BENCHMARKING INTERACTION-ORIENTED AERIAL MOBILITY ACROSS HETEROGENEOUS ENVIRONMENTS

Anonymous authors
Paper under double-blind review



Figure 1: **Overview of CROSSFLY benchmark.** It focuses on interaction-oriented aerial mobility tasks and consists of a large dataset for unmanned aerial vehicle (UAV) imitation learning, featuring multiple aerial embodiments, heterogeneous environments, photorealistic rendering, and diverse target objects. To enable systematic experimental analysis under interaction-oriented mobility task settings, we additionally provide an evaluation protocol and propose an interaction readiness metric.

ABSTRACT

Aerial mobile manipulation requires unmanned aerial vehicles (UAVs) to traverse heterogeneous environments such as indoor and outdoor domains, and connect navigation with subsequent physical interaction within the same trajectories. Existing aerial benchmarks are typically confined to single-environment settings, limiting evaluation of agents across continuous environmental transitions. When interaction is evaluated, agents are often initialized near the target, leaving the transition from navigation to interaction largely unassessed. This work introduces CROSSFLY, a benchmark for long-horizon interaction-oriented aerial mobility across heterogeneous environments. CROSSFLY contains **15K** trajectories across more than **20** reconstructed scenes under a unified simulation and evaluation framework, covering *Indoor*, *Outdoor*, and *Indoor↔Outdoor* trajectories. It further introduces Interaction Readiness Rate (IRR) to assess whether terminal navigation states provide suitable conditions for downstream interaction without assuming a specific manipulation policy. Experiments with **five** representative aerial vision-language-action (VLA) agents and the proposed CROSSFLY-AGENT show that *Indoor↔Outdoor* settings are more challenging than single-environment settings, with the highest success rate reaching 9.7% on unseen scenes. A downstream manipulation case study further validates the relevance of interaction-oriented arrival to subsequent physical interaction. We hope CROSSFLY helps advance aerial embodied intelligence toward unified mobility and physical interaction in heterogeneous environments. Related datasets and code will be released.

1 INTRODUCTION

Aerial mobile manipulation involves UAVs traversing three-dimensional spaces and interacting with objects within a single trajectory (Sun et al., 2026a;b; Wang et al., 2026; Tucker et al., 2026; He

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.

et al., 2025). Reaching an interaction site may require flight through rooms, corridors, open spaces, and building boundaries under varying spatial constraints. Such trajectories may span heterogeneous environments (Luo et al., 2026), where flight must not only reach the target region but also terminate in a state suitable for interaction (Zhang et al., 2025a). These requirements motivate interaction-oriented aerial mobility as a core capability for extending goal-directed flight toward interaction.

Existing aerial benchmarks are typically confined to single-environment settings, where evaluation centers on instruction-conditioned or goal-directed flight and generalization to unseen scenes (Wang et al., 2025a; Gao et al., 2026; Liu et al., 2023; Lee et al., 2025; Liu et al., 2026). These protocols largely characterize whether agents reach designated goals, limiting evaluation of their behavior across long-horizon, continuous environmental transitions. When interaction is included, agents are often initialized near the target, providing little assessment of how they progress from extended flight through diverse environments to physical interaction (Sun et al., 2026a;b; Wang et al., 2026).

Supporting this capability requires continuous trajectories that preserve environmental transitions and the approach to interaction sites (Wu et al., 2026a). Such trajectories provide supervision for learning under changing conditions within the same episode, while evaluation should assess whether terminal states support subsequent interaction without assuming a specific manipulation policy.

CROSSFLY instantiates these requirements as a large-scale benchmark for learning and evaluating interaction-oriented aerial mobility across heterogeneous environments. It contains 15K trajectories across more than 20 reconstructed scenes under a unified simulation framework, spanning diverse *Indoor*, *Outdoor*, and *Indoor↔Outdoor* task suites. Each trajectory records synchronized egocentric RGB observations, 6-DoF flight states, language instructions, waypoints, fine-grained subtasks, and interaction-related annotations throughout the trajectory. For evaluation, CROSSFLY introduces Interaction Readiness Rate (IRR) to assess whether terminal flight states provide suitable conditions for subsequent physical interaction, making it applicable to different manipulation strategies.

Experiments with five representative aerial agents and the proposed CROSSFLY-AGENT reveal substantial gaps in interaction-oriented aerial mobility. Even the strongest agent reaches only a 9.7% success rate on unseen *Indoor↔Outdoor* trajectories, indicating difficulty in sustaining goal-directed flight across environmental transitions. IRR shows that goal reaching does not necessarily translate into interaction-oriented arrival, exposing limitations beyond standard navigation metrics. A downstream manipulation case study validates the relevance of interaction readiness to subsequent physical interaction. We hope CROSSFLY will serve as a standardized platform for evaluating and advancing aerial interaction-oriented mobility. Related datasets and code will be publicly available.

2 RELATED WORK

Aerial Benchmarks. Language-guided aerial benchmarks cover diverse environments and task settings. AerialVLN (Liu et al., 2023) studies long-horizon UAV instruction following from egocentric observations, while AVDN (Fan et al., 2023) incorporates interactive dialogue. Subsequent benchmarks improve flight realism and expand real-world scene coverage (Wang et al., 2025b; Lee et al., 2025). OpenFly (Gao et al., 2026) integrates data generation, simulation, and agent evaluation across multiple engines, whereas IndoorUAV (Liu et al., 2026) focuses on continuous navigation in indoor environments. Beyond route following, UAV-Flow Colosseo (Wang et al., 2025a) evaluates fine-grained flight control, AIR-VLA (Sun et al., 2026a) couples aerial mobility with manipulation, and HUGE-Bench (Guo et al., 2026) introduces process- and safety-aware evaluation for multi-stage tasks. Ground-based last-mile navigation further targets manipulation-ready positioning (Zhang et al., 2025a; Lee et al., 2026). These benchmarks address complementary aspects of navigation and interaction, but separate indoor and outdoor evaluations do not characterize continuous execution across environmental boundaries. CROSSFLY brings these dimensions together by combining flight through connected indoor–outdoor environments with terminal interaction-readiness evaluation. Its single-domain and cross-domain suites support systematic comparison of navigation success and arrival quality across different environmental conditions and transitions.

Aerial Agents. Language-conditioned aerial agents span planning-oriented navigation methods and action-oriented VLA policies. Early navigation approaches predict waypoints or hierarchical trajectories from language and visual observations (Fan et al., 2023; Liu et al., 2023; Wang et al., 2025b). Foundation-model-based agents support more robust long-horizon decision making through seman-

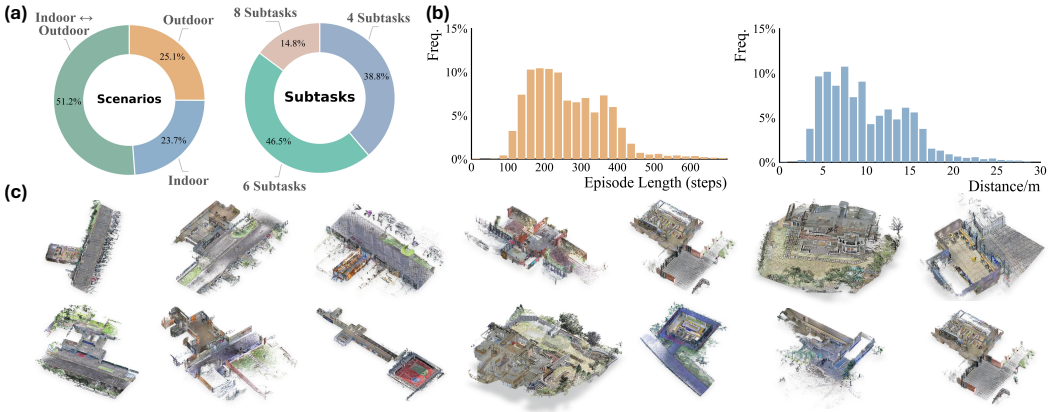

Figure 2: **Trajectory statistics and scene diversity.** (a) Distribution of scene types and task instructions with different numbers of subtasks (§3.1). (b) Distribution of trajectory lengths and path distances, summarizing the temporal and spatial scales of the dataset. (c) Representative environments in CROSSFLY, where each scene covers indoor and outdoor environments.

