THINK LIKE A PILOT: FINE-GRAINED LONG-HORIZON UAV NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-guided UAV agents must execute long-horizon semantic instructions while producing smooth, physically feasible continuous flight commands. Existing Vision-Language Navigation (VLN) benchmarks typically use discrete or coarse actions and existing UAV Vision-Language-Action (VLA) tasks focus on short, atomic maneuvers. To address this gap in UAV task settings, we introduce **FLIGHT**, a **F**ine-grained **L**ong-horizon **I**nstruction-**G**uided benchmark for **H**ybrid UAV navigation and reasoning **T**asks, which combines multi-stage instructions with dense 6-DoF trajectory annotations across two dataset splits: Fine-grained VLN and Long-horizon Flow. To endow the UAV agent with the capability of in-flight reasoning over task execution status and mission planning, while simultaneously accommodating high-frequency precise control, we further propose **FLIGHT VLA**, an asynchronous architecture that decouples a low-frequency Streaming Pilot Vision-Language Model (VLM) for task-state reasoning from a high-frequency diffusion action model for continuous control, supervised by explicit **Pilot Reasoning** texts that summarize the current flight state and anticipate the next subgoal. In closed-loop evaluation, FLIGHT VLA consistently surpasses representative VLN and VLA baselines on our FLIGHT benchmarks, achieving stronger multi-stage completion and subgoal adherence.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are currently undergoing a profound paradigm shift, evolving from specialized flight platforms into autonomous agents capable of executing complex tasks in open-ended environments. This evolution expands their utility across diverse scenarios, including logistics delivery and disaster rescue. Moving away from traditional operations that rely on waypoint planning or manual remote control, language-guided UAV control empowers users to directly specify mission objectives via natural language. Consequently, the UAV must autonomously comprehend commands, perceive its surroundings, and execute continuous flight maneuvers. This paradigm promises to simplify UAV deployment and enable autonomous aerial missions.

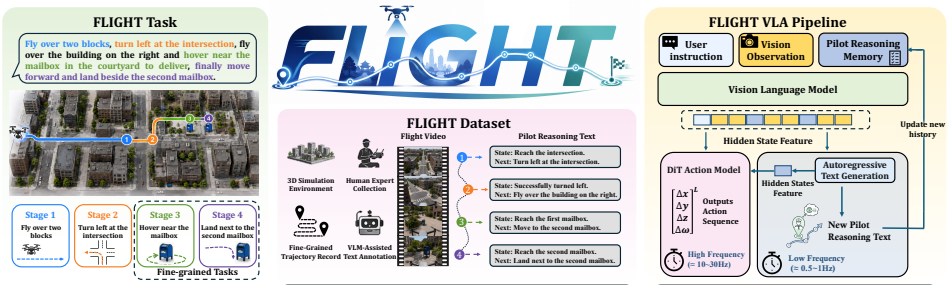


Figure 1: Overview of the FLIGHT benchmark and VLA framework. Left: FLIGHT targets UAV tasks requiring long-horizon navigation and fine-grained control. Middle: Flight videos with fine-grained trajectories, language annotations, and Pilot Reasoning captions. Right: FLIGHT VLA asynchronously combines a streaming video VLM with a VLA model, using Pilot Reasoning memory to bridge task-level reasoning and real-time control.

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.

Recent advances in Vision-Language Navigation (VLN) (Liu et al., 2023; Gao et al., 2025; Wang et al., 2025b;c) and Vision-Language-Action (VLA) (Wang et al., 2025a; Liu et al., 2026; Zhang et al., 2025) have demonstrated remarkable potential in grounding language instructions to embodied perception and action. However, existing VLN benchmarks primarily address destination finding, typically using discrete actions or simplified motion models that inadequately capture continuous trajectories and 3D flight constraints. Meanwhile, existing VLA approaches advance continuous UAV control but remain focused on short-horizon, atomic instructions. As aerial embodied agents, UAVs require both long-horizon semantic understanding and precise, smooth, physically feasible continuous control. As illustrated in the left panel of Fig. 1, complex missions such as "fly two blocks ahead, turn left at the intersection to hover and drop off a package near the mailbox in the courtyard, then proceed forward to finally land next to the second mailbox" can neither be reduced to traditional long-range navigation nor handled by short-term localized control alone.

