ACTIVEFLY-BENCH: ALIGNING EMBODIED QUESTION ANSWERING WITH VISION-LANGUAGE-ACTION FOR AERIAL EMBODIED PERCEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unmanned Aerial Vehicles (UAVs) have emerged as promising embodied agents for perception due to their free 3D mobility. A key requirement for embodied perception is the ability to navigate to task-relevant regions, acquire informative observations, and complete the task through scene understanding. Prior work has largely focused on either language-guided UAV control in the physical world or high-level scene understanding in cyberspace, leaving a clear gap between the two. In this work, we introduce ActiveFly-Bench, the first benchmark that aligns cyberspace and the physical world for UAV embodied perception. It decomposes the problem into three hierarchical tasks: Aerial Embodied Question Answering (Air-EQA) for cyberspace reasoning, Fine-grained Language-Guided UAV Control (FLUC) for physical-world interaction, and Observation Behavior Planning (OBP) to bridge the two domains. The datasets are collected from diverse simulated and real-world environments, supporting imitation learning for aerial vision-language-action tasks. We further develop ActiveFly, a closed-loop UAV agent that integrates visual-language reasoning with fine-grained control, and deploy it on a physical UAV platform. Experiments with representative agentic frameworks show that current UAV agents still struggle with behavior planning and viewpoint adjustment in embodied perception. These results establish ActiveFly-Bench as a new testbed for embodied aerial intelligence. We release datasets and reproducible codes in the anonymous project page: `https://lvmolvmo.github.io/ActiveFly/`.

1 INTRODUCTION

Bridging cyberspace and the physical world is a core capability of embodied AI. Language-guided embodied perception (Liu et al., 2025) is one of the representative tasks. Given a natural language instruction such as *"Check what's behind the wall"*, the agent must actively explore the environment, adjust its viewpoint, and answer the question based on the acquired visual information. Such requirements make UAVs with free 3D mobility (Chen et al., 2024; Zhou et al., 2021; 2020) a well-suited platform.

Recent work has explored several related tasks in low-altitude aerial scenarios. A representative line of research is aerial vision-and-language navigation (VLN) (Liu et al., 2023c; Wang et al., 2024; Gao et al., 2025; Lee et al., 2025; Lin et al., 2025), which originates from ground robotics (Anderson et al., 2018; Krantz et al., 2020; Chen et al., 2019; Ku et al., 2020). These efforts primarily focus on improving the agent's instruction-following capability, requiring UAVs to interpret natural language instructions to navigate to a target location or search for a target object. Another related direction is Embodied Question Answering (EQA) (Das et al., 2018; Islam et al., 2024; Wijmans et al., 2019), in which an embodied agent answers open-vocabulary questions through active exploration of the environment.

While these tasks partially assess an agent's ability to actively acquire information through environmental interaction, they remain limited in evaluating UAV intelligence for embodied perception. (1) **Gap between physical control and high-level reasoning.** Existing EQA benchmarks either omit action execution (Zhao et al., 2025; Ren et al., 2024) or provide exploration trajectories without explicit task-driven motivation (Majumdar et al., 2024; Jiang et al., 2025). Conversely, aerial VLN (Liu et al.,

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



Figure 1: **Overview of ActiveFly-Bench.** The benchmark targets language-guided embodied perception for UAVs. It decomposes the task into three hierarchical and semantically aligned subtasks: Aerial Embodied Question Answering, Observation Behavior Planning, and Fine-Grained Language-Guided UAV Control. Given a high-level task such as "What is under the tree?", the UAV first performs reasoning and planning to infer the observation behavior, then executes control actions to reach an appropriate viewpoint, and finally answers the question based on the visual information.

2023c; Wang et al., 2025) emphasizes action prediction but lacks high-level EQA objectives, while its lengthy, structured instructions differ from realistic human–UAV interaction. (2) **Coarse-grained action control.** Current language-guided UAV tasks (Chen et al., 2026; Xu et al., 2026; Wang et al., 2025) focus on reaching a target location while neglecting final orientation. Yet embodied perception requires the UAV to adjust both its position and viewpoint to obtain the desired observation. (3) **Lack of real-world validation.** Most language-guided UAV systems are evaluated only in simulation. Although several pioneering works (Wu et al., 2025; Wang et al., 2025) demonstrate real-world control with short instructions, systems supporting language-guided embodied perception in real environments remain lacking.

To address these limitations, we introduce **ActiveFly-Bench**, a benchmark designed to systematically evaluate UAV intelligence in language-guided embodied perception tasks. We characterize such intelligence as the UAV's ability to reason about appropriate observation behaviors from high-level instructions and current observations, generate executable control signals, and accomplish the task using visual information gathered during execution, as demonstrated in Figure 1.

Following this paradigm, ActiveFly-Bench comprises three hierarchical tasks for embodied perception: Aerial EQA (Air-EQA), Observation Behavior Planning (OBP), and Fine-grained Language-guided UAV Control (FLUC). Air-EQA is a high-level task where task-critical visual evidence is unavailable from the current aerial perspective, requiring the UAV to adjust its viewpoint to acquire additional information. OBP requires the agent to infer explicit observation behaviors conditioned on the question in Air-EQA. FLUC further requires the agent to generate executable fine-grained actions based on the inferred observation behaviors. Compared with aerial VLN (Liu et al., 2023c) and VLA (Wang et al., 2025), FLUC enables realistic viewpoint-aware 5-DoF control by jointly modeling UAV-body and gimbal actions while constraining both final position and orientation.

To validate whether a UAV agent can perform embodied perception tasks in real-world environments, we design the **ActiveFly** agent and deploy it on a physical UAV platform in a ground–drone collaborative manner (Wang et al., 2025).

In summary, our contributions are as follows:

- We introduce **ActiveFly-Bench**, a benchmark for evaluating UAV intelligence in language-guided embodied perception. It decomposes the task into hierarchical, semantically aligned subtasks, enabling the evaluation of reasoning, planning, and action execution in a unified framework.
- We provide the first fine-grained language-guided UAV control (FLUC) dataset, which jointly models UAV-body and gimbal control for viewpoint-aware action execution. Built on FLUC, we further construct aerial embodied question answering (Air-EQA) and Observation Behavior Planning (OBP) to bridge high-level scene understanding and low-level control.

Table 1: **Comparison between existing benchmarks and ActiveFly-Bench.**

Method	Environment	Data Source	QA			Language-guided action control				Annotation
			Active	Planning	# QA	QA-aligned instr.	Action space	Traj. feature	# Traj	
EQA-v1 (Das et al., 2018)	Indoor	Simulation	✓	✘	5.0k	✘	2 DoF	Short, fine-grained	-	Rule-based
MP3D-EQA (Wijmans et al., 2019)	Indoor	Real World	✓	✘	1.1k	✘	2 DoF	Short, fine-grained	-	Rule-based
HM-EQA (Ren et al., 2024)	Indoor	Real World	✓	✘	500	✘	2 DoF	-	-	VLM
OpenEQA (Majumdar et al., 2024)	Indoor	Real World	✓	✘	1.6k	✘	2 DoF	Short, fine-grained	152	Human
EXPRESS (Jiang et al., 2025)	Indoor	Real World	✓	✘	2.0k	✘	2 DoF	Short, fine-grained	777	VLM
CityEQA (Zhao et al., 2025)	Outdoor	Simulation	✓	✘	1.4k	✘	4 DoF	-	-	Human
AerialVLN (Liu et al., 2023c)	Outdoor	Simulation	-	-	-	✘	4 DoF	Long, coarse-grained	100k	Human
OpenUAV (Wang et al., 2024)	Outdoor	Simulation	-	-	-	✘	6 DoF	Long, coarse-grained	12.6k	VLM
OpenFly (Gao et al., 2025)	Outdoor	Simulation	-	-	-	✘	4 DoF	Long, coarse-grained	100k	VLM
UAV-Flow (Wang et al., 2025)	Outdoor	Real. & Sim.	-	-	-	✘	4 DoF	Short, fine-grained	40k	Human
ActiveFly-Bench	In. & Outdoor	Real. & Sim.	✓	✓	3.6k	✓	5 DoF	Short, fine-grained	10k	Human

- We develop the **ActiveFly** agent and systematically evaluate representative VLMs and VLAs on the benchmark. Through failure-mode analysis, we identify key bottlenecks in current UAV embodied perception systems and validate the framework through real-world UAV deployment.

2 Related Work

2.1 Benchmark for EQA

The EQA task (Das et al., 2018; Majumdar et al., 2024; Yu et al., 2019; Li et al., 2026; Zhai et al., 2025b; Tan et al., 2023) typically requires agents to actively explore the environment by placing question-relevant semantics outside the initial field of view. However, many benchmarks (Wijmans et al., 2019; Yu et al., 2019; Cangea et al., 2019) do not provide reference trajectories, making it difficult to disentangle exploration from reasoning based on QA accuracy alone. Others rely on pre-specified memory (Majumdar et al., 2024; Zhai et al., 2025a; Datta et al., 2022), which weakens their ability to reflect active perception. Although benchmarks such as EQA-v1 (Das et al., 2018) and OpenEQA (Majumdar et al., 2024) provide both QA pairs and reference trajectories, they still lack the intermediate reasoning process that links a question to a plausible exploration trajectory. In addition, most existing EQA benchmarks are limited to closed indoor environments (Tan et al., 2020; Wu et al., 2024), restricting their ability to evaluate embodied intelligence in open outdoor settings. CityEQA (Zhao et al., 2025) extends EQA to urban outdoor scenes, but its lack of reference trajectories and the non-uniqueness of valid solutions make exploration efficiency hard to assess. Finally, most EQA benchmarks remain confined to simulated environments, with little or no system-level validation in the real world.

2.2 VLA for Aerial Navigation

For UAVs, the primary objective of VLA tasks is to navigate to a target location based on language instructions. Early work formulates this problem as Aerial VLN (Liu et al., 2023c; Gao et al., 2025; Lee et al., 2025; Wang et al., 2025; Xiao et al., 2025), which mainly evaluates an aerial agent's ability to follow long instructions and perform long-horizon, coarse-grained navigation. More recent efforts (Liu et al., 2026; Sun et al., 2026), such as VLA-AN (Wu et al., 2025) and UAV-Flow (Wang et al., 2025), focus instead on short-horizon, fine-grained action control. Although existing aerial VLA benchmarks have enabled deployment in the physical world and interaction with real environments, they still remain semantically disconnected from high-level tasks. A fundamental question is: after the agent follows an instruction and reaches a location, what can it actually do there?