tic planning, memory, explicit reasoning, and directional grounding (Zhang et al., 2025b; Cai et al., 2025; Ning et al., 2026; Fan et al., 2026; Qi et al., 2026; Ding et al., 2026), while AerialVLA (Chen et al., 2026a) uses task progress and observation history to support proactive dialogue during flight. Action-oriented policies map multimodal observations more directly to flight commands: OpenFly-Agent (Gao et al., 2026) uses historical observations for action prediction, whereas AeroVLA (Xu et al., 2026) and AutoFly (Sun et al., 2026c) learn end-to-end continuous control from language instructions. Other work explores learning-free navigation, efficient onboard execution, unified indoor–outdoor training, and compressed long-horizon context (Hu et al., 2025; Chen et al., 2026b; Luo et al., 2026; Jiang et al., 2025). Recent aerial manipulation work couples navigation and manipulation over short spatial ranges, leveraging transfer from manipulation-pretrained VLA models (Tucker et al., 2026) and separate decoding of flight and manipulator actions (Sun et al., 2026b). Connecting longer-range navigation to such local interactions requires evaluating whether the terminal flight state supports the subsequent manipulation skill. CROSSFLY makes this connection explicit through interaction-readiness evaluation. We further introduce CROSSFLY-AGENT as a hierarchical agent, in which a high-level VLM planner specifies the active subtask and a low-level VLA policy generates 6-DoF flight actions and predicts a manipulation-invocation signal.

3 CROSSFLY BENCHMARK

Evaluating interaction-oriented aerial mobility across heterogeneous environments requires unified, consistent definitions of the collected data, execution environment, and evaluation criteria. We first summarize the benchmark composition, including trajectory statistics and scene categories. Then, the dataset construction pipeline describes how scenes, trajectories, and subtask annotations are constructed (§3.2). Next, the simulation environment specifies the physics simulator, UAV platform, and action space used for closed-loop execution (§3.3). Finally, the evaluation protocol defines the task suites and metrics used to assess navigation and interaction readiness (§3.4).

3.1 BENCHMARK STATS

CROSSFLY contains 15K trajectories with 3M frames, totaling 160+ hours and 120 km of flight. As shown in Fig. 2, the dataset spans diverse indoor, outdoor, and cross-environment *Indoor↔Outdoor* tasks, where cross-environment trajectories account for the largest portion (**51.2**%), compared with *Indoor* (**23.7**%) and *Outdoor* (**25.1**%) trajectories. The instructions are highly compositional, consisting of 4, 6, or 8 subtasks, with 6-subtask instructions being the most common (**46.5**%), followed by 4-subtask (**38.8**%) and 8-subtask (**14.8**%) instructions. The trajectories further exhibit substantial variations in both temporal horizon and spatial extent: most trajectories span 100–400 frames, while trajectory distances are broadly distributed across different ranges, with the majority falling below 20 m and a small number extending to nearly 30 m. Such diversity in trajectory length and spatial

Table 1: **Comparison with related benchmarks (§3.1).**

Dataset	Platform	Scene	N_{traj}	N_{scenes}	Environment	Interaction	Dynamics	DoF
R2R (Anderson et al., 2018)	Ground Agent	Matterport3D	7,189	90	Indoor	✗	✗	Graph-based
VLN-CE (Krantz et al., 2020)	Ground Agent	Habitat	4,475	90	Indoor	✗	✗	2
OpenUAV (Schmittle et al., 2018)	UAV	AirSim + UE	12,149	22	Outdoor	✗	✓	6
AVDN (Fan et al., 2023)	UAV	xView	6,269	–	Outdoor	✗	✗	3
CityNav (Lee et al., 2025)	UAV	SensatUrban	32,637	34	Outdoor	✗	✗	4
OpenFly (Gao et al., 2026)	UAV	AirSim, GTAV, etc.	100K	18	Outdoor	✗	✗	4
IndoorUAV (Liu et al., 2026)	UAV	Habitat	16,040	1,075	Indoor	✗	✓	4
CROSSFLY	UAV	Isaac Sim	15,263	23	Heterogeneous	✓	✓	6

scale introduces varying levels of long-horizon navigation and interaction complexity. Moreover, CROSSFLY covers a wide range of heterogeneous environments with substantially different layouts, scales, and visual appearances, providing diverse settings for evaluating embodied aerial agents.

As shown in Table 1, CROSSFLY presents a more challenging setting compared with existing benchmarks. The agent needs to navigate across heterogeneous environments with different spatial constraints, while ensuring that the destinations are suitable for subsequent physical interactions.

3.2 DATASET CONSTRUCTION PIPELINE

This pipeline is designed to construct large-scale, executable, and semantically annotated UAV trajectories in a unified framework across diverse heterogeneous environments, including indoor, outdoor, and indoor–outdoor domains. As shown in Fig. 3, it consists of three main stages. First, tasks are defined through objects, key waypoints, and language instructions in self-collected scenes. Next, the resulting task specifications are executed under realistic UAV dynamics to acquire synchronized egocentric observations and flight states, while infeasible rollouts are filtered out automatically. Finally, the waypoint structure and trajectory observations are used to derive fine-grained task annotations, providing structured supervision at both the trajectory and subtask level for effective learning.

Scene Collection. We reconstruct diverse heterogeneous environments from multi-view RGB images and point clouds captured by a handheld 3D scanner, including 3D Gaussian Splatting (3DGS) scenes and mesh representations (Kerbl et al., 2023). The collected scenes cover both individual indoor and outdoor spaces as well as connected regions that allow trajectories to transition between the two domains. These reconstructed environments provide the visual appearance and scene geometry used for subsequent trajectory construction and simulation. For each scene, we annotate target objects, spatial waypoints, and language instructions to define executable trajectories. Each trajectory is specified by four kinds of waypoints: *start*, *intermediate*, *terminal*, and *turning* points. By combining these annotations with different target objects and language instructions, each reconstructed scene can support diverse configurations with different routes, viewpoints, and interaction goals.

Waypoint Annotation. Within each reconstructed scene, we define a high-level trajectory by annotating target objects, language instructions, and a sparse set of key waypoints. The waypoints serve different roles along the path: the *start* waypoint specifies the initial UAV pose; *intermediate* waypoints guide UAV through important spatial regions along the intended route; the *terminal* waypoint specifies the final destination, typically near the target object. Together, these annotations define a coarse trajectory representation for subsequent trajectory refinement.

