INSIGHT-O3: EMPOWERING MULTIMODAL FOUNDATION MODELS WITH GENERALIZED VISUAL SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

The ability for AI agents to "think with images" requires a sophisticated blend of reasoning and perception. However, current open multimodal agents still largely fall short on the reasoning aspect that are crucial for real-world tasks like analyzing documents with dense charts/diagrams and navigating maps. To address this gap, we first introduce O3-BENCH, a new benchmark designed to evaluate multimodal reasoning with interleaved attention to visual details. O3-BENCH features challenging questions that require agents to gather subtle visual information from multiple distinct areas of an image while reasoning upon it. These tasks are highly challenging even for frontier systems like OpenAI o3, which only obtains 42.8% accuracy on O3-BENCH. To tackle these tasks, we propose INSIGHT-O3, a multi-agent framework that divides labor between a visual reasoning agent (vReasoner) and a visual search agent (vSearcher). As a concrete first step towards o3-like open systems, this work focuses on the latter (*i.e.*, vSearcher), for which we introduce the task of generalized visual search—locating relational, fuzzy, or conceptual regions described in free-form language, beyond just simple objects or figures in natural images. We present a multimodal LLM purpose-trained for this task via reinforcement learning. As a plug-and-play agent, our vSearcher empowers frontier multimodal models (as vReasoners) with generalized visual search, significantly improving their performance on a wide range of benchmarks.

1 INTRODUCTION

Thinking with images is an important and very useful skill for multimodal agents (OpenAI, 2025b). The skill rests on two crucial and fundamental cognitive abilities: reasoning and perception. Recent efforts at developing such a skill based on open models mainly focus on the perception component, *e.g.*, searching for a particular object or figure in natural images and then answering a simple visual query about them (Wu & Xie, 2024; Shen et al., 2024; Li et al., 2025; Zhang et al., 2025a; Su et al., 2025a; Zheng et al., 2025; Wang et al., 2025c; Zhu et al., 2025b; Wang et al., 2025a; Lai et al., 2025). While this feature is clearly useful in certain scenarios, it is still far from being able to handle many real-world tasks that require deeper and more abstract reasoning. Typical examples of such tasks include extracting key information from complex reports and navigating through intricate maps to reach a destination. Solving these tasks often require both organized reasoning and focused attention to various visual details scattered across an image.

Currently, the reasoning capability of open multimodal models is still relatively weak in comparison with frontier proprietary models (Yue et al., 2024a; Yuan et al., 2025; Hao et al., 2025). This makes it very difficult to replicate the kind of reasoning-driven image-thinking behavior demonstrated by OpenAI o3 (OpenAI, 2025b). In this work, we take a concrete step towards building such an intelligent system with open models. First, we propose a new multimodal benchmark, O3-BENCH, to help better evaluate the general capability of multimodal models to think with images. Complementary to most of the existing benchmarks which only deal with object attributes and spatial relations in natural images (Wu & Xie, 2024; Wang et al., 2025g; Lai et al., 2025; Wang et al., 2025a), O3-BENCH consists of a set of high-quality, *reasoning-oriented* questions on images of *high information density*. The questions involve real-world tasks such as map navigation and cross-chart/diagram analysis that are highly challenging even for frontier systems like OpenAI o3. Compared with benchmarks like MME-RealWorld (Zhang et al., 2024b), O3-BENCH is significantly harder, requiring the evaluated

Figure 1: A multi-step visual reasoning example of INSIGHT-O3 on O3-BENCH. For clarity, the internal reasoning processes of vReasoner and vSearcher are omitted. More examples can be found in Appendix E.

system to collect detailed visual information from *multiple* distinct image areas while performing *complex*, *interleaved* reasoning using the information collected in the process.

To make substantive progress on O3-BENCH, we introduce a multi-agent framework, INSIGHT-O3, that comprises a visual reasoning agent (**vReasoner**) and a visual search agent (**vSearcher**). The former is responsible for high-level reasoning and general image understanding, while the latter is to help vReasoner locate specific regions of interest and collect the visual information therein. As such, INSIGHT-O3 reduces the burden of a single agent, allowing us to build an o3-like system via divide-and-conquer. This kind of specialization has been shown to work in prior art (Dayan & Hinton, 1992; Zeng et al., 2023; Shen et al., 2023; Li et al., 2023a; Hong et al., 2023; Castrejon et al., 2024). In this work, we focus on vSearcher and how it should interact with a given vReasoner. Different from current practices (Wu & Xie, 2024; Lai et al., 2025) for natural images and discrete object references, we aim to solve *generalized visual search*, where the input image can be arbitrary, *e.g.*, a map, a poster, or a screenshot; and the referring description may specify a relational, fuzzy, or conceptual region, *e.g.*, "the area to the left of the wooden chair," and "the chart showing the company's revenue in the last decade," rather than a specific object or figure. Such fuzzy descriptions are more in line with how humans reason and direct their attention to a general region of interest.

To address this broader challenge, we present InSight-o3-vS, a vSearcher model specialized in generalized visual search through reinforcement learning. InSight-o3-vS combines multimodal understanding with spatial reasoning to localize regions described in completely free-form language. The

name of our model, InSight-o3-vS, reflects its dual role: providing deeper *insight* into multimodal semantics while bringing the target region *in sight* through precise localization. Our model empowers existing multimodal foundation models (as vReasoners) in a *plug-and-play* fashion, significantly improving the performance of frontier models across a wide range of benchmarks, *e.g.*, from 38.1% to 63.4% on O3-BENCH for GPT-5-mini (OpenAI, 2025a), and from 80.1% to 87.6% on V*-Bench for Gemini-2.5-Flash (Comanici et al., 2025).

To summarize, we make the following key contributions in this work:

- We propose a new benchmark, O3-BENCH, to evaluate complex, reasoning-oriented visual tasks. This benchmark features challenges like map navigation and cross-chart analysis, which require collecting information from multiple image areas and performing interleaved reasoning, making it significantly harder than existing benchmarks.
- We introduce INSIGHT-O3, a multi-agent framework that divides the task of "thinking with images" between a high-level reasoning agent (**vReasoner**) and a visual search agent (**vSearcher**). This divide-and-conquer design greatly simplifies the complex interleaved reasoning, allowing us to build o3-like systems that surpass OpenAI o3 across a variety of benchmarks.
- We present InSight-o3-vS, a specialized vSearcher model that excels at *generalized visual search*. It is designed to be a "plug-and-play" component that empowers existing multimodal foundation models, demonstrably and significantly improving the performance of frontier systems on a wide range of benchmarks (*e.g.*, raising GPT-5-mini's score on O3-BENCH from 38.1% to 63.4%).

2 RELATED WORK

We provide a brief overview of the most relevant related work in this section. For a more comprehensive discussion, please refer to Appendix A.

Multimodal benchmarks. Classical multimodal benchmarks (Goyal et al., 2017; Saikh et al., 2022; Liu et al., 2023; Ge et al., 2024) mainly test coarse image-level or salient-attribute recognition, where modern MLLMs are near-saturated (Bai et al., 2025; Wang et al., 2025e). Recent multimodal reasoning benchmark split into (i) cognition-centric STEM benchmarks (Lu et al., 2023; Yue et al., 2024a;b) that emphasize multi-step/world-knowledge reasoning but use visually simple images, and (ii) perception-centric datasets (Wu & Xie, 2024; Zhang et al., 2024b; Lai et al., 2025) that require fine-grained recognition in high-resolution, text-rich scenes yet often limited to single-region lookups. Motivated by the "think with images" paradigm (OpenAI, 2025b), O3-BENCH jointly evaluates search/localization and higher-level reasoning on high-information-density charts and maps, requiring cross-region evidence aggregation via interleaved, multi-hop reasoning.

Multimodal reasoning models. Reinforcement learning (RL) has long been used to align model behavior with human preferences (Schulman et al., 2017). DeepSeek-R1 applies group relative policy optimization (GRPO) (Guo et al., 2025a; Shao et al., 2024b), reliably eliciting planning, reflection, and long chain-of-thought reasoning under simple rewards. Building on this idea, recent multimodal models (Yang et al., 2025b) adopt GRPO-style training and report strong gains, while cascaded RL stages (*e.g.*, InternVL3.5 (Wang et al., 2025e), Keye-VL1.5 (Yang et al., 2025a)) further push reasoning, approaching proprietary models. There are also pioneering work utilizing Python code execution to call various vision tools to solve tasks via divide-and-conquer or to help with reasoning (Gupta & Kembhavi, 2023; Surís et al., 2023; Ke et al., 2024). Nevertheless, most multimodal reasoners remain text-centric, overlooking the distinctive demands of visual reasoning.

Visual search models. Visual search is a core multimodal capability, requiring active region perception for fine-grained understanding. Early methods relied on external detectors or scripted workflows to localize regions and triggered tools via instruction tuning, leading to rigid outputs and typically single-round search (Wu & Xie, 2024). The "think with images" paradigm (OpenAI, 2025b) internalizes zoom/crop operations and has inspired end-to-end search (DeepEyes (Zheng et al., 2025)), synthetic warm-starts (Pixel-Reasoner (Su et al., 2025a)), and multi-turn RL (Mini-o3 (Lai et al., 2025)). Nonetheless, most systems still emphasize finding a single region in natural images, with limited support for multi-hop reasoning. We broaden this scope by decoupling visual search from visual reasoning and enabling multi-region search on arbitrary images.

3 O3-BENCH

We conceptualize "thinking with images" as an iterative perception-reasoning process. Perception focuses on searching and localizing task-relevant visual details, while reasoning needs to organize these cues into structured facts and performs higher-order inference (*e.g.*, planning, arithmetic, use of world knowledge) to complete the task. These two critical skills should be executed effectively and cooperate tightly to achieve strong performance. Existing benchmarks (Wu & Xie, 2024; Zhang et al., 2024b; Lai et al., 2025) primarily emphasize perception, where their questions hardly require multi-step reasoning and thus induce short reasoning chains. To bridge this gap, we introduce O3-BENCH, a benchmark that jointly assesses high-resolution perception and multi-hop visual reasoning. O3-BENCH is designed with two principles:

- **High resolution & high information density.** Images are large, high-resolution, cluttered, and information-dense, making evidence gathering genuinely non-trivial.
- **Multi-hop solution paths.** Questions require decomposing the goal, retrieving evidence from multiple regions, and composing it via intermediate steps before answering.

To instantiate these principles, O3-BENCH comprises two complementary domains: (1) *Composite charts*. Each image contains multiple heterogeneous charts (*e.g.*, bar/line/pie/tables). Our crafted questions demand cross-chart retrieval (series, axes, units), lightweight calculations (differences, ratios, aggregates), and consistency checks (scale, legend, time ranges) to derive the final answer. (2) *High-resolution digital maps*. The images typically include a map along with auxiliary components such as legends and building indices. We meticulously design questions that require visual search for targets (*e.g.*, matching symbols, categories, or toponyms) and spatial reasoning about relations and routes (*e.g.*, proximity constraints or shortest paths), conditioned on the provided context.

Overall, O3-BENCH comprises 185 images (98 in charts, 87 in maps) and 318 QA samples (136 in chart, 182 in map) in total. The majority of samples fall into the more challenging *map* category, underscoring our prioritization of complex visual perception and multi-hop reasoning. The questions of O3-BENCH are multi-choice questions with six choices and one correct answer. Among the six choices, there are four distractors that appear in the image or look similar to the correct one. We also include an option F as *No Right Choice* if there are no correct options provided. Below, we present the construction process of O3-BENCH. For other details about O3-BENCH, see Appendix B.1.

3.1 SOURCE DATA COLLECTION

Chart. The chart images in O3-BENCH are curated from the "Diagram and Table" subset from MME-RealWorld (Zhang et al., 2024b) and the Internet. To ensure high information density, we run a layout detection model, PP-DocLayout_plus-L (Cui et al., 2025), on the candidate images and only keep those with at least 8 detected layouts. As a result, 256 of 2,539 images that contain sufficient number of sub-figures and rich recognizable texts are left.

Map. We manually collect high-resolution digital maps from the Internet via keyword search. We center on the venue-level maps that require reading the provided legend/index and visually locating entities within the image to answer the question. We exclude all the country-, state-, or city-scale cartography that could be potentially answered with world knowledge. Through this process, we end up with 87 high-density map images spanning the categories over bus routes, campus, park, *etc*.

3.2 ANNOTATION PIPELINE

After the collection and initial filtering process, all images then undergo further manual screening to ensure clarity and completeness of key visual cues (*e.g.*, axes, units and legends). Next, we combine automated machine pre-annotation with human verification and authoring to generate the question-answer (QA) instances. The detailed process of data annotation is presented as follows.

Machine pre-annotation. To relieve the burden of human annotators and increase the data diversity, we first apply a three-step automated data pipeline to generate five questions for each image. (1) *Layout detection*. We divide the high-resolution images into several structured layouts (*e.g.*, tables, charts, legends) using PP-DocLayout_plus-L (Cui et al., 2025). For map images, we review

the predictions, correct erroneous regions, and supplement missing areas via manual annotation. (2) *Information extraction*. For each layout, we prompt Qwen2.5-VL-32B (Bai et al., 2025) to produce a detailed caption for the layout and extract OCR text from it. In addition, we obtain global context by generating a caption and extracting the OCR text for the full image. (3) *Automated question synthesis*. For each image, we provide the layout set (with captions and OCR texts) and the global context to GPT-5 (OpenAI, 2025a) to generate five questions (with answers and explanations) that compose evidence from the provided layout regions. Note that we do not provide the full image to GPT-5, compelling it to focus on region-level details and encourages multi-hop composition. More details about the whole pre-annotation process can be found in Appendix B.2.

Human annotation. (1) *Filtering and validation*. Annotators start by discarding ill-posed or low-quality machine-generated QAs (*e.g.*, those with factual inconsistencies, ambiguous prompts, or spurious multi-hop reasoning). For the retained QAs, annotators verify that the six-option set contains exactly one unambiguous correct answer and confirm that the explanation faithfully, step by step, justifies the choice. The annotators also ensures that the target layouts are relevant to answering the question; these layouts are either derived from the explanation or added via manual annotation. (2) *Human-authored questions*. For information-dense images, machine-generated QAs often contain logical errors, wrong answers, or missed visual details. These QAs are reworked or completely rewritten by the annotators, adhering to our design principles: requiring fine-grained detail retrieval and multi-hop reasoning. Each QA includes a detailed explanation to aid verification and have exactly one unambiguous correct choice among the six.

Difficulty filtering and secondary review. We evaluate all candidate items with three strong proprietary MLLMs, *i.e.*, GPT-5-mini (OpenAI, 2025a), Gemini-2.5-Flash (Comanici et al., 2025) and Doubao-Seed-1.6 (Bytedance, 2025), using the same evaluation prompt. We discard any items solved by all three models to ensure difficulty. Subsequently, independent reviewers (distinct from the original annotators) conducts cross-verification: a final pass over the QAs and the explanations to confirm factual correctness, clarity, and formatting consistency. Finally, we confirm with experiments that attention to visual details is vital to good performance on O3-BENCH (see Appendix B.3).

4 INSIGHT-O3

In the previous section, we introduced O3-BENCH, a meticulously-crafted benchmark that require problem-solving systems to have both good reasoning and perception capabilities, as well as the ability to integrate them in a natural, synergetic manner. Recent approaches towards such systems mostly build upon a single MLLM agent which handles both reasoning and perception workloads within a single context window (Su et al., 2025a; Zheng et al., 2025; Lai et al., 2025). While this is reasonable for tasks primarily focusing on either reasoning or perception, the agent may struggle when the workloads are heavy and intertwined.

To address the issue, we propose INSIGHT-O3, a two-agent system that largely decouples the aforementioned burden by a visual reasoning agent (**vReasoner**) and a visual search agent (**vSearcher**). The former specializes in high-level, abstract reasoning (with some general image understanding capability), while the latter is mainly responsible for extracting detailed visual information and presenting them to vReasoner. For instance, given a question, vReasoner first decomposes the question through reasoning, and, if needed, issues relevant image region descriptions to vSearcher; vSearcher then localizes the requested evidence (with help from tools like image cropping) and returns it for subsequent rounds until a final answer is produced. This process is illustrated in Figure 2(a). However, jointly training both agents in a system like INSIGHT-O3 is notoriously difficult—their objectives differ yet are highly interdependent, causing difficulties such as credit assignment across calls, and non-stationary updates when both policies learn. Additionally, in our case, even the state-of-the-art open MLLMs, *e.g.*, Qwen2.5-VL-Instruct (Bai et al., 2025), tend to produce overly concise replies with little thinking process (Lai et al., 2025).

To avoid overcomplication, we consider a simpler, more manageable setting in this paper. Specifically, we delegate higher-order reasoning at training time to a strong external model (*e.g.*, GPT-4o as vReasoner) and focus on training vSearcher to cooperate with the given vReasoner effectively. This separation helps simplify optimization and improve data efficiency. Our experiment shows that a well-trained vSearcher can significantly improve the performance of a wide range of vReasoners

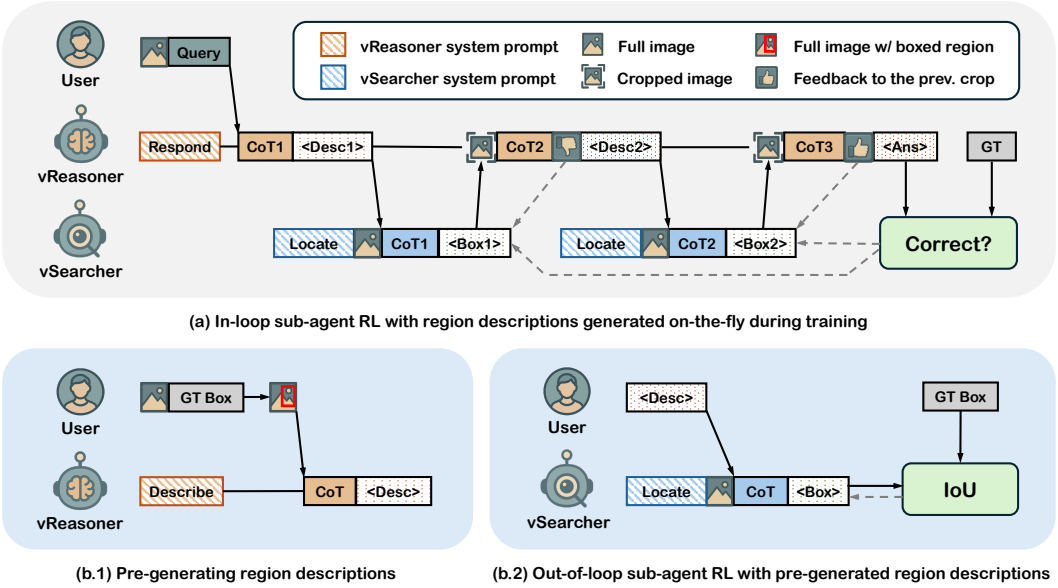


Figure 2: **Training pipeline.** We use a hybrid RL algorithm to train vSearcher. **(a)** In the in-loop component, vReasoner generates visual search tasks on-the-fly during training as it tries to answer a user query. We use vReasoner's feedback and final answer correctness as supervision (denoted by dashed arrows) for vSearcher. **(b)** In the out-of-loop component, we use pre-generated descriptions with ground-truth bounding boxes, allowing us to train vSearcher efficiently via IoU supervision.

as a plug-and-play callable agent. Furthermore, the resulting system may help synthesize multi-turn supervised-finetuning (SFT) traces with interleaved reasoning and visual search, paving the way for a larger, potentially unified model. While prior works (Su et al., 2025a; Jiang et al., 2025a) have explored role-playing to construct similar SFT traces, the data quality is often not very high as the role-playing agents are only loosely coordinated. In this respect, our approach helps strength this coordination with reinforcement learning (RL), as detailed next.

4.1 Training Algorithm

We propose a hybrid sub-agent RL algorithm that consists of an *in-loop* component and an *out-of-loop* component to train vSearcher (see Figure 2). In the out-of-loop component, we pre-generate region descriptions with predefined bounding boxes, allowing us to train vSearcher very efficiently via direct IoU supervision. In the in-loop component, we use real descriptions generated on-the-fly during training by vReasoner. Compared with pre-generated tasks, these dynamically generated tasks are more aligned in nature with the tasks that vSearcher will see during inference time.

Reward design. For the out-of-loop RL, we use the following reward function for vSearcher:

$$r = \mathbb{I}[n_{\text{tool}} > 0] \cdot (\lambda_{\text{format}} \cdot r_{\text{format}} + \lambda_{\text{IoU}} \cdot r_{\text{IoU}}), \tag{1}$$

where n_{tool} is the number of tool calls made by vSearcher, $r_{\text{format}} \in \{0, 1\}$ is the format reward, and $r_{\text{IoU}} \in (0, 1)$ is the IoU reward (λ_{format} and λ_{IoU} are weighting coefficients for the rewards). In Eq.(1), the IoU reward r_{IoU} encourages vSearcher to propose an accurate region that matches the description. In addition, we encourage vSearcher to use image cropping[1] at least once to verify that the returned region matches the given region description. For every predicted box b and the ground-truth box b^*, we define the IoU reward as $r_{\text{IoU}} = \max\{0, \text{IoU}(b, b^*) - \alpha\}/(1 - \alpha)$ where $\alpha \in (0, 1)$ controls the reward threshold. Boxes with an IoU less than α are not rewarded.

