b>25.1 (+4.9)</b>    | <b>29.3 (+6.8)</b> | <b>45.1 (+4.9)</b> | <b>55.9 (+11.0)</b> |

we replace `<object>` in instruction  $\mathcal{I}_O$  with the target category. We sample 10,000 training images and 1,200 validation images, ensuring each category appears no more than three times in the test set for balance.

**Results.** This benchmark is challenging due to small, densely packed objects. As shown in Table 1, both GDINO and Qwen2.5-VL struggle in this setting. In contrast, **ACTIVE-O3** outperforms Qwen2.5-VL and its CoT variant, improving APs/ARs by +1.0/+2.8 on **LVIS<sub>small</sub>**, and by +2.7/+3.5 on **LVIS<sub>dense</sub>**. It also improves AR<sub>l</sub> by +14.6 in large-object retrieval. When paired with GDINO, **ACTIVE-O3+GDINO** achieves 7.0 AP<sub>s</sub> and 7.9 AR<sub>s</sub>, surpassing GDINO by +1.3/+1.6. These results highlight ACTIVE-O3 as a strong and generalizable sensing policy  $\mathcal{M}_O$  for complex, open-world scenarios.

### 5.3 Domain-Specific Small Object Detection

**Dataset.** To evaluate domain generalization, we use the SODA benchmark [42], which includes two large-scale datasets for small object detection: SODA-D (autonomous driving) and SODA-A (aerial imagery). SODA-D has 24,828 traffic images with 278,433 instances in 9 categories, while SODA-A offers 2,513 aerial images with 872,069 instances across 9 classes like vehicles and buildings. These datasets cover diverse and practical small-object detection scenarios.

**Results.** Table 2 shows that **ACTIVE-O3** achieves strong performance across both domains, with 9.2/10.4 AP<sub>s</sub>/AR<sub>s</sub> on **SODA-A** and 15.1/22.0 on **SODA-D**. Despite the larger domain gap in the aerial scenario, **ACTIVE-O3** still outperforms Qwen2.5-VL by +8.5 AP<sub>s</sub> on SODA-A, indicating robust generalization. Performance on SODA-D is even higher, suggesting that our learned sensing policy  $\mathcal{M}_O$  effectively transfers across distinct visual domains.

Table 2: Performance comparison on **SODA-A** and **SODA-D** for small object detection. Numbers in parentheses denote improvement over Qwen2.5-VL.

| Method                | SODA-A            |                    | SODA-D              |                     |
|-----------------------|-------------------|--------------------|---------------------|---------------------|
|                       | AP <sub>s</sub>   | AR <sub>s</sub>    | AP <sub>s</sub>     | AR <sub>s</sub>     |
| <b>Qwen2.5-VL</b>     | 0.7               | 1.5                | 2.1                 | 4.5                 |
| <b>GDINO</b>          | 0.5               | 1.2                | 8.0                 | 8.7                 |
| <b>Qwen2.5-VL-CoT</b> | 3.2               | 5.2                | 7.8                 | 15.2                |
| <b>ACTIVE-O3</b>      | <b>9.2 (+8.5)</b> | <b>10.4 (+8.9)</b> | <b>15.1 (+13.0)</b> | <b>22.0 (+17.5)</b> |

### 5.4 Fine-Grained Interactive Segmentation

**Dataset and Setup.** We use the ThinObjects dataset for its fine-grained segmentation masks and semantic labels, ideal for evaluating zoom-in interactive segmentation. Due to the lack of a strong public task model  $\mathcal{M}_A$ , we use an oracle version that simulates perfect click-based feedback to isolate the impact of our sensing policy  $\mathcal{M}_O$ . Each sample allows up to 3 zoom-in steps, and performance is measured by mean IoU between predicted and ground-truth masks after interaction.

**Effect of Zoom-in Budget.** Figure 5 compares QWEN2.5-VL-COT and ACTIVE-O3 under different zoom-in budgets. While both start at the same initial mIoU, QWEN2.5-VL-COT suffers performance degradation as budget increases, dropping to 0.561 at budget 3. This is due to its tendency to zoom into incorrect regions, compounding errors in subsequent steps. In contrast, ACTIVE-O3 progressively improves to 0.863, demonstrating that our reinforcement learning policy effectively learns to identify and correct errors by selectively zooming in on challenging regions.

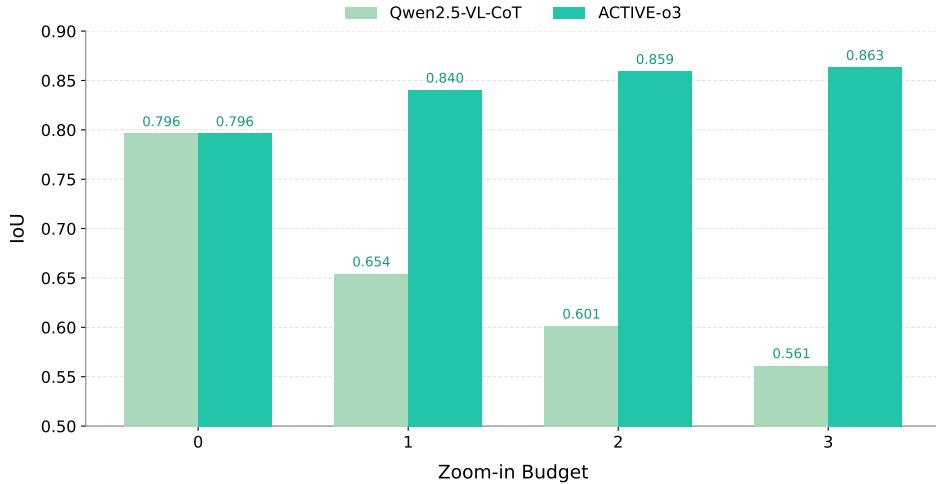


Figure 5: Comparison of segmentation performance (mIoU) under different zoom-in budgets.