Long-horizon, fine-grained missions pose two challenges. First, long-range instructions require continuous tracking of sub-goals, completed stages, and upcoming actions in relation to environmental semantics. However, rapid maneuvers drastically alter UAV observations, increasing the risk of semantic drift, sub-goal forgetting, and incorrect planning. Second, robust semantic understanding and reasoning rely on computationally intensive VLMs, whose slow inference limits high-frequency UAV control. Language-guided UAV control must therefore reconcile the temporal mismatch between long-horizon semantic planning and high-frequency continuous control.

To advance research on UAV flight missions, we introduce FLIGHT - a novel benchmark for Fine-grained Long-horizon Instruction-Guided Hybrid UAV Tasks. FLIGHT challenges autonomous agents to execute closed-loop flights guided by multi-stage language instructions. To support it, we collected a large-scale dataset of flight trajectories. Unlike traditional navigation benchmarks using sparse, simplified waypoints, our approach captures dense, fine-grained control points to generate high-fidelity trajectories consistent with flight dynamics. Beyond destination success, FLIGHT evaluates intermediate sub-goal execution over long horizons with continuous flight control.

To address the cognitive challenges inherent to such long-horizon missions, we introduce the **Pilot Reasoning mechanism**. Leveraging large-scale VLM-assisted annotation coupled with human verification, we externalize the "decision-making chain" of human pilots. Specifically, at each critical junction, the mechanism explicitly outputs an analysis of the current flight state along with a proactive planning of the subsequent action. As illustrated in the middle of Fig. 1, these Pilot Reasoning Text examples demonstrate how this explicit reasoning signal acts as a "cognitive anchor," effectively bridging the gap between high-level mission objectives and low-level control commands.

Furthermore, we propose FLIGHT VLA, which asynchronously decouples a slow module—dedicated to streaming video understanding—from a fast module tailored for high-frequency continuous control. The slow module maintains long-horizon context and periodically updates the semantic planning state, while the fast module operates at a significantly higher frequency to generate fine-grained control commands based on the immediate input and the latest planning state. This dual-system enables real-time UAV control while preserving long-horizon semantic comprehension.

We construct a comprehensive evaluation suite for FLIGHT in simulation, adapting representative VLN models, including LAG (Liu et al., 2023) and NaVid (Zhang et al., 2024), alongside VLA models such as OpenVLA (Kim et al., 2024) and MemoryVLA (Shi et al., 2025), as baselines. Empirical results show that existing methods struggle with multi-stage UAV missions, mainly due to limited long-horizon state maintenance and continuous control capabilities. In contrast, FLIGHT VLA integrates Streaming VLM based on Pilot Reasoning chains of thought paradigm with an asynchronous, high-frequency real-time action model, achieving substantial performance gains. Our contributions are summarized as follows:

- We propose **FLIGHT**, a long-horizon fine-grained hybrid-control paradigm and benchmark for UAV navigation, featuring semantically rich multi-stage trajectories and **Pilot Reasoning** supervision to bridge high-level navigation and low-level continuous control.
- We introduce **FLIGHT VLA**, an asynchronous framework that separates streaming semantic reasoning from real-time control, enabling long-context understanding and stable continuous flight.
- We provide a closed-loop evaluation of VLN and VLA baselines, demonstrating the challenges of FLIGHT and the effectiveness of FLIGHT VLA in long-horizon UAV tasks.

2 Related Work

2.1 Language-Guided UAV Tasks

Language-guided UAV benchmarks primarily focus on vision-language navigation (VLN), requiring agents to interpret instructions, plan, and navigate. Aerial VLN (Liu et al., 2023) uses discrete action sequences for language-guided long-range flight; Travel (Wang et al., 2025b) introduces instruction-assisted waypoint-level navigation; and OpenFly (Gao et al., 2025) broadens cross-scene UAV VLN to 18 scenes and 100K trajectories.

Beyond VLN, UAV-Flow (Wang et al., 2025a) first formulates fine-grained language-guided UAV control as a vision-language-action (VLA) task. Its Flow (Flying-on-a-word) dataset pairs short instructions with maneuvers such as circling, traversing, and landing, demonstrating VLA's feasibility for local fine-grained control. Indoor UAV (Liu et al., 2026) builds an indoor navigation dataset and derives IndoorUAV-VLA by selecting keyframes and regenerating concise instructions to split long trajectories into shorter sub-trajectories. However, these benchmarks do not effectively unify long-horizon navigation and fine-grained VLA control. FLIGHT bridges this gap through **Long-horizon Flow**, which composes multiple fine-grained instructions into long-horizon VLA tasks, and **Fine-grained VLN**, which integrates fine-grained control into conventional navigation.