For the in-loop RL component, we replace the IoU reward r_{IoU} in Eq.(1) with a pseudo IoU reward $\hat{r}_{\text{IoU}} \in \{0, 1\}$. We obtain $\hat{r}_{\text{IoU}}$ by asking vReasoner to rate each vSearcher's prediction, with the rating criterion being whether the prediction is relevant to the assigned task and can help vReasoner

[1]For simplicity, we only allow vSearcher to use the most essential image cropping tool. Our framework, however, does not impose such a constraint. It is easy to incorporate other kinds of tools for vSearcher to use.

answer the user query. The rating $s \in \{0, 1\}$ is a binary score indicating if the prediction is helpful. However, as this rating is not always reliable (vReasoner may sometimes make mistakes), we further incorporate outcome supervision as a safeguard. Let $c \in \{0, 1\}$ stand for whether the final answer of vReasoner is correct. The pseudo IoU reward is defined as $\hat{r}_{\text{IoU}} = \mathbb{I}[s = c = 1]$.

Advantage estimation. We follow GRPO (Shao et al., 2024b) to estimate the advantages for the out-of-loop RL component. As for the in-loop component, we normalize the rewards with respect to the *global* mean and standard deviation instead of the *group* mean and standard deviation since the concept of "group" no longer exists for the dynamically generated tasks. Formally, the advantage of an output token o_t at time step t is computed as $\hat{A}_t = [r - \text{mean}(\boldsymbol{r})]/\text{std}(\boldsymbol{r})$, where for the out-of-loop component, $\boldsymbol{r} = \{r_i\}_{i=1}^{G}$ with G being the group size; and for the in-loop component, $\boldsymbol{r} = \{r_i\}_{i=1}^{N}$ with N being the total number of visual search tasks generated on-the-fly by vReasoner.

Objective function. The objective function we use is based on GRPO (Shao et al., 2024b), with some modifications (*e.g.*, global advantage estimations) to incorporate the in-loop RL component. Given a policy model π_θ, the old policy $\pi_{\theta_{\text{old}}}$, and a reference policy π_{ref}, the objective function for a batch of M vSearcher outputs (including both in-loop and out-of-loop ones) is defined as

$$J(\theta) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\gamma_t(\theta)\hat{A}_t, \text{clip}\left(\gamma_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}]\right\}, \quad (2)$$

where $\gamma_t(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}$ is the importance ratio of the output token $o_{i,t}$ given a query q and all previous output tokens $o_{i,<t}$ including tool-response tokens. During training, we mask the loss for tool-response tokens as they are not generated by the policy model.

4.2 Training Data Construction

As high-resolution, information-dense images with good questions are scarce and difficult to collect on a large scale, we construct training data by synthesizing collages (for the in-loop RL) and generating pseudo visual search targets (for the out-of-loop RL). The source data are mostly from existing VQA training datasets, which follow a largely different distribution from O3-Bench and other evaluation benchmarks we consider in our experiments.

In-loop RL data. A key criterion for in-loop RL data is that they must be difficult enough to incentivize visual search; otherwise vReasoner could simply answer on its own and vSearcher would receive no reward. To raise search difficulty and ensure meaningful credit assignment, we build image collages by stitching multiple low-to-medium-resolution images into a canvas. We construct collages from a filtered combination of Visual CoT (Shao et al., 2024a) and the V* training data (Wu & Xie, 2024), where each item provides a QA pair and a target bounding box. For each collage, we choose one target image (carrying the QA) and add several filler images as distractors. After difficulty filtering, we obtain 15,303 hard problems that vReasoner must rely on vSearcher to solve reliably. For more construction details and visualizations of the data, see Appendix C.1.

Out-of-loop RL data. We use InfographicVQA (Mathew et al., 2022) as the image source of the out-of-loop RL data. Most InfographicVQA images have high information density and feature more organic and diverse layouts than collages. We detect layout components in the source images with PP-DocLayout_plus-L (Cui et al., 2025). The candidate layout boxes are filtered and further processed, resulting in 10,186 high-quality layout boxes. We then use GPT-5-nano to generate concise, high-level region descriptions for each box, as illustrated in Figure 2(b.1). Through prompting, we make GPT-5-nano mimic the style it would use when invoking vSearcher in the in-loop setting. This process yields a set of *(image, region description, bbox)* which enables the out-of-loop RL. More construction details and visualizations are provided in the Appendix C.2.

5 Experiment

In our main experiments, we train Qwen2.5-VL-7B-Instruct (Bai et al., 2025) as vSearcher under GPT-5-mini-2025-08-07 (OpenAI, 2025a) as vReasoner for balanced efficiency and reasoning

Table 1: **Performance comparison with frontier models/systems.** All models/systems are evaluated under their default configurations unless specified otherwise. Performance of open models are mostly cited from the literature (Wu & Xie, 2024; Zheng et al., 2025; Wang et al., 2025a; Lai et al., 2025). Other results are averaged over 3 trials, except for MME-RW_{Lite} (single-trial). Small-size numbers indicate performance gaps between vReasoners with and without access to vSearchers.

Model/System	V*-Bench	HR-$Bench_{4K}$	Tree-Bench	$VProbe_{Hard}$	MME-RW_{Lite}	O3-Bench	Average
LLaVA-OV-7B	70.9	62.0	37.3	13.4	48.5	-	-
InternVL3-8B	-	-	38.8	-	-	-	-
Qwen2.5-VL-7B	75.5	68.2	37.0	23.9	46.7	29.9	46.9
Qwen3-VL-8B	86.4	78.9	48.3	31.6	53.0	45.6	57.3
Pixel Reasoner	86.3	74.0	28.8	28.8	-	-	-
DeepEyes	83.3	73.2	37.5	35.1	-	29.6	-
Mini-o3	88.2	77.5	-	48.0	-	28.4	-
OpenAI o3	76.4	74.3	52.3	23.6	55.2	42.8	54.1
GPT-4o	68.6	65.1	47.4	26.4	51.2	29.9	48.1
+ Qwen2.5-VL-7B	75.2 +6.6	69.7 +4.6	45.9 −1.6	15.4 −11.0	44.6 −6.6	33.8 +3.9	47.4 −0.7
+ InSight-o3-vS	80.4 +11.8	76.2 +11.1	49.5 +2.1	25.5 −1.1	50.1 −1.1	41.9 +12.0	53.9 +5.8
GPT-5-nano	64.0	60.6	45.4	21.7	47.7	28.9	44.7
+ Qwen2.5-VL-7B	70.1 +6.1	67.3 +6.7	45.7 +0.3	18.2 −3.5	44.9 −2.8	29.4 +0.5	45.9 +1.2
+ InSight-o3-vS	75.1 +11.1	72.3 +11.7	47.7 +2.3	31.4 +9.7	48.4 +0.7	34.8 +5.9	51.0 +6.3
GPT-5-mini	73.8	72.0	54.6	26.4	56.1	38.1	53.5
+ Qwen2.5-VL-7B	80.6 +6.8	83.2 +11.2	53.1 −1.5	37.7 +11.3	58.1 +2.0	52.3 +14.2	56.4 +2.9
+ InSight-o3-vS	86.9 +13.1	86.7 +14.7	54.1 −0.5	41.2 +14.6	59.0 +2.9	63.4 +25.3	65.2 +11.7
+ InSight-o3-vS†	86.2 +12.4	85.7 +13.7	55.0 +0.4	39.6 +13.2	58.4 +2.3	64.0 +25.9	64.8 +11.3
Gemini-2.5-Flash#	72.8	75.0	48.9	17.9	55.6	57.2	54.6
+ Qwen2.5-VL-7B	76.3 +3.5	76.7 +1.7	51.3 +2.4	16.7 −1.2	50.9 −4.7	58.2 +1.0	55.0 +0.4
+ InSight-o3-vS	80.8 +8.0	80.2 +5.2	52.1 +3.2	19.8 +1.9	55.1 −0.5	65.4 +8.2	58.9 +4.3
+ InSight-o3-vS†	85.5 +12.7	82.7 +7.7	52.6 +3.7	26.4 +8.5	56.1 +0.5	65.7 +8.5	61.5 +6.9
Gemini-2.5-Flash	80.1	83.5	49.9	39.6	56.5	60.7	61.7
+ Qwen2.5-VL-7B	80.9 +0.8	79.0 −4.5	49.1 +0.8	31.4 −8.2	52.0 −4.5	61.6 +0.9	59.0 −2.7
+ InSight-o3-vS	87.6 +7.5	82.3 −1.2	50.1 +0.2	36.2 −3.4	56.3 −0.2	67.0 +6.3	63.3 +1.6
+ InSight-o3-vS†	88.3 +8.2	83.0 −0.5	53.6 +3.7	38.3 −1.3	56.4 −0.1	67.4 +6.7	64.5 +2.8

† Trained with Gemini-2.5-Flash as vReasoner.
# Image-size constraint tightened to 1280×1280px, same as the maximum supported size for GPT models.

capability. The resulting vSearcher, named InSight-o3-vS, is evaluated under various vReasoners including Gemini-2.5-Flash (Comanici et al., 2025). For comparison, we evaluate these vReasoners normally as standalone models as well (and with the untrained Qwen2.5-VL-7B-Instruct being the vSearcher). We use the default configuration for proprietary models/systems (per official API) except from setting image detail to `high`[2]. More implementation details can be found in Appendix D.

Evaluation datasets. We evaluate a range of open and proprietary models/systems on the following benchmarks: (1) Natural-image benchmarks: V*-Bench (Wu & Xie, 2024), Tree-Bench (Wang et al., 2025a), and VisualProbe-Hard (Lai et al., 2025). (2) Mixed benchmarks: HR-Bench (Wang et al., 2025f) and MME-RealWorld (Zhang et al., 2024b). For efficient evaluation, we use the lite version of MME-RealWorld, which has 1,919 questions, still much heavier than the other benchmarks. (3) Our O3-BENCH. More information about the benchmarks can be found in Appendix J.

5.1 MAIN RESULTS

Cross-domain performance improvement for frontier models. As shown in Table 1, INSIGHT-O3 significantly improves the performance of frontier models such as GPT-5-mini and Gemini-2.5-Flash on most benchmarks. On average, the performance of GPT-5-mini has improved by 22%. In particular, the accuracy of GPT-5-mini on O3-BENCH has nearly doubled (from 38.1% to 63.4%) with the help of our vSearcher (InSight-o3-vS). Meanwhile, INSIGHT-O3 also significantly outperforms their pre-RL counterparts, *i.e.*, vReasoner + Qwen2.5-VL-7B, across all the benchmarks. The results suggest that InSight-o3-vS is able to generalize *out-of-distribution* across various domains since its training data distribution is distinct from the evaluation data distributions. To gain more insight on how INSIGHT-O3 improves the baselines, see Appendix F for a comparative analysis.

[2] When image detail is set to `high`, OpenAI scales down oversize images to roughly 1280×1280px, as per OpenAI API (https://platform.openai.com/docs/guides/images-vision#calculating-costs). This is the maximum supported image resolution for GPT models. Gemini-2.5-Flash API does not impose such a strict constraint, so we use a much larger, 3500×3500px budget that is ample for the purpose.

Table 2: **Performance of Gemini-2.5-Flash (+ InSight-o3-vS†) under different maximum training/test image resolutions.** All results are averaged over 3 trials, except for MME-RW$_{\text{Lite}}$ (which is based on a single trial). Small-size numbers indicate performance gaps between settings with and without vSearcher. † Trained with Gemini-2.5-Flash as vReasoner.

Train res.	Test res.	V*-Bench	HR-Bench$_{4K}$	Tree-Bench	VProbe$_{\text{Hard}}$	MME-RW$_{\text{Lite}}$	O3-Bench	Average
-	1280^2	72.8	75.0	48.9	17.9	55.6	57.2	54.6
1280^2	1280^2	85.5 +12.7	82.7 +7.7	52.6 +3.7	26.4 +8.5	56.1 +0.5	65.7 +8.5	61.5 +6.9
3500^2	1280^2	85.3 +12.5	81.3 +6.3	53.1 +4.2	22.6 +4.7	55.1 −0.5	66.9 +9.7	60.7 +6.1
-	3500^2	80.1	83.5	49.9	39.6	56.5	60.7	61.7
1280^2	3500^2	87.8 +7.7	84.3 +0.8	52.1 +2.2	39.6 +0.0	56.4 −0.1	67.5 +6.8	64.6 +2.9
3500^2	3500^2	88.3 +8.2	83.0 −0.5	53.6 +3.7	38.3 −1.3	56.4 −0.1	67.4 +6.7	64.5 +2.8

Table 3: **Ablation study on reward design and advantage estimation.** All results are averaged over 3 trials. Small-size numbers indicate performance changes w.r.t. the proposed setting.

Setting	V*-B.	VP$_{\text{Hard}}$	O3-B.	Avg.
Proposed	86.9	**41.2**	**63.4**	**63.8**
w/o tool cond.	86.4 −0.5	39.3 −1.9	62.6 −0.8	62.8 −1.0
w/o feedback	86.5 −0.4	37.1 −4.1	61.4 −2.0	61.7 −2.1
w/o outcome	86.9 +0.0	38.7 −2.5	63.3 −0.1	63.0 −0.8
w/o GN	**87.3** +0.4	36.8 −4.4	60.9 −2.5	61.7 −2.1

Table 4: **Sensitivity analysis w.r.t. max. input resolution of vSearcher.** "#vSearch" is the number of vSearcher calls made by vReasoner per QA.

Max. pixels	V*-B.	O3-B.	#vSearch
0.8M	85.3	61.5	2.77
1.6M	86.7	63.2	2.60
3.2M	89.4	63.2	2.65
6.4M	86.4	64.0	2.58
12.8M	86.9	63.4	2.55

Generalization under different vReasoners. We observe that InSight-o3-vS, which was trained as a sub-agent under GPT-5-mini, generalizes under other vReasoner models as well. As shown in Table 1, InSight-o3-vS improves the performance of a much smaller model, GPT-5-nano, from 21.7% to 31.4% on VisualProbe-Hard, from 28.9% to 34.8% on O3-BENCH, and from 44.7% to 51.0% overall. Under Gemini-2.5-Flash (a different model family), the advantage remains significant, showing about 6-7% lead over the baselines on V*-Bench and O3-BENCH. We have also explored training InSight-o3-vS under Gemini-2.5-Flash instead of GPT-5-mini, and observed similar generalization (see "+ InSight-o3-vS†" rows in Table 1). In few cases where InSight-o3-vS fails to improve the performance of the vReasoner, *e.g.*, GPT-4o and Gemini-2.5-Flash on VisualProbe-Hard, we see a sharp decrease in performance as we allow these models to call Qwen2.5-VL-7B. This suggests that these models are relatively weak at tool calling and multi-turn reasoning. In Appendix G, we present typical failure cases of INSIGHT-O3; we find that even GPT-5-mini (despite its great performance) still makes a lot of mistakes.

Performance gaps on O3-BENCH. Interestingly, on O3-BENCH, we observe that Gemini-2.5-Flash has a huge edge over GPT-5-mini (when they have no access to vSearcher), but on the other benchmarks, the edge is not so prominent—Gemini-2.5-Flash is even slightly worse than GPT-5-mini on Tree-Bench. This suggests that O3-BENCH is indeed quite different from the other benchmarks, and Gemini-2.5-Flash is particularly good at solving the kind of tasks in O3-BENCH on its own. Notably, with InSight-o3-vS, GPT-5-mini is able to drastically reduce the gap (from 22.6% to 4.0%) with Gemini-2.5-Flash, demonstrating the importance of thinking with images for addressing O3-BENCH, and also highlighting the effectiveness of our approach.

Effect of input image resolution. Comparing the results of Gemini-2.5-Flash in Table 1 under different maximum input image resolutions, we see that a much higher resolution offers clear advantages. However, the improvement brought by vSearcher is less when vReasoner can see clearer. In addition, we find that training under one resolution and evaluating under another seems to have little impact on the performance (see Table 2). Meanwhile, the input image resolution for vSearcher has less impact. Table 4 shows the performance of GPT-5-mini + InSight-o3-vS under varying maximum image resolution of InSight-o3-vS during evaluation. From the results, we can see that InSight-o3-vS is not sensitive to the resolution, maintaining decent performance on V*-Bench and O3-BENCH even when the maximum image resolution is only 0.8M (25% of that during training). As the resolution decreases, the average number of vSearcher calls increases. This is reasonable as low-resolution images often obscure fine details, making it harder for vSearcher to locate the targets.

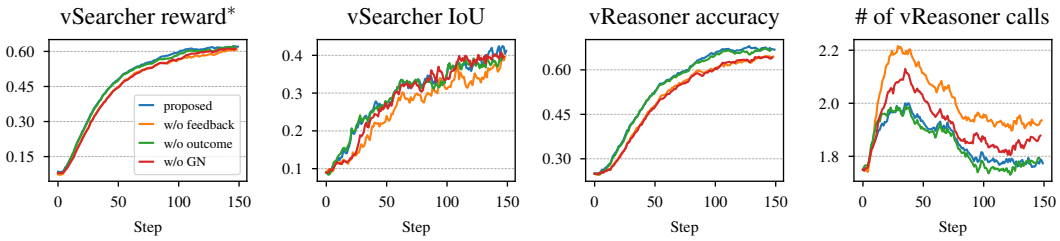


Figure 3: **Training dynamics of INSIGHT-O3.** The rightmost chart, "# of vReasoner calls", shows the average number of times vReasoner calls vSearcher per QA. * For fair comparison, the reward curves are plotted under the same setting ("w/o feedback") for all the settings.

5.2 ABLATION STUDY

Hybrid RL training. Table 5 shows the results of ablating the in-loop and the out-of-loop sub-agent RL components. Without the in-loop RL component, training is much faster (81-89% reduction in time per training step) but the final performance is worse on average. Dropping the out-of-loop RL component also hurts the performance; moreover, the training time increases due to more in-loop training. As mentioned in Section 4.2, the two RL components use different training data. The better performance of hybrid RL training can be partly explained by the combined use of the training data. Another contributing factor is the combination of two different sources of supervision (high-level vReasoner feedback and low-level IoU with ground-truth boxes). Overall, combining the two components leads to the best result.

Table 5: **Ablation study on hybrid RL training.** "I." and "O." stand for the in-loop and out-of-loop RL components, respectively. "T/step" is the average time per training step. All results are averaged over 3 trials. Small-size numbers indicate performance changes w.r.t. the untrained baselines.

	I.	O.	V*-B.	VP_{Hard}	O3-B.	T/step
GPT-5-nano			70.1	18.2	29.4	-
	✓		**74.9** +4.8	23.9 +5.7	33.2 +3.8	846s
		✓	73.7 +3.6	25.1 +6.9	34.6 +5.2	130s
	✓	✓	74.5 +4.4	**27.4** +9.2	**35.9** +6.5	693s
GPT-5-mini			80.6	37.7	52.3	-
	✓		86.4 +5.8	39.0 +1.3	61.9 +9.6	1223s
		✓	84.8 +4.2	**41.2** +3.5	60.8 +8.5	105s
	✓	✓	**86.9** +6.3	**41.2** +3.5	**63.4** +11.1	941s

Reward design and advantage estimation. In Table 3, we compare the setting we *proposed* in Section 4.1 on GPT-5-mini + InSight-o3-vS with the following ablated variants: "*w/o tool*" drops the tool condition in the reward function; "*w/o feedback*" removes vReasoner feedback, only using outcome supervision for pseudo IoU reward; "*w/o outcome*" is the opposite of "w/o feedback"; and "*w/o GN*" drops the global normalization for advantage estimation. The originally proposed setting outperforms all the variants with a small average lead on the three benchmarks. The training dynamics under these settings are shown in Figure 3. As vSearcher learns to better locate the regions described by vReasoner, we observe that both the out-of-loop localization IoU and the in-loop vReasoner accuracy improve. The non-monotonic "# of vReasoner calls" shows two RL phases of INSIGHT-O3: vSearcher first learns to obey the formatting instructions, and then learns to localize more accurately (so vReasoner could solve the same problem with less vSearcher calls).

Although we encourage vSearcher to use the image-cropping tool, we find the average tool call count often ends up close to 1. There are two underlying reasons for this behavior. First, as mentioned by Zheng et al. (2025); Lai et al. (2025), Qwen2.5-VL-Instruct is often reluctant to call the tool, and does not seem to know how to use the tool properly. Second, vReasoner usually describe a rough region that is not very hard for vSearcher to locate, so the tool has little utility for vSearcher.

6 CONCLUSION

In this work, we introduced O3-BENCH, a high-information-density benchmark that jointly evaluates visual localization and multi-hop reasoning. To advance the research on this challenging benchmark, we proposed INSIGHT-O3, a multi-agent framework that decomposes the "think with images" workflow into high-level reasoning (vReasoner) and visual search (vSearcher). We focus on the training of vSearcher via reinforcement learning to seamlessly cooperate with vReasoner. The specialized InSight-o3-vS can be used as a "plug-and-play" component for existing multimodal foundation models and helps significantly improve the performance of frontier models.