To address this gap, we propose ActiveFly-Bench, which bridges high-level reasoning in cyber space and low-level execution in the physical world, enabling a more comprehensive evaluation of an agent's active exploration and semantic reasoning capabilities in real-world environments.

3 Problem Formulation

We decompose the language-guided embodied perception task into three fundamental and correlated subtasks: *Observation Behavior Planning (OBP)*, *Fine-grained Language-Guided UAV Control (FLUC)*, and *Aerial Embodied Question Answering (Air-EQA)*.

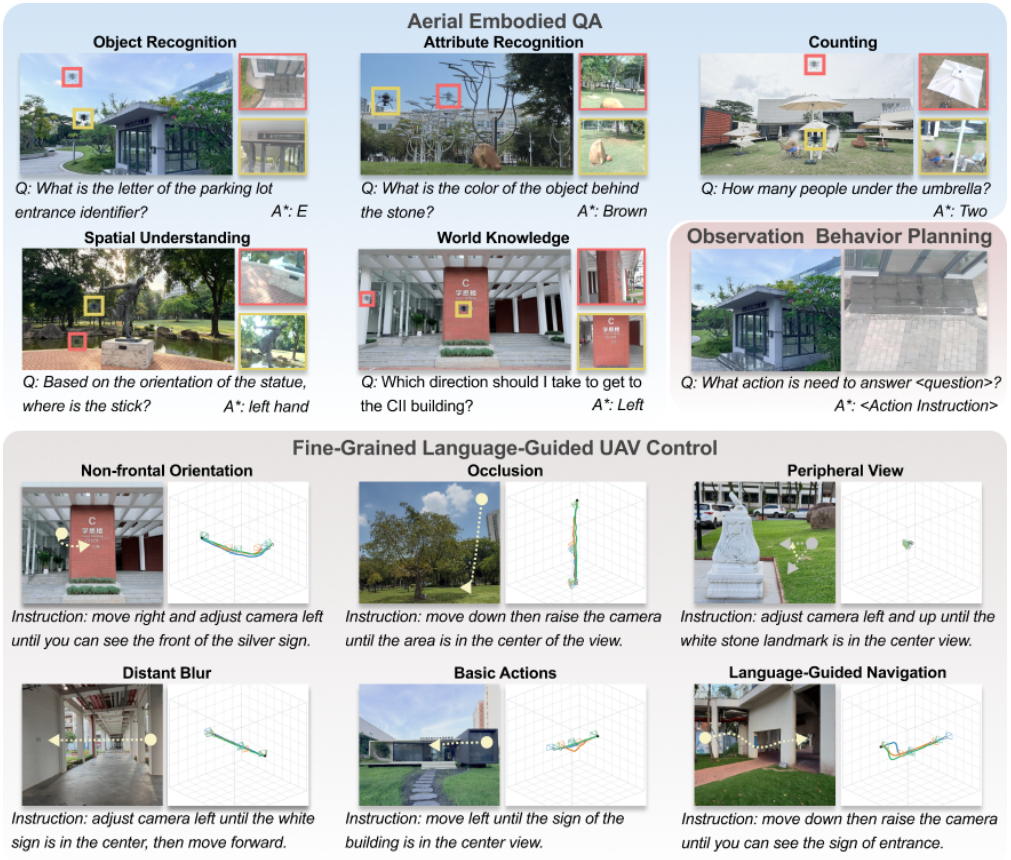

Figure 2: **Examples of the three tasks in ActiveFly-Bench.** Air-EQA spans five question categories: object recognition, attribute recognition, counting, spatial understanding, and world knowledge, with each example illustrated by the first and last frames. Observation Behavior Planning bridges Air-EQA and FLUC. FLUC includes 10 elementary and 4 complex active perception trajectories.

Air-EQA is the final task to be completed by the UAV agent, and serves as the high-level objective that drives embodied perception. Specifically, an Air-EQA instance is defined as a tuple (Q, A^*, O_0), where Q is an open-vocabulary question, A^* is the ground-truth answer, and O_0 denotes the agent's initial visual observation. The agent must explore the environment and produce an answer A based on its historical observations $O_{0:N}$, which is formulated as

$$A = \texttt{ActiveFlyAgent}(Q, O_{0:N}), \tag{1}$$

where N is the total number of exploration steps.

OBP serves as an intermediate step for solving Air-EQA. It is defined as a tuple $(O_0, Q, L^*_{\text{ob}})$, where L^*_{ob} denotes the observation behavior description annotated by humans. OBP evaluates whether the agent can translate high-level task intentions into task-relevant observation behaviors based on the current visual context. Formally, the agent predicts the observation behavior description as

$$L_{\text{ob}} = \texttt{ActiveFlyAgent}(Q, O_0). \tag{2}$$

FLUC corresponds to the low-level action execution stage conditioned on the observation behavior description from OBP. To better reflect real-world UAV embodied perception, unlike conventional UAV VLN (Liu et al., 2023c) or VLA (Wang et al., 2025) settings, we additionally introduce gimbal pitch control, enabling more fine-grained action execution and complex observation behaviors. The agent maps visual observations and textual instructions to embodied actions, similar to the standard VLA formulation (Kim et al., 2024; Intelligence et al., 2025; Black et al., 2024):

$$a_t = \texttt{ActiveFlyAgent}(L^*_{\text{ob}}, O_t, \mathbf{x}_t), \tag{3}$$

where $a_t \in \mathbb{R}^5$ consisting of 3D translational motion, yaw, and pitch. After executing N actions, the agent collects a history of observations $O_{0:N}$, which is used to answer the embodied question Q.

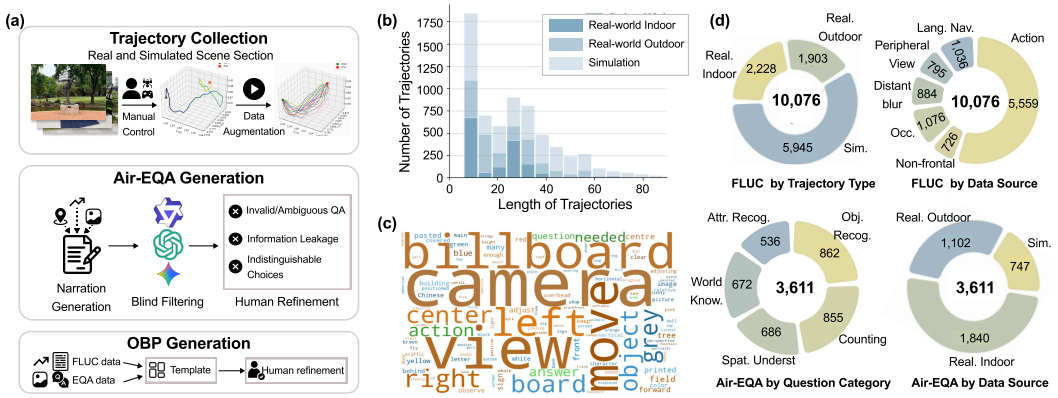

Figure 3: **Dataset Generation Pipeline and Statistics.** (a) The data generation pipeline of three tasks. (b) The distribution of FLUC trajectory lengths. (c) A word cloud generated from the language annotations. (d) Dataset composition statistics.

4 Dataset Collection and Validation

4.1 FLUC Data Collection

High-Quality Trajectory Collection. Each trajectory satisfies three criteria: (1) a well-defined target object or region; (2) the target is unclear from the initial view but clearly observable at the final view; and (3) the motion follows human intuition. We consider semantically meaningful targets, such as billboards, landmarks, and open spaces, under four representative constraints: *Occlusion*, *Non-Frontal Orientation*, *Peripheral View*, and *Distant Blur*.

Experienced pilots jointly control the UAV body and gimbal to reach informative viewpoints with minimal movement. Beyond these complex observation behaviors, we design ten elementary actions including forward/backward, left/right, up/down, turn left/right, and gimbal up/down to support fine-grained 5-DoF control learning. In total, we collect 10k trajectories and corresponding first-person videos across one real-world campus, one indoor environment, and two high-fidelity urban environments (Gao et al., 2024; Liu et al., 2023c).

Language Instruction Annotation. We adopt a hybrid annotation pipeline to generate language instructions for each trajectory and remove invalid samples. For elementary observation behaviors, we design instruction templates of the form "`<Action> until <target object> is in the center of the field of view`", which enables automatic generation of clear instructions by filling in the action type and target object. In contrast, complex observation behaviors often involve both translational motion and orientation adjustment. For these cases, we employ experienced annotators to produce unambiguous instructions.

Data Augmentation. To support sample-efficient imitation learning, we augment real-world trajectories following LIBERO (Liu et al., 2023a) and UAV-Flow (Wang et al., 2025). For each expert trajectory, we perturb the positions and orientations of the start, end, and intermediate waypoints with Gaussian noise to generate trajectories with similar observation behaviors but slight motion variations. A path planner then smooths the perturbed waypoints, which are replayed on the UAV to automatically collect 5–40 demonstrations for each instruction.

4.2 Air-EQA Data Collection

Question Generation. We build Air-EQA samples from high-quality trajectories in the FLUC dataset. Given the first-person observations along each trajectory, human annotators generate question–answer pairs based on the changes in visual semantics. We define five categories of Air-EQA questions to evaluate the UAV's abilities in (1) object recognition (e.g., "What is under the tree?"), (2) attribute recognition (e.g., "What is the color of the object under the tree?"), (3) counting (e.g., "How many people are beneath the pavilion?"), (4) spatial reasoning (e.g., "What is on the left side of the red tree?"), and (5) world-knowledge reasoning, as illustrated in Figure 2. To ensure the embodied

nature of the task, each question is constructed such that the task-critical visual evidence required to determine the answer is not directly observable from the initial viewpoint, but becomes clearly accessible after the UAV actively adjusts its position or camera viewpoint.

Data Validation. To ensure that Air-EQA requires active visual information acquisition, we employ three strong commercial VLMs as auxiliary leakage detectors with only the question and initial observation, and discard samples correctly answered by all three models. Human reviewers further remove ambiguous questions or samples where task-critical visual evidence is already observable from the initial viewpoint, and refine distractors to avoid trivial elimination from context. More details are in Appendix A.3.