Trajectory Acquisition. Given the annotated waypoints, UAV is initialized at the *start* pose. Then the Ego-Planner (Zhou et al., 2022) generates a collision-free path between consecutive waypoints, transforming the sparse waypoints into an executable reference trajectory. A position-velocity PD tracking controller then converts the trajectory into acceleration commands executed by the low-level flight controller. The resulting motion is simulated under UAV dynamics, introducing realistic flight effects such as small oscillations, tracking deviations, response delays, and viewpoint perturbations.

During each rollout, we record synchronized egocentric RGB observations and 6-DoF UAV states. The complete state sequence provides a temporally aligned representation of how the agent progresses through the task and supports fine-grained evaluation of navigation and interaction behaviors. Rollouts that collide with the reconstructed scene geometry or fail to reach the designated waypoint are automatically rejected. This execution-and-filtering procedure ensures that retained task instances are physically feasible and can be completed under the simulator dynamics.

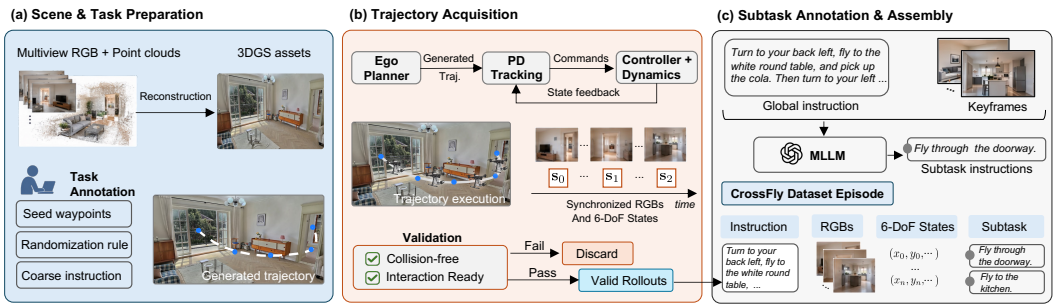

Figure 3: **Overview of the dataset construction pipeline (§3.2).** CROSSFLY begins with reconstructed scenes from point clouds and realistic rendering, followed by waypoint annotation and trajectory acquisition through UAV simulation. The collected trajectories are then assembled with RGB observations, UAV states, and task-related annotations to construct the benchmark dataset.

Task Decomposition. The waypoint structure provides a natural basis for partitioning each trajectory into semantically coherent subtasks. Each trajectory is partitioned into a sequence of subtasks, where each subtask corresponds to the trajectory segment between two adjacent waypoints. These waypoints include *start*, *intermediate*, and *terminal* waypoints. For each segment, we extract the egocentric observations at its start and end frames together with the global instruction ℓ, and prompt a pretrained VLM (GPT-5.5) to generate a concise description of the segment-level intent, which serves as the ground-truth subtask for training high-level planning models after human verification.

3.3 SIMULATION ENVIRONMENT

Physical Simulator. CROSSFLY is built on NVIDIA Isaac Sim, which provides physics simulation, collision detection, and closed-loop UAV control. UAV dynamics are simulated with PhysX, and the reconstructed 3D Gaussian Splatting scenes serve as the visual environments for navigation. At each simulation step, egocentric observations are rendered according to the current UAV pose, while the predicted actions are executed under the simulator dynamics. The updated UAV state is subsequently used to generate the next observation, forming a closed-loop interaction cycle between the agent and the simulated environment. This setup enables closed-loop evaluation across indoor, outdoor, and heterogeneous environments with realistic visual appearance and physical motion.

UAV Platform. CROSSFLY supports diverse aerial embodiments through a unified control interface. Our main benchmark experiments are conducted on two mobile manipulation platforms: a quadrotor equipped with SO100-ARM (Cadene et al., 2024) and a hand-like UAV (Wu et al., 2026b).

Action Space. We adopt a 6-DoF action space for UAV navigation, consisting of position (x, y, z) and orientation (roll, pitch, and yaw). Motion commands predicted by the agent are executed through a low-level PD controller. For object-interaction tasks, agents can additionally issue an interaction command to invoke the corresponding manipulation policy after reaching the target.

3.4 EVALUATION PROTOCOL

Task Suites. While many existing benchmarks focus on either indoor or outdoor domains, CROSSFLY provides a cross-domain dataset with aerial interaction-oriented mobility scenarios spanning heterogeneous environments. It provides three fixed suites for evaluation: ***Indoor***, ***Outdoor***, and ***Indoor↔Outdoor***. The first two measure navigation and interaction within a single environmental domain, and the last one evaluates long-horizon execution across environmental transitions.

- ***Indoor.*** Indoor tasks are executed inside buildings. Given a natural-language instruction, the agent must navigate through rooms, corridors, and doorways, approach the specified interaction location, and output a special token such as `<Manip>` when manipulation is required. The suite evaluates instruction following, object grounding, and target approach in constrained indoor spaces.
- ***Outdoor.*** Outdoor tasks follow the same formulation in courtyards, building surroundings, and other open areas. They emphasize long-range navigation, instruction grounding, and target localization under larger spatial scales and different visual and geometric conditions.

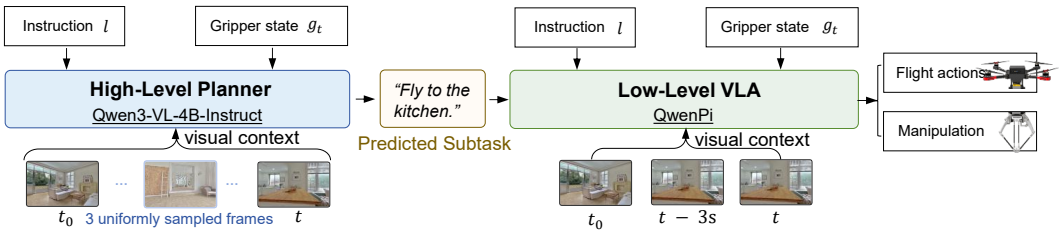

Figure 4: **Architecture of CROSSFLY-AGENT (§4.1).** A high-level VLM planner uses visual context spanning the observed trajectory to predict the active language subtask. Conditioned on this subtask, a low-level VLA policy combines the initial observation with recent visual context to generate flight action chunks and determine when to predict a manipulation-invocation signal.

- ***Indoor↔Outdoor.*** These tasks require the agent to navigate across indoor and outdoor environments in either direction, starting in one environment and reaching a target located in the other. The agent must identify and approach an appropriate transition region, such as a doorway or building entrance, cross the environmental boundary, and continue toward the target. This suite evaluates robust navigation under substantial changes in appearance, illumination, and spatial constraints, while maintaining effective navigation performance across heterogeneous environments.

Interaction Readiness Rate. Alongside standard navigation metrics, we report the proposed Interaction Readiness Rate (IRR) to evaluate whether navigation terminates in a state suitable for subsequent physical interaction. Let $\mathcal{R}$ denote the set of episodes in which UAV successfully reaches the target interaction region. We define the interaction-ready terminal state set as:

$$\mathcal{X}_{\text{int}} = \{s \mid d(s) \le d_{\max},\ \psi(s) \in [\psi_{\min}, \psi_{\max}]\}\,, \tag{1}$$

where $d(s)$ denotes the distance from the terminal UAV state to the interaction point, and $\psi(s)$ denotes the relative yaw offset between UAV heading and the target-facing direction. IRR is then defined as:

$$\text{IRR} = \begin{cases} \frac{1}{|\mathcal{R}|}\sum_{i\in\mathcal{R}} \mathbb{I}\left[s_i^T \in \mathcal{X}_{\text{int}}\right], & |\mathcal{R}| > 0, \\ 0, & |\mathcal{R}| = 0. \end{cases} \tag{2}$$

Here s_i^T denotes the terminal state of trajectory i and $\mathbb{I}[\cdot]$ is the indicator function. We set $d_{\max} = 0.626\,\text{m}$ and $[\psi_{\min}, \psi_{\max}] = [-50.5°, 47.5°]$. These thresholds define the required spatial proximity and viewpoint alignment for downstream interaction, and are further analyzed in §4.4.