REFERENCES

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 126–135, 2017. 51

Anonymous. Agentorchestra: Orchestrating hierarchical multi-agent intelligence with the tool-environment-agent (TEA) protocol. In *Submitted to The Fourteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YcnKdeI9pp. under review. 18

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. 3, 5, 7, 18, 20, 41

Bytedance. Doubao-Seed-1.6. https://console.volcengine.com/ark/region:ark+cn-beijing/model/detail?Id=doubao-seed-1-6/, 2025. 5

Lluis Castrejon, Thomas Mensink, Howard Zhou, Vittorio Ferrari, Andre Araujo, and Jasper Uijlings. Hammr: Hierarchical multimodal react agents for generic vqa. In *NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward*, 2024. 2, 18

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*, 2025a. 18

Shuang Chen, Yue Guo, Zhaochen Su, Yafu Li, Yulun Wu, Jiacheng Chen, Jiayu Chen, Weijie Wang, Xiaoye Qu, and Yu Cheng. Advancing multimodal reasoning: From optimized cold start to staged reinforcement learning. *arXiv preprint arXiv:2506.04207*, 2025b. 18

Zihui Cheng, Qiguang Chen, Xiao Xu, Jiaqi Wang, Weiyun Wang, Hao Fei, Yidong Wang, Alex Jinpeng Wang, Zhi Chen, Wanxiang Che, et al. Visual thoughts: A unified perspective of understanding multimodal chain-of-thought. *arXiv preprint arXiv:2505.15510*, 2025. 19

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 3, 5, 8, 18

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595. 4, 7, 20

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, et al. Multi-agent collaboration via evolving orchestration. *arXiv preprint arXiv:2505.19591*, 2025. 18

Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. *Advances in neural information processing systems*, 5, 1992. 2

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement, 2025. URL https://arxiv.org/abs/2503.17352. 18

Yue Fan, Xuehai He, Diji Yang, Kaizhi Zheng, Ching-Chen Kuo, Yuting Zheng, Sravana Jyothi Narayanaraju, Xinze Guan, and Xin Eric Wang. Grit: Teaching mllms to think with images. *arXiv preprint arXiv:2505.15879*, 2025. 19

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 18

Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 3, 18

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017. 3, 18

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*, 2025a. 3, 18

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, et al. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025b. 18

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14953–14962, 2023. 3, 19

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018. 18

Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. In *Forty-second International Conference on Machine Learning*, 2025. 1

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2023. 2, 18

Zhipeng Hou, Junyi Tang, and Yipeng Wang. Halo: Hierarchical autonomous logic-oriented orchestration for multi-agent llm systems. *arXiv preprint arXiv:2505.13516*, 2025. 18

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 2025. 18

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024. 19

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. 18

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019. 18

Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. Llvip: A visible-infrared paired dataset for low-light vision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3496–3504, 2021. 51

Chaoya Jiang, Yongrui Heng, Wei Ye, Han Yang, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. Vlm-r3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. *arXiv preprint arXiv:2505.16192*, 2025a. 6, 19

Qing Jiang, Xingyu Chen, Zhaoyang Zeng, Junzhi Yu, and Lei Zhang. Rex-thinker: Grounded object referring via chain-of-thought reasoning. *arXiv preprint arXiv:2506.04034*, 2025b. 19

Fucai Ke, Zhixi Cai, Simindokht Jahangard, Weiqing Wang, Pari Delir Haghighi, and Hamid Rezatofighi. Hydra: A hyper agent for dynamic compositional visual reasoning. In *European Conference on Computer Vision*, pp. 132–149. Springer, 2024. 3, 19

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023. 50

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. *arXiv preprint arXiv:2509.07969*, 2025. 1, 2, 3, 4, 5, 8, 10, 18, 19, 50

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 9098–9108, 2025. 1, 19

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023a. 2, 18

Kaican Li, Kai Chen, Haoyu Wang, Lanqing Hong, Chaoqiang Ye, Jianhua Han, Yukuai Chen, Wei Zhang, Chunjing Xu, Dit-Yan Yeung, et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. In *European conference on computer vision*, pp. 406–423. Springer, 2022. 51

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023b. 18

Jiaying Liu, Dong Liu, Wenhan Yang, Sifeng Xia, Xiaoshuai Zhang, and Yuanying Dai. A comprehensive benchmark for single image compression artifact reduction. *IEEE Transactions on image processing*, 29:7845–7860, 2020. 51

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 3, 18

Ziyu Liu, Yuhang Zang, Yushan Zou, Zijian Liang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual agentic reinforcement fine-tuning. *arXiv preprint arXiv:2505.14246*, 2025. 19

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 3, 18, 52

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022. 7

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025. 18

Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. *arXiv preprint arXiv:2505.19702*, 2025. 19

OpenAI. OpenAI GPT-5. https://openai.com/index/introducing-gpt-5/, 2025a. 3, 5, 7, 18, 20, 31, 42, 44, 50

OpenAI. OpenAI o3 and o4-mini. https://openai.com/index/introducing-o3-and-o4-mini/, 2025b. 1, 3, 18, 19

Yi Peng, Peiyu Wang, Xiaokun Wang, Yichen Wei, Jiangbo Pei, Weijie Qiu, Ai Jian, Yunzhuo Hao, Jiachun Pan, Tianyidan Xie, et al. Skywork r1v: Pioneering multimodal reasoning with chain-of-thought. *arXiv preprint arXiv:2504.05599*, 2025. 18

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Bin Xu, Lei Hou, Juanzi Li, Yuxiao Dong, et al. Cogcom: Train large vision-language models diving into details through chain of manipulations. *arXiv preprint arXiv:2402.04236*, 2024. 19

Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. Mia-bench: Towards better instruction following evaluation of multimodal llms, 2024. URL https://arxiv.org/abs/2407.01509. 52

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. 18

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, Daniel Toyama, Robert Berry, Divya Tyamagundlu, Timothy Lillicrap, and Oriana Riva. Androidworld: A dynamic benchmarking environment for autonomous agents, 2024. URL https://arxiv.org/abs/2405.14573. 52

Enna Sachdeva, Nakul Agarwal, Suhas Chundi, Sean Roelofs, Jiachen Li, Mykel Kochenderfer, Chiho Choi, and Behzad Dariush. Rank2tell: A multimodal driving dataset for joint importance ranking and reasoning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 7513–7522, 2024. 51

Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23(3):289–301, 2022. 3, 18

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3, 18

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Unleashing chain-of-thought reasoning in multi-modal language models. *CoRR*, 2024a. 7, 19, 27

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b. 3, 7, 18

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. *arXiv preprint arXiv:2411.16044*, 2024. 1

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*, 2025a. 18

Wei Shen, Jiangbo Pei, Yi Peng, Xuchen Song, Yang Liu, Jian Peng, Haofeng Sun, Yunzhuo Hao, Peiyu Wang, Jianhao Zhang, et al. Skywork-r1v3 technical report. *arXiv preprint arXiv:2507.06167*, 2025b. 18

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023. 2, 18, 19

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024. 31

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025a. 1, 3, 5, 6, 19

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, et al. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025b. 19

Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022. 51

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11888–11898, 2023. 3, 19

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*, 2025a. 52

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025b. 18

Kwai Keye Team, Biao Yang, Bin Wen, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl technical report. *arXiv preprint arXiv:2507.01949*, 2025c. 18

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025d. URL https://arxiv.org/abs/2507.01006. 18

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*, 2025a. 1, 8, 18, 50

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025b. 18

Jiacong Wang, Zijian Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, et al. Vgr: Visual grounded reasoning. *arXiv preprint arXiv:2506.11991*, 2025c. 1, 19

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024. 18, 52

Peiyu Wang, Yichen Wei, Yi Peng, Xiaokun Wang, Weijie Qiu, Wei Shen, Tianyidan Xie, Jiangbo Pei, Jianhao Zhang, Yunzhuo Hao, et al. Skywork r1v2: Multimodal hybrid reinforcement learning for reasoning. *arXiv preprint arXiv:2504.16656*, 2025d. 18

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*, 2025e. 3, 18

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025f. 8, 50

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025g. 1, 18

Ye Wang, Qianglong Chen, Zejun Li, Siyuan Wang, Shijie Guo, Zhirui Zhang, and Zhongyu Wei. Simple o3: Towards interleaved vision-language reasoning. *arXiv preprint arXiv:2508.12109*, 2025h. 19

Yana Wei, Liang Zhao, Jianjian Sun, Kangheng Lin, Jisheng Yin, Jingcheng Hu, Yinmin Zhang, En Yu, Haoran Lv, Zejia Weng, et al. Open vision reasoner: Transferring linguistic cognitive behavior for visual reasoning. *arXiv preprint arXiv:2507.05255*, 2025. 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*, pp. 13084–13094, 2024. 1, 2, 3, 4, 7, 8, 18, 19, 27, 50

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024. 18

LLM-Core-Team Xiaomi. Mimo-vl technical report, 2025. URL https://arxiv.org/abs/2506.03569. 18

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. 52

Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl 1.5 technical report. *arXiv preprint arXiv:2509.01563*, 2025a. 3, 18

Qinhong Yang, Dongdong Chen, Zhentao Tan, Qiankun Liu, Qi Chu, Jianmin Bao, Lu Yuan, Gang Hua, and Nenghai Yu. Hq-50k: A large-scale, high-quality dataset for image restoration. *arXiv preprint arXiv:2306.05390*, 2023. 51

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025b. 3, 18

Jiakang Yuan, Tianshuo Peng, Yilei Jiang, Yiting Lu, Renrui Zhang, Kaituo Feng, Chaoyou Fu, Tao Chen, Lei Bai, Bo Zhang, et al. Mme-reasoning: A comprehensive benchmark for logical reasoning in mllms. *arXiv preprint arXiv:2505.21327*, 2025. 1

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, 2024a. 1, 3, 18

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. *arXiv preprint arXiv:2409.02813*, 2024b. 3, 18

Andy Zeng, Maria Attarian, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, et al. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 18

Kaihao Zhang, Dongxu Li, Wenhan Luo, Wenqi Ren, Björn Stenger, Wei Liu, Hongdong Li, and Ming-Hsuan Yang. Benchmarking ultra-high-definition image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14769–14778, 2021. 51

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, 2024a. 18

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Chain-of-focus: Adaptive visual search and zooming for multimodal reasoning via rl. *arXiv preprint arXiv:2505.15436*, 2025a. 1, 19

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint arXiv:2408.13257*, 2024b. 1, 3, 4, 8, 18, 51

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*, 2025b. 19

Shitian Zhao, Haoquan Zhang, Shaoheng Lin, Ming Li, Qilong Wu, Kaipeng Zhang, and Chen Wei. Pyvision: Agentic vision with dynamic tooling. *arXiv preprint arXiv:2507.07998*, 2025. 19

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025. 1, 3, 5, 8, 10, 19, 48

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *arXiv preprint arXiv:2506.04308*, 2025. 52

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025a. 18

Muzhi Zhu, Hao Zhong, Canyu Zhao, Zongze Du, Zheng Huang, Mingyu Liu, Hao Chen, Cheng Zou, Jingdong Chen, Ming Yang, et al. Active-o3: Empowering multimodal large language models with active perception via grpo. *arXiv preprint arXiv:2505.21457*, 2025b. 1, 19

Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7380–7399, 2021. 51

A More Discussion on Related Work

A.1 Multimodal Benchmarks

Classical multimodal benchmarks (Goyal et al., 2017; Hudson & Manning, 2019; Gurari et al., 2018; Saikh et al., 2022; Li et al., 2023b; Fu et al., 2023; Liu et al., 2023; Ge et al., 2024) primarily focus on coarse image-level understanding or target at salient-object attributes, on which current multimodal models (Bai et al., 2025; Wang et al., 2025e; Yang et al., 2025a) show near-saturated performance. With growing attention to multimodal reasoning, more challenging benchmarks have emerged, which could be categorized as two groups. (1) Cognition-centric benchmarks. STEM (science, technology, engineering, and mathematics) benchmarks (Lu et al., 2023; Wang et al., 2024; Zhang et al., 2024a; Yue et al., 2024a;b) evaluate the model's multi-step reasoning, integration of world knowledge, and complex calculations to solve scientific problems, whereas the accompanying images are generally straightforward to interpret. (2) Perception-centric benchmarks. These benchmarks (Wu & Xie, 2024; Wang et al., 2025g; Zhang et al., 2024b; Lai et al., 2025) require fine-grained perception in high-resolution images and strong OCR recognition on text-rich scenes. Nevertheless, many questions become routine once the model precisely localizes the target region, allowing for single-glance solutions. Though the recent TreeBench (Wang et al., 2025a) evaluates second-order reasoning over object spatial transformations, depth ordering, and *etc*, it still centers on a single region in natural images, leaving cross-region evidence aggregation largely underexplored. With the emergence of the "think with images" paradigm (OpenAI, 2025b), we argue that a well-designed benchmarks should evaluate the joint perceptual and cognitive skills. Our proposed O3-Bench fills the research gap by meticulously constructing hard questions on high-information density images (*e.g.*, composite graphs, maps), therefore requiring models to gather information from multiple, spatially distinct regions and to perform complex, interleaved reasoning.

A.2 Multimodal Reasoning Models

Reinforcement learning (RL) (Schulman et al., 2017; Rafailov et al., 2023; Hu et al., 2025) has long been used to align the response of large language models (LLMs) and multimodal LLMs (MLLMs) with human preferences. Recently, DeepSeek-R1 (Guo et al., 2025a) creatively applied group relative policy optimization (GRPO) (Shao et al., 2024b) to LLMs, estimating the mean and variance of advantages across response groups under a simple reward signal. This strategy reliably elicits behaviors such as planning, thinking, and self-reflection, enabling long chain-of-thought (CoT) reasoning and moving toward more general-purpose reasoning capabilities. Building on this success, several works (Huang et al., 2025; Yang et al., 2025b; Meng et al., 2025; Chen et al., 2025a; Wang et al., 2025b; Shen et al., 2025a; Chen et al., 2025b; Deng et al., 2025; Wei et al., 2025; Peng et al., 2025; Wang et al., 2025d; Shen et al., 2025b) explore cold-start initialization and GRPO-based RL training for multimodal models (*e.g.*, Qwen2.5-VL (Bai et al., 2025)) and report substantial gains on science- and math-oriented benchmarks. Concurrently, InternVL3.5 (Zhu et al., 2025a; Wang et al., 2025e) and Keye-VL1.5 (Team et al., 2025c; Yang et al., 2025a) further leverage cascaded, iterative RL stages to push the frontier of reasoning ability, achieving performance competitive with proprietary models (OpenAI, 2025a; Comanici et al., 2025). Nevertheless, current multimodal reasoning models (Xiaomi, 2025; Team et al., 2025d; Guo et al., 2025b; Team et al., 2025b) still focus on text-centric reasoning, neglecting the distinctive demands of visual reasoning in multimodal scenarios.

A.3 Hierarchical Agent Frameworks and Tool-Using Multimodal Agents

Recent work has shown that hierarchical collaboration among specialized agents can significantly improve performance on complex tasks. Socratic Models (Zeng et al., 2023) and HuggingGPT (Shen et al., 2023) demonstrate early examples of LLM-based orchestration over expert models via language. More structured frameworks like CAMEL (Li et al., 2023a), MetaGPT (Hong et al., 2023), and HAMMR (Castrejon et al., 2024) explore role-based or modular specialization with coordinated task decomposition. Others, including AutoGen (Wu et al., 2024), HALO (Hou et al., 2025), Puppeteer (Dang et al., 2025), and AgentOrchestra (Anonymous, 2025), further extend this paradigm with explicit planning hierarchies, adaptive execution, and learned orchestration, consistently outperforming flat-agent baselines across diverse domains.

Recent works have also explored equipping multimodal models with tool-use and programmatic reasoning to tackle complex visual tasks. VisProg (Visual Programming) (Gupta & Kembhavi, 2023) and ViperGPT (Surís et al., 2023) are two pioneering approaches that use large language models to generate and execute code for orchestrating vision modules. Beyond static program generation, other agents use LLMs as high-level controllers. For example, HuggingGPT (Shen et al., 2023) demonstrates an LLM (ChatGPT) orchestrating numerous specialized models (for vision, language, etc.) More recently, HYDRA (Ke et al., 2024) introduces a dynamic multi-stage framework for visual reasoning: it integrates an LLM-based planner and reasoner with a reinforcement learning–based controller that adapts the sequence of operations via feedback loops, yielding more reliable step-by-step reasoning. These tool-augmented systems highlight the power of combining learned vision-language models with external modules or code execution to improve flexibility and compositional reasoning. In contrast, our work (INSIGHT-O3) targets a complementary gap by introducing a dedicated visual search agent that can be invoked by reasoning agents to locate fine-grained, conceptually described regions within images. This specialized capability, absent in prior tool-using frameworks, allows an INSIGHT-O3-enabled system to pinpoint relevant visual details based on free-form descriptions, thereby enhancing multimodal reasoning with more precise visual understanding.

A.4 VISUAL SEARCH MODELS

Visual search is an important functionality in the multimodal domain, requiring the models to perform active perception over regions of interest (RoIs) for fine-grained visual understanding. Early approaches (Wu & Xie, 2024; Shao et al., 2024a; Qi et al., 2024; Hu et al., 2024; Li et al., 2025) rely on external tools or predefined workflows for region localization and use instruction tuning to trigger tool use. These models exhibit rigid output patterns and typically support only a single round of visual search, which limits their effectiveness in complex scenes. Recently, the milestone OpenAI o3 (OpenAI, 2025b) established the "think with images" paradigm, in which image manipulations (*e.g.*, zooming, cropping) are internalized as intrinsic capabilities, enabling image–text interleaved reasoning. The community has rapidly turn the attention to the promising field. DeepEyes (Zheng et al., 2025) exploits the inherent grounding ability of MLLMs and incentivizes visual search via end-to-end reinforcement learning. Pixel-Reasoner (Su et al., 2025a) improves search accuracy by warm-start instruction tuning on synthesized data with error-induced self-correction trajectories. Mini-o3 (Lai et al., 2025) introduces an over-turn masking technique during RL to encourage multi-turn interaction, markedly enhancing reasoning adaptability and diversity. Other lines of work (Zhao et al., 2025; Zhang et al., 2025b; Liu et al., 2025) resort to write codes for executing multiple image manipulations (*e.g.*, cropping, rotation, enhancement), pointing to an open-ended toolkit for visual reasoning. Although effective, recent advanced methods (Zhu et al., 2025b; Fan et al., 2025; Zhang et al., 2025a; Su et al., 2025b; Ni et al., 2025; Cheng et al., 2025; Jiang et al., 2025a;b; Wang et al., 2025c;h) still largely prioritize locating a single region on natural images, which sidelines the model's capacity for reasoning. This paper broadens the capability scope of visual search models by decoupling visual reasoning and visual search agents, allowing the receiving of any images and searching of multiple distinct regions.

B ADDITIONAL INFORMATION ON O3-BENCH

B.1 BENCHMARK STATISTICS

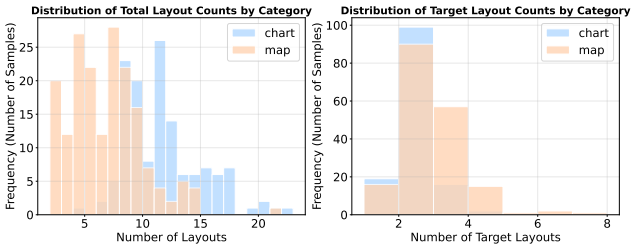


Figure 4: Distribution of layout numbers in O3-BENCH.

Figure 5: Distribution of resolution.

We summarize the benchmark statistics from following three aspects. **(1) Distribution of layouts.** The benchmark features 8.5 layouts and 2.3 target layouts for each sample on average, indicating high information density and the need for multi-step reasoning. And the layout distribution by category is displayed in Figure 4. It can be seen that chart images typically exhibit a larger set of total layouts, whereas map images require more target layouts for reasoning. **(2) Distribution of resolution.** We collect high-resolution imagery in O3-BENCH. As shown in Figure 5, most images have side lengths in the 2K–4K range, while map images reach up to $\sim$10K pixels on the longer side, yielding high information density. On average, image height and width are 3,967 and 4,602 pixels, respectively. **(3) Distribution of options.** We randomly shuffle options A–E, ensuring an approximately uniform distribution of correct-answer positions. And a small portion (8.2%) of samples use option F (*No Right Choice*) as correct answer, which compels models to aggregate evidence across the entire image and determine that none of the other options is valid.

B.2 DETAILS OF MACHINE PRE-ANNOTATION

(1) *Layout detection*. We first divide the high-resolution images into several structured layouts (*e.g.*, tables, charts, legends). We use PP-DocLayout_plus-L (Cui et al., 2025) to detect layout bounding boxes in the image and construct the set $\mathcal{L} = \{l_i\}_{i=1}^{m}$, where $l_i \in \mathbb{R}^4$ denotes layout coordinates. For chart images, we directly use the detector outputs. For map images, we review the predictions, correct erroneous regions, and supplement missing areas via manual annotation.

(2) *Information extraction*. For each detected layout l_i in image $\boldsymbol{I}$, we obtain the cropped image $\boldsymbol{I}_{l_i}$ according to its coordinates. We then prompt Qwen2.5-VL-32B (Bai et al., 2025) to produce a detailed caption c_i and extract OCR text o_i for $\boldsymbol{I}_{l_i}$, thereby forming the layout triplet $\tau_i = (\boldsymbol{I}_{l_i}, c_i, o_i)$. And then we aggregate all region triplets into $\mathcal{T} = \{\tau_i\}_{i=1}^{m}$. In addition, we obtain global context by generating a caption and OCR text for the full image, denoted as $\mathcal{G} = (c_g, o_g)$. The prompts we use are provided in Appendix H.1.

(3) *Automated question synthesis*. We provide the layout set $\mathcal{T}$ and the global context $\mathcal{G}$ to GPT-5 (OpenAI, 2025a) and explicitly prompt it to generate five questions that compose evidence from multiple regions. For each question, GPT-5 must produce six options (A–F) with exactly one correct answer, and option F is reserved for *No Right Choice*. It also need to supply a step-by-step explanation that interprets the reasoning chain. It is noted that we do not provide the full image to GPT-5, which compels the model to focus on region-level details and encourages multi-hop composition. The prompts we use are provided in Appendix H.2.

B.3 MORE EXPERIMENTAL RESULTS FOR O3-BENCH

We present additional results for O3-BENCH in this section. Because our annotation pipeline supplies the target layouts most relevant to each question, we can provide these region crops alongside the original full image at test time. We evaluate GPT-5-Mini (OpenAI, 2025a) and Qwen2.5-VL-7B (Bai et al., 2025), with results shown in Table 6. Both models exhibit significant performance gains when given the additional target layouts, underscoring the need for models to actively locate task-critical regions and perform interleaved visual reasoning.

Table 6: Ablation on target layouts in O3-BENCH.