4.3 Bridging the Gap Between FLUC and Air-EQA

Air-EQA does not explicitly model how the exploration should be conducted, resulting in exploration behaviors that often lack human-like priors. Thus, we construct the OBP dataset to guide the agent in reasoning about appropriate movements for completing Air-EQA tasks.

OBP Data Generation. Given the FLUC dataset and the corresponding Air-EQA annotations, annotators construct OBP question–answer pairs by reformulating the Air-EQA questions and FLUC instructions. We first design a template of the form: `Q: You are a UAV agent. You can control your motion, yaw, and gimbal pitch. How should you move to answer the question: <Air-EQA question>? A: <FLUC instruction>`. The generated OBP pairs are then refined by human annotators to remove information inconsistency and leakage.

4.4 Dataset Analysis

Trajectory and Instruction Analysis. As depicted in Figure 3, the proposed benchmark contains 10k FLUC trajectories, 3.6k Air-EQA pairs, and 1k unique OBP pairs. FLUC includes 5.8k simulated trajectories, 2.2k real-world indoor trajectories, and 1.9k real-world outdoor trajectories. Most trajectories range from 10 to 40 meters. The instruction lengths are mainly distributed between 15 and 30 words. Air-EQA questions are generally shorter, mostly ranging from 5 to 20 words. Each OBP instance corresponds to one Air-EQA instance.

Dataset Splits. Simulated FLUC trajectories are independently collected by human pilots and randomly split into 80% training and 20% testing sets. For real-world FLUC, expert and augmented trajectories are used exclusively for training, while testing is conducted through closed-loop rollouts from independently initialized UAV states, with 10 trials performed for each scene. All Air-EQA and OBP samples are reserved for evaluation.

5 ActiveFly Agent

In this section, we introduce **ActiveFly**, a closed-loop UAV system designed for language-guided active perception, and describe how its capabilities are comprehensively evaluated. We also present its real-world deployment on a physical UAV platform for active perception in real environments.

Closed-loop System and Real-world Deployment. ActiveFly couples a VLM (Singh et al., 2025; Comanici et al., 2025; Yang et al., 2025; Liu et al., 2023b) with a VLA (Kim et al., 2024; 2025; Intelligence et al., 2025) in a closed-loop framework. Given an embodied question and the initial observation, the VLM first generates a textual observation plan, which is then used as the language instruction for the VLA to predict fine-grained UAV actions conditioned on the current observation and UAV state. After trajectory execution, the VLM answers the question using $n = 16$ uniformly sampled observations from the collected history. For real-world deployment, we adopt a ground–drone collaborative architecture similar to UAV-Flow: the VLM and VLA run on an RTX A6000 server, while SLAM (Xu et al., 2022) and path planning (Zhou et al., 2020) are executed onboard. The UAV streams 1K first-person video and state information to the ground station over WiFi at 30 Hz and receives predicted actions for execution. To reduce control mismatch caused by communication and inference latency, we employ a simple *stop-and-infer* strategy, where the UAV waits for the next action after completing the previous one.

Table 2: Performance comparison of different baselines on EP, Air-EQA, OBP[1], and FLUC tasks.

Method	EP	Air-EQA		OBP	FLUC			
	SR	Acc.	APL	Acc.	SR	OSR	nDTW	NE
VLM + VLA								
Gemini-2.5-Pro + OpenVLA	10.4	74.8	62.3	70.4	11.7	15.1	15.7	9.0
Gemini-2.5-Pro + Pi-0.5	43.6	78.4	37.9	70.4	25.1	56.0	14.7	13.7
Qwen3-VL-Max + OpenVLA	13.4	77.4	64.8	63.9	14.4	16.8	**16.8**	9.3
Qwen3-VL-Max + Pi-0.5	43.8	79.7	36.7	63.9	21.3	54.6	14.6	14.4
GPT-5.6-Sol + OpenVLA	12.0	77.2	**65.3**	**72.5**	12.4	15.5	16.4	**8.3**
GPT-5.6-Sol + Pi-0.5	**46.3**	**81.7**	38.6	**72.5**	**25.4**	**57.4**	14.8	12.6
Human + VLA								
Human + OpenVLA	21.8	82.3	**58.3**	**99.1**	16.8	24.4	**23.1**	**6.0**
Human + Pi-0.5	**60.3**	**85.0**	41.2	**99.1**	**26.8**	**77.7**	16.9	10.1
VLM + Human								
Gemini-2.5-Pro + Human	60.5	80.2	69.3	70.4	71.8	72.9	70.9	7.4
Qwen3-VL-Max + Human	54.4	78.6	65.3	63.9	66.0	68.4	64.3	8.9
GPT-5.6-Sol + Human	**64.7**	**82.2**	**71.9**	**72.5**	**74.2**	**75.6**	**72.7**	**6.1**
Human Agent	94.3	98.2	91.3	99.1	89.1	95.6	77.9	5.4

6 EXPERIMENTS

We conduct a comprehensive evaluation of popular VLMs and VLA models on our benchmark. We adapt VLA models to the fine-grained action space of FLUC and evaluate them in simulated environments. We further evaluate ActiveFly through real-world closed-loop deployment and analyze failure modes to identify key bottlenecks in UAV active perception.

6.1 EXPERIMENTAL SETUP

Baselines. We construct different ActiveFly agent baselines by combining different VLM and VLA models. For VLMs, we select three commercial models with strong reasoning ability: GPT-5.6-Sol (Singh et al., 2025), Gemini-2.5 Pro (Comanici et al., 2025), and Qwen3-VL-Max (Bai et al., 2025). For VLA models, we adopt the representative discrete-action prediction model OpenVLA (Kim et al., 2024) and the continuous-action prediction model Pi-0.5 (Intelligence et al., 2025). We modify the output dimension of VLA models to adapt to the UAV action space. We further replace either the VLM or the VLA module with human agents to establish upper-bound performance.

Metrics. For Air-EQA, we use multiple-choice accuracy and Accuracy weighted by Path Length (APL) (Majumdar et al., 2024) to evaluate correctness and efficiency. For FLUC, we report Success Rate (SR), Oracle Success Rate (OSR), Navigation Error (NE), and normalized Dynamic Time Warping (nDTW). We define EP success as $\mathrm{S}_{ep} = \mathrm{S}_{obp} \cdot \mathrm{OS}_{fluc} \cdot \mathrm{S}_{eqa}$, where S_{obp}, S_{eqa}, and $\mathrm{OS}_{fluc} \in \{0, 1\}$ indicate OBP correctness, Air-EQA correctness, and FLUC oracle success, respectively. EP SR is the ratio of successful cases. We adopt this strict criterion to require success at all three stages, while separately analyzing escape cases in Section 6.4 to capture downstream success despite intermediate failure.

6.2 OVERALL PERFORMANCE

We report the end-to-end EP results along with the intermediate subtask results in Table 2. For end-to-end EP, Pi-0.5-based agents consistently achieve substantially higher SR than their OpenVLA-based counterparts across all VLMs, with improvements of over 30%. Meanwhile, all non-human agents obtain much higher Air-EQA accuracy than EP success rate, reflecting the compounded difficulty of jointly completing observation planning, action execution, and question answering. This gap is further amplified by the *Air-EQA escape* phenomenon, where an agent may answer the question correctly despite failing to exactly complete the intermediate OBP or FLUC objective, which we analyze in Section 6.4. Although Pi-0.5-based agents generally achieve higher Air-EQA accuracy,

[1]The same VLM yields the same OBP results.

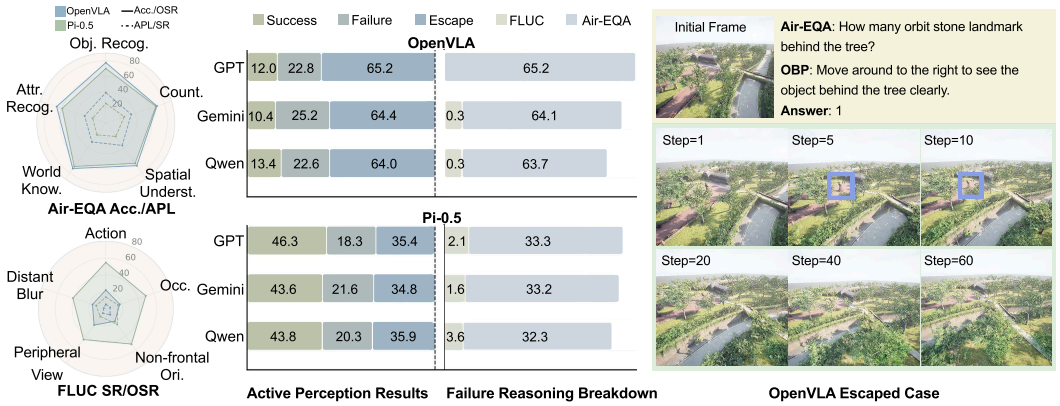

Figure 4: **Category-level performance (left), error breakdown (middle) and case study of an *"escaped"* case (right).** The escaped case shows that although the agent deviates from the reference trajectory, it can still acquire the critical visual information (purple box) to answer the question.

their APL is consistently lower than that of OpenVLA-based agents, suggesting that successful information acquisition often comes at the cost of longer exploration trajectories.

On OBP, GPT-5.6-Sol achieves the highest accuracy among the evaluated VLMs, followed by Gemini-2.5-Pro and Qwen3-VL-Max. For FLUC, Pi-0.5 consistently achieves higher SR and OSR than OpenVLA across different VLM-generated observation plans. However, Pi-0.5 produces lower nDTW and larger NE, suggesting weaker trajectory fidelity and less precise final stopping.

As presented in Table 3, text-only performance remains close to chance, while providing the initial frame improves accuracy to around 50%, indicating that VLMs can exploit partial scene context before movement. With full trajectory observations, all three models achieve around 80% accuracy, supporting that viewpoint adjustment provides additional task-relevant visual evidence for Air-EQA.