## 6 Conclusion

We propose **ACTIVE-O3**, a reinforcement learning framework that empowers MLLMs with active perception via a two-module policy for sensing and action. Trained with task-aware and exploratory rewards, ACTIVE-O3 enables MLLMs to reason about where to look and how to act more effectively. Experiments across open-world grounding, fine-grained segmentation, and domain-specific small object detection show that ACTIVE-O3 consistently improves accuracy and efficiency under limited computational budgets, while generalizing well across diverse domains. We hope that this work encourages further research on active vision with MLLMs.

## Acknowledgement

This work is supported by Ant Group Research Intern Program.

## A Appendix Overview

This appendix provides additional technical details, implementation insights, and extended results to supplement the main paper. It is organized as follows:

- **Section B: Heuristic Reward Formulations**  
Describes the manually designed reward components used to evaluate MLLM outputs, including format validity, spatial overlap, area constraints, and coverage metrics.
- **Section C: Task-Aware Reward Formulation**  
Defines the reward signals computed using downstream task-specific models (e.g., object detection and interactive segmentation).
- **Section D: Discussion: Framework Considerations and Insights**  
Discusses the design choices and considerations behind our MLLM-based active perception framework.
- **Section E: Method Details**  
Discusses implementation details of our active perception system, including MLLM prompt design, reward integration, evaluation metrics, and model configuration.
- **Section F: Ablation Studies**  
Presents ablation experiments on different reward combinations and dataset configurations to understand the contribution of each component.

- **Section G: Qualitative Visualization**

Visual comparisons of model outputs, including correct cases and failure modes, to highlight model behavior under different conditions.

## B Heuristic Reward Formulations

In this section, we detail the heuristic reward functions used to evaluate the quality of region proposals generated by the MLLM. Each reward component is applied to a single MLLM response  $y$ , which typically includes multiple bounding boxes  $\{b_i\}_{i=1}^N$  and optional reasoning traces. The final reward  $\mathcal{R}_{\text{heuristic}}(y)$  is a weighted combination of the components described below.

### B.1 Format Validity Reward $\mathcal{R}_{\text{format}}$

This reward ensures the response adheres to expected syntax and structure. It includes two checks:

- **JSON validity:** the output must be parseable as a list of objects with bounding box fields `bbox_2d`.
- **Response structure:** the output should include the required reasoning and answer format using tags `<think>` and `<answer>`.

$$\mathcal{R}_{\text{format}}(y) = \begin{cases} 1, & \text{if } y \text{ is valid JSON and contains both } <\text{think}>, <\text{answer}> \\ 0, & \text{otherwise} \end{cases}$$

### B.2 Non-overlapping Reward $\mathcal{R}_{\text{no-overlap}}$

This reward penalizes overlapping region proposals to promote spatial diversity:

$$\mathcal{R}_{\text{no-overlap}}(\{b_i\}) = \begin{cases} 1, & \text{if } \text{IoU}(b_i, b_j) \leq \tau, \forall i \neq j \\ 0, & \text{otherwise} \end{cases} \quad \text{with } \tau = 0.3$$

### B.3 Area Range Reward $\mathcal{R}_{\text{area}}$

We encourage region proposals whose areas fall within a reasonable proportion of the image:

$$\text{AreaRatio}(b_i) = \frac{(x_2 - x_1 + 1)(y_2 - y_1 + 1)}{W \cdot H}$$

$$\mathcal{R}_{\text{area}}(\{b_i\}) = \begin{cases} 1, & \text{if } \forall i, r_{\min} \leq \text{AreaRatio}(b_i) \leq r_{\max} \\ 0, & \text{otherwise} \end{cases} \quad \text{with } r_{\min} = 0.01, r_{\max} = 0.5$$

### B.4 Coverage-Based Reward $\mathcal{R}_{\text{coverage}}$

This reward evaluates how well the proposed regions align with task-relevant areas. It is defined in multiple modes:

- **Ground-truth mask coverage:** for binary mask  $M \in \{0, 1\}^{H \times W}$ , we compute the average proportion of mask pixels covered:

$$\text{Coverage}(b_i, M) = \frac{\sum_{(x,y) \in b_i} M(x, y)}{\text{Area}(b_i)}$$

$$\mathcal{R}_{\text{mask}}(\{b_i\}) = \frac{1}{N} \sum_{i=1}^N \mathbf{1} [\text{Coverage}(b_i, M) \geq \theta]$$

- **Ground-truth box coverage:** we count how many ground-truth boxes have at least one matching predicted box ( $\text{IoU} \geq \delta$ ), producing a coverage ratio:

$$\mathcal{R}_{\text{gt-box}} = \frac{\#\text{matched GT boxes}}{\#\text{total GT boxes}}$$

- **Mask-to-mask alignment:** if both predicted and ground-truth masks are available, we compute Dice or IoU over the merged regions.

The final coverage reward can be defined as a soft combination of the above modes when applicable.

## B.5 Overall Heuristic Reward

We define the total heuristic reward as a weighted sum of the components:

$$\mathcal{R}_{\text{heuristic}}(y) = \lambda_1 \mathcal{R}_{\text{format}} + \lambda_2 \mathcal{R}_{\text{no-overlap}} + \lambda_3 \mathcal{R}_{\text{area}} + \lambda_4 \mathcal{R}_{\text{coverage}}$$

where  $\lambda_i$  are all set to 1.

## C Task-Aware Reward Formulation

We provide task-specific definitions of the reward signal computed from the outputs of the task model  $\mathcal{M}_A$ .