IRR therefore measures the fraction of target-reaching trajectories whose terminal states satisfy the defined interaction-ready conditions, thereby quantifying terminal-state suitability beyond navigation completion. Conditioning on successful target reaching isolates terminal-state quality from navigation success, making IRR complementary to standard navigation metrics such as SR.

4 EXPERIMENT

CROSSFLY is proposed to support long-horizon aerial interaction-oriented mobility across heterogeneous environments with diverse task settings. To demonstrate its broad applicability and evaluate the capabilities of existing methods, extensive experiments are conducted. First, the evaluation metrics and baseline methods are introduced in §4.1. Second, representative methods are benchmarked on seen scenes and unseen scenes in §4.2. Then, qualitative analyses of representative trajectories are presented in §4.3. Finally, we conduct diagnostic experiments to analyze the effects of environmental transitions and examine the connection between IRR and manipulation in §4.4.

4.1 EXPERIMENTAL SETUP

Evaluation Metric. Following standardized protocols (Wang et al., 2025a; Liu et al., 2026), we use three standard metrics for evaluation: i) Success Rate (SR); ii) Navigation Error (NE); and iii) Oracle Success Rate (OSR). In addition, the proposed Interaction Readiness Rate (IRR) is reported to evaluate whether UAV reaches a state suitable for interaction after completing navigation tasks.

Table 2: **Quantitative comparison on seen scenes (§4.2).**

Model	*Indoor*				*Outdoor*				*Indoor↔ Outdoor*			
	NE ↓	SR ↑	OSR ↑	IRR ↑	NE ↓	SR ↑	OSR ↑	IRR ↑	NE ↓	SR ↑	OSR ↑	IRR ↑
π_0 (Black et al., 2024)	4.07	17.7	23.6	37.2	3.79	20.4	27.8	37.0	7.21	13.2	19.7	29.1
π_0-FAST (Pertsch et al., 2025)	3.81	18.5	19.2	39.0	3.42	22.7	30.1	38.9	6.63	15.8	22.1	30.7
OpenFly-Agent (Gao et al., 2026)	3.48	15.1	21.4	46.7	3.39	18.1	20.6	43.4	5.93	13.9	17.7	34.2
OpenVLA-OFT (Kim et al., 2025)	3.84	17.2	21.7	47.9	3.51	19.5	24.4	38.1	6.87	14.9	21.3	38.8
GR00T-N1.7 (Bjorck et al., 2025)	3.66	18.9	19.7	40.2	3.43	21.3	24.2	40.0	6.37	11.7	15.6	31.5
Ours	**3.32**	**23.5**	**30.5**	**51.0**	**3.17**	**25.5**	**31.7**	**50.3**	**5.42**	**19.8**	**24.3**	**43.0**
Error Bar (±)	(0.11)	(1.6)	(1.7)	(1.1)	(0.10)	(1.5)	(1.6)	(1.0)	(0.15)	(1.8)	(1.9)	(1.2)

Table 3: **Quantitative comparison on unseen scenes (§4.2).**

Model	*Indoor*				*Outdoor*				*Indoor↔ Outdoor*			
	NE ↓	SR ↑	OSR ↑	IRR ↑	NE ↓	SR ↑	OSR ↑	IRR ↑	NE ↓	SR ↑	OSR ↑	IRR ↑
π_0 (Black et al., 2024)	5.72	10.8	13.4	26.7	5.26	13.5	15.8	27.8	9.38	2.4	4.1	21.7
π_0-FAST (Pertsch et al., 2025)	5.18	13.2	15.7	28.4	4.83	15.6	17.2	29.1	8.74	3.7	6.3	25.2
OpenFly-Agent (Gao et al., 2026)	4.49	10.6	13.9	32.7	4.22	13.4	15.1	26.4	7.61	7.2	7.9	26.2
OpenVLA-OFT (Kim et al., 2025)	5.45	11.9	14.6	27.5	4.97	14.1	16.4	28.2	9.02	3.1	4.7	27.9
GR00T-N1.7 (Bjorck et al., 2025)	5.03	13.8	16.2	28.8	4.61	16.2	18.0	29.7	8.56	4.2	5.6	23.8
Ours	**4.05**	**17.3**	**18.6**	**35.5**	**3.83**	**18.5**	**22.7**	**35.3**	**6.91**	**9.7**	**11.4**	**29.3**
Error Bar (±)	(0.17)	(1.4)	(1.5)	(1.1)	(0.16)	(1.5)	(1.5)	(1.1)	(0.24)	(0.9)	(1.1)	(0.7)

Baselines. We evaluate several representative open-source agents spanning different action-generation paradigms and training distributions. π_0 (Black et al., 2024) is a Flow Matching-based foundation model pretrained on large-scale cross-embodiment robot data. π_0-FAST (Pertsch et al., 2025) uses FAST action tokenization for efficient autoregressive action generation. OpenFly-Agent (Gao et al., 2026) represents agents developed for outdoor aerial scenarios, OpenVLA-OFT (Kim et al., 2025) represents pretrained VLA agents adapted to downstream control tasks through optimized fine-tuning, and GR00T-N1.7 (Bjorck et al., 2025) represents generalist VLA agents developed for humanoid robot control. For a fair comparison, all baselines are retrained on CROSSFLY training datasets following unified training recipes. More details are provided in §C.

CROSSFLY-Agent. We construct CROSSFLY-AGENT as a hierarchical agent for long-horizon interaction-oriented aerial mobility (Fig. 4), consisting of a high-level VLM planner and a low-level VLA policy. The planner predicts the subtask from the instruction, gripper state, and visual context comprising the initial frame, three uniformly sampled historical frames, and the current frame. The policy uses the initial frame, a frame from 3 s earlier, and the current frame. Conditioned on this visual context, the predicted subtask, global instruction, and gripper state, the policy adopts QwenPi (StarVLA Community, 2026) to generate action chunks together with a manipulation-invocation signal. The planner and policy are trained independently and coupled through the language subtask interface. The planner is fine-tuned on annotated subtasks, while the policy learns action chunks via flow matching, with supervision for manipulation invocation. The policy uses ground-truth subtasks during training and planner predictions at inference. See §A for more details.

4.2 PERFORMANCE BENCHMARKING

Performance on Seen Scenes. Table 2 reports the performance of different methods on seen scenes. These results characterize the capability of existing state-of-the-art methods in environments included in the training distribution. CROSSFLY-AGENT achieves the best performance across all metrics and task suites, indicating that explicit high-level task decomposition is useful for interaction-oriented aerial mobility tasks. Compared with existing methods without high-level planning (Black et al., 2024; Pertsch et al., 2025; Gao et al., 2026; Kim et al., 2025; Bjorck et al., 2025), CROSSFLY-AGENT obtains consistent improvements across different task suites. For example, on the *Indoor* suite, it achieves SR of **23.5%** compared to 17.7% from π_0 (Black et al., 2024), while also improving IRR from 37.2% to **51.0%**. These results show that CROSSFLY evaluates both target reaching and interaction readiness, revealing performance differences among aerial navigation agents.