2.2 VLA with Reasoning

Explicit reasoning in VLA introduces intermediate language, visual, or action representations before execution to improve task decomposition, long-horizon planning, and interpretability. ECoT (Zawalski et al., 2024), CoT-VLA (Zhao et al., 2025), dVLA (Wen et al., 2025), and ACoT-VLA (Zhong et al., 2026) respectively explore embodied chain-of-thought, visual subgoal prediction, multimodal CoT, and action-intention reasoning. Grounding intermediate reasoning in environmental states and task objectives improves generalization and decision stability in complex manipulation.

For high-level semantic understanding and task structure, $\pi_{0.5}$ (Intelligence et al., 2025) and ChatVLA-2 (Zhou et al., 2025) preserve or enhance pretrained VLMs' open-world understanding and introduce semantic prediction or task decomposition for multi-stage control. MEM (Torne et al., 2026), VLingNav (Wang et al., 2026), and LongNav-R1 (Hu et al., 2026) explore linguistic memory, adaptive reasoning, and reinforcement learning to mitigate context forgetting and model cross-step dependencies in long-horizon interaction. ThinkAct (Huang et al., 2025) and DeepThinkVLA (Yin et al., 2025) adopt a "think-before-act" perspective, aligning reasoning and execution through reinforcement learning or causal alignment to improve few-shot adaptation, long-horizon planning, and autonomous error correction.

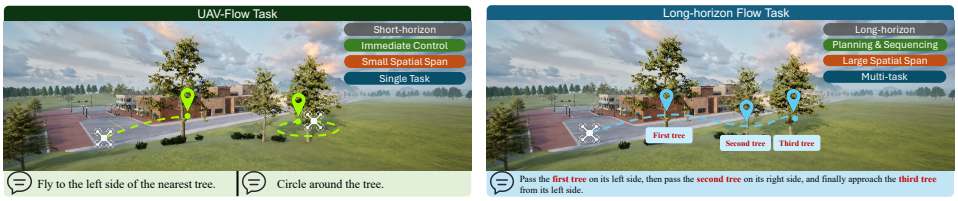
Nevertheless, long-duration UAV navigation and substantial viewpoint changes challenge VLA task understanding, temporal context modeling, and real-time control. We address these challenges with a fast–slow VLA architecture incorporating streaming video reasoning for long-horizon planning and contextual understanding.

3 FLIGHT Benchmark and Dataset

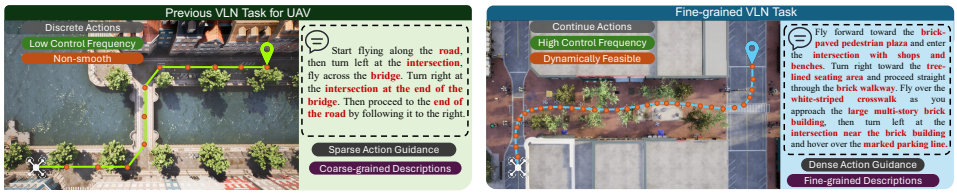
3.1 FLIGHT Task Definition

FLIGHT comprises two types of UAV tasks: Long-horizon Flow and Fine-grained VLN. Long-horizon Flow inherits the Flying-on-a-Word task setting proposed in (Wang et al., 2025a) and further extends it from atomic actions to multi-stage compositional action tasks. Fine-grained VLN, built upon the conventional VLN paradigm, introduces denser textual instruction descriptions and requires continuous fine-grained control.

At time step t, the task input consists of the language instruction I, the recent visual observation window $V_{[t-W,t]}$ of length W, and the UAV sensor state S_t. The model is required to generate a fine-grained action sequence (a_{t+1}) to execute the task specified by the language instruction. Moreover, unlike typical short-horizon VLA tasks, long-sequence tasks typically require historical information (M_t) alongside current observations to support decision-making.



(a) Long-horizon Flow.



(b) Fine-grained VLN.

Figure 2: Comparison with prior UAV benchmarks. (a) FLIGHT extends UAV-Flow to multi-stage trajectories. (b) Fine-grained VLN provides more detailed instructions and finer action granularity.

$$\pi : (S_t, V_{[t-W,t]}, I, M_t) \mapsto a_{t+1}, M_{t+1}. \tag{1}$$

Long-horizon Flow extends UAV-Flow's atomic maneuvers into multi-stage compositions (Fig. 2a). For example, instructions like "traverse the first tree from the left, then traverse the second tree from the right," and "traverse the tree ahead and then land," define long-horizon motion paradigms that integrate multiple atomic actions. Appendix D (Table 5) provides additional instruction examples.