**Object Detection.** Let  $\hat{B} = \{b_i\}_{i=1}^K$  be the predicted bounding boxes and  $B^* = \{b_j\}_{j=1}^J$  be the ground-truth boxes. The reward is computed using standard detection metrics:

$$\mathcal{R}_{\text{detect}} = \text{AP}@\text{IoU}=0.5 + \text{AR}@\text{IoU}=0.5$$

**Interactive Segmentation.** Let  $\hat{M}$  be the predicted mask returned by the SAM [43] API and  $M^*$  be the ground-truth mask. The segmentation reward is defined as:

$$\mathcal{R}_{\text{seg}} = \text{mIoU}(\hat{M}, M^*) = \frac{|\hat{M} \cap M^*|}{|\hat{M} \cup M^*|}$$

We generate the SAM prediction using positive and negative points inferred by  $\mathcal{M}_A$ .

## D Discussion: Framework Considerations and Insights

In this section, we provide further insights into the design of our MLLM-based active perception framework, building upon the main formulation introduced in Section 3 of main paper. The following remarks highlight critical architectural choices and theoretical simplifications made to improve performance, efficiency, and generalization.

**Remark D.1 (MLLM-Driven Action and Sensing Modules).** Unlike prior approaches that use specialist models for each module, we adopt a single multi-modal large language model (MLLM) to jointly handle both action and sensing. This design offers several advantages. First, MLLMs exhibit strong capabilities in following natural language instructions and generalizing to open-ended semantic goals. Second, they can leverage rich contextual information, including long-term observation history, to make more informed and coherent decisions. Finally, in addition to predicting  $a_t^{\text{env}}$  and  $a_t^{\text{cam}}$ , MLLMs can also generate intermediate reasoning steps, which not only enhance interpretability but have also been shown to improve task performance in prior work (e.g., chain-of-thought prompting).

**Remark D.2 (Optimization Strategy).** In principle, the action model  $\mathcal{M}_A$  and the sensing model  $\mathcal{M}_O$  can be jointly optimized. However, this requires  $\mathcal{M}_A$  to already possess sufficient baseline capability. A common alternative is to perform staged or iterative optimization, where one alternately updates  $\mathcal{M}_A$  and  $\mathcal{M}_O$  in a bootstrapping manner. In this work, we assume access to a reasonably strong  $\mathcal{M}_A$  and focus on optimizing  $\mathcal{M}_O$  accordingly, since our goal is to investigate how to equip MLLMs with effective active perception strategies. Furthermore, to simplify the problem, we reformulate the perceptual cost term as a fixed sensing budget. That is, under a given number of allowed sensing actions, the objective becomes maximizing task reward. This is the setup we adopt in our experiments.

**Remark D.3 (2D Setting as a Single-Step Active Perception Problem).** A key property of the 2D visual scenario is that the environment state  $s^{\text{env}}$  remains static across time (since the interaction action  $a_t^{\text{env}}$  does not change the image). As a result, the task reduces to a single-step decision problem ( $T = 1$ ), and the agent's objective becomes repeatedly selecting an initial sensing action  $a_0^{\text{cam}}$ . This

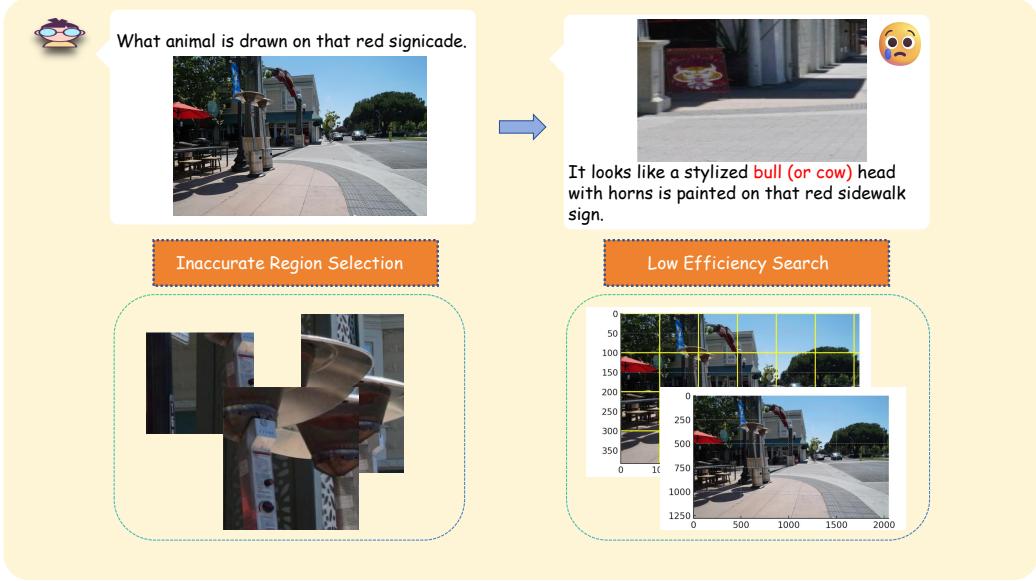


Figure 6: A failure case of GPT-o3 in answering the question: What animal is drawn on that red sign? The reasoning trajectory reveals two key limitations: inaccurate region selection (left), and inefficient, near-exhaustive search patterns (right).

reframing allows for a significantly more efficient implementation: multiple candidate sensing actions can be evaluated in parallel, enabling broader exploration of the observation space without relying on sequential interaction. In this sense, 2D active perception can be viewed as a parallelized search over viewpoints within a fixed scene.