Model	O3-BENCH		
	chart	map	overall
GPT-5-mini	38.2	37.9	38.1
+ target layouts	74.3	53.3	62.3
Qwen2.5-VL-7B	36.0	25.3	29.9
+ target layouts	46.3	28.0	35.9

B.4 VISUALIZATION OF O3-BENCH

In Figures 6–11, we present six representative visualizations (four map items and two chart items), showing how O3-BENCH couples high-resolution perception with multi-step reasoning. Each annotation includes: (i) the multiple-choice question and answer; (ii) highlighted target layouts that mark the regions consulted along the solution path; and (iii) a concise, ordered explanation that composes the evidence, which allows readers to verify the answer quickly.

Annotation

Split: Map

Q&A: What is the building opposite the main entrance of the School of Business Postgraduate Annexe?

A. MInT Study; ✅ **B. Commerce Building;** C. Psychology, Darwin House; D. Philosophy, Union St East; E. Pacific Islands Centre; F. None of above

Target Layouts:

Explanation: **Step 1.** Locate the School of Business Postgraduate Annexe in the building list, identified as No. 26 at F7. **Step 2.** Find this building at F7 on the East Campus map. **Step 3.** Refer to the Map Legend to confirm that the triangle represents the Main Entrance to the Building. Observe that the main entrance of the School of Business Postgraduate Annexe faces Building No. 27. **Step 4.** Look up Building No. 27 in the building list, identified as the Commerce Building, thus the answer is B.

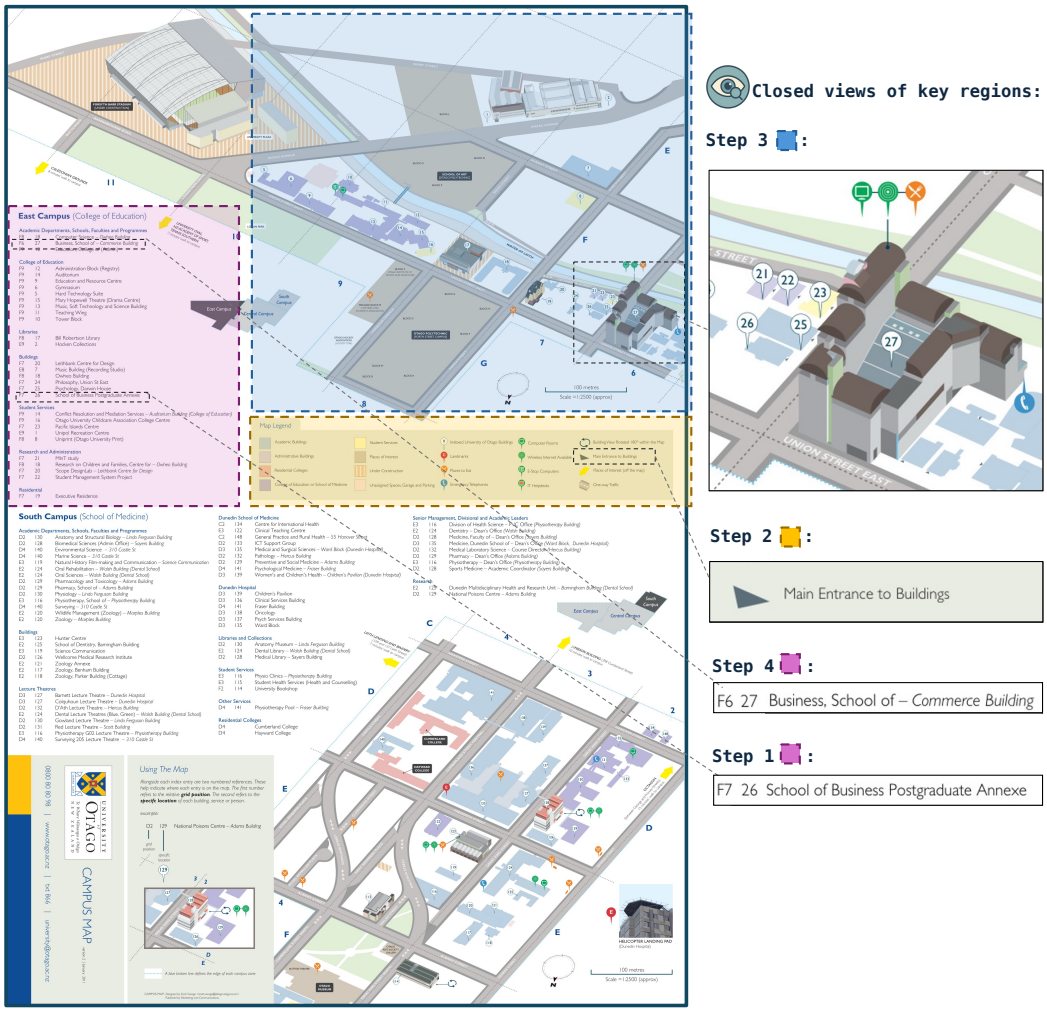


Figure 6: Example from O3-BENCH (Map-1). Each annotation comprises a six-choice QA and a brief explanation with highlighted target layouts for quick verification; additionally, we also provide step-wise close-ups (outside the annotation) to reveal the evidence chain in large images where fine details may be hard to see.

Annotation

Split: Map

Q&A: Which floor has the Prayer room, and near which zone is it located?

A. 2F, South zone; ✅ **B. 2F, North zone;** C. 3F, North zone; D. 4F, North zone; E. 4F, West zone; F. None of above

Target Layouts:

Explanation: **Step 1.** Refer to the Legend to identify the symbol for the Prayer room. **Step 2.** Use the Floor Guide in the bottom-right corner to find that the Prayer room is located on 2F. **Step 3.** On the 2F map , locate the Prayer room to the right of the H&M store in the North zone. Select option B accordingly.

Figure 7: Example from O3-BENCH (Map-2). Each annotation comprises a six-choice QA and a brief explanation with highlighted target layouts for quick verification; additionally, we also provide step-wise close-ups (outside the annotation) to reveal the evidence chain in large images where fine details may be hard to see.

Annotation

Split: Map

Q&A: Which restaurant has an ATM located next to it?

A. Burger Federation; B. Fulton Street Café; C. Wolfgang Express; D. Goose Island Beer Company; E. Facades Bar; ✅ **F. The Bronze Tap**

Target Layouts: 🟦 🟪 🟧

Explanation: **Step 1.** Locate the ATM symbol in PASSENGER SERVICES 🟪. **Step 2.** Match the boarding gate numbers of the restaurants listed in the options using the restaurant list 🟧: Burger Federation (L20), Fulton Street Cafe (H14), Wolfgang Express (K12), Goose Island Beer Company (L10A), Facades Bar (K15), The Bronze Tap (K5). **Step 3.** Compare the gate locations on the terminal map 🟦 and identify which gate has an ATM symbol nearby. Only K5 has an ATM symbol, so the answer is The Bronze Tap, F

Closed views of key regions:

Step 2 🟧:

Burger Federation	L20
Fulton Street Café	H14
Wolfgang Express	K12
Goose Island Beer Company	L10A
Facades Bar	K15
The Bronze Tap	K5

Step 1 🟪:

Automated Teller Machine (ATM)

Step 3 🟦:

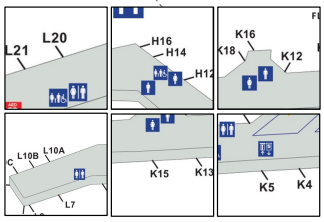


Figure 8: Example from O3-BENCH (Map-3). Each annotation comprises a six-choice QA and a brief explanation with highlighted target layouts for quick verification; additionally, we also provide step-wise close-ups (outside the annotation) to reveal the evidence chain in large images where fine details may be hard to see.

Annotation

Split: Map

Q&A: After having lunch at Brookfield Cafe, I want to take a bus to visit Boole Library. Which route should I take?

A. 201; B. 202; ✅ **C. 205;** D. 208; E. 209; F. 216

Target Layouts: 🟦 🟪 🟨

Explanation: **Step 1.** Locate Brookfield Cafe in the Catering list of 🟪; it is building 13 in Grid C7. **Step 2.** Locate Boole Library in 🟨; it is building 11 in Grid F6. **Step 3.** On the map 🟦, find that Route 205 bus from College Road below building 13 leads to the vicinity of building 11.",

Figure 9: Example from O3-BENCH (Map-4). Each annotation comprises a six-choice QA and a brief explanation with highlighted target layouts for quick verification; additionally, we also provide step-wise close-ups (outside the annotation) to reveal the evidence chain in large images where fine details may be hard to see.

Annotation

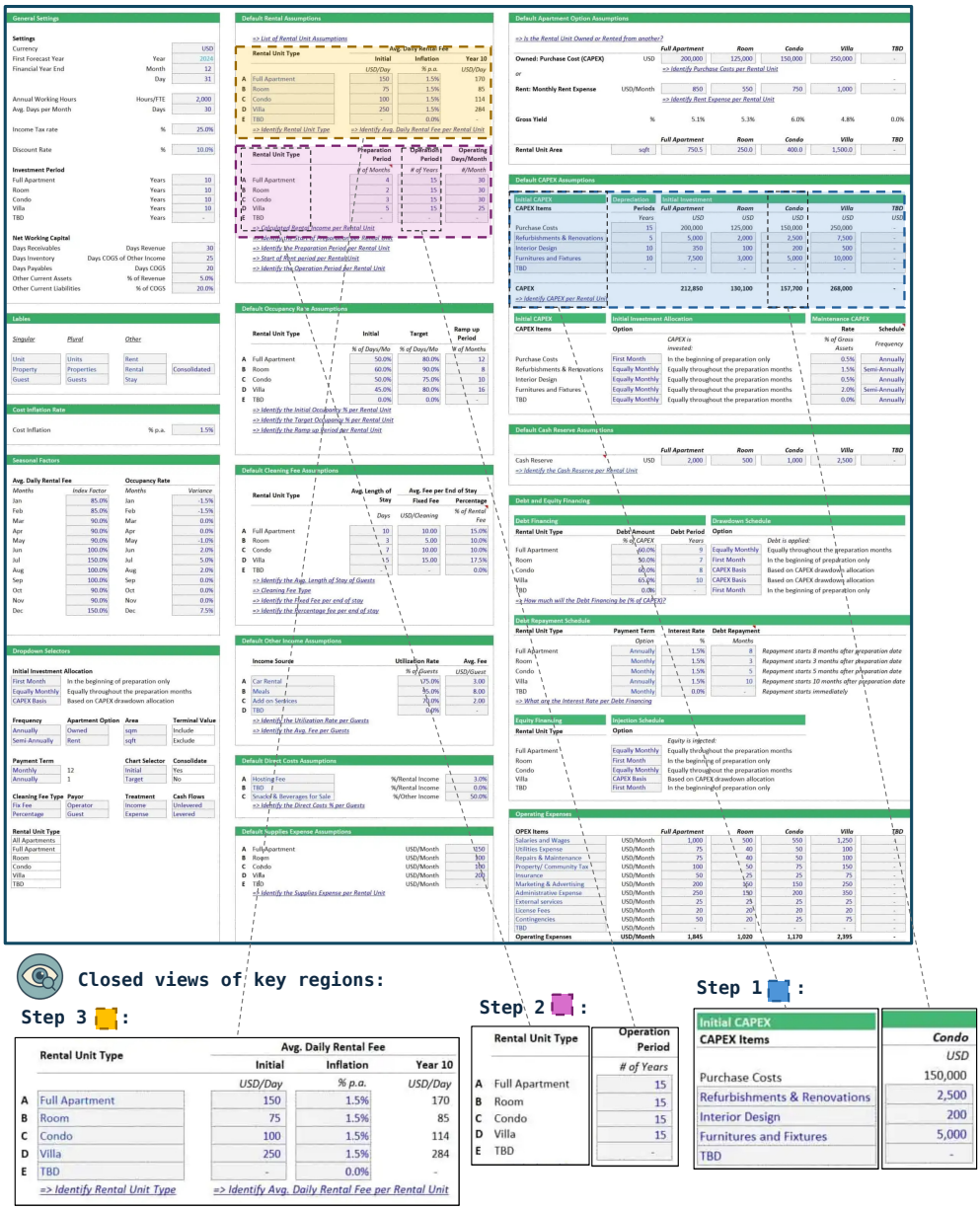
Split: Chart

Q&A: If the Condo unit's Purchase Cost is depreciated evenly over its stated operation period, what percentage does this depreciation amount account for of its Avg. Annual Rental Fee (annualized based on the Year 10 Avg. Daily Rental Fee and 30 days/month)?

A. 35.8%; B. 36.0%; C. 37.5%; D. 38.5%; E. 39.0% ✅ **F. No right choice**

Target Layouts:

Explanation: **Step 1:** From , Condo Purchase Costs = 150,000 USD. **Step 2:** From , depreciation period = 15 years. Therefore, Annual depreciation = 150,000 / 15 = 10,000 USD/year. **Step 3:** From , Condo Year 10 Avg. Daily Rental Fee = 114 USD/day. **Step 4:** Annual rental income at Year 10 rate with 30 days/month = 114 * 30 * 12 = 41,040 USD/year. **Step 5:** Percentage = (10,000 / 41,040) * 100 ≈ 24.37%. Step 6: 24.37%\nis not listed in options A–E, so the correct choice is F.



Closed views of key regions:

Step 3 :

	Rental Unit Type	Avg. Daily Rental Fee		
		Initial	Inflation	Year 10
		USD/Day	% p.a.	USD/Day
A	Full Apartment	150	1.5%	170
B	Room	75	1.5%	85
C	Condo	100	1.5%	114
D	Villa	250	1.5%	284
E	TBD	-	0.0%	-
	=> Identify Rental Unit Type	=> Identify Avg. Daily Rental Fee per Rental Unit		

Step 2 :

	Rental Unit Type	Operation Period
		# of Years
A	Full Apartment	15
B	Room	15
C	Condo	15
D	Villa	15
E	TBD	-

Step 1 :

Initial CAPEX	
CAPEX Items	Condo
	USD
Purchase Costs	150,000
Refurbishments & Renovations	2,500
Interior Design	200
Furnitures and Fixtures	5,000
TBD	-

Figure 10: Example from O3-BENCH (Chart-1). Each annotation comprises a six-choice QA and a brief explanation with highlighted target layouts for quick verification; additionally, we also provide step-wise close-ups (outside the annotation) to reveal the evidence chain in large images where fine details may be hard to see.

Annotation

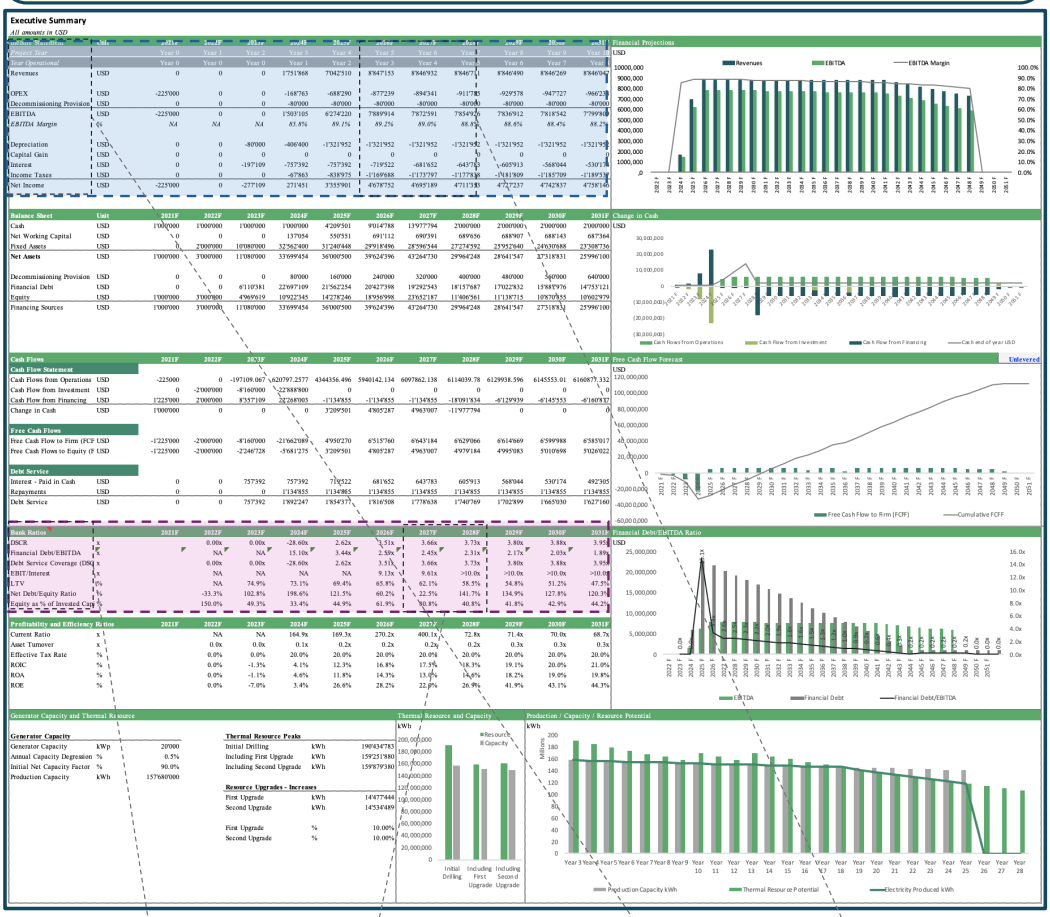
Split: Chart

Q&A: What is the LTV in the first year when Net Income exceeds 4,700,000 USD?

A. 65.8%; B. 62.1%; ✅ **C. 58.5%;** D. 54.8%; E. 51.2%; F. No right choice

Target Layouts:

Explanation: **Step 1:** From the Income Statement table, find first year Net Income > 4,700,000: 2028F = 4,711,353 is the earliest that exceeds. **Step 2:** From Bank Ratios, LTV in 2028F is shown as 58.5%



Closed views of key regions:

Step 2:

Bank Ratios	2027F	2028F
DSCR	3.66x	3.73x
Financial Debt/EBITDA	2.45x	2.31x
Debt Service Coverage (DSC	3.66x	3.73x
EBIT/Interest	9.61x	>10.0x
LTV	62.1%	58.5%
Net Debt/Equity Ratio	22.5%	141.7%
Equity as % of Invested Capi	80.8%	40.8%

Step 1:

Income Statement	2026F	2027F	2028F
Project Year	Year 5	Year 6	Year 7
Year Operational	Year 3	Year 4	Year 5
Revenues	8'847'153	8'846'932	8'846'711
OPEX	-877'239	-894'341	-911'785
Decommissioning Provision	-80'000	-80'000	-80'000
EBITDA	7'889'914	7'872'591	7'854'926
EBITDA Margin	*89.2%*	*89.0%*	*88.8%*
Depreciation	1'321'952	-1'321'952	-1'321'952
Capital Gain	0	0	0
Interest	-719'522	-681'652	-643'783
Income Taxes	1'169'688	-1'173'797	-1'177'838
Net Income	4'678'752	4'695'189	4'711'353

Figure 11: Example from O3-BENCH (Chart-2). Each annotation comprises a six-choice QA and a brief explanation with highlighted target layouts for quick verification; additionally, we also provide step-wise close-ups (outside the annotation) to reveal the evidence chain in large images where fine details may be hard to see.

B.5 COMPARISON WITH MME-REALWORLD (CHART)

The chart images of O3-BENCH are from the "Diagram & Table" subset of MME-RealWorld. The original questions from MME-RealWorld are relatively simple, usually focusing on a single value in a chart, which do not require any kind of multi-hop reasoning. For example, the original question of MME-RealWorld for the chart we show in Figure 10 is:

```
What is the cost inflation rate in the General Settings section of the
General Assumptions table?
```

In comparison, the new question in O3-BENCH is:

```
If the Condo unit's Purchase Cost is depreciated evenly over its stated
operation period, what percentage does this depreciation amount account
for of its Avg. Annual Rental Fee (annualized based on the Year 10 Avg.
Daily Rental Fee and 30 days/month)?
```

As illustrated in Figure 10, answering this question requires gathering detailed information from *three* different tables and connecting the information through multi-step reasoning and calculations.

Overall, the questions of O3-BENCH (chart) are much more difficult than the MME-RealWorld counterparts. This can also be seen from the following statistics:

- The average accuracy of GPT-5-mini on O3-BENCH (chart) is about 38.2%, whereas on MME-RealWorld (chart), the accuracy is about 82.4%.
- The average number of vSearch steps of InSight-o3 on O3-BENCH (chart) is about 3.1, whereas on MME-RealWorld (chart), the number is about 1.1.
- The average response length of GPT-5-mini vReasoner on O3-BENCH (chart) is about 1942.3 characters, whereas on MME-RealWorld (chart), the average response length is about 730.0 characters.

C TRAINING DATA CONSTRUCTION DETAILS

C.1 IN-LOOP RL DATA

Our collage sources come from the training split of Visual CoT (Shao et al., 2024a) and V* (Wu & Xie, 2024). We first filter both datasets by target bounding box size, retaining items with $\text{area}(\text{bbox})/\text{area}(\text{image}) < 0.04$. From Visual CoT, we keep all Chart/OCR-centric subsets (`dude`, `cub`, `textvqa`, `docvqa`, `infographicsvqa`, `sroie`, `vsr`, `textcap`), and treat the natural-image subsets (`flickr30k`, `gqa`, `openimages`, `v7w`) together with V* as a separate stream due to lower QA reliability (*e.g.*, weaker question–image alignment and non-unique answers). To ensure stable RL rewards, we filter this stream with an MLLM check using Qwen2.5-VL-7B and GPT-5-nano under a deterministic prompt. An item is retained only if both models return correct answer; otherwise it is discarded, including ambiguous or poorly aligned cases. After this pipeline, we retain $\sim$100K items as the source pool $\mathcal{D}$ for collage synthesis.