6.3 Category-Level Performance

In this section, we present the category-level performance of different VLA models across the three tasks. On Air-EQA, the Pi-0.5-based agent has lower APL than the OpenVLA-based agent across all question categories due to its longer exploration trajectory. On OBP, VLMs achieve substantially higher planning success rates in distant blur, peripheral view, and elementary trajectory settings than in occlusion and non-frontal orientation scenarios. On FLUC, Pi-0.5 achieves higher SR and OSR than OpenVLA across all subcategories, while performing comparably to OpenVLA in terms of nDTW. However, the gap between SR and OSR is larger for Pi-0.5, especially on simple tasks such as elementary trajectories. We attribute this to the tendency of Pi-0.5 trajectories to pass through the target position rather than stop precisely at it.

6.4 Error Analysis

We further analyze the outcome distribution across different VLM–VLA combinations. As shown in Figure 4, OpenVLA-based agents achieve EP SR of around 12%, with approximately 64% of the episodes falling into the escape category. In comparison, Pi-0.5-based agents achieve EP SR of around 45%, while reducing the escape rate to around 35%.

We further decompose the escape cases according to their source by tracking OBP, FLUC, and Air-EQA binary indicators. We define *FLUC escape* as cases where OBP fails but FLUC achieves oracle success, and *Air-EQA escape* as cases where FLUC fails but Air-EQA is correct. For OpenVLA, nearly all escape cases are associated with Air-EQA, accounting for over 63% of all episodes, while FLUC-related escape is negligible. A similar trend is observed for Pi-0.5, where Air-EQA escape accounts for about 32% of the episodes, compared with about 3% for FLUC-related escape. As illustrated by the OpenVLA example in Figure 4, the agent may deviate from the reference trajectory but still acquire sufficient task-relevant visual information to answer the embodied question correctly. This suggests that some trajectories failing the intermediate criteria can nevertheless provide useful

Table 3: Air-EQA accuracy under different observation settings.

VLM	Text-Only	Initial Frame	Full Frames
Gemini-2.5-Pro	23.4	54.2	80.2
Qwen3-VL-Max	24.6	51.5	78.6
GPT-5.6-Sol	22.5	50.2	82.2

Table 4: Real-world evaluation of a physical UAV platform deployed with OpenVLA and different VLMs.

VLM	EP SR	Air-EQA Acc.	OBP Acc.	FLUC SR	FLUC OSR
Gemini-2.5-Pro	39.5	79.4	82.8	21.4	45.2
Qwen3-VL-Max	35.8	89.0	89.5	22.0	46.3
GPT-5.6-Sol	44.0	77.3	77.0	21.4	45.8



Figure 5: **Visualization of real-world deployment of ActiveFly agent.** The agent comprises GPT-5.6-Sol and OpenVLA. The top and bottom rows show the first- and third-person views, respectively.

observations, reflecting the non-uniqueness of valid observation behaviors in active perception. More details and results are in Appendix A.5.4.

6.5 Real-World Validation

We present the real-world deployment and validation results of the ActiveFly Agent in real-world environments. To validate the feasibility of the complete ActiveFly pipeline on a physical platform, we deploy OpenVLA as the action policy and pair it with different VLMs for OBP and Air-EQA. The entire system is implemented in a ground–drone collaborative framework and operates in a closed-loop manner. As shown in Figure 5, the UAV agent can autonomously perform reasoning and fine-grained action control, and answer the embodied question. However, the EP SR remains limited, with physical action execution being the primary bottleneck as presented in Table 4.

We also report the system latency of a single control loop in Table 5. The latency is measured under a 5 Mbps Wi-Fi connection. The primary latency comes from image transmission between the UAV and the ground station, as well as VLA model inference. Nevertheless, the total latency of one control cycle remains within 1 second, indicating that the system can execute embodied perception tasks in a relatively smooth and responsive manner under common network conditions.

Table 5: **System latency breakdown of a single control loop.**

	Image transmission	VLA inference	Motion planning	PID response	Total
Latency	~ 120	~ 250	~ 10	~ 10	~ 390

7 Conclusion

In this work, we introduce ActiveFly-Bench, the first benchmark bridging high-level EQA and low-level action control for UAV embodied perception. It decomposes embodied perception into three hierarchical tasks and provides 10k FLUC trajectories together with 4.6k QA pairs for reasoning and planning evaluation. We further develop ActiveFly, a closed-loop UAV agent for real-world deployment. We hope ActiveFly-Bench will serve as a useful testbed for future research on UAV embodied perception.

AI USE STATEMENT

Generative AI tools were used to assist with code development for experiment execution and to refine the text for improved readability. The authors reviewed and verified all AI-assisted code and text. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed task formulations and benchmark definitions in Section 3, and describe the dataset collection, annotation, validation, and data splits in Section 4 and Appendix A.1–A.4. The implementation details of the physical UAV platform and navigation system are provided in Appendix A.2, while the evaluation metrics and real-world evaluation protocol are specified in Appendix A.5. To facilitate reproduction of the reported results, we provide the benchmark datasets, evaluation scripts, and implementation code through the anonymous project page: `https://lvmolvmo.github.io/ActiveFly/`.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

Cătălina Cangea, Eugene Belilovsky, Pietro Liò, and Aaron Courville. VideoNavQA: Bridging the gap between visual and embodied question answering. In *British Machine Vision Conference (BMVC)*, 2019.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12538–12547, 2019.

Jinyu Chen, Hongyu Li, Zongheng Tang, Xiaoduo Li, Wenjun Wu, and Si Liu. Aerialvla: A vision-language-action model for aerial navigation with online dialogue. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18161–18169, 2026.

Xuecheng Chen, Haoyang Wang, Yuhan Cheng, Haohao Fu, Yuxuan Liu, Fan Dang, Yunhao Liu, Jinqiang Cui, and Xinlei Chen. Ddl: Empowering delivery drones with large-scale urban sensing capability. *IEEE Journal of Selected Topics in Signal Processing*, 18(3):502–515, 2024.

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.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–10, 2018.

Samyak Datta, Sameer Dharur, Vincent Cartillier, Ruta Desai, Mukul Khanna, Dhruv Batra, and Devi Parikh. Episodic memory question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19119–19128, 2022.

Chen Gao, Baining Zhao, Weichen Zhang, Jinzhu Mao, Jun Zhang, Zhiheng Zheng, Fanhang Man, Jianjie Fang, Zile Zhou, Jinqiang Cui, et al. Embodiedcity: A benchmark platform for embodied agent in real-world city environment. *arXiv preprint arXiv:2410.09604*, 2024.

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, et al. Openfly: A comprehensive platform for aerial vision-language navigation. *arXiv preprint arXiv:2502.18041*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Md Mofijul Islam, Alexi Gladstone, Riashat Islam, and Tariq Iqbal. Eqa-mx: Embodied question answering using multimodal expression. In *International Conference on Learning Representations*, volume 2024, pp. 48079–48102, 2024.

Kaixuan Jiang, Yang Liu, Weixing Chen, Jingzhou Luo, Ziliang Chen, Ling Pan, Guanbin Li, and Liang Lin. Beyond the destination: A novel benchmark for exploration-aware embodied question answering. *arXiv preprint arXiv:2503.11117*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision – ECCV 2020*, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Jungdae Lee, Taiki Miyanishi, Shuhei Kurita, Koya Sakamoto, Daichi Azuma, Yutaka Matsuo, and Nakamasa Inoue. Citynav: A large-scale dataset for real-world aerial navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5912–5922, 2025.

Yifan Li, Yuhang Chen, Anh Dao, Lichi Li, Zhongyi Cai, Zhen Tan, Tianlong Chen, and Yu Kong. Industryeqa: Pushing the frontiers of embodied question answering in industrial scenarios. *Advances in Neural Information Processing Systems*, 38, 2026.

Peican Lin, Gan Sun, Chenxi Liu, Fazeng Li, Weihong Ren, and Yang Cong. Openvln: Open-world aerial vision-language navigation. *arXiv preprint arXiv:2511.06182*, 2025.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023b.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15384–15394, 2023c.

Xu Liu, Yu Liu, Hanshuo Qiu, Yang Qirong, and Zhouhui Lian. Indooruav: Benchmarking vision-language uav navigation in continuous indoor environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23864–23872, 2026.

Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, and Liang Lin. Aligning cyber space with physical world: A comprehensive survey on embodied ai. *IEEE/ASME Transactions on Mechatronics*, 2025.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16488–16498, 2024.

Allen Z Ren, Jaden Clark, Anushri Dixit, Masha Itkina, Anirudha Majumdar, and Dorsa Sadigh. Explore until confident: Efficient exploration for embodied question answering. In *Robotics: Science and Systems (RSS)*, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Jianli Sun, Bin Tian, Qiyao Zhang, Chengxiang Li, Zihan Song, Zhiyong Cui, Yisheng Lv, and Yonglin Tian. Air-vla: Vision-language-action systems for aerial manipulation. *arXiv preprint arXiv:2601.21602*, 2026.

Sinan Tan, Weilai Xiang, Huaping Liu, Di Guo, and Fuchun Sun. Multi-agent embodied question answering in interactive environments. In *European Conference on Computer Vision*, pp. 663–678. Springer, 2020.

Sinan Tan, Mengmeng Ge, Di Guo, Huaping Liu, and Fuchun Sun. Knowledge-based embodied question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 11948–11960, 2023.

Xiangyu Wang, Donglin Yang, Ziqin Wang, Hohin Kwan, Jinyu Chen, Wenjun Wu, Hongsheng Li, Yue Liao, and Si Liu. Towards realistic UAV vision-language navigation: Platform, benchmark, and methodology. *CoRR*, abs/2410.07087, 2024.

Xiangyu Wang, Donglin Yang, Yue Liao, Wenhao Zheng, Bin Dai, Wenjun Wu, Hongsheng Li, and Si Liu. UAV-Flow colosseo: A real-world benchmark for flying-on-a-word uav imitation learning. *arXiv preprint arXiv:2505.15725*, 2025.

Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied Question Answering in photorealistic environments with point cloud perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6659–6668, 2019.

Tao Wu, Chuhao Zhou, Yen Heng Wong, Lin Gu, and Jianfei Yang. Noisyeqa: Benchmarking embodied question answering against noisy queries. *arXiv preprint arXiv:2412.10726*, 2024.

Yuze Wu, Mo Zhu, Xingxing Li, Yuheng Du, Yuxin Fan, Wenjun Li, Zhichao Han, Xin Zhou, and Fei Gao. Vla-an: An efficient and onboard vision-language-action framework for aerial navigation in complex environments. *arXiv preprint arXiv:2512.15258*, 2025.