**Remark D.4 (GPT-o3 vs. ACTIVE-O3).** The zoom-in search strategy used in GPT-o3 can be seen as a special case of the active perception framework defined in this paper. However, it suffers from two major limitations. First, its search process is purely sequential—only one region can be selected and zoomed in at a time—which leads to low efficiency. Second, its region selection is often inaccurate, resulting in unnecessary zooms and missed critical areas. In contrast, ACTIVE-O3 enables parallel selection of multiple candidate regions, improving search coverage and efficiency. Moreover, by leveraging the reasoning capability of MLLMs and optimizing the sensing policy  $\mathcal{M}_O$  through reinforcement learning, ACTIVE-O3 is able to identify more informative regions under a fixed sensing budget.

We further illustrate the limitations of GPT-o3 with a failure case shown in Figure 6. The task is to answer the question “What animal is drawn on that red sign?”. The correct answer is tiger, as a stylized tiger face is clearly visible on the red sidewalk sign.

However, GPT-o3 fails to accurately locate the relevant region. It initially zooms into irrelevant parts of the image—such as metallic structures and background textures—due to its limited context and short-horizon planning. As shown in the left panel of Figure 6, the chosen regions completely miss the actual sign.

Moreover, as highlighted on the right side, GPT-o3’s sensing process becomes inefficient, closely resembling exhaustive grid-based search in some cases. This leads to redundant actions and poor use of the limited zoom-in budget. In contrast, ACTIVE-O3 identifies more informative regions early by reasoning over spatial layout and task context, significantly improving efficiency and accuracy.

## D.1 Limitations and Future Work

Despite the promising results, our framework has several limitations that open avenues for future research. (see Figure 16).

First, the domain gap remains a challenge, particularly for specialized domains such as remote sensing. Current MLLMs may struggle to accurately identify domain-specific categories (e.g., windmills,

## Prompt for ACTIVE-O3 Detection

- "Find up to three different regions in the image that likely contain a high number of '{object}'."
- "Even if the '{object}' are not clearly visible, infer where they are most likely to appear."
- "Each region should cover multiple '{object}' and include some visual context."
- "The selected regions should be as distinct as possible, with minimal or no overlap between them."
- "Return the coordinates in JSON format as: {"bbox\_2d": [x1, y1, x2, y2], "label": "{object}-dense region"}."
- "Explain your reasoning in <think>...</think> and output the final result in <answer>...</answer>."
- "Example: <think> thinking process here </think> <answer> JSON format here </answer>"

Figure 7: Prompt for ACTIVE-O3-DET.

## Prompt for ACTIVE-O3 Segmentation

- "Identify exactly three distinct regions in the image that illustrate segmentation inaccuracies in the translucent green mask for the '{object}'."
- "The selected regions should be as distinct as possible, with minimal or no overlap between them."
- "Check whether the mask accurately covers the '{object}', meaning it should fully include the object without significant over-segmentation (mask extends into background) or under-segmentation (parts of the object are not covered)."
- "Each region should represent a clear segmentation mistake and include enough surrounding context for verification."
- "Return the results in JSON format as: {"bbox\_2d": [x1, y1, x2, y2], "label": "{object} segmentation error"}."
- "Explain your reasoning in <think>...</think> and output the final result in <answer>...</answer>."
- "Example: <think> reasoning process here </think> <answer> JSON format here </answer>"

Figure 8: Prompt for ACTIVE-O3-Seg.

storage tanks), which can lead to inaccurate task-aware reward estimation due to the limited capability of the task model.

Second, the current action space is constrained. Our framework only allows zooming into three target regions per step. However, certain applications may require more flexible control, such as selecting a larger number of regions or introducing transformations like rotation—especially relevant for tasks like OCR, though less critical for tasks such as grounding.

Third, the input to the sensing model is limited to the current observation. In practice, incorporating a memory mechanism to store past actions and observations could enable more informed decision-making. This extension may support more sophisticated strategies, such as trajectory-level planning, long-term search, and rollback operations.

Addressing these limitations could further improve the adaptability, generalization, and decision quality of the proposed sensing policy in more complex or specialized scenarios.

## E Method Details

### E.1 Prompt Design

In this section, we provide the prompts used to guide the MLLM in both detection (Figure 3) and segmentation (Figure 8) tasks as the sensing policy  $\mathcal{M}_O$ . The prompts are designed to elicit specific behaviors from the model, ensuring that it generates appropriate region proposals and reasoning. For the task model  $\mathcal{M}_A$ , we use a simple instruction to ask the model to perform the task (Figure 9).

## Prompt for Task Model

- "Please find all instances of ‘**{object}**’ in the image and return the bounding box coordinates in JSON format."

Figure 9: Prompt for the task model  $\mathcal{M}_A$ .

### E.2 Implementation Details

We use Qwen2.5-VL-7B-Instruct as the shared policy backbone  $\pi_\theta$ . All experiments are conducted using GRPO with KL regularization coefficient  $\beta = 0.04$ , group size 8, and a learning rate of  $1e-6$  using the AdamW optimizer with weight decay 0.01.

Training is performed on 8 GPUs with 80–90GB memory each, using bf16 precision, a per-device batch size of 1, gradient accumulation of 1, and gradient checkpointing enabled. Training is performed with DeepSpeed ZeRO-3 for memory efficiency. Each experiment typically completes within 24 hours. For the sensing model  $\mathcal{M}_O$ , we resize the input image such that the shorter side is 1024 pixels, while preserving the original aspect ratio. For the task model  $\mathcal{M}_A$ , all images are resized to a fixed resolution of  $840 \times 840$ . For Grounding DINO, we follow the official preprocessing pipeline provided by the authors.