Given the filtered source pool $\mathcal{D}$, we synthesize collage-style training images around one primary target (the image the model should attend to) and auxiliary fills (other images used to occupy remaining space and control background complexity). The full procedure is presented in Algorithm 1. Specifically, we sample and grid-quantize a canvas, then determine a feasible target scale by intersecting global bounds with a bbox-to-canvas cap and a minimum short-edge constraint after light aspect jitter (Steps 1–2). We plan & place the target using a fit-then-shrink heuristic with a single enlarge-canvas fallback (Steps 3–4). Remaining area is panelized into grid-aligned regions under simple aspect/size guards (Step 5). Panels are then filled (largest-first) by sampling images $\tilde{t} \in \mathcal{D}$ using usage-aware weights (favoring less-frequently used candidates) that also roughly match panel aspect; when needed, we apply a light center crop and bounded scaling (Step 6). If packing remains incomplete after a brief extra fill pass, we resample from the canvas; otherwise we finalize the collage (Steps 7–8). To avoid ambiguity when querying the target image, we annotate each collage tile with an ID and include this ID in the question as a reference. Figure 12 shows representative visualizations of synthesized collages.

The canvas is sized so that the target box occupies only a tiny fraction of the canvas area, enforcing $\text{area(bbox)}/\text{area(canvas)} < 0.0002$. We filter out items that vReasoner can already solve without calling vSearcher using a pass@3 check (three attempts; any success leads to removal).

Algorithm 1 Target-and-Fill Collage Synthesis (High-level)

Require: Metadata table $\mathcal{D}$ (image path, $W_{\text{src}}, H_{\text{src}}$, object bbox), **target image** $t^\star \in \mathcal{D}$, grid G, min short edge M, canvas area/aspect ranges $[A_{\min}, A_{\max}]$ and $[a_{\min}, a_{\max}]$, target scale bounds $[\lambda_{\min}, \lambda_{\max}]$, target aspect jitter τ_{tgt}, fill jitter τ_{fill}, fill scales $[\lambda^{\text{fill}}_{\min}, \lambda^{\text{fill}}_{\max}]$, panel aspect range $[\text{AR}^{\text{panel}}_{\min}, \text{AR}^{\text{panel}}_{\max}]$, max effective source area $S^{\text{eff}}_{\max}$, **bbox coverage cap** $\rho_{\text{cap}}{=}2{\times}10^{-4}$, placement retries R, max attempts T

Ensure: Canvas $\mathcal{C}$ and placements $\mathcal{P} = \{\text{target}, \text{fills}\}$

1: **Precompute target meta.** From $t^\star$, read $W_{\text{src}}, H_{\text{src}}$, bbox ratio ρ_{src}; set $S_\star \leftarrow (W_{\text{src}}H_{\text{src}}) \cdot \min\big(1, \sqrt{S^{\text{eff}}_{\max}/(W_{\text{src}}H_{\text{src}})}\big)^2$, and $r_{\text{src}} \leftarrow W_{\text{src}}/H_{\text{src}}$.

2: **for attempt** $= 1$ to T **do** ▷ rejection loop

3: **Step 1 — Sample canvas.** Draw $A_{\text{canvas}} \sim [A_{\min}, A_{\max}]$, $a \sim [a_{\min}, a_{\max}]$; snap to grid to obtain (W, H).

4: **Step 2 — Compute feasible target scale interval.**
Bounds: $\lambda \in [\lambda_{\min}, \lambda_{\max}]$; occupancy: $\lambda S_\star \le A_{\text{canvas}}$; bbox: $\dfrac{\rho_{\text{src}}\lambda S_\star}{A_{\text{canvas}}} \le \rho_{\text{cap}}$.
Choose $r_\star$ by log-jittering r_{src} within $\pm\tau_{\text{tgt}}$ and raise the lower bound on λ so that $\min(\sqrt{\lambda S_\star r_\star}, \sqrt{\lambda S_\star / r_\star}) \ge M$.
Let I_λ be the intersection of the above constraints; if $I_\lambda = \varnothing$, optionally enlarge A_{canvas} once and recompute; if still empty, **continue** ▷ reject → restart at Step 1

5: **Step 3 — Plan target box.** Pick $\lambda \in I_\lambda$ (*e.g.*, midpoint); set $w_\star{=}\sqrt{(\lambda S_\star) r_\star}$, $h_\star{=}\sqrt{(\lambda S_\star)/r_\star}$; snap $(w_\star, h_\star)$ to multiples of G.

6: **Step 4 — Place target.** Best-fit on the free list (grid-aligned). If no fit, iteratively shrink $(w_\star, h_\star)$ and update $\lambda{=}w_\star h_\star / S_\star$, keeping feasibility in I_λ, up to R retries; if still not placed, optionally enlarge canvas once and re-plan; if it fails, **continue** ▷ reject → restart at Step 1

7: **Step 5 — Normalize free space.** Recursively split free regions into grid-aligned panels subject to aspect $\text{AR}_P \in [\text{AR}^{\text{panel}}_{\min}, \text{AR}^{\text{panel}}_{\max}]$ and minimal size.

8: **Step 6 — Fill panels.** For each panel (largest-first), sample a fill image $\tilde{t} \in \mathcal{D}$ within a $\pm\tau_{\text{fill}}$ aspect band around AR_P; if needed, center-crop $\tilde{t}$ to AR_P; scale with $\lambda^{\text{fill}} \in [\lambda^{\text{fill}}_{\min}, \lambda^{\text{fill}}_{\max}]$ and place.

9: **Step 7 — Resample if needed.** If residual free space remains after one extra fill pass, **continue** ▷ reject → restart at Step 1

10: **Step 8 — Finalize. return** $\mathcal{C}, \mathcal{P}$

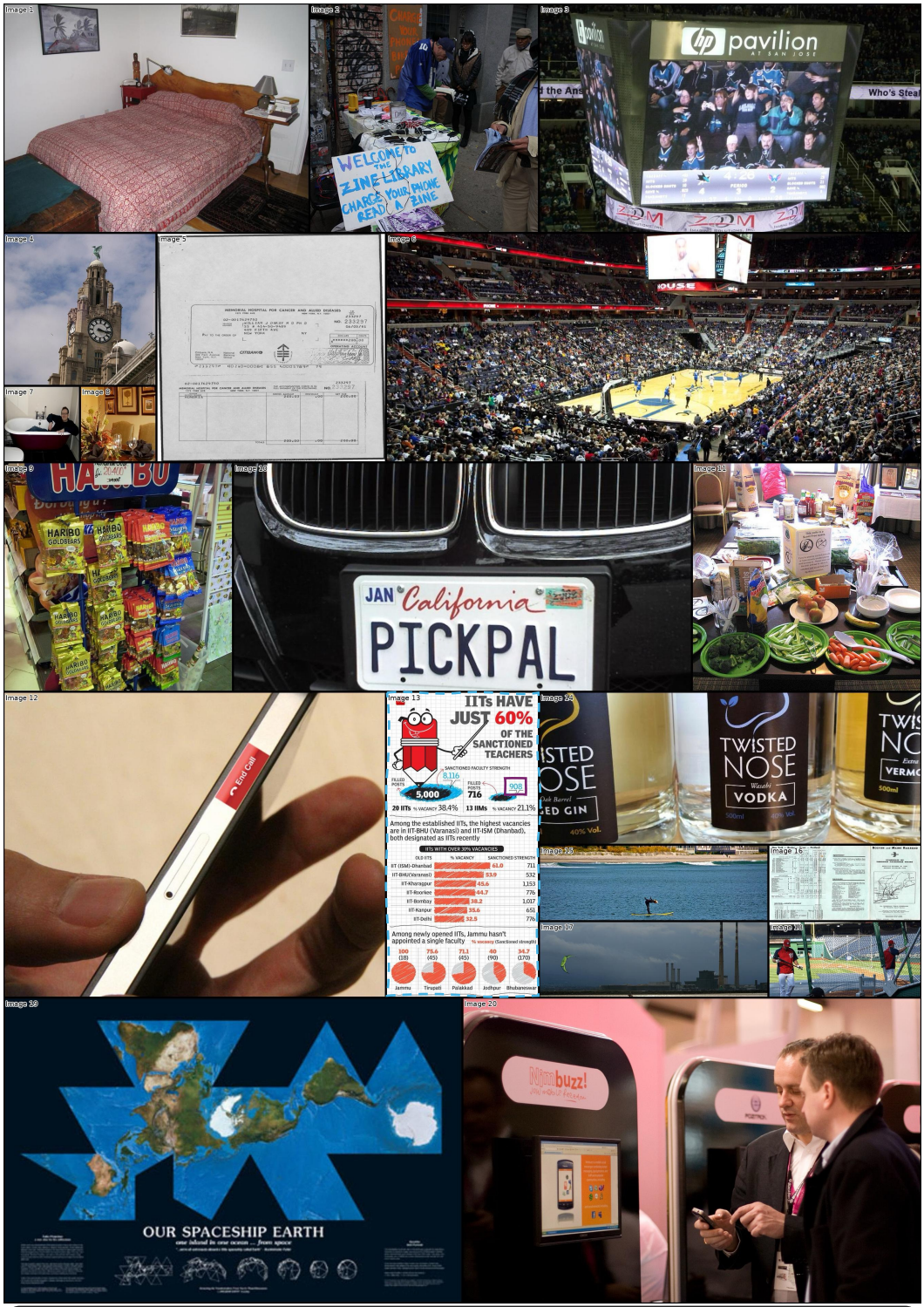
11: **end for**

12: **return None** ▷ no feasible collage after T attempts

C.2 Out-of-Loop RL Data

For PP-DocLayout_plus-L, we use the following configuration: `{"threshold": 0.01, "layout_nms": True, "layout_merge_bboxes_mode": "union"}`.

To construct meaningful visual search targets from the boxes produced by PP-DocLayout_plus-L, we start by dropping boxes of trivial layout classes such as `header` and `footer`, keeping only `text`, `image`, `table`, `chart`, `figure_title`, and `paragraph_title` boxes. We then drop boxes that are too large as they usually merge disparate things together. Boxes with area larger than a quarter of the whole image area are dropped, except for charts which are usually clean and we use a much higher threshold (0.8) for them. Next, we merge boxes that are very close to each other, measured by the effort required to enclose them in one box. The effort is computed as `1 - summed_area / min_enclosing_area`. We start with the boxes that require the



Question: In image13, what is the sanctioned faculty strength in 13 IIMS?

Answer: 908

Target Image Target Bbox

Figure 12: Example of synthesized collage for the in-loop RL. Multiple low-resolution images are stitched to raise visual density. The blue dashed box highlights the target tile; the magenta box marks the target bbox. Remaining tiles are distractors.

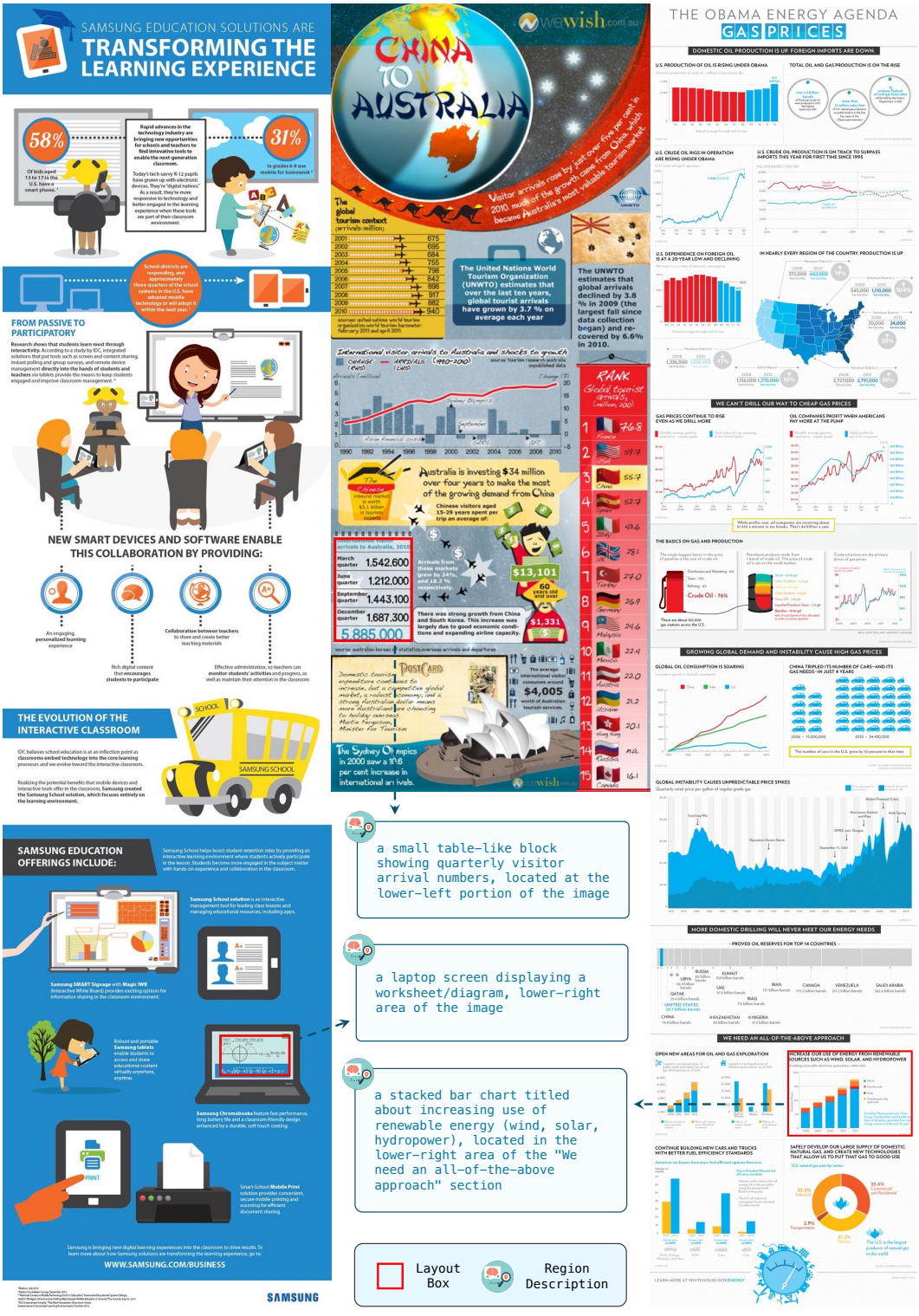


Figure 13: Examples of InfographicVQA images with pre-generated layout boxes and region descriptions for the out-of-loop RL.

least effort to merge, and stop until the required effort reaches a threshold. For `figure_title` and `paragraph_title` boxes, the threshold is 0.15, while for other boxes, it is 0.1. This helps merging small, auxiliary boxes such as figure titles and chart legends with their closest neighbors in the vicinity, avoiding truncating important context and preventing trivial search targets dominating the dataset. We skip a merge if the new box would be too large (box-to-image area ratio more than 0.25) or have an extreme aspect ratio (not within 1:5 and 5:1). We do not merge charts and tables.

After merging the boxes, we drop (1) charts/tables enclosing other charts/tables, (2) unmerged images that do not contain any text or titles, (3) unmerged titles, (4) unmerged texts, (5) boxes that are too small (box-to-image area ratio less than 0.001), and (6) boxes with extreme aspect ratios (not within 1:5 and 5:1). These boxes often contain little information (*e.g.*, icons, short texts). In the end, an image may still have multiple visual search targets; they are treated as separate data entries.

To generate region descriptions for the visual search targets obtained earlier, we first draw a red box around the target on the image, and then prompt GPT-5-nano as follows:

```
[SYSTEM]
You are a visual assistant. Your goal is to help the user to locate the region indicated by
 the red bounding box in an image.

When the user asks you to describe the region, you must follow the following rules:
- Keep it super simple and short as if you can't see clearly what is in the region.
- Don't mention any details, specific content, or small text in the region.
- Use concise, visually grounded targets (e.g., a chart, an object, a text block, a
distinct area).
- Optionally include approximate location (e.g., top-left of the image, bottom-right of the
 big chart, center column).
- Optionally include the title of the region (e.g., the table about XXX, the section titled
 XXX).
- Avoid non-visual or ordinal references (e.g., "the third largest bar", "the second row's
number").
- Don't mention the red bounding box.

Output format: region_description={...}

[USER]
Describe the region in the red bounding box.
```

In Figure 13, three examples of the final data are shown. Note that unlike collages, these examples are not stitched together; they are simply displayed side-by-side to save space.

D INSIGHT-O3 IMPLEMENTATION DETAILS

The maximum image resolution of vSearcher is set to ∼3.2M pixels (4K tokens/image) during training, and ∼12.8M pixels (16K tokens/image) during evaluation. Oversize images are downsampled to meet the constraint. We allow both vReasoner and vSearcher to make at most 6 sub-agent/tool calls during both training and evaluation. Image crops returned by sub-agent/tool calls are obtained from original images, and then resized if they exceed the size limit. For vSearcher, we use a maximum response length (including results returned by sub-agent/tool calls) of 9K and 32K tokens for training and evaluation, respectively. The sampling temperature of vSearcher is set to 1 and 0, respectively. Other hyperparameters include: training batch size 24, rollout number 8, learning rate 10^{-6}, KL loss coefficient 0.01, reward weights $\lambda_{\text{format}} = 0.2$, $\lambda_{\text{IoU}} = 0.8$, and IoU reward threshold $\alpha = 0.25$. The composition of in-loop/out-of-loop training data is 1:1. We train vSearcher fully on-policy for 150 steps. We freeze the vision tower and the adapter of Qwen2.5-VL-7B-Instruct during the whole training process. We use GPT-5-nano (OpenAI, 2025a) for evaluating answer correctness. Our code is based on verl (Sheng et al., 2024). The prompts we use can be found in Appendix I.

E MORE VISUALIZATIONS OF INSIGHT-O3 REASONING PROCESS

Figure 14 and 15 show examples of inference process of INSIGHT-O3 (GPT-5-mini as vReasoner). The vReasoner issues natural-language target descriptions; the vSearcher localizes evidence and returns them. Across a few rounds, the pair composes multi-step evidence and produces the final answer, demonstrating that INSIGHT-O3-VS plugs in cleanly and supports effective reasoning.

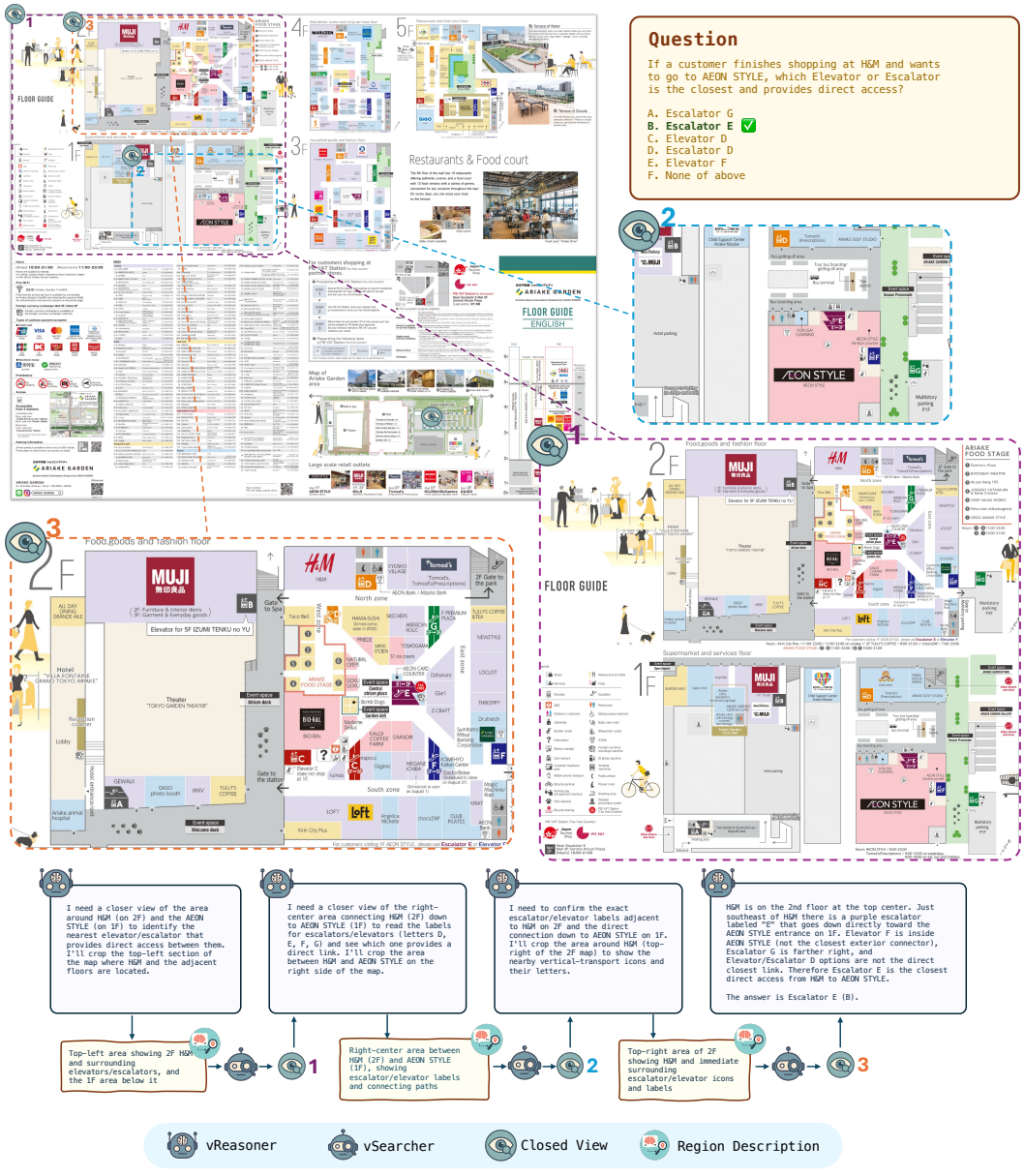


Figure 14: Qualitative result 1 with GPT-5-mini as vReasoner and INSIGHT-O3-VS as vSearcher. The reasoner requests venue-level cues (*e.g.*, legend/index lookups); the searcher returns localized regions and snippets, iterating to a correct answer.