Jianqiang Xiao, Yuexuan Sun, Yixin Shao, Boxi Gan, Rongqiang Liu, Yanjin Wu, Weili Guan, and Xiang Deng. Uav-on: A benchmark for open-world object goal navigation with aerial agents. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 13023–13029, 2025.

Peng Xu, Zhengnan Deng, Jiayan Deng, Zonghua Gu, and Shaohua Wan. Aerialvla: A vision-language-action model for uav navigation via minimalist end-to-end control. *arXiv preprint arXiv:2603.14363*, 2026.

Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L Berg, and Dhruv Batra. Multi-Target Embodied Question Answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6309–6318, 2019.

Mingliang Zhai, Zhi Gao, Yuwei Wu, and Yunde Jia. Memory-centric embodied question answering. *arXiv preprint arXiv:2505.13948*, 2025a.

Mingliang Zhai, Hansheng Liang, Xiaomeng Fan, Zhi Gao, Chuanhao Li, Che Sun, Xu Bin, Yuwei Wu, and Yunde Jia. Multi-step reasoning for embodied question answering via tool augmentation. *arXiv preprint arXiv:2510.20310*, 2025b.

Yong Zhao, Kai Xu, Zhengqiu Zhu, Yue Hu, Zhiheng Zheng, Yingfeng Chen, Yatai Ji, Chen Gao, Yong Li, and Jincai Huang. CityEQA: A hierarchical llm agent on embodied question answering benchmark in city space. *arXiv preprint arXiv:2502.12532*, 2025.

Boyu Zhou, Yichen Zhang, Xinyi Chen, and Shaojie Shen. Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning. *IEEE Robotics and Automation Letters*, 6 (2):779–786, 2021.

Xin Zhou, Zhepei Wang, Hongkai Ye, Chao Xu, and Fei Gao. Ego-planner: An esdf-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2):478–485, 2020.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10012–10022, 2020.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A.1 DATASET DETAILS

A.1.1 DETAILS OF TRAJECTORY/IMAGE CURATION IN SIMULATOR

Advantages of the Selected Simulators:
a) Realistic environmental modeling: Both Beijing and Citypark are built on the Unreal Engine, encompassing diverse architectural styles and street layouts. They feature over a hundred micro-city elements, significantly enriching the semantic information of the acquired embodied agent video data.
b) Support for aerial agents: Both simulators are equipped with the built-in AirSim plugin, facilitating the control of aerial agents.
c) Existing route references: Previous research on vision-language navigation has been conducted in these simulators (Zhu et al., 2020), enabling us to obtain specific route coordinates and instruction data. Although most of these routes cannot be directly utilized (e.g., the majority of flight paths in AerialVLN contain numerous meaningless, repetitive maneuvers and lack logical progression, conflicting with the purpose of this benchmark), they provide valuable reference for our data collection process.
Drone Settings in the Simulator: The drone is equipped with a gimbal-mounted camera supporting pitch movements ranging from 0 to 90 degrees.
Collection Standards: The drone agents were operated by a total of five experienced pilots to ensure the rationality of the flight maneuvers. We carefully balanced the efficiency and diversity of the flight paths to ensure that the collected data comprehensively represents drone motion patterns in real-world Vision-Language-Action (VLA) tasks. This approach serves to validate the VLA and embodied Question & Answer (EQA) capabilities of Video Large Language Models (Video-LLMs).

A.1.2 DETAILS OF TRAJECTORY/IMAGE CURATION IN REAL WORLD

For the real-world tasks, we first collect expert seed trajectories with professional UAV pilots, following the same task categories as in simulation. Based on these expert demonstrations, we further generate perturbed trajectories through automated replay, resulting in approximately 4,000 indoor and outdoor real-world trajectories. The trajectory design also follows the simulation setup and comprises ten atomic-action categories—move forward, move backward, move left, move right, turn left, turn right, move up, move down, camera up, and camera down; four active-perception categories—occlusion, non-frontal orientation, peripheral view, and distant blur; and one general language-guided navigation category. Since a real drone cannot achieve the precise and efficient motion of a simulated drone, manually controlling the drone to collect a large number of images is impractical. Therefore, we designed an automated data collection pipeline based on expert demonstrations, detailed as follows:

First, at the expert trajectory collection level, for each task our professional pilot flies an expert trajectory according to the task requirements. We extract the odometry data (x, y, z, yaw) and gimbal state data (gimbal-state, including roll, pitch, yaw) from the recorded ROS bag. To distill the core information that characterizes the motion trend of the trajectory from the dense raw trajectory points, we perform adaptive sampling on the trajectory in specified planes (e.g., X-Y, X-Z, Y-Z): we first compute the rate of curvature change at each point to identify key motion phases such as turning and speed changes, while simultaneously imposing a minimum physical distance constraint between adjacent points to avoid overly dense sampling. By adjusting the curvature threshold and the minimum spacing, the sampling granularity can be flexibly controlled (defaulting to 8 key points). The resulting key-points are represented as a five-tuple (x, y, z, yaw, gimbal-state), capable of effectively capturing the spatial path and attitude variation trend with only a small number of points.

Next, to increase dataset diversity, we perturb these key points: Gaussian noise with zero mean and a standard deviation of 0.1 m is added to the position coordinates, and Gaussian noise with zero mean and a standard deviation of 0.05 rad is added to the yaw angle. This yields multiple new trajectories that slightly differ in position and orientation yet share the overall motion pattern of the expert trajectory. This perturbation strategy enriches the state-action distribution for subsequent policy learning without destroying task semantics. Finally, at the control level, we realize trajectory tracking along two dimensions. For the drone body, we modify EGO-Planner so that it can constrain height (z) and yaw angle, and use a PID controller to drive the drone to move smoothly between

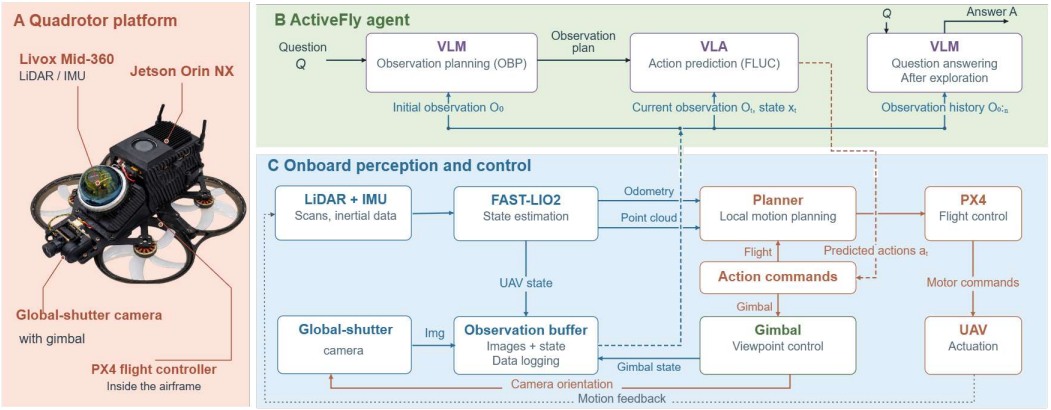

Figure 6: **System design of ActiveFly agent**

adjacent key-points. For the gimbal, we design a dedicated servo controller according to its motion characteristics, making it follow the gimbal state corresponding to the key points in a coordinated manner. The two controllers work together so that the drone moves along the expert trajectory while the gimbal maintains a camera angle consistent with the task requirements, thereby automatically completing the image data collection.

Through this pipeline, we can use a small number of human demonstrations as seeds to generate annotated training samples at scale, providing a data foundation for subsequent learning. All expert and augmented real-world trajectories are used exclusively for policy training and are not included in the test set. Real-world evaluation is instead conducted through closed-loop policy rollouts from independently reinitialized UAV states, with each scene evaluated over 10 independent trials.

A.1.3 Instruction Details

Consistent with the dataset structure presented in the main text, the text instructions in our dataset are categorized into Atomic Actions and VLA Tasks. For Atomic Actions, the main body of the instruction comprises one of ten single actions: ”move up”, ”move down”, ”move forward”, ”move backward”, ”move left”, ”move right”, ”turn left”, ”turn right”, ”camera up”, ”camera down”. Meanwhile, the instructions for VLA Tasks are composed of atomic actions as the minimal units. This compositional approach ensures that the fine-tuning and testing phases of VLA Tasks can maximally benefit from those of the Atomic Action phase. Specific examples are provided in Figure 9.

All instructions were manually annotated by the pilots involved in the trajectory/image curation process to guarantee precise alignment between the text instructions and their corresponding trajectories.

A.2 Platform Details

A.2.1 Hardware Configuration

As depicted in Figure 6, the UAV hardware platform is constructed to support agile flight and high-quality data collection in complex environments. The core airframe utilizes a compact 280-mm wheelbase carbon fiber design. Propulsion is driven by T-Motor F90 motors, providing sufficient thrust for responsive maneuvers. Onboard computation is handled by an Intel NUC 13 ANKB mini PC, which processes real-time perception, trajectory planning, and data logging. Low-level flight control is executed by a flight control unit running the PX4 Autopilot.

For environmental perception, a Livox Mid-360 3D LiDAR is integrated to provide omnidirectional depth measurements. To capture stable and viewpoint-adjustable visual observations, a high-resolution camera is mounted on an XF C-200T 3-axis gimbal. This gimbal system allows independent adjustment of the camera’s pitch, roll, and yaw. Such configuration directly supports the collection of diverse observation behaviors and captures the precise gimbal states required for real-world trajectory curation.