### E.3 Datasets Details

**LVIS.** We construct our benchmark for open-world small and dense object grounding based on the LVIS [41] dataset, which offers the richest long-tail object vocabulary and the highest prevalence of small and densely packed instances among existing segmentation datasets. To assess small object grounding, we identify all instances with an area less than 100 pixels and retain their corresponding categories as test queries. For dense object grounding, we select images that contain more than 15 annotated instances and treat all instance categories within such images as query targets. In both cases, we replace the placeholder `<object>` in the original instruction  $\mathcal{I}_O$  with the chosen category name. We sample 10,000 training images from the LVIS training set using this strategy, and 1,200 images from the validation set for evaluation. During test set construction, we ensure that each category appears at most three times to promote category balance. We adopt standard COCO evaluation metrics using the official COCO API. Specifically, we report average precision (AP) across IoU thresholds from 0.5 to 0.95 (in 0.05 increments), as well as AP for small ( $AP_s$ ), medium ( $AP_m$ ), and large ( $AP_l$ ) object sizes.

**SODA.** To further evaluate the generalization of our framework in specialized visual domains, we adopt the SODA [42] benchmark, which includes two large-scale datasets designed for small object detection: **SODA-D** (autonomous driving) and **SODA-A** (aerial imagery). SODA-D contains 24,828 traffic images with 278,433 annotated instances across nine traffic-related categories. SODA-A includes 2,513 high-resolution aerial images with 872,069 object instances across nine categories such as vehicles and buildings. These datasets present a wide range of realistic and challenging small-object detection scenarios. During training, we randomly select 1,000 images from each dataset as the training set. For SODA-A, whose annotations are originally provided as polygons, we convert them into bounding boxes to serve as ground truth for training and evaluation. Due to the significant domain shift compared to LVIS, direct use of standard evaluation settings (e.g., COCO-style AP at IoU 0.5–0.95) leads to very low scores and poor comparability. To better capture performance under such domain-specific conditions, we lower the IoU threshold to 0.1 when computing detection metrics. This adjustment allows a fairer evaluation of the model’s generalization ability in these more challenging domains.

**ThinObjects.** We adopt the ThinObjects [44] dataset for this task, as it provides both semantic annotations and high-quality, fine-grained segmentation masks, making it suitable for evaluating interactive segmentation under zoom-in conditions. One core challenge is the lack of a robust existing task model  $\mathcal{M}_A$  for click-based interactive segmentation. To focus on evaluating the effectiveness of our method as a sensing policy  $\mathcal{M}_O$ , we construct an oracle variant of  $\mathcal{M}_A$  as a proxy. This oracle

Table 3: Impact of training data combinations on small object detection performance. We report  $\text{AP}_s/\text{AR}_s$  on SODA-A and SODA-D.

| Training Set      | SODA-A        |               | SODA-D        |               |
|-------------------|---------------|---------------|---------------|---------------|
|                   | $\text{AP}_s$ | $\text{AR}_s$ | $\text{AP}_s$ | $\text{AR}_s$ |
| SODA-A            | 3.7           | 7.5           | —             | —             |
| SODA-D            | —             | —             | 11.4          | 18.9          |
| LVIS + SODA-A     | 6.4           | 8.8           | 14.0          | 17.9          |
| LVIS + SODA-A + D | <b>9.2</b>    | <b>10.4</b>   | <b>15.1</b>   | <b>22.0</b>   |

Table 4: Ablation study on reward design. Comparison of task reward, heuristic reward, and their combination across different object sizes (small, medium, large). Metrics are AP and AR.

| Reward Type      | $\text{AP}_s$ | $\text{AR}_s$ | $\text{AP}_m$ | $\text{AR}_m$ | $\text{AP}_l$ | $\text{AR}_l$ |
|------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Task Reward      | 3.6           | 5.0           | 12.1          | 15.7          | 16.4          | 25.2          |
| Heuristic Reward | 3.0           | 4.2           | 9.7           | 13.8          | 13.2          | 21.7          |
| Combined Reward  | <b>4.4</b>    | <b>5.8</b>    | <b>15.4</b>   | <b>20.2</b>   | <b>19.1</b>   | <b>27.4</b>   |

simulates perfect feedback during interaction. We set a maximum budget of 3 zoom-in steps per sample. The final performance is evaluated using the mean Intersection over Union (mIoU) between the predicted and ground-truth masks after the interaction sequence.

## F Ablation Studies

**Training Data Combination.** Table 3 presents the effect of different training data combinations on small object detection performance, evaluated on SODA-A and SODA-D. When incorporating LVIS into the training set, the performance improves significantly across both domains. For example, adding LVIS to SODA-A yields a +2.7  $\text{AP}_s$  and +1.3  $\text{AR}_s$  gain on SODA-A, and also enables reasonable generalization to SODA-D. Finally, using the full combination of LVIS, SODA-A, and SODA-D leads to the best overall performance, achieving 9.2/10.4 on SODA-A and 15.1/22.0 on SODA-D. These results demonstrate that ACTIVE-O3 serves as a general and flexible framework capable of leveraging heterogeneous domain-specific datasets to learn a unified sensing policy  $\mathcal{M}_O$ . By incorporating diverse training sources such as LVIS, SODA-A, and SODA-D, ACTIVE-O3 is able to generalize effectively across multiple domains, highlighting its scalability and adaptability in open-world scenarios.

**Reward Design.** As mentioned in Section 4, we adopt a dual-form reward design that combines heuristic and task-aware rewards. To evaluate the impact of each component, we conduct an ablation study on the reward design. As shown in Table 4, the combined reward achieves the best performance across all object sizes, especially for small objects ( $\text{AP}_s$ : 4.4,  $\text{AR}_s$ : 5.8). Compared to using only task or heuristic rewards, the combination leads to consistent improvements, indicating that it effectively balances exploration (via heuristics) and task-driven optimization. This validates the effectiveness of our dual-form reward design in guiding better policy learning.