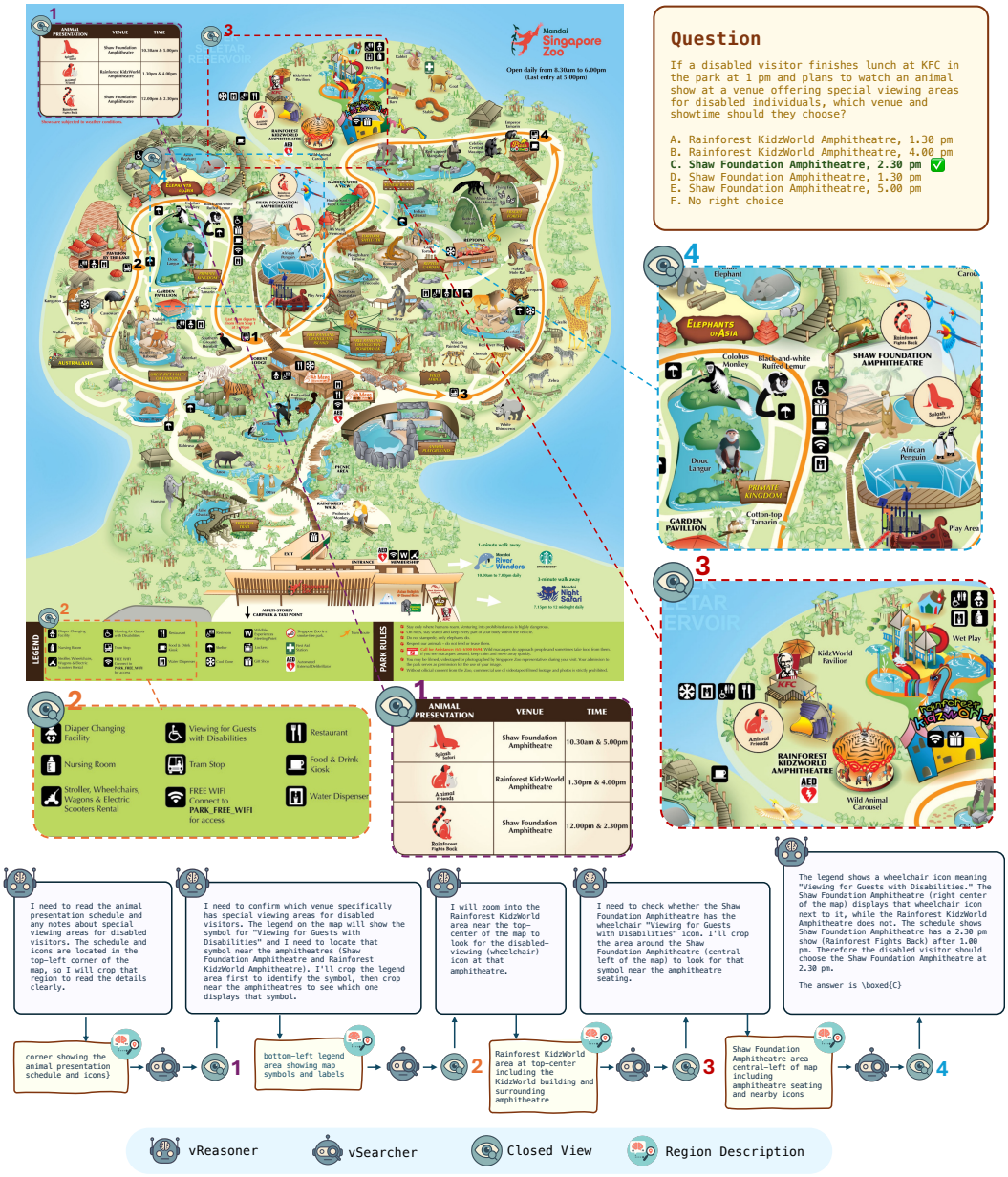


Figure 15: Qualitative result 2 with GPT-5-mini as vReasoner and INSIGHT-O3-VS as vSearcher. The reasoner requests venue-level cues (*e.g.*, legend/index lookups); the searcher returns localized regions and snippets, iterating to a correct answer.

F Comparative Analysis between InSight-o3 and Baselines

In Figure 16-19, we compare the behavior of InSight-o3 (GPT-5-mini + InSight-o3-vS) with two baselines: (1) GPT-5-mini and (2) GPT-5-mini + Qwen2.5-VL-7B. This comparative analysis is based on three examples of O3-Bench. We rate each crop returned by vSearcher on three levels:

- High-quality crops tightly enclose the visual search targets and the relevant context.
- Medium-quality crops contain the visual search targets but include too much context.
- Low-quality crops miss or truncate the visual search targets or relevant context.

In the most basic setting where vReasoner does not have access to vSearcher, it often uses similar reasoning patterns as follows to reach its conclusion: "I first locate ... I see ... I then look for ... There is ... Therefore ..." (see the top parts of Figure 17-19). During this process, vReasoner often hallucinates and makes factual errors about what it sees, suggesting that it does not really see the relevant visual details clearly but still pretends so anyway.

With the vanilla Qwen2.5-VL-7B vSearcher, vReasoner is able to take closer looks at regions of interest and makes less factual errors (see the bottom part of Figure 17 and the middle part of Figure 18-19). However, the vanilla vSearcher is often unreliable, returning inaccurate/wrong crops to vReasoner or simply concluding that the target is not in the image, usually after a minimal amount of (sometimes none) reasoning. In such cases, vReasoner would eventually give up and resort to its own perception after multiple failed visual search attempts, leading to wrong final answers.

In comparison, our visual search agent, InSight-o3-vS, would usually first reason about the visual search request and then crop the candidate region to verify before returning it to vReasoner. For example, in Figure 16, InSight-o3-vS first reasons about what the bounding box should cover:

```
Based on the description, the right section of the map includes the
legend/index and the "Catering venues" list with numbered cafes. The
legend/index is located at right of the map, and the "Catering venues"
list is further down, under the "Catering venues" heading. The bounding
box should cover these areas.
```

Then, after viewing the cropped region, it concludes:

```
Based on the tool response, the right section of the map showing the
legend/index and the "Catering venues" list with numbered cafes is
already covered by the bounding box provided. Therefore, no further
zooming is necessary.
```

In some cases, InSight-o3-vS is able to correct an initial bad crop after reviewing the crop:

```
Based on the response, the zoomed-in area does not match the intended
area around grid B4. The bounding box needs to be adjusted to better
capture the car parks and nearby cafe icons and labels.
```

The returned crops are usually medium-to-high-quality crops as shown in Figure 16. From these cases, we can see that vSearcher mostly helps vReasoner by (i) precisely locating the regions requested by vReasoner and (ii) presenting them to vReasoner and directing its attention to those regions, thereby reducing hallucination and facilitating evidence-based reasoning.

G Failure Cases of InSight-o3

Figure 20-21 show typical failure cases of InSight-o3. In the first three failure cases, vSearcher (InSight-o3-vS) provided the correct crops for the search targets but vReasoner (GPT-5-mini) answers incorrectly due to its own errors, e.g., ignoring visual evidence due to internal knowledge bias, and jumping to conclusion without examining key visual information. This suggests that existing frontier models, even the proprietary ones, are still not very good at thinking with images in complex scenarios. The last failure cases is due to vSearcher failing to understand the structure of the map and/or the relatively complicated region descriptions involving multiple visual cues. Finally, we note that, at least on O3-Bench, most wrong answers are due to vReasoner, not vSearcher.

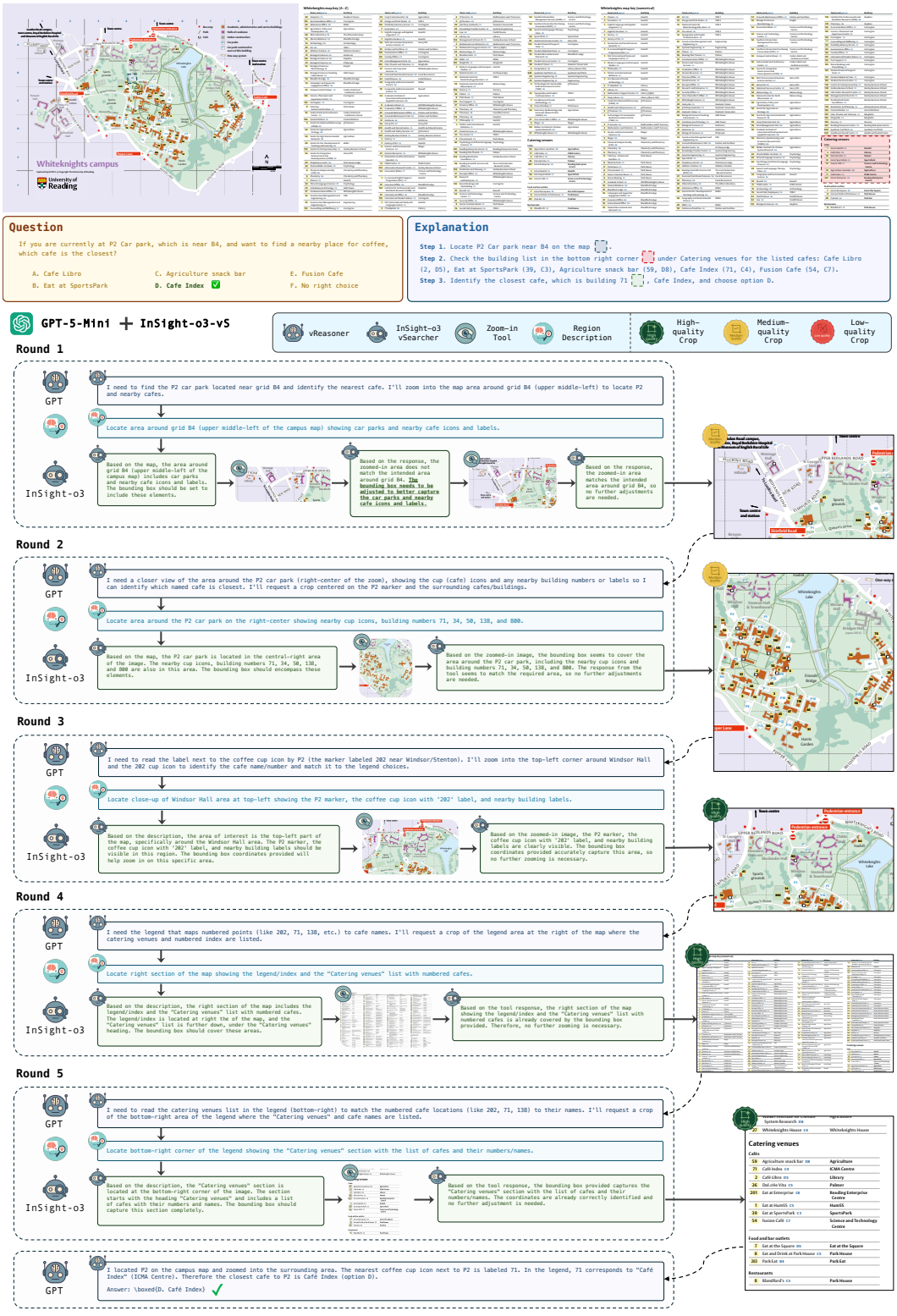


Figure 16: Example 1 (MAP case 1) of complete reasoning trace of our model (GPT-5-mini as vReasoner and INSIGHT-O3-VS as vSearcher). The vReasoner iteratively guides vSearcher through natural-language region descriptions. INSIGHT-O3-VS precisely retrieves high-quality image crops that match the described regions, effectively supporting vReasoner in locating the target area and producing the correct answer.

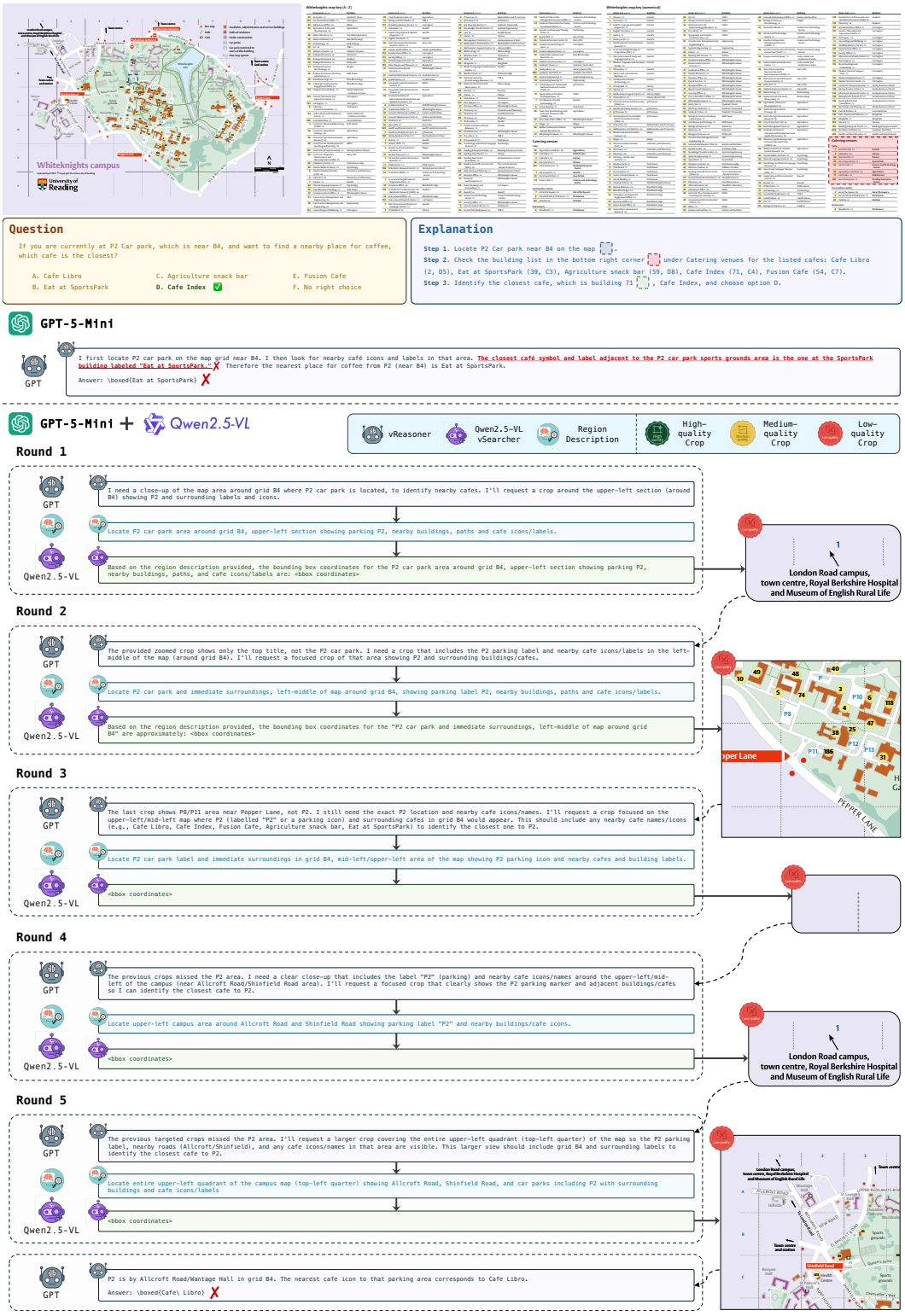


Figure 17: Example 1 (MAP case 1, continuation of Fig. 16): reasoning traces of GPT-5-mini and GPT-5-mini + Qwen2.5-VL-7B-Instruct. GPT-5-mini exhibits misperception and reasoning drift, while Qwen2.5-VL-7B-Instruct frequently fails to follow vReasoner's instructions, producing low-quality crops misaligned with the described regions. Consequently, both baselines yield incorrect answers.

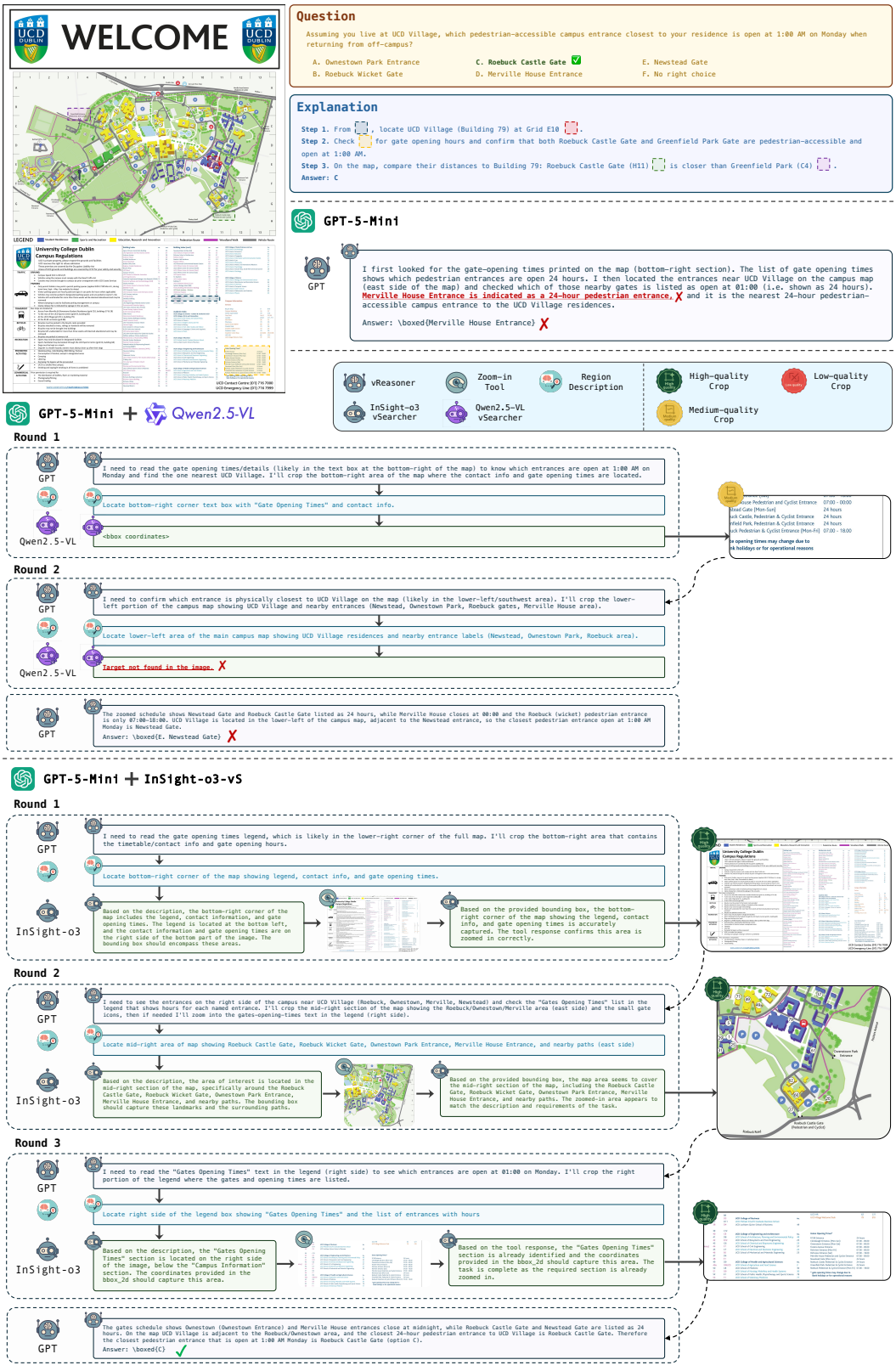


Figure 18: Example 2 (MAP case 2): reasoning traces of our model, GPT-5-mini, and GPT-5-mini + Qwen2.5-VL-7B-Instruct. Our INSIGHT-O3-VS accurately follows vReasoner's instructions and returns high-quality crops aligned with the described regions, leading to a correct answer. In contrast, Qwen2.5-VL-7B fails to return a valid crop in the final reasoning round, resulting in an incorrect answer.