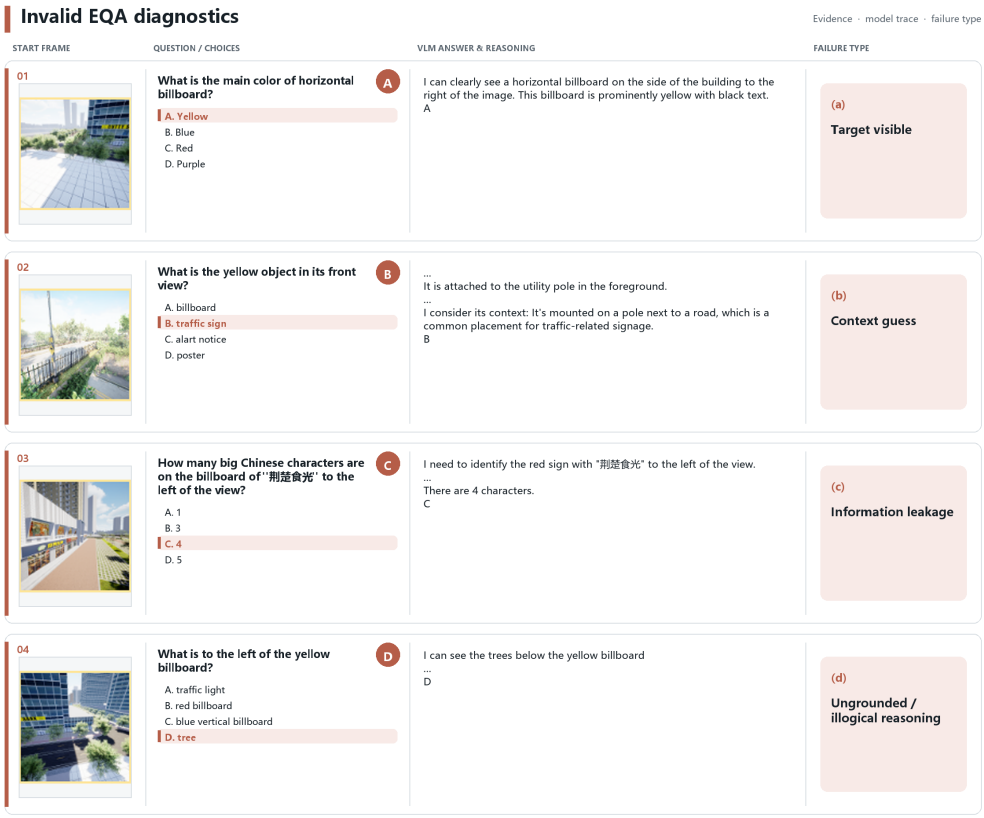

Figure 7: **Example of Invalid EQA**

A.2.2 Navigation and Control System

The autonomous navigation system integrates efficient state estimation and local planning modules to ensure safe obstacle avoidance during flight. State estimation is powered by FAST-LIO2, a tightly-coupled LiDAR-inertial odometry framework that fuses LiDAR scans and IMU measurements. It provides high-frequency, low-latency pose estimates, maintaining accurate localization during continuous and agile flight.

Real-time obstacle avoidance and trajectory generation are achieved using Ego Planner. By operating directly on LiDAR point clouds without constructing a global Euclidean Signed Distance Field (ESDF), Ego Planner generates smooth and collision-free local paths with minimal computational overhead. The integration of FAST-LIO2 and Ego Planner guarantees that the UAV can autonomously navigate through cluttered spaces and safely execute the required flight maneuvers during the automated data collection process.

A.3 EQA Details

A.3.1 Details of EQA Generation

As described in the main text, all Embodied Question & Answer (EQA) instances are categorized into five types: ”Object Recognition”, ”Attribute Recognition”, ”World Knowledge”, ”Counting”, and ”Spatial Understanding”. The EQA tasks are formulated as Multiple Choice Questions (MCQs), consisting of one correct option and three distractor options; the model is required to select the correct answer from these four choices. Each EQA is assigned a unique ”question_id” and corresponds to a unique test sample based on the ”data_id”. Notably, not every test sample is associated with an EQA, nor does every sample encompass all types of EQAs. Only scenarios and questions that strictly meet our predefined criteria are adopted.

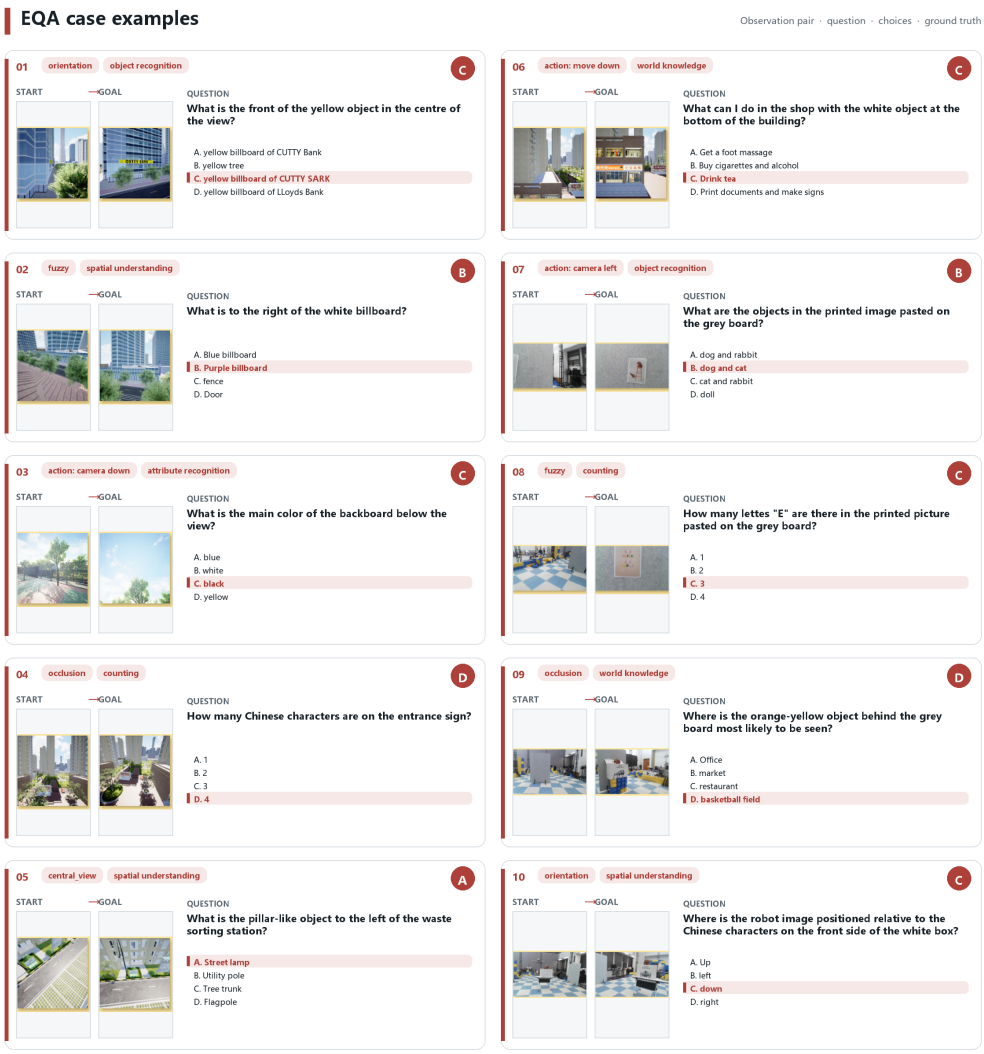

Figure 8: **More Examples of EQA**

EQA Criteria:
a) Task-Critical Evidence Criterion: Each EQA requires visual evidence that is unavailable or insufficiently observable from the initial viewpoint but becomes clearly observable after reaching the target viewpoint. Human annotators verify that the answer-determining object, attribute, count, or spatial relation cannot be directly perceived from the initial observation.
b) Unambiguity Principle: Each EQA has exactly one correct answer. Since Vision-Language Models (VLMs) generate answers based on historical observations, the three misleading options in the EQA must not introduce any potential confusion with other objects observable along the entire trajectory.
c) Indisputability Principle: The correctness and validity of all EQAs are strictly based on human judgment. Therefore, all personnel involved in the creation and evaluation of the EQAs must reach a unanimous consensus on each question and its corresponding correct answer.

In the initial phase of formulating multiple EQAs for each data sample, state-of-the-art commercial VLMs were evaluated for automated EQA generation. However, the majority of the VLM-generated EQAs required major human refinement. After carefully weighing efficiency and cost, all EQAs were ultimately authored and evaluated manually. During the attempts at VLM-based EQA generation, we observed that Large Language Models (LLMs) still struggle with comprehending complex urban environments within videos. These limitations become particularly pronounced when attempting