## G Qualitative Visualization

### G.1 Zero-shot Transfer on $V^*$ Benchmark

We demonstrate that ACTIVE-O3 is capable of zero-shot transfer to fine-grained VQA tasks, such as those in the  $V^*$  [18] benchmark. By learning effective reasoning and search strategies through reinforcement learning on small object detection tasks, ACTIVE-O3 generalizes well to previously unseen tasks. We highlight several challenging cases involving OCR (Figures 10, 1) and attribute recognition (Figures 11, 12) where base models struggle. In contrast, ACTIVE-O3 can successfully complete the task by leveraging its ability to reason and zoom in adaptively.

## G.2 Small Object Detection on SODA-A and SODA-D

Figure 13 presents qualitative results of ACTIVE-O3 on the SODA-A and SODA-D datasets. Compared with several baselines, ACTIVE-O3 consistently selects more relevant regions to zoom into, leading to improved detection performance on small objects. These results demonstrate that our sensing model can effectively identify task-critical regions and enhance performance in both aerial and driving scenarios.

## G.3 Small Object Detection on LVIS

We further evaluate ACTIVE-O3 on the LVIS dataset and visualize its performance in Figure 14. Compared with alternative methods, ACTIVE-O3 demonstrates superior ability in selecting semantically meaningful regions for zoom-in, resulting in improved detection of small and rare object instances. These examples validate the general applicability of our approach to long-tail and fine-grained detection benchmarks.

## G.4 Interactive Segmentation on ThinObjects

We show in Figure 15 the performance of ACTIVE-O3 on the ThinObjects dataset for interactive segmentation. Our sensing model effectively identifies and focuses on regions with poor initial segmentation quality, enabling more precise refinement. These results highlight the utility of ACTIVE-O3 beyond detection, extending to segmentation tasks that require spatial reasoning and adaptive focus.

 Tell me the number on the police car.

 **Qwen2.5 VL**

The image does not clearly show the number on the police car. The police car is visible in the foreground, but the details of the number plate are not discernible due to the angle and distance from which the photo was taken. 

 **ACTIVE-O3**

**<think>**  
To find the number on the police car, I need to look for any part of the image where a police car is visible. Given the context, it's likely that the police car is located near the center of the image, possibly in front of the large red building. The number on the police car would typically be displayed on its side or rear, so I'll focus on the area around the police car for potential answers.  
**</think>**

The number on the police car is 102.

Figure 10: Zero-shot reasoning on the  $V^*$  benchmark (Example 2). Given the question “Tell me the number on the police car”, the baseline model (Qwen2.5 VL) fails to locate the relevant visual evidence due to limited resolution and reasoning capability. In contrast, our method (ACTIVE-O3) identifies the appropriate region through contextual reasoning and zoom-in selection. It successfully locates the number 102 on the police car, demonstrating strong spatial inference and fine-grained visual understanding.



What is the color of the van?




Qwen2.5 VL

The image does not show a van. Instead, it features a river with several boats, including a large white and red ferry in the foreground. The background includes a cityscape with a prominent cathedral and other buildings. If you have another image or context related to a van, please provide that information!


ACTIVE-O3

**<think>**

To determine the color of the van, we need to look for any vehicles in the image. Since there are no vans visible in the provided image, we can infer that the van's color might be present in the background or on the left side of the image where there are some buildings and structures. However, without a clear view of a van, we can only make educated guesses based on the context. Here are three regions that might contain relevant visual cues.

**</think>**



The van in the image is red.

Figure 11: Zero-shot reasoning on the  $V^*$  benchmark (Example 3). For the question “What is the color of the van?”, the baseline model (Qwen2.5 VL) fails to detect the presence of the van and incorrectly claims that no such object is visible. In contrast, ACTIVE-O3 accurately identifies the small red van in the background and correctly answers red, demonstrating its ability to localize and reason over subtle visual cues that are easily overlooked.

What is the color of the watchband?



The watchband in the image appears to be black.

**ACTIVE-03**

<think>

To determine the color of the watchband, we need to look for a person who might be wearing a watch on their wrist. In this image, there are two people visible, but only one appears to have a watchband that could be seen. The woman in the foreground seems to be wearing a watch on her left wrist, which is partially visible. Therefore, focusing on the woman in the foreground will provide the necessary information.

</think>



The watchband in the picture appears to be purple.

Figure 12: Zero-shot reasoning on the  $V^*$  benchmark (Example 4). Given the question “What is the color of the watchband?”, baseline predictions are inconsistent. ACTIVE-03 focuses on the wrist of the foreground figure, providing the accurate answer (purple) by effectively zooming in on the fine-grained detail.