Generalization on Unseen Scenes. As shown in Table 3, all agents experience performance degradation in unseen environments, while CROSSFLY-AGENT maintains the best performance across all metrics and task suites. In particular, on the *Outdoor* task suite, CROSSFLY-AGENT improves SR

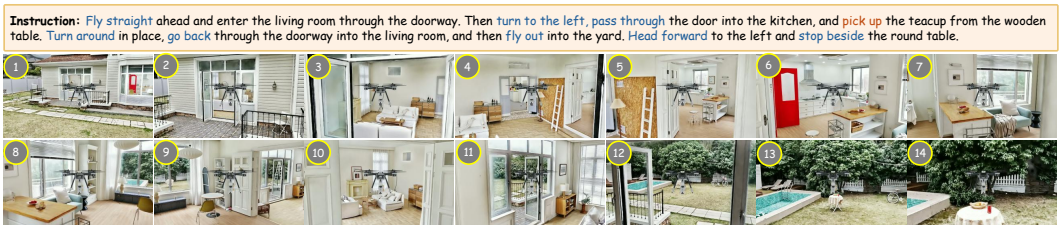

Figure 5: **Qualitative results of CROSSFLY-AGENT (§4.3).** This example shows the instruction and selected third-person views. Blue text denotes navigation actions, while red text denotes manipulation actions. The top row shows the trajectory from the start to the interaction-ready state, while the bottom row covers the trajectory from the interaction-ready state to task completion.

from 16.2% to **18.5%**, and reduces NE from 4.22 to **3.83**. These results suggest that hierarchical planning and subtask supervision in CROSSFLY-AGENT provide useful guidance in unseen scenes.

4.3 QUALITATIVE RESULTS

Fig. 5 presents the qualitative results of CROSSFLY-AGENT. The example illustrates a challenging *Indoor↔Outdoor* task, requiring the agent to adapt its navigation behavior when transitioning between indoor and outdoor environments. More visualizations are provided in §A.2. These qualitative results further demonstrate the effectiveness of the hierarchical design and subtask supervision in handling navigation tasks with varying scene constraints and planning horizons.

4.4 DIAGNOSTIC EXPERIMENTS

Controlled Comparison across Task Suites. To investigate whether the performance degradation of *Indoor↔Outdoor* tasks is associated with environmental transitions rather than differences in other trajectory properties, we conduct a controlled comparison with matched trajectory statistics. Specifically, we construct matched subsets for the *Indoor* and *Outdoor* suites and the *Indoor↔Outdoor* suite by controlling for trajectory length, number of subtasks, and instruction length. We select 5 reconstructed scenes and sample 30 matched trajectories from each suite within each scene. This design minimizes the influence of variations in navigation horizon, scene gaps, and language instruction variations, enabling a more controlled comparison across the three task suites.

Table 4: **Performance comparison under matched trajectories.**

Task Suite	Traj. Length	# Subtasks	Instr. Length	NE ↓	SR ↑
Indoor	11.2	6.8	18.9	4.15	23.64
Outdoor	11.8	6.7	18.5	4.36	25.28
Indoor↔Outdoor	11.5	6.8	18.8	6.43	10.98

As shown in Table 4, all task suites contain comparable trajectories. Despite receiving similar training supervision, agents consistently achieve lower performance on the *Indoor↔Outdoor* task suite compared with single-domain tasks. These results suggest that the performance gap cannot be fully explained by differences in training exposure or trajectory properties. Instead, cross-domain navigation presents additional challenges that require agents to adapt to changes in visual appearance and spatial constraints when transitioning between different domains.

Manipulation Case Study. To examine whether IRR reflects the suitability of a navigation terminal state for subsequent manipulation, we train a reinforcement learning policy for grasping on a hand-like UAV platform, using the target object pose estimated from the depth map as input. The grasping policy is initialized from the terminal state produced by the navigation agent. As shown in Fig. 6, the top case satisfies the distance and target-view constraints defined by IRR, enabling the policy to successfully grasp the target object. In contrast, the bottom case violates IRR criteria, and the same policy fails to complete the grasp. These examples demonstrate that IRR can distinguish navigation terminal states that lead to different outcomes under the same downstream grasping policy.

Empirical Validation of IRR Criteria. To ground IRR criteria in downstream manipulation capability, we evaluate manipulation success across multiple aerial embodiments, each with its own manipulation policy. For each embodiment, we randomly sample terminal UAV states around fixed target objects and record grasp outcomes while varying the distance to the target and rela-



Figure 6: **Manipulation case study (§4.4).** Each row represents a manipulation case. The leftmost image shows the initial observation of UAV. The top case satisfies IRR criteria and achieves a successful grasp, whereas the bottom case violates IRR criteria and fails to grasp.

tive yaw. We then estimate the interaction readiness boundaries through maximum-likelihood fitting of the grasp outcomes, using candidate spatial and viewpoint partitions induced by the sampled terminal states. Aggregating the manipulation outcomes across all embodiments and their corresponding policies yields the spatial threshold $\hat{d}_{\max} = 0.626$ m and the valid viewpoint interval $[\hat{\psi}_{\min}, \hat{\psi}_{\max}] = [-50.5°, 47.5°]$. The resulting region provides an empirical approximation of the terminal configurations that support downstream grasping. Thus, IRR criteria are grounded in manipulation feasibility across multiple embodiments rather than manually specified thresholds.

5 Limitations and Impact

Limitations. Despite the effectiveness of CrossFly in providing a unified evaluation framework for interaction-oriented aerial mobility across heterogeneous environments, it still has several limitations that could be addressed in future work. **First**, CrossFly is currently developed in simulation environments. While the benchmark incorporates realistic rendering and physics-based UAV dynamics, the simulation-to-real gap caused by sensor noise, aerodynamic disturbances, and hardware constraints remains unexplored. Future work will investigate real-world deployment and transfer protocols. **Second**, CrossFly evaluates agents under feasible trajectories generated from successful demonstrations, while real-world long-horizon autonomy requires robust failure recovery and online adaptation. Future benchmarks could incorporate interactive learning, recovery behaviors, and continual adaptation to better evaluate embodied intelligence in open-world environments.

Social Impact. CrossFly advances the evaluation of interaction-oriented aerial mobility by assessing goal reaching and whether terminal states are suitable for interaction across heterogeneous environments. On the positive side, it may facilitate the development of aerial embodied agents. For potential negative impact, CrossFly is simulation-based and does not fully capture hardware constraints or failure recovery. Hence, its utility for real-world deployment should be further examined.

6 Conclusion

In this work, we introduce CrossFly, a benchmark for interaction-oriented aerial mobility across heterogeneous environments. It provides 15K trajectories across 23 reconstructed scenes and evaluates both navigation performance and interaction readiness through a unified protocol. We propose the Interaction Readiness Rate to measure whether the terminal state satisfies the spatial and viewpoint requirements for interaction. Experiments with five aerial agents and CrossFly-Agent show that existing methods face challenges in indoor–outdoor transition tasks, highlighting the difficulty of connecting long-horizon navigation with physical interaction. We hope CrossFly facilitates the development of interaction-oriented aerial mobility by providing a standardized benchmark.

7 AI Use Statement

AI tools are used for language polishing of the manuscript and subtask annotation. All AI-assisted edits are reviewed by the authors, who take full responsibility for the final content of this work.