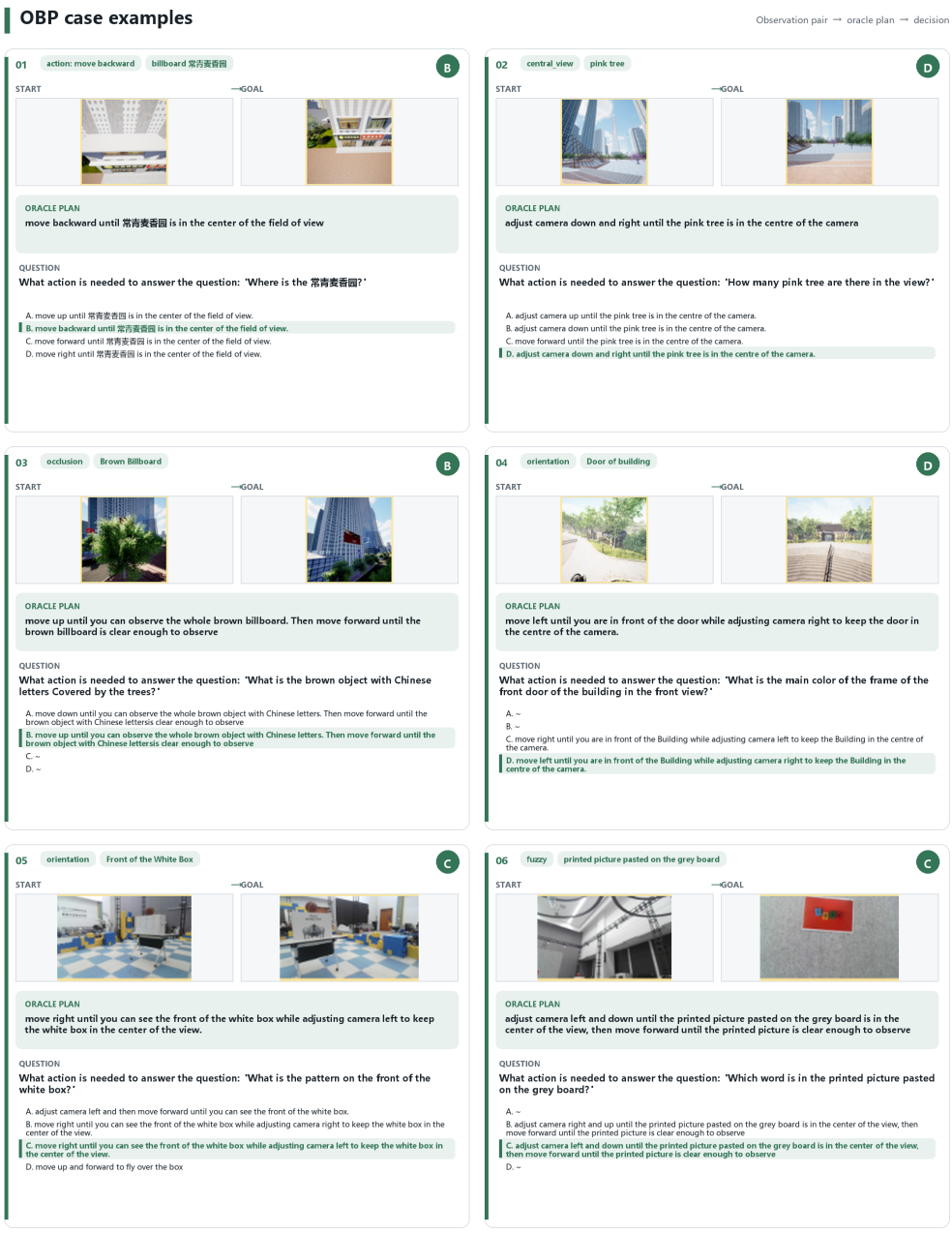

Figure 9: **More Examples of OBP**

to interpret the motion patterns of first-person agents and when applying the aforementioned EQA criteria to video understanding.

A.3.2 Analysis of Failed EQAs Judged by a Single VLM and Corresponding Solutions

After completing the first manual iteration of EQAs for all simulated data, we conducted a blind screening using Gemini 3.1 Pro: we formulated a scenario where the model acts as a static observer. The EQA and the first frame of the corresponding data were fed into the model, prompting it to return a single option in a standardized format. If the model answered correctly, the EQA was flagged as invalid. As a result, 33% of the EQAs were filtered out. We analyzed the failure modes of these

invalid EQAs, categorizing and summarizing them as follows. These failures stem from both inherent quality issues in the manually crafted EQAs and the features of the VLMs themselves. Examples of each failure mode are illustrated in Figure 7.

a) Object Appears/Partially Appears in the Start Frame: When the target object is fully or partially visible in the start frame, the VLM might correctly answer the question even under extremely poor viewing angles (e.g., severe occlusion or oblique angles). These are considered incidental cases and require human judgment to determine whether they should be retained or revised. Furthermore, even if the object is only partially visible, certain questions (e.g., querying the object's color) can still be answered directly. Such invalid EQAs must be discarded.

b) Guessing Answers Based on Scene and Commonsense: Even if the start frame lacks a valid observation of the target object, the VLM might guess the most plausible option based on semantic information extracted from the scene. Addressing this issue requires modifying the distractor options into choices that cannot be easily eliminated using mere scene context. However, further evaluation revealed that modifying distractors only mitigates this issue. Regardless of the option design, VLMs actively utilize scene context to eliminate unlikely choices and make guesses, occasionally answering EQAs correctly by chance. If human evaluators determine that such an EQA still strictly requires "embodied actions to supplement necessary information," it is retained as a valid EQA.

c) Information Leakage: Imprecise or non-compliant EQAs might inadvertently suffer from information leakage. For instance, the question text of the third example in Figure 7 has already exposed the number of letters on the target object. Such EQAs must be either heavily refined or entirely removed.

d) Illogical Reasoning by the Model: Models occasionally arrive at the correct answer through illogical means. For example, a model might rely on pure random guessing, or, if the target object (such as a specific tree) is absent in the start frame, it might anchor its answer on other similar objects (e.g., another tree) present in the scene (Figure 7). These cases require careful manual identification and should be retained.

As demonstrated above, EQAs cannot be filtered through simple binary judgments: some necessitate deletion, others require refinement, and those still meeting the EQA criteria should be retained. Consequently, a more comprehensive VLM-based Blind Screening pipeline was deployed to address the aforementioned failure modes and to ensure the validity of the EQAs in this benchmark, as detailed in Section A.3.3.

A.3.3 VLM-BASED BLIND SCREENING OF EQA

The complete VLM-based Blind Screening pipeline utilized in this benchmark is outlined below:

- **Multi-Model Voting:** The single-model judgment was replaced with a multi-model voting mechanism. Only when all models answer correctly on the start frame is the EQA flagged as "invalid". This effectively minimizes the probability of models guessing the correct answer by chance. The models employed include Gemini_2.5_Flash, GPT_5.4_Nano, and Qwen3-VL_Flash.
- **"IDK" and "Action Needed" Labels:** An "I don't know" (IDK) option was introduced into the prompt, empowering the model to assess whether the question is answerable. Furthermore, the concept of "Embodied" was explicitly defined in the prompt, and an "Action needed" option was added to allow the model to gauge its self-confidence in answering.
- **Chain of Thought (CoT) Output:** The models are required to output their Chain of Thought (CoT), thereby ensuring the interpretability of their answers.
- **Human Cross-Refinement of EQAs:** Based on the CoT generated in the previous step, annotators exchange their EQAs and conduct rigorous peer reviews.
- **Re-evaluation of Refined EQAs:** The CoT from this stage is utilized to determine whether to discard the previously flagged invalid EQAs. This process allows for the retention of

Algorithm 1 EQA Filter via Multi-Model Voting

Require: EQA dataset $\mathcal{D}$, Set of VLMs $\mathcal{M} = \{M_{\text{Gemini}}, M_{\text{GPT}}, M_{\text{Qwen}}\}$, Max retries N_{retry}
Ensure: Set of invalid EQA IDs $\mathcal{I}$, Evaluation logs $\mathcal{L}$

Initialize $\mathcal{I} \leftarrow \emptyset$, $\mathcal{L} \leftarrow \emptyset$
Define Prompt Template $\mathcal{P}(q)$:

You are an expert visual QA assistant. Look closely at the provided image and answer the question. Your task is to determine the correct choice based on the given image.
If the image does not contain enough clear information to definitively answer the question, you MUST choose "I don't know".
Alternatively, if you can guess a vague answer but your confidence is low, and you determine that taking an action to change the camera's position or pose would help acquire more information to answer accurately, you MUST choose "action needed".
If you can confidently determine the correct answer based on this image, please provide your final choice among "A", "B", "C", "D".
First, think step-by-step and provide your reasoning (Chain of Thought).
Then, you MUST output your final choice wrapped between `<answer>[Your Choice]</answer>` *tags. Do not include any other punctuation, explanations, or extra text. (This rule must not be violated, as we will use regex to extract the answer from these tags.)*
The final choice MUST be exactly one of: "A", "B", "C", "D", "I don't know", or "action needed".
restricted format:
Reasoning: [your step-by-step analysis here]
`<answer>A</answer>`
Question: $\{q\}$

for each EQA instance $x \in \mathcal{D}$ **do**
  Extract initial frame V_0, question q, and ground truth a_{gt} from x
  $C \leftarrow \emptyset$ {Track models that answered correctly}
  for each model $M \in \mathcal{M}$ in parallel **do**
    $ans_M \leftarrow$ None
    for $i = 1$ **to** N_{retry} **do**
      $r_M \leftarrow \text{GenerateContent}(M, V_0, \mathcal{P}(q))$
      $ans_M \leftarrow \text{RegexExtract}(r_M, \texttt{<answer>(.*?)</answer>})$
      if $ans_M \neq$ Error **then**
        break
      end if
    end for
    Append CoT and ans_M to $\mathcal{L}$
    if $ans_M == a_{gt}$ **then**
      $C \leftarrow C \cup \{M\}$ {Model guessed correctly on initial frame}
    end if
  end for
  if $|C| == |\mathcal{M}|$ **then**
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{x.\text{id}\}$ {Invalid if all models answer correctly}
  end if
end for
return $\mathcal{I}, \mathcal{L}$

certain edge cases that failed the EQA evaluation but still fundamentally align with the EQA criteria.

To ensure that the CoT reasoning process does not interfere with the judgment of answer correctness, the script utilizes an `extract_answer()` function to isolate the model's output options, demarcated by the `<answer>` tags. Algorithm 1 demonstrates the partial core code and prompt of this script.

A.3.4 More EQA Examples

To better illustrate the proposed EQAs, we provide additional examples encompassing all task and question types across both simulators and the real world. These are showcased in Figure 8.

A.4 OBP Details

Observation Behavior Planning (OBP) is presented in a QA format. The standard template for the question is *"What action is needed to answer the question: "¡EQA question¿"*. The choices consist of the actual trajectory instruction as the correct answer, alongside three misleading instructions that are either physically impossible to execute or would result in a trajectory that fails to answer the EQA question. Figure 9 presents illustrative examples of OBP.

A.4.1 Details of OBP Generation

OBP generation must similarly adhere to the unambiguity and indisputability principles applied in EQA generation (Appendix A.3.1). Furthermore, preventing information leakage is critical in OBP generation. In many data samples, the *object name* and the *EQA question* intrinsically contain spatial cues or hints that inadvertently reveal the answer, as depicted by the third example in Figure 9.: The name of target object has already indicated the answer. When necessary, the text of the OBP question and choices must be meticulously refined to ensure the problem's validity. Crucially, an effective OBP relies simultaneously on the essential information present in both the start frame image and the text-based target description. If the correct action instruction can be deduced from either the text or the image alone, the OBP fails to genuinely evaluate the model's embodied capabilities.

A.4.2 More OBP Examples

To further clarify the proposed OBP, we provide additional examples encompassing all task types across both simulators and the real world. These are presented in Figure 9.