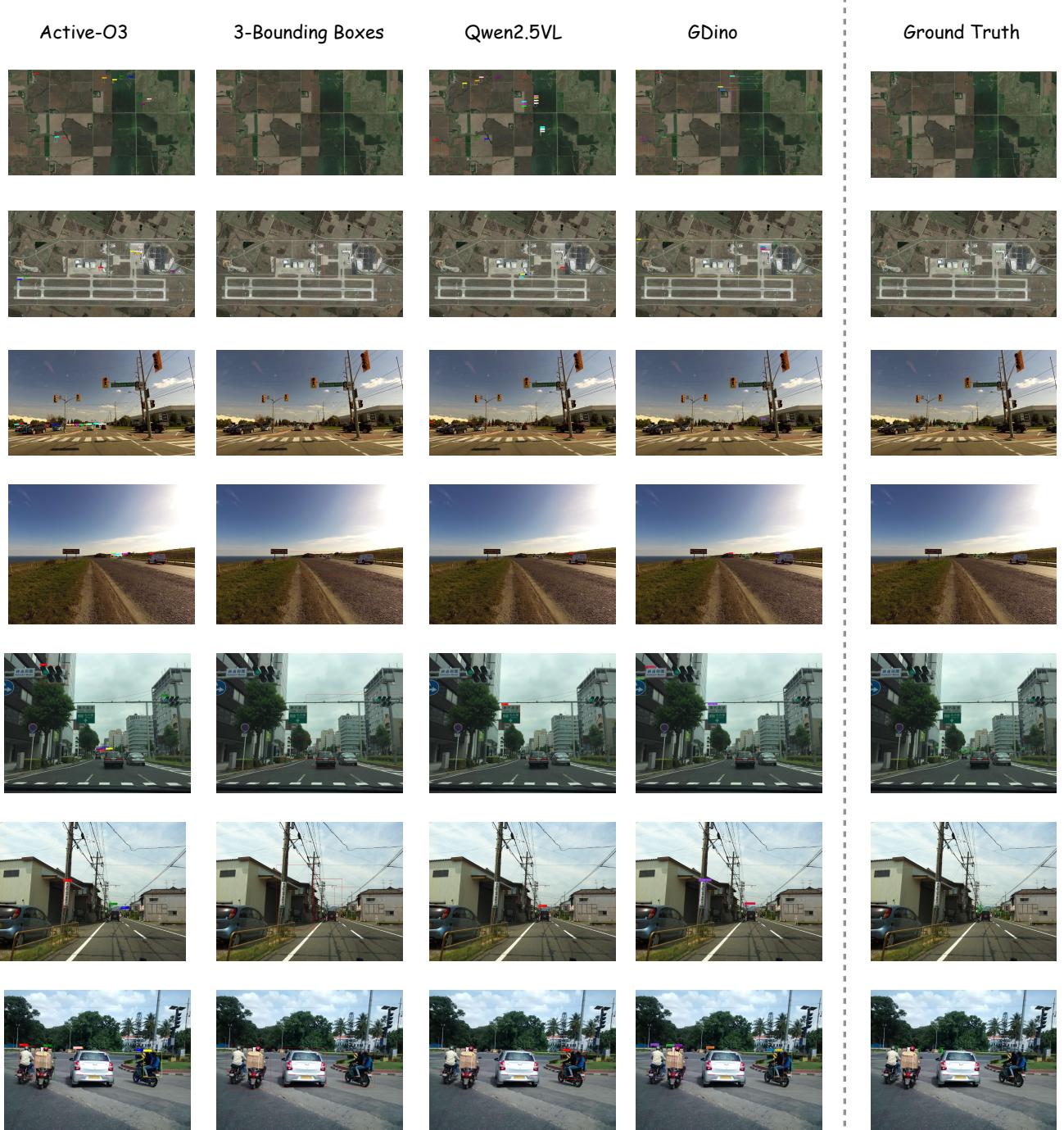


Figure 13: Visualization of Small Object Detection results on SODA-A and SODA-D datasets. Each row shows a different example from either SODA-A (top two rows) or SODA-D (remaining rows). The second column illustrates the candidate regions selected by our sensing model. Zoom in for better visibility of fine details and small objects.

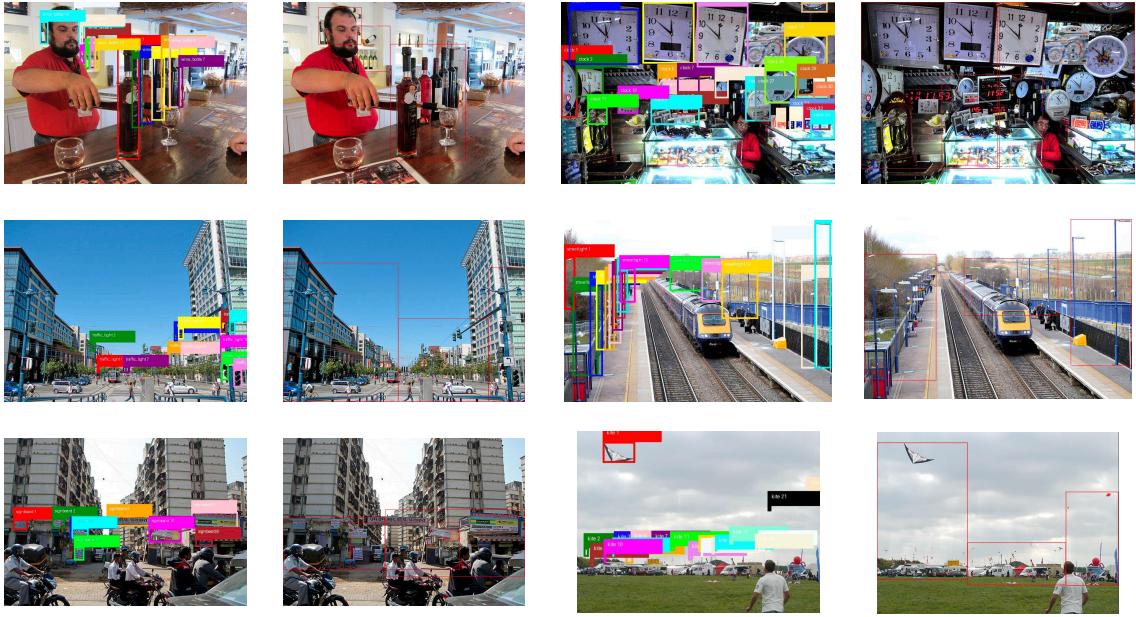


Figure 14: Visualization of object detection results on various scenes from the LVIS dataset. The left column shows the candidate regions selected by our sensing model.