REFERENCES

Jianli Sun, Bin Tian, Qiyao Zhang, Chengxiang Li, Zihan Song, Zhiyong Cui, Yisheng Lv, and Yonglin Tian. AIR-VLA: A benchmark for aerial vision-language-action. *arXiv preprint arXiv:2601.21602*, 2026a.

Jianli Sun, Bin Tian, Qiyao Zhang, Zijian Liu, Yutong Wang, Zhiyong Cui, Bai Li, Yisheng Lv, and Yonglin Tian. AIR-VLA+: Decoupling movement and manipulation via cascaded dual-action decoders with asymmetric MoE for aerial robots. *arXiv preprint arXiv:2606.12859*, 2026b.

Yutong Wang, Dongjae Lee, Xiaofeng Guo, Yuanzhu Zhan, Yufei Jiang, Bavin Saravanan, Muqing Cao, Jia Xie, Chenyang Mao, Sebastian Scherer, Junyi Geng, and Guanya Shi. AM-Bench: A modular simulation suite and benchmark for aerial manipulation policy learning. In *CoRL*, 2026.

Johnathan Tucker, Denis Liu, Aiden Swann, Allen Ren, Javier Yu, Jiankai Sun, Brandon Kim, Lachlain McGranahan, Quan Vuong, and Mac Schwager. π, but make it fly: Physics-guided transfer of VLA models to aerial manipulation. In *CoRL*, 2026.

Guanqi He, Xiaofeng Guo, Luyi Tang, Yuanhang Zhang, Mohammadreza Mousaei, Jiahe Xu, Junyi Geng, Sebastian Scherer, and Guanya Shi. Flying hand: End-effector-centric framework for versatile aerial manipulation teleoperation and policy learning. In *RSS*, 2025.

Yuzhao Luo, Ming Zhu, Xinghan Liu, Hengzhi Su, Yapeng Hu, Yuncheng Liu, and Tian Chen. AirUniNav: Unified vision-language navigation for UAVs in indoor and outdoor scenes. *TechRxiv preprint*, 2026.

Pingrui Zhang, Xianqiang Gao, Yuhan Wu, Kehui Liu, Dong Wang, Zhigang Wang, Bin Zhao, Yan Ding, and Xuelong Li. MoMa-Kitchen: A 100k+ benchmark for affordance-grounded last-mile navigation in mobile manipulation. In *ICCV*, 2025a.

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

Yunpeng Gao, Chenhui Li, Zhongrui You, Junli Liu, Zhen Li, Pengan Chen, Qizhi Chen, Zhonghan Tang, Liansheng Wang, Penghui Yang, Yiwen Tang, Yuhang Tang, Shuai Liang, Songyi Zhu, Ziqin Xiong, Yifei Su, Xinyi Ye, Jianan Li, Yan Ding, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. OpenFly: A comprehensive platform for aerial vision-language navigation. In *ICLR*, 2026.

Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. AerialVLN: Vision-and-language navigation for UAVs. In *ICCV*, 2023.

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 *ICCV*, 2025.

Xu Liu, Yu Liu, Hanshuo Qiu, Yang Qirong, and Zhouhui Lian. IndoorUAV: Benchmarking vision-language UAV navigation in continuous indoor environments. In *AAAI*, 2026.

Junzhe Wu, Yue Hu, Zeyu Han, Po-Hsun Chang, Yinan Dong, Behrad Rabiei, and Maani Ghaffari. NavVerse: Benchmarking indoor-to-outdoor embodied navigation in continuous robot simulation. *arXiv preprint arXiv:2607.19695*, 2026a.

Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, and Xin Wang. Aerial vision-and-dialog navigation. In *ACL*, 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. In *ICLR*, 2025b.

Jingyu Guo, Ziye Chen, Ziwen Li, Zhengqing Gao, Jiaxin Huang, Hanlue Zhang, Fengming Huang, Yu Yao, Tongliang Liu, and Mingming Gong. HUGE-Bench: A benchmark for high-level UAV vision-language-action tasks. *arXiv preprint arXiv:2603.19822*, 2026.

Tzu-Hsien Lee, Fidan Mahmudova, and Karthik Desingh. Learning category-level last-meter navigation from RGB demonstrations of a single-instance. *IEEE Robotics and Automation Letters*, 2026.

Weichen Zhang, Chen Gao, Shiquan Yu, Ruiying Peng, Baining Zhao, Qian Zhang, Jinqiang Cui, Xinlei Chen, and Yong Li. CityNavAgent: Aerial vision-and-language navigation with hierarchical semantic planning and global memory. In *ACL*, 2025b.

Hengxing Cai, Jinhan Dong, Jingjun Tan, Jingcheng Deng, Sihang Li, Zhifeng Gao, Haidong Wang, Zicheng Su, Agachai Sumalee, and Renxin Zhong. FlightGPT: Towards generalizable and interpretable UAV vision-and-language navigation with vision-language models. In *EMNLP*, 2025.

Yuwei Ning, Ganlong Zhao, Yipeng Qin, Si Liu, Yang Liu, Liang Lin, and Guanbin Li. Lookaside-VLN: Direction-aware aerial vision-and-language navigation. In *CVPR*, 2026.

Chengjie Fan, Cong Pan, Zijian Liu, Ningzhong Liu, and Jie Qin. HTNav: A hybrid navigation framework with tiered structure for urban aerial vision-and-language navigation. In *CVPR*, 2026.

Yu Qi, Hongyu Li, Shaofei Huang, Tianrui Hui, Yaxiong Wang, Lechao Cheng, Zhun Zhong, Si Liu, and Meng Wang. Parse, search, and confirmation: Training-free aerial vision-and-dialog navigation with chain-of-thought reasoning and structured spatial memory. In *CVPR*, 2026.

Xichen Ding, Jianzhe Gao, Cong Pan, Wenguan Wang, and Jie Qin. History-enhanced two-stage transformer for aerial vision-and-language navigation. In *AAAI*, 2026.

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 *AAAI*, 2026a.

Peng Xu, Zhengnan Deng, Jiayan Deng, Zonghua Gu, and Shaohua Wan. AeroVLA: A vision-language-action model for UAV navigation via minimalist end-to-end control. In *ECCV*, 2026.

Xiaolou Sun, Wufei Si, Wenhui Ni, Yuntian Li, Dongming Wu, Fei Xie, Runwei Guan, He-Yang Xu, Henghui Ding, Yuan Wu, Yutao Yue, Yongming Huang, and Hui Xiong. AutoFly: Vision-language-action model for UAV autonomous navigation in the wild. In *ICLR*, 2026c.

Chih Yao Hu, Yang-Sen Lin, Yuna Lee, Chih-Hai Su, Jie-Ying Lee, Shr-Ruei Tsai, Chin-Yang Lin, Kuan-Wen Chen, Tsung-Wei Ke, and Yu-Lun Liu. See, point, fly: A learning-free VLM framework for universal unmanned aerial navigation. In *CoRL*, 2025.

Qianzhong Chen, Naixiang Gao, Suning Huang, JunEn Low, Timothy Chen, Jiankai Sun, and Mac Schwager. GRaD-Nav++: Vision-language model enabled visual drone navigation with gaussian radiance fields and differentiable dynamics. *IEEE Robotics and Automation Letters*, 11(2):1418–1425, 2026b.