A.5 Experiment Details

A.5.1 Metrics

Let a predicted trajectory be denoted as

$$\hat{\tau} = \{\hat{\mathrm{x}}_1, \hat{\mathrm{x}}_2, \ldots, \hat{\mathrm{x}}_T\}, \tag{4}$$

where $\hat{\mathrm{x}}_t$ is the UAV pose at step t, and let the ground-truth trajectory be

$$\tau^* = \{\mathrm{x}_1^*, \mathrm{x}_2^*, \ldots, \mathrm{x}_N^*\}. \tag{5}$$

Here, each pose $\mathrm{x} = [\mathrm{x}_{\mathrm{loc}} : \mathrm{x}_{\mathrm{ori}}]$ includes both position $\mathrm{x}_{\mathrm{loc}}$ and orientation $\mathrm{x}_{\mathrm{ori}}$. We use $d_{\mathrm{Man}}(\cdot,\cdot)$ and $d_{\mathrm{Euc}}(\cdot,\cdot)$ to denote the Manhattan and Euclidean distance, respectively.

The success rate (SR) follows the definition used in aerial VLN (Liu et al., 2023c). However, since FLUC targets short-horizon, viewpoint-aware navigation, we additionally take the UAV's yaw and pitch angles into account when determining whether the target has been successfully reached. Accordingly, SR is defined as follows:

$$\mathrm{SR} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}\Big[d_{\mathrm{Euc}}\Big(\hat{\mathrm{x}}_{T,\mathrm{loc}}^{(i)}, \mathrm{x}_{N,\mathrm{loc}}^{*(i)}\Big) < \delta_{\mathrm{loc}} \ \wedge \ d_{\mathrm{Man}}\Big(\hat{\mathrm{x}}_{T,\mathrm{ori}}^{(i)}, \mathrm{x}_{N,\mathrm{ori}}^{*(i)}\Big) < \delta_{\mathrm{ori}}\Big], \tag{6}$$

where M is the number of evaluation samples. We set $\delta_{\mathrm{loc}} = 3$ and $\delta_{\mathrm{ori}} = 10^\circ$.

Oracle Success Rate (OSR) measures whether the predicted trajectory reaches the target pose at any step. Under the viewpoint-aware FLUC setting, a trajectory is regarded as oracle-successful if there exists at least one step whose position and orientation are both sufficiently close to the ground-truth target pose. Formally, OSR is defined as

$$\mathrm{OSR} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{1}\Big[\exists\, t \in \{1,\ldots,T\},\ d_{\mathrm{Euc}}\Big(\hat{\mathrm{x}}_{t,\mathrm{loc}}^{(i)}, \mathrm{x}_{N,\mathrm{loc}}^{*(i)}\Big) < \delta_{\mathrm{loc}} \ \wedge \ d_{\mathrm{Man}}\Big(\hat{\mathrm{x}}_{t,\mathrm{ori}}^{(i)}, \mathrm{x}_{N,\mathrm{ori}}^{*(i)}\Big) < \delta_{\mathrm{ori}}\Big]. \tag{7}$$

Navigation Error (NE) is defined as the distance between the final predicted pose and the ground-truth final pose:

$$\mathrm{NE} = \frac{1}{M}\sum_{i=1}^{M} d_{\mathrm{Euc}}\left(\hat{\mathrm{x}}_{T,\mathrm{loc}}^{(i)}, \mathrm{x}_{N,\mathrm{loc}}^{*(i)}\right). \tag{8}$$

Normalized Dynamic Time Warping (nDTW) evaluates trajectory similarity by comparing the predicted trajectory with the ground-truth trajectory using Dynamic Time Warping (DTW). Let $\mathrm{DTW}(\hat{\tau}, \tau^*)$ denote the DTW distance between the two trajectories. Then nDTW is defined as

$$\mathrm{nDTW} = \frac{1}{M}\sum_{i=1}^{M} \exp\left(-\frac{\mathrm{DTW}(\hat{\tau}^{(i)}, \tau^{*(i)})}{\eta\,|\tau^{*(i)}|}\right), \tag{9}$$

where $|\tau^{*(i)}|$ is the length of the ground-truth trajectory, and η is the normalizing success threshold. We set $\eta = 1$.

Accuracy weighted by Path Length (APL) jointly evaluate correctness and efficiency.APL rewards correct answers obtained with shorter exploration trajectories. Let $a_i \in \{0, 1\}$ denote whether the answer to the i-th question is correct, and let L_i denote the length of the corresponding exploration trajectory. Then APL is defined as

$$\mathrm{APL} = \frac{1}{M}\sum_{i=1}^{M} a_i \cdot \frac{|\tau^{*(i)}|}{\max\left(|\hat{\tau}^{(i)}|, |\tau^{*(i)}|\right)}, \tag{10}$$

In practice, a higher APL indicates that the agent can answer the question correctly with a shorter exploration path, and therefore reflects better overall efficiency in embodied perception.

A.5.2 Real-World Evaluation Protocol

For real-world evaluation, the UAV is reinitialized with independently specified initial poses and performs closed-loop policy rollouts. No augmented training trajectories are reused for evaluation, and each scene is evaluated over 10 independent trials.

A.5.3 VLA Adaptation and Training Details

Both policies control five degrees of freedom, with task actions ordered as

$$a_t = (\Delta x_t, \Delta y_t, \Delta z_t, \Delta\psi_t, \Delta\theta_t).$$

Here, (x, y, z) denotes position, ψ vehicle yaw, and θ camera pitch. Each policy uses its own training-data quantiles for action normalization:

$$\widetilde{a}_j = 2\frac{a_j - q_{0.01,j}}{q_{0.99,j} - q_{0.01,j} + \epsilon} - 1.$$

The quantities $q_{0.01,j}$ and $q_{0.99,j}$ are the first and 99th percentiles of action dimension j. Inference uses the normalization statistics stored with the corresponding checkpoint and the policy-specific inverse transform.

OpenVLA. At inference, the policy receives the current onboard RGB image resized to 224×224, the navigation instruction, and the five-dimensional pose (x, y, z, ψ, θ) relative to the initial pose. The pose is serialized into the language prompt with one decimal place, using meters for position and degrees for the angular state variables. OpenVLA generates five action tokens with its autoregressive decoder. These tokens are decoded and inverse-normalized to obtain translation and angular increments. Translation increments are expressed in the current heading-aligned frame. The action normalization uses $\epsilon = 10^{-8}$ and clips normalized training targets to $[-1, 1]$. Evaluation uses deterministic token decoding and executes one predicted action before acquiring a new observation and predicting the next action.

We fine-tune the model using AdamW with batch size 8, one gradient-accumulation step, and learning rate 5×10^{-4}. LoRA is applied to linear layers with rank 32, scaling parameter 16, and zero dropout. The model is trained over 30,000 optimization steps.

Table 6: Detailed outcome decomposition of different baselines over embodied perception episodes.

OBP Acc.	FLUC OSR	EQA Acc.	GPT		Gemini		Qwen	
			Count	Ratio	Count	Ratio	Count	Ratio
OpenVLA								
1	1	1	90	12.1	78	10.4	100	13.4
1	1	0	13	1.7	16	2.1	14	1.9
1	0	1	373	49.9	335	44.9	306	40.9
1	0	0	85	11.4	92	12.3	56	7.5
0	1	1	0	0.0	2	0.3	2	0.3
0	1	0	0	0.0	0	0.0	0	0.0
0	0	1	114	15.3	144	19.3	170	22.8
0	0	0	72	9.6	80	10.7	99	13.3
Pi-0.5								
1	1	1	346	46.3	326	43.6	327	43.8
1	1	0	45	6.0	46	6.2	43	5.8
1	0	1	133	17.8	113	15.1	87	11.7
1	0	0	37	5.0	36	4.8	19	2.5
0	1	1	15	2.0	12	1.6	27	3.6
0	1	0	2	0.3	1	0.1	4	0.5
0	0	1	116	15.5	135	18.1	154	20.6
0	0	0	53	7.1	78	10.4	86	11.5

Pi-0.5. The policy is conditioned on the navigation instruction , current state and the onboard RGB image, resized with aspect-ratio-preserving padding to 224×224.

The model uses a 32-dimensional internal action representation and a continuous flow-matching objective. Normalized five-dimensional task actions are zero-padded to this internal dimension; the output transform inverse-normalizes and retains the first five coordinates. The prediction horizon is 10, yielding a 10×5 task-action chunk. The policy replans from the current observation after every two executed actions. For each execution step, predictions for that step from at most four overlapping chunks are combined with normalized weights proportional to 0.5^i, where $i = 0$ denotes the newest available prediction.

Fine-tuning uses a global batch size of 8 and a 30,000-step budget. LoRA ranks and scaling parameters are 16 for the language backbone and 32 for the action expert. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 10^{-10}, and gradient-norm clipping at 1.0. The learning rate warms up over 10,000 steps to 5×10^{-5} and remains constant thereafter. Exponential moving averaging is disabled.

Evaluation protocol. In the full pipeline, the navigation instruction is supplied by the OBP prediction. Rollouts use the scene-specific simulator in ComputerVision mode. Both policies predict the same canonical 5-DoF task actions, while model-specific adapters transform the translational increments into the execution coordinate frame. Each rollout is limited to 80 executed actions and terminates early after five consecutive small-motion comparisons, each requiring less than 0.03 m change along every position axis and less than 1° change in both yaw and camera pitch.

A.5.4 Failure and Escape Decomposition

For each embodied perception episode, we record three binary indicators: $S_{\text{obp}} \in \{0, 1\}$ for OBP correctness, $OS_{\text{fluc}} \in \{0, 1\}$ for FLUC oracle success, and $S_{\text{eqa}} \in \{0, 1\}$ for Air-EQA correctness. The end-to-end episode is successful only when all three stages succeed:

$$S_{\text{ep}} = S_{\text{obp}} \cdot OS_{\text{fluc}} \cdot S_{\text{eqa}}. \tag{11}$$

To analyze cases where a downstream stage succeeds despite failure at an intermediate stage, we further define two types of escape. *FLUC escape* occurs when the observation plan is incorrect but the executed trajectory nevertheless reaches a valid observation region:

$$S_{\text{obp}} = 0, \qquad OS_{\text{fluc}} = 1. \tag{12}$$

Air-EQA escape occurs when FLUC fails to satisfy the oracle-success criterion but the agent still obtains sufficient visual information to answer the embodied question correctly:

$$OS_{\text{fluc}} = 0, \qquad S_{\text{eqa}} = 1. \tag{13}$$

Episodes that satisfy neither the end-to-end success condition nor either escape condition are categorized as failures.

The results are depicted in Table 6.