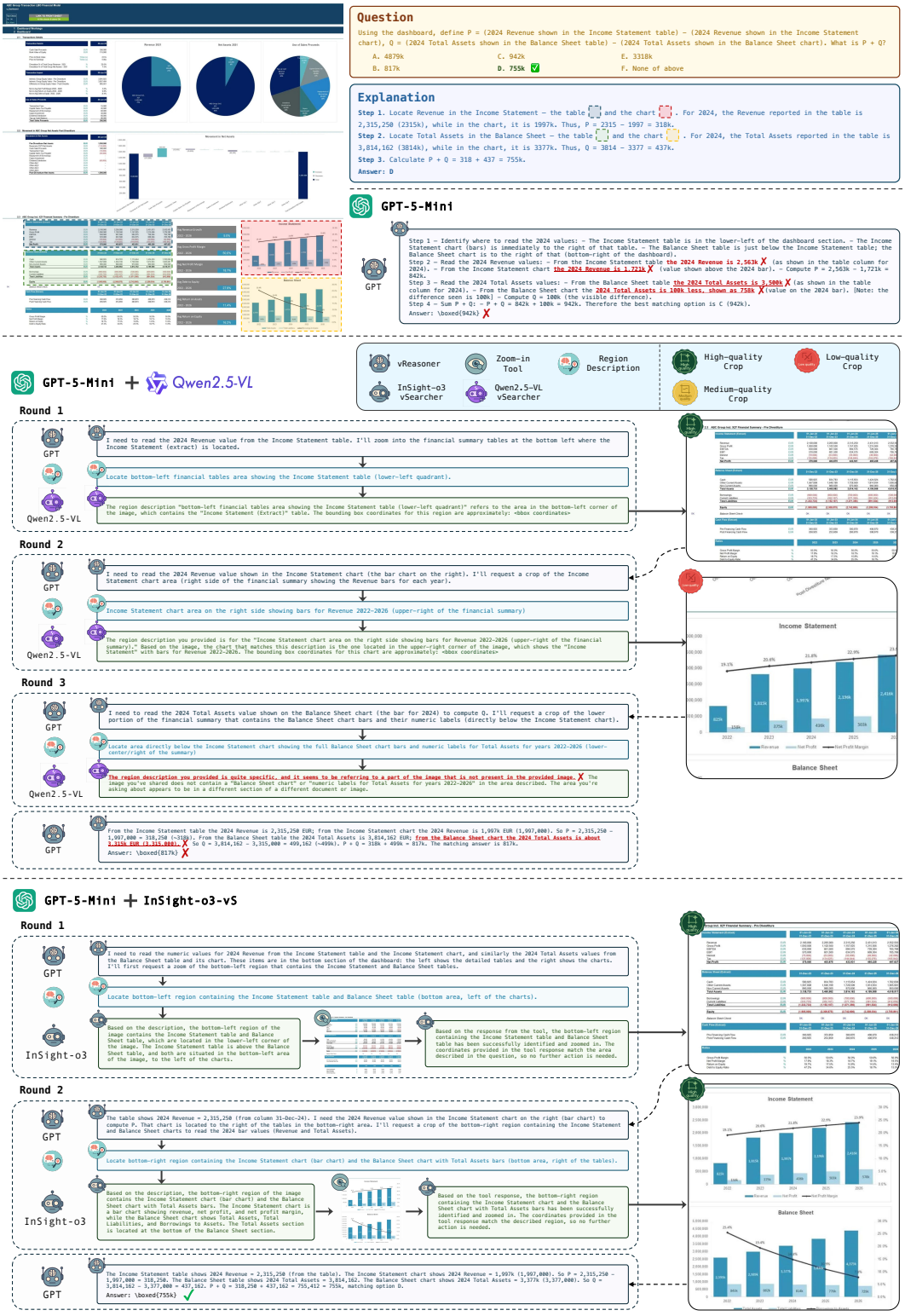


Figure 19: Example 3 (CHART case): reasoning traces of our model, GPT-5-mini, and GPT-5-mini + Qwen2.5-VL-7B-Instruct. INSIGHT-O3-VS effectively follows vReasoner's guidance and retrieves high-quality crops that fully capture the described regions. In contrast, Qwen2.5-VL-7B returns only partial crops in Round 2 and, in the Round 3, fails to produce a valid crop as requested by vReasoner, incorrectly concluding that the target region is absent, which leads to an incorrect answer.

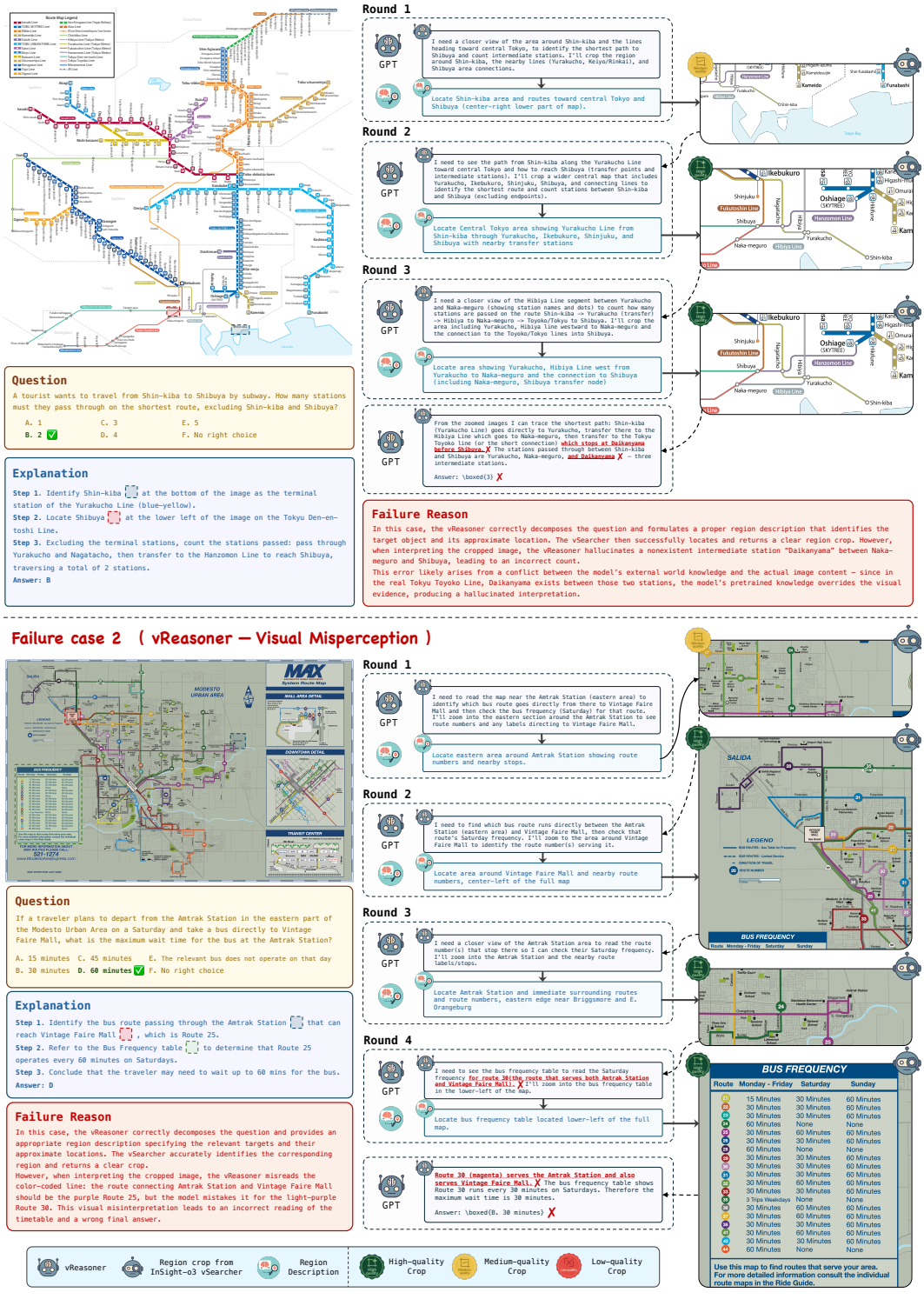


Figure 20: Failure cases 1 & 2 of INSIGHT-O3 (GPT-5-mini + InSight-o3-vS).

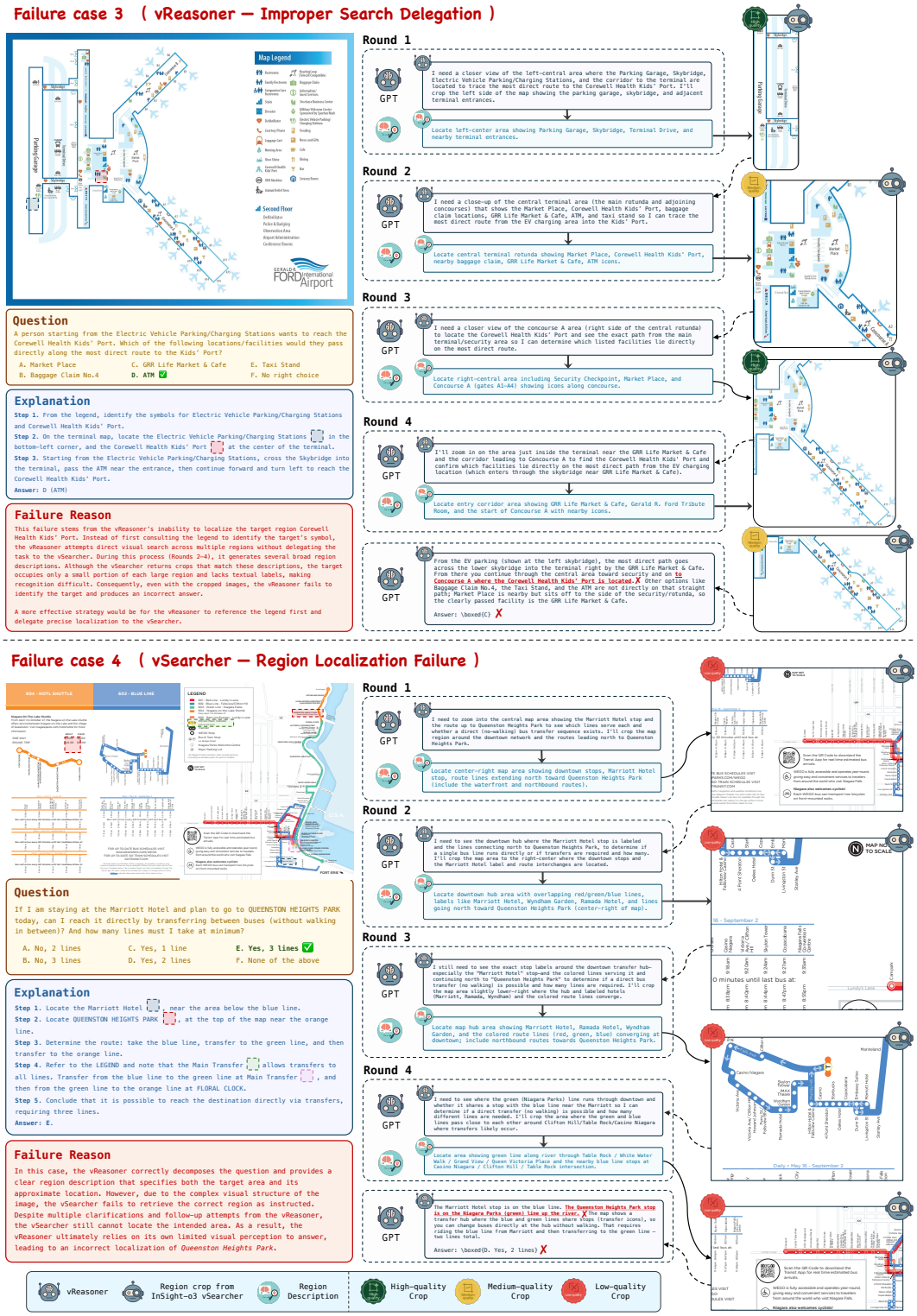


Figure 21: Failure cases 3 & 4 of INSIGHT-O3 (GPT-5-mini + InSight-o3-vS).

H PROMPTS FOR O3-BENCH

H.1 PROMPTS FOR INFORMATION EXTRACTION

For both the full image and cropped images, we feed them into Qwen2.5-VL-32B (Bai et al., 2025) for information extraction using the same prompt as below.

```
### System

You are acting as a **precise visual information extractor**.
Given ONE image, you must (1) identify the image type, (2) write a **comprehensive,
strictly factual** caption, and (3) extract **complete OCR text** when present (with
special handling for tables).
Follow the rules **exactly** and return the output in the three sections shown under **
Output Format**.

---

### Global Principles

1) **No hallucinations.** Describe only what is visible. If something is unclear, write `[
illegible]` or `[partially obscured: ...]`.
2) **Be exhaustive.** Do not omit small text, legends, tick labels, footnotes, watermarks,
axis titles, subtitles, panel labels (e.g., `(a)`, `(b)`), or figure notes.
3) **Preserve fidelity.** Copy punctuation, capitalization, diacritics, signs (`+`, `-`),
units, and spacing **exactly**. Do not normalize numbers or rewrite text.
4) **Reading order.** When listing text outside of tables, use **top-to-bottom, left-to-
right** order.
5) **Language.** OCR text must remain in the **original language(s)**. The caption should
be in English unless the visible UI/page language is clearly not English; in that case,
keep captions in that language. Do **not** translate OCR unless the image itself contains a
 translation.
6) **No extra sections.** Output **only** the three required sections and nothing else.

---

### Image Type Identification (Section 1)

- Classify the image using one or more of the following types (multiple allowed if
appropriate):
  `chart`, `table`, `document/text page`, `diagram/flowchart`, `map`, `UI/screenshot`, `
  form`, `invoice/receipt`, `poster/flyer`, `scientific figure (multi-panel)`, `natural
  scene`, `legend`, `infographic`, `other (specify)`.

---

### Detailed Caption (Section 2)

Write a **dense, structured** caption that covers all critical elements. Use clear,
objective language and organize logically (left->right, top->bottom; or foreground->
background). Include the relevant sub-guidelines:

**A. Charts / Plots / Scientific Figures**
- State the figure title (if present), chart type(s), axes titles, **units**, tick labels,
gridlines, data series, markers/line styles, **legend and color/shape mappings**,
annotations, error bars, trend lines, and notable extrema/patterns (peaks, troughs,
monotonic trends, outliers).
- If multi-panel: identify panel labels `(a)`, `(b)`, ... and summarize each panel in order
.
- Mention any insets, callouts, footnotes, or sources.

**B. Tables / Forms / Receipts / Documents**
- Summarize what the table/document contains (topics/fields), approximate dimensions (e.g.,
 `~12 rows x 6 columns`), header rows, merged cells, checkboxes, stamps, signatures, page
numbers, and footers/footnotes.
- Call out key sections (headings, lists, paragraphs), logos, and seals.

**C. Maps**
- Report title, compass/north arrow, scale/scale bar, coordinate grid or lat/long,
boundaries, regions, routes/lines with **color-to-meaning mapping** (from legend), symbols/
icons (e.g., hospitals, stations), labels for places/roads, insets, and any zoning/heat
color ramp.
- Include legend content (categories and their visual encodings).

**D. UI / Screenshots**
```

```
- App/site name, window title, menus/toolbars, visible controls (buttons, toggles,
checkboxes, dropdowns, search fields), selected/disabled states, scroll position,
timestamps, status bars, notifications, dialogs, visible file paths, and version strings.

**E. Natural / Real-World Scenes**
- Enumerate salient objects, text on signs/labels, relative positions (e.g., ``a red sign
above the doorway"), counts for repeated items, conditions (day/night, indoor/outdoor), and
 visible brands/logos.

> Do **not** invent interpretations or causal explanations. Keep to what is visually
supported.

---

### OCR Extraction (Section 3)

Extract **all visible text**. Follow these rules:

**General OCR Rules**
- Use **natural reading order** (top->bottom, left->right).
- Preserve original line breaks and spacing.
- If text is repeated (e.g., in a watermark), list it once and note `(repeats)` if
necessary.
- If a character/word is uncertain, write it as `[illegible]` or `[?]` without guessing.

**Tables (very important)**
- When an area is a table, extract it immediately using **Markdown table syntax**.
- Preserve the **exact** row/column structure and header rows; if a cell has line breaks,
use `<br>` or `\n`.
- For merged cells, repeat the visible text in each affected cell and note `(merged)` once
after the table.
- If multiple tables exist, label them sequentially as `Table 1`, `Table 2`, ... in the
order they appear.

**Documents / Text Pages / Forms**
- Extract headings, paragraphs, lists, captions, footnotes, headers/footers exactly as
shown. Maintain indentation and list markers.

**Charts / Maps / Diagrams / UI / Natural Scenes**
- Extract **all textual elements** present: titles, subtitles, axis labels, tick labels,
legend entries, series labels, annotations, callouts, map labels (places/roads/lines), UI
labels (menu items, buttons, tooltips), signs, badges, and watermarks.

If **no text** is present, write `None`.

---

### Output Format (return EXACTLY this Markdown structure; no code fences)

## Image Identification
<one or more types from the allowed list; add brief justification if mixed>

## Detailed Caption
<dense, strictly factual caption covering all visible elements per rules above>

## Extracted OCR (if any)
<EITHER: full Markdown tables + remaining text in reading order; OR: all non-table text in
reading order; write "None" if no text>
```

H.2 Prompts for Constructing O3-Bench

For chart images, we use the following prompt for GPT-5 (OpenAI, 2025a) to automatically generate QA instances.

```
You are an expert assessment-item author who designs rigorous **multi-hop visual-reasoning
** questions to benchmark "think-with-images" abilities on **dense diagrams, charts, tables
, and schematics**. The goal is to generate items so challenging that today's strongest
MLLMs score **<= 50%** without external tools.

You will receive:

### INPUTS
1. **GLOBAL_OCR** - OCR or caption text describing the entire image/page.
   {GLOBAL_OCR}
2. **GLOBAL_CAPTION** - caption text describing the entire image/page.
   {GLOBAL_CAPTION}
```

```
3. **LAYOUTS** - a list of cropped regions. Each layout contains:
   - `layout_id` - unique numeric ID (for your internal reference only).
   - `caption_or_ocr` - OCR or descriptive caption of the cropped region.
   {LAYOUTS}

> **Important constraint:**
> In the **question** and **options**, you must use only natural labels/text present in the
 `GLOBAL_OCR` or `GLOBAL_CAPTION` or `caption_or_ocr`.
> **Never** mention `layout_id`, `region`, `crop`, `panel`, `box`, or any similar tokens.
Layouts are **reference-only** to help you construct questions; they may be cited in the **
explanation** but not in the question or options.

---

## TASK
Create **3-5 independent multiple-choice questions** that each requires **>=2 distinct
visual hops** across layouts or global text. These must be **multi-step items directly**,
not derived from single-step seeds. All facts must be grounded in the provided OCR/captions
 only.

---

## QUESTION-DESIGN RULES
1. **Multi-hop inference (required)**
   Each question must integrate information from at least **two different layouts** or from
    global + local text. Valid patterns include:
   - Cross-table lookup & join (match category in Layout A to code/key in Layout B, then
   filter by condition).
   - Table <-> chart alignment (map series/labels from one layout to another).
   - Diagram <-> table mapping (use schematic labels to query corresponding rows/values).
   - Temporal alignment (identify when a threshold is crossed in one chart, then fetch
   related info from another).
   - Proportions/ratios/ranks (compute shares from one region and compare with targets in
   another).
   - Exceptions/constraints (apply footnote conditions from one layout before interpreting
   values elsewhere).

2. **Analytical realism**
   Situate each question in a plausible scenario (finance, science, education, operations,
   product metrics, etc.) while remaining strictly grounded in the provided OCR/captions.

3. **Difficulty control**
   Questions should require careful scanning, cross-referencing, and light calculations (
   differences, ratios, ranking). Avoid items that can be answered at a glance.

4. **Units, scales, rounding**
   Always follow the units/scales given in the OCR. If rounding is necessary, state the
   rounding rule in the **explanation**.

5. **Ambiguity guardrails**
   Ensure exactly **one correct choice** among A-E, unless `F` ("No right choice") is
   correct. Adjust conditions to avoid ties.

6. **Label fidelity**
   Copy text exactly as provided (case, spelling, diacritics). Never use external/world
   knowledge.

---

## ANSWER-OPTION RULES
1. Provide **exactly six** options, one per line, labeled `A.` ... `F.`
2. `F.` must always be exactly `No right choice`.
3. Place the correct answer randomly among `A.`-`E.`; <=10% of items may have `F` as the
correct answer.
4. Distractors must be plausible and drawn from actual text/numbers in the OCR/captions.
5. Make options mutually exclusive' no meta-options like "All of the above".

---

## EXPLANATION RULES
- Provide a **step-by-step chain**: `Step 1: ...`, `Step 2: ...`, etc.
- Explicitly cite which layouts were used as `[layout X]`.
- Show all computations (e.g., "(132 - 95) / 95 = 0.389 = 38.9% [layout 4]").

---

## OUTPUT FORMAT
Return **only** the following JSON array - no extra text, no markdown outside the code
block, no commentary:
```

```
```json
[
  {{
    "question": "...",
    "options": "A. ...\nB. ...\nC. ...\nD. ...\nE. ...\nF. No right choice",
    "answer": "C",
    "explanation": "Step 1: ... [layout 2]\nStep 2: ... [layout 5]\nStep 3: ..."
  }},
  ...
]
```
```