Wen Jiang, Li Wang, Kangyao Huang, Wei Fan, Jinyuan Liu, Shaoyu Liu, Hongwei Duan, Bin Xu, and Xiangyang Ji. LongFly: Long-horizon aerial vision-language navigation. *arXiv preprint arXiv:2512.22010*, 2025.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

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

Matt Schmittle, Anna Lukina, Lukas Vacek, Jnaneshwar Das, Christopher P Buskirk, Stephen Rees, Janos Sztipanovits, Radu Grosu, and Vijay Kumar. OpenUAV: A UAV testbed for the CPS and robotics community. In *ICCPS*, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Xin Zhou, Xiangyong Wen, Zhepei Wang, Yuman Gao, Haojia Li, Qianhao Wang, Tiankai Yang, Haojian Lu, Yanjun Cao, Chao Xu, and Fei Gao. Swarm of micro flying robots in the wild. *Science Robotics*, 7(66):eabm5954, 2022.

Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Khalil Meftah, Maxime Ellerbach, Jess Moss, and Thomas Wolf. LeRobot: State-of-the-art machine learning for real-world robotics in PyTorch. https://github.com/huggingface/lerobot, 2024.

Yuze Wu, Fan Yang, Rui Jin, Yuhang Zhong, Junjie Wang, Xuankang Wu, and Fei Gao. Hand-like autonomous flying robot for airborne grasping and interaction. *Nature Communications*, 17(1), 2026b.

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.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

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

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

In the appendix, we provide the following items that shed deeper insight into our contributions:

- §A: Additional details on the CROSSFLY-AGENT.
- §B: Supplementary information of benchmark splits.
- §C: Implementation details.
- §D: Discussion.

A ADDITIONAL DETAILS ON THE CROSSFLY-AGENT

A.1 METHOD

Overview. We introduce CROSSFLY-AGENT, a hierarchical agent for aerial mobile manipulation. A high-level VLM planner predicts language subtasks using visual observations sampled across the trajectory. Conditioned on the predicted subtask, a low-level VLA policy uses the initial and recent observations to generate flight action chunks and determine when to invoke manipulation.

High-Level Planning. Let ℓ denote the global instruction, o_t the current RGB observation, and g_t the gripper state. The high-level VLM planner uses Qwen3-VL-4B-Instruct (Qwen Team, 2025) to predict the active language subtask. Its visual context consists of the initial observation o_0, three historical frames $\mathcal{R}_t^{\text{plan}}$ uniformly sampled between the initial and current observations, and the current observation o_t:

$$z_t = P_\theta(\ell, g_t, o_0, \mathcal{R}_t^{\text{plan}}, o_t). \tag{3}$$

This five-frame context captures task progress over the observed trajectory.

Low-Level Action Generation. The low-level VLA policy adopts QwenPi (StarVLA Community, 2026), with its vision-language backbone initialized from Qwen3-VL-4B-Instruct. Conditioned on the instruction, predicted subtask, and gripper state, it uses the initial observation, an observation from $\Delta = 3$ s earlier, and the current observation to generate a flight action chunk:

$$\mathbf{A}_t = (a_t, \ldots, a_{t+H-1}) \sim \pi_\phi(\cdot \mid \ell, z_t, g_t, o_0, o_{t-\Delta}, o_t), \tag{4}$$

where H is the chunk length and each flight action specifies a 6-DoF command. The policy also determines when to invoke downstream manipulation, ending the current flight segment and transferring control for grasping or placing.

Training. The planner and policy are trained independently. The planner is fine-tuned on annotated subtasks using a language-modeling loss. The policy learns action chunks through conditional flow matching. Let $c_t = (\ell, z_t, g_t, o_0, o_{t-\Delta}, o_t)$ denote the policy inputs. Given a demonstrated action chunk $\mathbf{A}_t$, noise $\epsilon \sim \mathcal{N}(0, I)$, and flow time τ, we construct $\mathbf{A}_t^\tau = (1-\tau)\mathbf{A}_t + \tau\epsilon$ and minimize

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{A}_t, c_t, \epsilon, \tau}\left[\left\| v_\phi(\mathbf{A}_t^\tau, \tau \mid c_t) - (\epsilon - \mathbf{A}_t) \right\|_2^2\right], \tag{5}$$

where v_ϕ predicts the conditional velocity field. The policy additionally receives supervision for manipulation invocation. It uses ground-truth subtasks during training and planner predictions at inference. Each component uses the same visual sampling strategy during training and inference.

A.2 More Qualitative Results

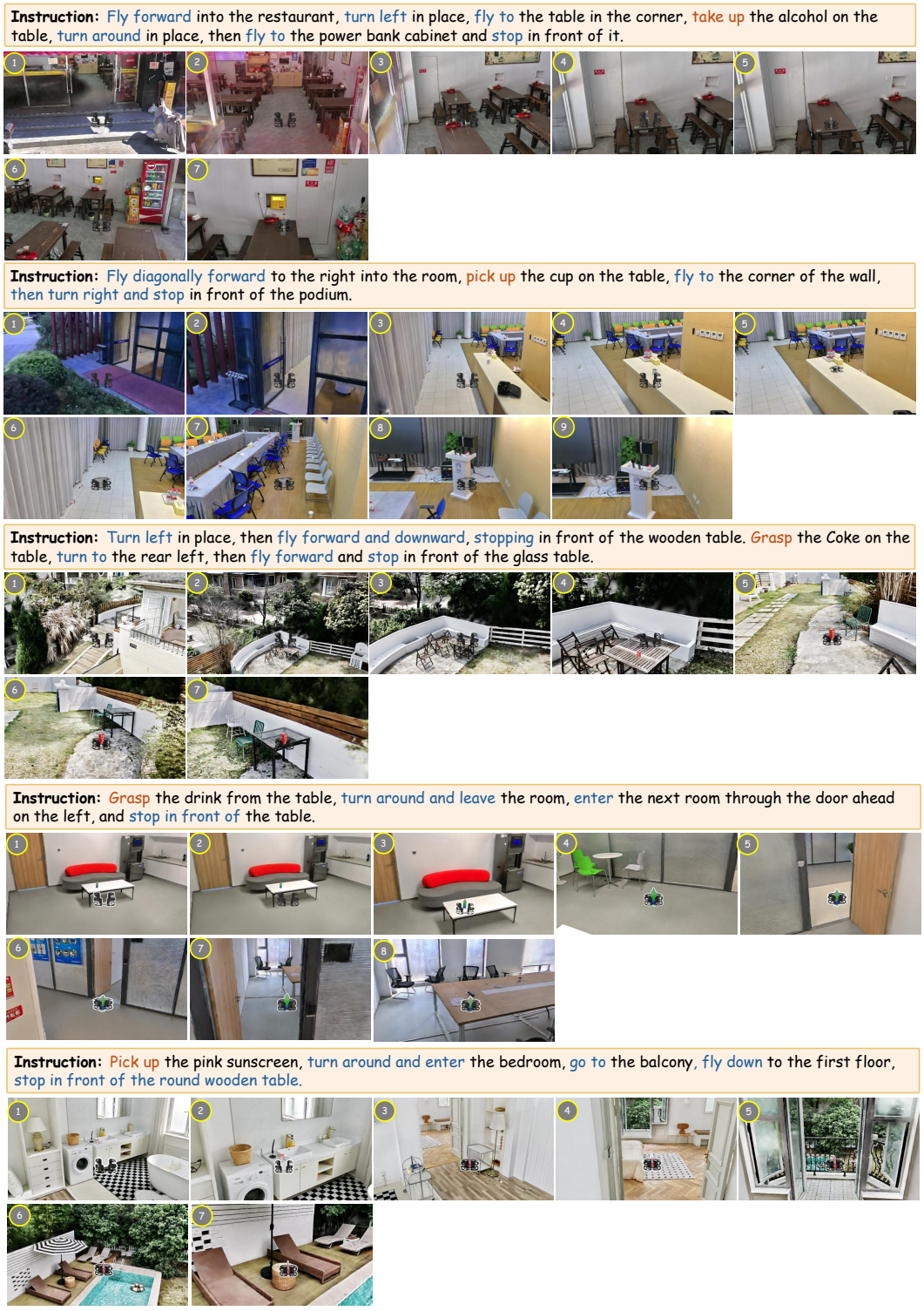

Figure 7: **Qualitative results of CrossFly-Agent.**

Table 5: **Scene-level partition used in CROSSFLY (§B.1).**

Split	# Scenes	Scene overlap with training	Used for training
Training	18	–	Yes
Seen	18	Same scenes	No
Unseen	5	None	No