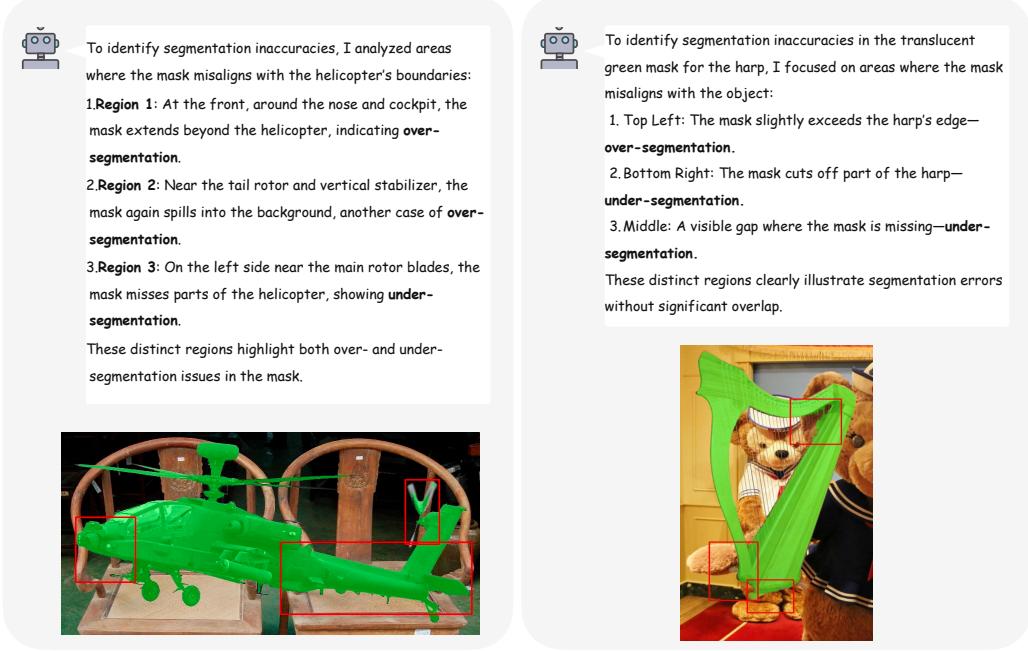
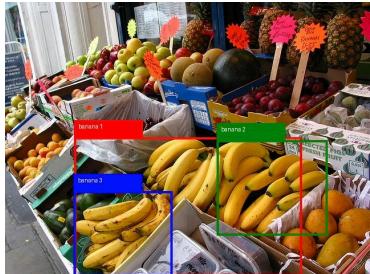


Figure 15: Interactive segmentation analysis on ThinObjects. ACTIVE-O3 identifies specific regions with segmentation inaccuracies by reasoning over visual cues. The left example (helicopter) reveals both over-segmentation (e.g., mask spilling beyond the nose and tail) and under-segmentation (e.g., missing rotor parts). The right example (harp) similarly highlights areas where the mask exceeds or misses the object boundary. These results demonstrate ACTIVE-O3’s capability to localize fine-grained segmentation errors, facilitating efficient and targeted mask refinement.

LVIS



SODA-A



Figure 16: Failure cases. **Left (LVIS)**: When objects are densely packed, the model fails to distinguish between them, resulting in inaccurate segmentation. **Right (SODA-A)**: For small objects in aerial images, domain gap issues lead to poor localization—even if the object is roughly boxed, the model can fail to identify it correctly.

## References

- [1] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International journal of computer vision*, 1:333–356, 1988. [1](#), [3](#)
- [2] Dana H Ballard. Animate vision. *Artificial intelligence*, 48(1):57–86, 1991. [1](#), [3](#)
- [3] Peter Whaite and Frank P Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997. [1](#), [3](#)
- [4] Ermano Arruda, Jeremy Wyatt, and Marek Kopicki. Active vision for dexterous grasping of novel objects. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2881–2888. IEEE, 2016. [1](#), [4](#)
- [5] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural modular control for embodied question answering. In *Conference on robot learning*, pages 53–62. PMLR, 2018. [1](#), [4](#)
- [6] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020. [1](#), [3](#), [4](#)
- [7] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. [2](#), [3](#)
- [8] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. [2](#)
- [9] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. [2](#), [3](#)
- [10] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. [2](#), [3](#), [9](#)
- [11] Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinqiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, et al. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. *arXiv preprint arXiv:2502.13143*, 2025. [2](#)
- [12] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. *arXiv preprint arXiv:2502.13130*, 2025. [2](#)
- [13] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. [2](#)
- [14] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. [2](#), [4](#)
- [15] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. [2](#), [4](#)
- [16] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. \pi\_0.5: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. [2](#)
- [17] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. [3](#)
- [18] 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. [3](#), [17](#)

- [19] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3
- [20] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 3
- [21] Xiao Bi, Deli Chen, Guanting Chen, Shanhua Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. 3
- [22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 3
- [23] Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024. 3
- [24] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhusuo Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024. 3
- [25] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024. 3
- [26] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. 3
- [27] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Huawei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3
- [28] Jiaxing Zhao, Xihan Wei, and Liefeng Bo. R1-omni: Explainable omni-multimodal emotion recognition with reinforcement learning, 2025. 3
- [29] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms, 2025. 3
- [30] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025. 3
- [31] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025. 3
- [32] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 3
- [33] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4089–4098, 2018. 3
- [34] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 3
- [35] Dinesh Jayaraman and Kristen Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1238–1247, 2018. 3, 4

- [36] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017. 3
- [37] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020. 3
- [38] Zekun Qi, Wenyao Zhang, Yufei Ding, Runpei Dong, Xinjiang Yu, Jingwen Li, Lingyun Xu, Baoyu Li, Xialin He, Guofan Fan, Jiazhao Zhang, Jiawei He, Jiayuan Gu, Xin Jin, Kaisheng Ma, Zhizheng Zhang, He Wang, and Li Yi. Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation, 2025. 4
- [39] 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*, pages 38–55. Springer, 2024. 9
- [40] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 9
- [41] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 9, 16
- [42] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023. 10, 16
- [43] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädele, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 13
- [44] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 305–314, 2021. 16