The **Fine-grained** VLN task requires the UAV to perform navigation according to natural language instructions. In terms of instruction design, we follow the textual paradigm of natural language navigation instructions used in conventional VLN benchmarks, while preserving rich and diverse semantic expressions. As illustrated in Fig. 2b, compared with existing VLN datasets, our dataset enhances the fine-grained nature of the task from the following two aspects.

First, FLIGHT adopts **finer-grained subtask decomposition**, increasing decision density per unit path length through local referential constraints. For example, in addition to instructions such as "go straight through the road," we further incorporate local referential constraints such as "pass by the flower bed" and "move along the right side of the fountain". This design increases the density of spatial grounding in the instructions, requiring the model to more frequently reason about the relationship between the environment and the navigation goal.

Second, **a continuous and fine-grained action space** is employed by FLIGHT. Unlike conventional VLN datasets, which primarily formulate navigation as discrete action classification, we record fine-grained action sequences from continuous UAV trajectories as the action space, thereby preserving the continuous control process in real-world flight.

See Appendix D for quantitative comparisons with Aerial VLN (Liu et al., 2023) in subtask decomposition and action supervision per unit trajectory length (Table 3), while Table 4 shows the instruction examples contrasting prior VLN tasks.

3.2 Data Collection and Annotation

We construct two datasets for Fine-grained VLN and Long-horizon Flow tasks, containing 6,689 and 4,098 trajectories, respectively. The statistics of the dataset are shown in Appendix A.

3.2.1 UAV Flight Trajectory and Video Data Collection

We collect data across diverse, semantically rich simulation environments (urban, industrial, and rural) built via Unrealzoo (Zhong et al., 2024). The dataset covers diverse trajectories mainly collected by human pilots, with a small portion of simple-route scenarios supplemented by path-planning models. Unlike UAV VLN benchmarks such as Aerial VLN (Liu et al., 2023) and OpenFly (Gao

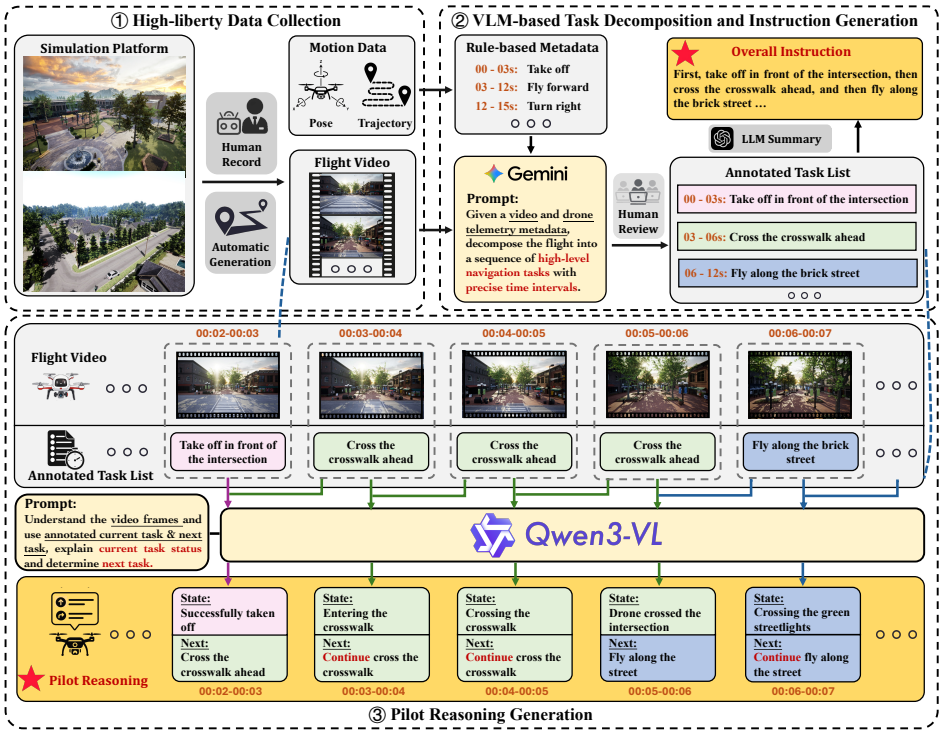


Figure 3: The data pipeline of FLIGHT. (1) Simulated trajectory data collection followed by a rule-based extraction to generate metadata tags; (2) VLM-driven generation of task semantic labels for each video segment assisted by metadata, which are subsequently fused into task instruction; (3) Generation of Pilot Reasoning texts encompassing state analysis and look-ahead reasoning, achieved by combining video segments with the fused current and next-step task semantic labels.

et al., 2025), which use discrete motion instructions, our dataset records full 6-DoF trajectories comprising position (latitude, longitude, altitude) and orientation (roll, pitch, yaw), alongside temporally aligned FPV videos. We further compute the relative displacement between consecutive trajectory points to obtain VLA-style action sequences.