For map images, we use the following prompt for GPT-5 (OpenAI, 2025a) to automatically generate QA instances.

```
You are an expert evaluation-item author who designs rigorous **multi-hop visual-reasoning
** questions to benchmark "think-with-images" abilities on **maps** (bus/metro networks,
terminals, malls, festivals, parks, etc.). Items should be challenging enough that today's
strongest MLLMs achieve **<= 50%** accuracy.

You will receive:

### INPUTS
1. **GLOBAL_OCR** - OCR or caption text describing the entire map.
   {GLOBAL_OCR}
2. **GLOBAL_CAPTION** - caption text describing the entire map.
   {GLOBAL_CAPTION}
3. **LAYOUTS** - a list of cropped map regions. Each layout contains:
   - `layout_id` - unique numeric ID (for your internal reference only).
   - `caption_or_ocr` - OCR or descriptive caption of the cropped region.
   {LAYOUTS}

> **Hard requirement:** In the **question** and **options**, you must use only natural
labels/text found in `GLOBAL_OCR` or `GLOBAL_CAPTION` or `caption_or_ocr`.
> Do **not** mention `layout_id`, `region`, `crop`, `box`, `panel`, or similar. Layouts are
 **reference-only** for your reasoning; you may cite them in the **explanation**, but never
 in the question or options.

---

## TASK
Generate **3-5 independent multiple-choice questions**.
Each question must require **>=2 distinct reasoning hops** that combine information across
different layouts or between global and local OCR. All facts must be image-grounded.

---

## 1) QUESTION-DESIGN RULES
1. **Multi-hop reasoning (mandatory).** Examples of valid hops:
   - Legend <-> line color <-> stop/zone.
   - Grid index <-> label <-> adjacency.
   - Level/floor marker <-> facility <-> inset.
   - Route <-> timetable <-> destination.
   - Symbol <-> restriction <-> path feasibility.
   - Distance/scale <-> number of segments <-> travel time.

2. **Image/OCR grounded only.** Do not use external/world knowledge.

3. **Diversity.** Vary question styles (routing, conditional reachability, transfer logic,
adjacency/containment, count/compare, scale-based).

4. **Difficulty target.** Avoid "at-a-glance" items. Require cross-checking, counting, or
lightweight calculation.

5. **Label fidelity.** Copy map labels exactly (case, spelling, diacritics).

6. **Uniqueness.** Ensure exactly **one correct answer** among A-E, unless F ("No right
choice") is deliberately correct.

7. **Units & scale.** If computing length/time/segments, use the map's own scales, symbols,
 or counts.

---

## 2) ANSWER-OPTION RULES
1. Provide **exactly six** options labeled `A.` ... `F.`.
```

```
2. 'F.' must always be 'No right choice'.
3. Normally, the correct answer is among A-E; only rarely (<10%) should 'F' be correct.
4. Distractors must be plausible, drawn from real map text/numbers, and mutually exclusive.
5. No meta-options ("All of the above").

---

## 3) EXPLANATION RULES
- Provide a **step-by-step reasoning chain**.
- Explicitly cite layouts used as '[layout X]'.
- Make hops and computations explicit (e.g., "Count 5 stops along Red Line [layout 3] and
compare to 4 stops in Zone B [layout 5]").

---

## 4) OUTPUT FORMAT
Return **only** the following JSON array-no extra commentary or markdown:

'''json
[
  {{
    "question": "...",
    "options": "A. ...\nB. ...\nC. ...\nD. ...\nE. ...\nF. No right choice",
    "answer": "B",
    "explanation": "Step 1: ... [layout 1]\nStep 2: ... [layout 4]\nStep 3: ..."
  }},
  ...
]
'''
```

H.3 Prompts for Evaluation

For the evaluation of API models, we adopt the commonly used thinking prompt as follows.

```
A conversation between User and Assistant. The user asks a question, and the Assistant
solves it.
The assistant first thinks about the reasoning process in the mind and then provides the
user with the answer.
The reasoning process and answer are enclosed within <think> </think> and <answer> </answer
> tags, respectively,
And the final answer letter is enclosed within \\boxed{{option letter}}.
i.e., <think> reasoning process here </think><answer> answer here \\boxed{{option letter}}
</answer>.
```

I Prompts for InSight-o3

I.1 vReasoner Prompts

We use the following prompts for vReasoner during training.

```
[SYSTEM]
You are a visual assistant. Your goal is to answer a question based on an image.

First, think step by step to identify which visual facts you need from the image to answer
the question. If the visual information is insufficient or unclear, call the visual search
tool by providing a concise region description:
<tool_call>region_description={...}</tool_call>

The tool will search the image and return a cropped view of the target region. You may
repeat this process until you have enough evidence to answer confidently. The tool is not
always precise -- evaluate its output critically. If it looks incorrect or off-target,
refine your description and try again.

Region description guidance:
- Use concise, visually grounded targets (e.g., a chart, an object, a text block, a
distinct area)
- Optionally include approximate location (e.g., top-left, bottom-right, center)
- Avoid non-visual or ordinal references (e.g., "the third largest bar", "the second row's
number")
- Describe only one region per tool call; do not request multiple regions in a single
description
```

```
Output format:
- Put your reasoning process inside <think>...</think>.
- Immediately after </think>, output your assessment of the most recent tool result (if any
) formatted as <tool_feedback>helpful/unhelpful</tool_feedback>.
  This should indicate whether the result returned by the previous tool call is relevant to
   your prior region description and helpful to answering the question. If it misses the
  key information you are looking for, it is unhelpful. If no previous tool result exists (
  e.g., the first turn), output <tool_feedback>NA</tool_feedback>.
- Immediately after </tool_feedback>, do exactly one of:
  1) Call the tool; or
  2) Provide the final answer (no tool call) -- include the result in \\boxed{...}. Do not
  mix tool calls and answers in the same turn.
- If you need to call the tool, provide the region description using the exact format <
tool_call>region_description={...}</tool_call>.
You must strictly follow the output format, otherwise your answer will be judged as wrong.

A multi-turn format example:
Assistant:
<think>{your step-by-step analysis; decide if more detail is needed}</think>
<tool_feedback>NA</tool_feedback>
<tool_call>region_description={concise, visually grounded target (optionally with location)
}</tool_call>

User:
[Zoomed-in image + guidance (e.g., "Based on your description, here is the zoomed-in image.
 Please continue your analysis; you may call the tool again or provide your final answer if
 sufficient.")]

Assistant:
<think>{updated analysis based on the zoomed-in view; decide whether to refine or answer}</
think>
<tool_feedback>unhelpful</tool_feedback>
<tool_call>region_description={next concise target (optionally with location)}</tool_call>

(Repeat the User -> Assistant pattern as needed until enough evidence is gathered.)

Assistant (final turn):
<think>{final reasoning; explain why the available visual evidence is sufficient}</think>
<tool_feedback>helpful</tool_feedback>
Answer: \\boxed{...}

[USER]
{question}<image>

[ASSISTANT]
...

[USER]
Based on your description, here is the zoomed-in image.

Please continue your analysis. After the analysis, state your assessment of the previous
tool result using <tool_feedback>helpful/unhelpful</tool_feedback>, then do one of the
following:
- Call the tool again if you believe more visual detail is needed; or
- Provide your final answer if the current information is sufficient.
<image>

[ASSISTANT]
...

[USER]
Based on your description, here is the zoomed-in image.

You have reached the limit for using the visual tool and cannot call it again.
In this turn, after reasoning step by step, output your assessment of the previous tool
result using exactly <tool_feedback>helpful/unhelpful</tool_feedback>, and then, based on
the available information, provide your final answer using the required format.
<image>

[ASSISTANT]
...
```

In case that vSearcher is unable to find a region that matches the region description provided by vReasoner, we use the following user prompt to notify vReasoner.

```
[USER]
The visual searcher could not locate the requested target in the image based on your
description.

Please adjust or refine your region description (for example, refer to a larger, clearly
visible area) and continue your analysis. Think first, then state your assessment of the
previous tool result using <tool_feedback>helpful/unhelpful</tool_feedback>. Finally, do
exactly one of the following:
- Call the tool again with a revised description; or
- Provide your final answer if the current information is sufficient.
```

Occasionally, vReasoner may fail to follow the format instructions. When this happens, we use the following user prompt to ask vReasoner to generate a new response:

```
[USER]
In your previous response, neither a tool call nor a final boxed answer was detected, or
you didn't output your assessment of the previous tool result in the correct format.

Think first, and then include your assessment of the previous tool result using exactly <
tool_feedback>helpful/unhelpful</tool_feedback> (or <tool_feedback>NA</tool_feedback> if
there is no previous result). Finally, do exactly one of the following:
- If you still need more visual detail, call the tool using the exact format:
  <tool_call>region_description={...}</tool_call>
- Otherwise, provide the final answer now and include the result in \\boxed{...}.
```

During inference, we do not ask vReasoner to provide any feedback to the tool. The prompts are slightly simplified as follows:

```
[SYSTEM]
You are a visual assistant. Your goal is to answer a question based on an image.

First, think step by step to identify which visual facts you need from the image to answer
the question. If the visual information is insufficient or unclear, call the visual search
tool by providing a concise region description:
<tool_call> region_description={...} </tool_call>

The tool will search the image and return a cropped view of the target region. You may
repeat this process until you have enough evidence to answer confidently. The tool is not
always precise -- evaluate its output critically. If it looks incorrect or off-target,
refine your description and try again.

Region description guidance:
- Use concise, visually grounded targets (e.g., a chart, an object, a text block, a
distinct area)
- Optionally include approximate location (e.g., top-left, bottom-right, center)
- Avoid non-visual or ordinal references (e.g., "the third largest bar", "the second row's
number")
- Describe only one region per tool call; do not request multiple regions in a single
description

Output format:
- Put your reasoning process inside <think>...</think>.
- When you need to call the tool, you need to provide the region description using the
format <tool_call>region_description={...}</tool_call>.
- Immediately after each </think>, do exactly one of:
  1) Call the tool; or
  2) Provide the final answer (no tool call) -- include the result in \\boxed{...}. Do not
  mix tool calls and answers in the same turn.
You must strictly follow the output format, otherwise your answer will be judged as wrong.

A multi-turn format example:
Assistant:
<think>{your step-by-step analysis; decide if more detail is needed}</think>
<tool_call> region_description={concise, visually grounded target (optionally with location
)} </tool_call>

User:
[Zoomed-in image + guidance (e.g., "Based on your description, here is the zoomed-in image.
 Please continue your analysis; you may call the tool again or provide your final answer if
 sufficient.")]

Assistant:
```

```
<think>{updated analysis based on the zoomed-in view; decide whether to refine or answer}</
think>
<tool_call> region_description={next concise target (optionally with location)} </tool_call
>

(Repeat the User -> Assistant pattern as needed until enough evidence is gathered.)

Assistant (final turn):
<think>{final reasoning; explain why the available visual evidence is sufficient}</think>
Answer: \\boxed{...}

[USER]
{question}<image>

[ASSISTANT]
...

[USER]
Based on your description, here is the zoomed-in image.

Please continue your analysis. You may:
- Call the tool again if you believe more visual detail is needed; or
- Provide your final answer if the current information is sufficient.
<image>

[ASSISTANT]
...

[USER]
Based on your description, here is the zoomed-in image.

You have reached the limit for using the visual tool and cannot call it again.
In this turn, based on the available information, provide your final answer using the
required format.
<image>

[ASSISTANT]
...
```

When vSearcher can't find the target region, we use the following user prompt to notify vReasoner:

```
[USER]
The visual searcher could not locate the requested target in the image based on your
description.

Please adjust or refine your region description (for example, refer to a larger, clearly
visible area) and continue your analysis. You may:
- Call the tool again with a revised description; or
- Provide your final answer if the current information is sufficient.
```

When vReasoner fails to follow the format instruction described in the system prompt, we the following user prompt to ask vReasoner to generate a new response:

```
[USER]
In your previous response, neither a tool call nor a final boxed answer was detected.

Please do exactly one of the following:
- If you still need more visual detail, call the tool using the exact format:
  <tool_call>region_description={...}</tool_call>
- Otherwise, provide the final answer now and include the result in \\boxed{...}.
```

I.2 vSEARCHER PROMPTS

We use the following prompts for vSearcher during both training and evaluation after training. The prompts are adapted from DeepEyes (Zheng et al., 2025). For the last turn, we notify vSearcher in the user prompt that it has reached tool call limit.

```
[SYSTEM]
You are a helpful assistant.
```

```
# Tools
You may call one or more functions to assist with the user query.
You are provided with function signatures within <tools></tools> XML tags:
<tools>
{"type":"function","function":{"name":"image_zoom_in_tool","description":"Zoom in on a
specific region of an image by cropping it based on a bounding box (bbox) and an optional
object label.","parameters":{"type":"object","properties":{"bbox_2d":{"type":"array","items
":{"type":"number"},"minItems":4,"maxItems":4,"description":"The bounding box of the region
 to zoom in, as [x1, y1, x2, y2], where (x1, y1) is the top-left corner and (x2, y2) is the
 bottom-right corner."},"label":{"type":"string","description":"The name or label of the
object in the specified bounding box (optional)."}},"required":["bbox"]}}}
</tools>

# How to call a tool
Return a json object with function name and arguments within <tool_call></tool_call> XML
tags:
<tool_call>
{"name": <function-name>, "arguments": <args-json-object>}
</tool_call>

Example:
<tool_call>
{"name": "image_zoom_in_tool", "arguments": {"bbox_2d": [10, 20, 100, 200], "label": "the
apple on the desk"}}
</tool_call>

[USER]
<image>
Locate {target}.
Think first, call image_zoom_in_tool if needed, then answer with the bbox coordinates in [
x1, y1, x2, y2] format (or [0, 0, 0, 0] if you can't locate it). Format strictly as:  <
think>...</think>  <tool_call>...</tool_call> (if tools needed)  <answer>[x1, y1, x2, y2]</
answer> (otherwise)

[ASSISTANT]
...

[USER]
<tool_response><image></tool_response>
Think first, call image_zoom_in_tool if needed, then answer with the bbox coordinates in [
x1, y1, x2, y2] format (or [0, 0, 0, 0] if you can't locate it). Format strictly as:  <
think>...</think>  <tool_call>...</tool_call> (if tools needed)  <answer>[x1, y1, x2, y2]</
answer> (otherwise)

[ASSISTANT]
...

[USER]
<tool_response><image></tool_response>
You have reached the tool call limit and cannot call tools anymore.
Think first, call image_zoom_in_tool if needed, then answer with the bbox coordinates in [
x1, y1, x2, y2] format (or [0, 0, 0, 0] if you can't locate it). Format strictly as:  <
think>...</think>  <answer>[x1, y1, x2, y2]</answer>

[ASSISTANT]
...
```

Without RL, Qwen2.5-VL-7B has difficulty following the instructions given in the above prompts. So, for evaluating vReasoner + Qwen2.5-VL-7B, we use the following simple prompt:

```
[SYSTEM]
You are a helpful assistant.

[USER]
Given an image and a region description, locate the region that best matches the
description and output its bounding box coordinates as [x_min, y_min, x_max, y_max].

If the target cannot be found, output [0, 0, 0, 0].

Region description:
{target}

Now, output the coordinates in format [x_min, y_min, x_max, y_max]:
<image>

[ASSISTANT]
...
```

I.3 Prompts for Answer Verification

For answer verification, we use GPT-5-nano (OpenAI, 2025a) and feed it with the following prompt.

```
You are given an image-based question, the ground truth (GT) answer, and a model's answer.

Compare the model's answer with the GT answer:

- If the model's answer matches the GT answer visually or semantically, reply with <correct
>.
- If it doesn't match, or if uncertain, reply with <wrong>.

Only reply with <correct> or <wrong>, no explanations.

Question: {question}
GT Answer: {gt_answer}
Model Answer: {model_answer}
```

J Evaluation Benchmarks

J.1 Natural-Image Benchmarks

- **V*-Bench** (Wu & Xie, 2024) is designed to test a model's ability to attend to high-resolution, detail-rich images. V*-Bench consists of 191 challenging natural images (sourced from the SA-1B Segment Anything dataset (Kirillov et al., 2023)) and focuses on two fine-grained visual tasks: attribute recognition (identifying specific object attributes like color or material) and spatial relationship reasoning (determining the relative positions of objects). By requiring accurate visual grounding of small details, V*-Bench exposes the limitations of models that rely on coarse image understanding.
- **Tree-Bench** (Wang et al., 2025a), like V*-Bench, uses high-quality, object-dense natural images (drawn from the SA-1B dataset) to evaluate fine-grained visual reasoning. However, Tree-Bench places additional emphasis on traceable evidence and complex reasoning. Each of its 405 visual question-answer pairs is annotated with bounding-box evidence for the correct answer, and many questions require second-order reasoning about object interactions or spatial hierarchies rather than simple identification.
- **VisualProbe** (Lai et al., 2025) pushes visual reasoning to an even harder regime. It features high-resolution images with very small target objects and many distractors, making it "super challenging" and necessitating iterative, trial-and-error search by the model. VisualProbe is organized into easy, medium, and hard subsets; VisualProbe-Hard denotes the toughest set of questions (106 in total) that often cannot be solved in a single glance. Compared to V*-Bench and Tree-Bench, VisualProbe-Hard scenarios demand an active visual search strategy: the model may need to zoom in on different regions or sequentially explore the image to find relevant details.

J.2 Mixed Benchmarks

- **HR-Bench** (Wang et al., 2025f) is a benchmark deliberately designed to evaluate models on ultra high-resolution images (up to 4K-8K pixels). It addresses a key limitation of prior multimodal tests (which max out at $\sim$2K resolution) by presenting tasks that cannot be solved with downsampled images. HR-Bench is split into two sub-task categories: (1) Fine-grained Single-instance Perception (FSP), with tasks like identifying detailed attributes of a single object, reading text via OCR, or responding to visual prompts on an image; and (2) Fine-grained Cross-instance Perception (FCP), which includes more complex multi-object challenges such as analyzing maps, interpreting charts/graphs, and assessing spatial relationships among multiple items. Each sub-task contains 100 queries on 8K-resolution images (with a downsampled 4K version also provided for efficiency).

Table 7: Comparison of O3-BENCH and related benchmarks.

Benchmark	# of QAs	Image domains	Layout/target boxes	Detailed explanations	Multi-hop	Avg. resolution
V*-Bench	191	natural (100%)	✓	✗	✗	2246×1583
Tree-Bench	405	natural (100%)	✗	✗	✗	2152×1615
VisualProbe$_{\text{Hard}}$	106	natural (100%)	✗	✗	✗	4944×3980
HR-Bench$_{4K}$	200*	natural (89%), chart (5%), map (6%)	✗	✗	✗	4024×3503
MME-RealWorld	∼29K	natural (>60%), chart (25%), map (6%), etc.	✗	✗	✗	2708×1844
O3-Bench	318	chart (43%), map (57%)	✓	✓	✓	4602×3967

* Roughly 200 distinct QA pairs. The original paper reported 800 but most are the same questions with scrambled options.

Table 8: Comparison of O3-BENCH and related benchmarks on chart & map.

Benchmark	Avg. resolution	Avg. acc. of GPT-5-mini	Avg. # of vSearch steps
HR-Bench$_{4K}$	4032×2509	79.6	2.3
MME-RealWorld	1875×1269	83.8*	1.0
O3-Bench	**4602×3967**	**38.1**	**2.9**

* Based on 500 random samples.

- **MME-RealWorld** (Zhang et al., 2024b) is a large-scale, comprehensive benchmark that evaluates models across a wide spectrum of real-world visual tasks. It comprises 13,366 high-quality images (average ∼2000×1500 resolution) and 29,429 QA pairs, spanning 43 distinct task types grouped into five real-world scenarios, curated from various datasets (Agustsson & Timofte, 2017; Liu et al., 2020; Zhang et al., 2021; Yang et al., 2023; Li et al., 2022; Sun et al., 2022; Sachdeva et al., 2024; Zhu et al., 2021; Jia et al., 2021). These scenarios cover diverse applications such as autonomous driving (*e.g.*, interpreting traffic scenes), video surveillance (*e.g.*, counting vehicles in an overhead street video), remote sensing (*e.g.*, identifying and counting tiny objects in satellite maps), sports and entertainment (*e.g.*, reading a scoreboard in a broadcast image), and others. MME-RealWorld is notable as the largest fully human-annotated multimodal benchmark to date. Given the benchmark's scale, we employ the official MME-RealWorld lite version[3] for efficiency, which uses a subset of 50 samples per task to speed up evaluation without significantly altering the task distribution.

J.3 COMPARISON OF O3-BENCH AND RELATED BENCHMARKS

In Table 7 and Table 8, we compare O3-BENCH with related benchmarks and highlight their key differences. Compared with the most closely related MME-RealWorld, we note that on the overlapping domains (i.e. chart & map): (i) the average image resolution of O3-BENCH is *significantly higher* (4602×3967 of ours *vs* 1875×1269 of MME-RealWorld); (ii) the average accuracy of GPT-5-mini on O3-BENCH is much lower (38.1% of ours *vs* 83.8% of MME-RealWorld); and (iii) the average number of visual search steps produced by InSight-o3 for O3-BENCH is *2.9*× that of MME-RealWorld. In addition, O3-BENCH provides layout boxes and detailed explanations for each QA pair, while most of the benchmarks do not. The explanations can help the community easily verify the correctness of the answers. These differences show that O3-BENCH is of exceptional quality and much harder to solve than the other related benchmarks.

Apart from quality, the scale of O3-BENCH is on par with most of the fine-grained perception multimodal understanding benchmarks, e.g., V*-Bench (191), Tree-Bench (405), VisualProbe-Hard

[3] https://huggingface.co/datasets/yifanzhang114/MME-RealWorld-Lite

(106), HR-Bench-4K (200*), and commonly-used benchmarks in other multimodal research areas; to list a few: Math: MathVision test-mini (304) (Wang et al., 2024), MathVista test-mini (1K) (Lu et al., 2023); VQA: RealWorldQA[4] (765); Embodiment/Spatial Understanding: ERQA (400) (Team et al., 2025a), RefSpatial-Bench (277) (Zhou et al., 2025); Agent: MIA Bench (400) (Qian et al., 2024), OSWorld (389) (Xie et al., 2024), AndroidWorld (116) (Rawles et al., 2024); Fine-grained Perception: V*-Bench. These benchmarks were used to evaluate the performance of Qwen3-VL[5]. As with O3-BENCH, these benchmarks are relatively small mainly because of the difficulty in data collection. Nevertheless, they have served the timely purpose of evaluating frontier models in the respective areas which are fast developing.

O3-BENCH only focuses on composite charts and digital maps for two main reasons. First, they are *representative* of most use cases of thinking with images in the digital domain (as opposed to the natural domain). More specifically, composite charts represent *structured* images (with clear delineations between different layout regions) and often require more *abstract* reasoning (e.g., computing the difference of two quantities). On the other hand, digital maps represent images with less structure and more *organic* layouts. They usually require more *visual* reasoning (e.g., finding the shortest route from location A to B). Together, we argue that O3-BENCH is generally sufficient for evaluating the thinking-with-image capability of current multimodal models in the digital domain. As for the natural domain, there are already a number of high-quality benchmarks for thinking with natural images (some are listed in Table 7). O3-BENCH precisely fills the gap of existing benchmarks while striking a balance between evaluation efficiency and generality.

[4] https://huggingface.co/datasets/xai-org/RealworldQA
[5] https://github.com/QwenLM/Qwen3-VL