A.3 FAILURE CASE

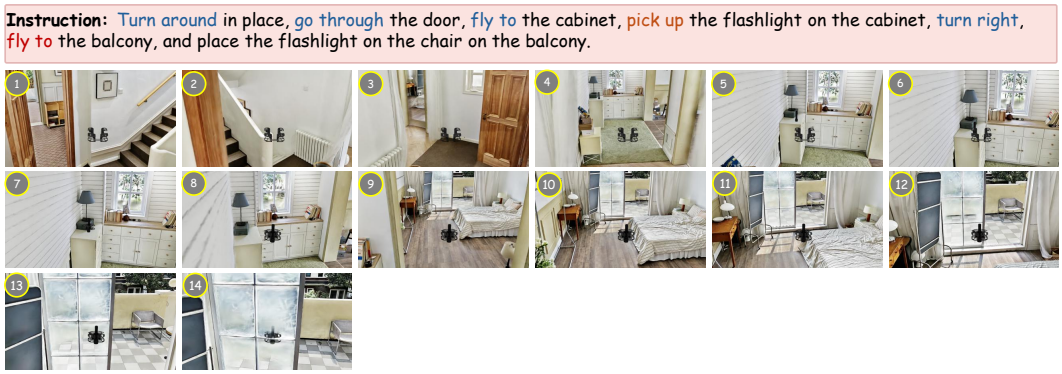

Figure 8: **Failure case.** Sometimes, UAV collides with obstacles when traversing passages across indoor–outdoor boundaries, indicating that navigation in transition regions remains challenging.

B SUPPLEMENTARY INFORMATION OF BENCHMARK SPLITS

B.1 SCENE-LEVEL SPLIT

CROSSFLY contains 23 independently reconstructed scenes and 15,263 trajectories in total. We construct the benchmark split at the scene level, rather than randomly splitting individual trajectories. Let $\mathcal{S}$ denote the complete set of reconstructed scenes. As shown in Table 5, we divide it into a training-scene set $\mathcal{S}_{\text{train}}$ and an unseen-scene set $\mathcal{S}_{\text{unseen}}$:

$$\mathcal{S} = \mathcal{S}_{\text{train}} \cup \mathcal{S}_{\text{unseen}}, \qquad \mathcal{S}_{\text{train}} \cap \mathcal{S}_{\text{unseen}} = \varnothing, \tag{6}$$

where $|\mathcal{S}_{\text{train}}| = 18$ and $|\mathcal{S}_{\text{unseen}}| = 5$, with $18 + 5 = 23$. The split is fixed and shared by all evaluated methods.

Scene assignment is atomic: all trajectories associated with the same reconstructed scene are assigned to the same scene partition, regardless of whether they belong to the *Indoor*, *Outdoor*, or *Indoor↔Outdoor* task suite. In particular, trajectories from different regions or task suites of the same reconstructed scene are never separated between the training and unseen partitions.

The partition is therefore strictly isolated at the scene level. All trajectories generated from scenes in $\mathcal{S}_{\text{unseen}}$, together with their RGB observations, 6-DoF UAV states, waypoint annotations, language instructions, and subtask annotations, are excluded from model training. Consequently, no trajectory or observation from an unseen scene is exposed to an agent during training.

B.2 SEEN- AND UNSEEN-SCENE EVALUATION

We distinguish between seen-scene and unseen-scene evaluation according to whether the underlying reconstructed environment is represented in the training set.

Seen-Scene Evaluation. The seen split consists of held-out evaluation trajectories collected from scenes in $\mathcal{S}_{\text{train}}$. Although the underlying scenes are represented during training, the evaluation trajectories themselves are excluded from optimization. Denoting the training and seen evaluation trajectory sets by $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{seen}}$, respectively, we enforce

$$\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{seen}} = \varnothing. \tag{7}$$

Therefore, seen-scene evaluation measures generalization to new episodes, routes, and task configurations within environments observed during training, rather than performance on trajectories used for optimization.

Unseen-Scene Evaluation. The unseen split consists exclusively of trajectories collected from $\mathcal{S}_{\text{unseen}}$. Let $\mathcal{D}_{\text{unseen}}$ denote the corresponding evaluation trajectory set. By construction,

$$\text{scene}(\mathcal{D}_{\text{train}}) \cap \text{scene}(\mathcal{D}_{\text{unseen}}) = \varnothing. \tag{8}$$

Thus, unseen-scene evaluation introduces both trajectory-level and scene-level separation from the training data. Agents are evaluated in reconstructed environments whose visual observations, scene geometry, and spatial layouts are not encountered during training. This setting measures generalization to novel environments rather than to new trajectories within familiar scenes.

C Implementation Details

For reproducibility, we provide the detailed training configuration used for all VLA models. Unless otherwise specified, all VLA models follow the same training protocol, as summarized in Table 6.

Table 6: **Training configuration for VLA models.**

Training Configuration	Setting
Batch size	32
Learning rate	1×10^{-5}
Learning rate schedule	Warmup + cosine decay
Optimizer	AdamW
AdamW β	$(0.9, 0.95)$
Weight decay	1×10^{-8}
Max. training steps	100,000
Action horizon	36
Precision	bf16
Training hardware	$4\times$ NVIDIA H100 GPUs

We apply this training protocol consistently to all VLA-based models in our experiments.

D Discussion

Aerial mobile manipulation naturally couples two stages operating at different spatial and control scales: long-range mobility and local physical interaction. While mobility requires UAV to follow instructions, traverse potentially heterogeneous environments, and approach a target, manipulation is typically executed after the platform reaches a suitable pose. The key challenge is the handoff from mobility to manipulation. CrossFly makes this handoff explicit through interaction-oriented mobility and the Interaction Readiness Rate (IRR), which evaluates whether the terminal navigation state satisfies the spatial and viewpoint conditions required for downstream interaction. Once such a state is reached, an interaction-invocation signal can trigger a task-specific manipulation policy.

This formulation provides a modular pathway for extending aerial VLA agents toward mobile manipulation without requiring a single policy to jointly learn long-horizon flight and local interaction. Existing aerial VLA agents can retain their role in instruction following and target approach, while downstream policies handle grasping or other local skills. Our manipulation experiments provide initial evidence for this interface: under the same grasping policy, interaction-ready states can support successful manipulation, whereas unsuitable arrival states may lead to failure. More broadly, this modular view enables mobility and manipulation to be developed independently while remaining connected within a unified embodied execution pipeline.