3.2.2 UAV Task-Semantic Annotation

Segmentation Annotation and Instruction Generation. As shown in Fig. 3, a VLM annotates each video segment with motion behavior and environmental context, such as "taking off from a plaza, flying along a street past traffic signs, or turning left near a roadside kiosk". This segment-level annotation strategy can more accurately capture fine-grained semantic changes throughout the flight process and is therefore better suited for fine-grained navigation and reasoning tasks. Vision-only annotation models often struggle to infer UAV attitude and motion trends from FPV videos. We therefore use UAV motion data as auxiliary prompts: raw signals such as attitude angles are analyzed to locate key action intervals, including takeoff, landing, and turning, and converted into metadata prompts alongside video clips for the VLM. For the episode-level instructions, we use an LLM to merge segment-level subtask descriptions into a coherent, natural high-level instruction, encouraging semantic understanding rather than keyword matching or template-based reasoning.

Construction of Pilot Reasoning Texts. This module further generates the corresponding **Pilot Reasoning** texts based on the segment-level semantic annotations obtained in the previous stage. The core objective of Pilot Reasoning is to simulate task planning and decision-making during flight. Therefore, the generated text is expected not only to describe the UAV's current flight state, but also to perform **prospective reasoning** about the actions that may be executed in the next time interval. To achieve this goal, an annotation VLM combines video content with semantic descriptions of the current and subsequent segments, using the latter as oracle planning priors (Fig. 3). It generates Pilot Reasoning text that describes the current flight state and anticipates the next action, yielding a temporally coherent, planning-oriented representation of flight decision-making.

4 FLIGHT VLA PIPELINE

FLIGHT VLA is an asynchronous VLA architecture for complex, long-horizon UAV flight, designed to preserve long-term semantic context while meeting the demands of high-frequency continuous control. We therefore decompose the policy into two modules. A low-frequency **Streaming Pilot VLM** performs online video understanding, task-stage recognition, and high-level planning, while a high-frequency **Diffusion Action Model** generates continuous control trajectories from real-time visual observations and UAV states. The two modules communicate through asynchronous semantic feature transfer, allowing semantic reasoning to run at a lower rate than the control loop.

We begin by formulating the continuous control objective. Given a language instruction I, FPV frames $\{V_t\}$, and IMU and pose states $\{S_t\}$, the agent predicts future actions at each control step t:

$$A_t = \pi_\theta(I, V_{\le t}, S_{\le t}, M_{\le t}) = \{a_t, a_{t+1}, \ldots, a_{t+H-1}\}, \tag{2}$$

where H is the prediction horizon, each action a_t is a relative motion command in the local UAV coordinate frame, and $\{M_t\}$ denotes the long-term contextual memory formed by historical reasoning texts. Directly using a VLM to instantiate π_θ at every control step is impractical because autoregressive reasoning introduces substantial latency. We instead factorize Eq. 2 into a slow semantic reasoning process and a fast action generation process:

$$(z_\tau, M_\tau) = g_\phi(I, V_{\tau-W_i:\tau}, S_{\tau-W_i:\tau}, M_{\tau-W_M:\tau}), \tag{3}$$

$$A_t = f_\theta\big(V_t, S_t, z_{\tau(t)}\big), \tag{4}$$

Here, g_ϕ is the Streaming Pilot VLM and f_θ is the Action Model. At reasoning time (τ), the VLM consumes a recent video-state window of length (W_i) and a reasoning-memory window of length (W_M), and outputs a Pilot Reasoning text (M_τ) along with a semantic latent feature (z_τ), which encodes the hidden states of both the input and output reasoning texts, as illustrated in Fig. 4. The function $\tau(t)$ indexes the most recent VLM output available to the controller at time t. This definition explicitly permits $\tau(t) < t$, thereby modeling VLM inference latency while preserving high-frequency action generation.

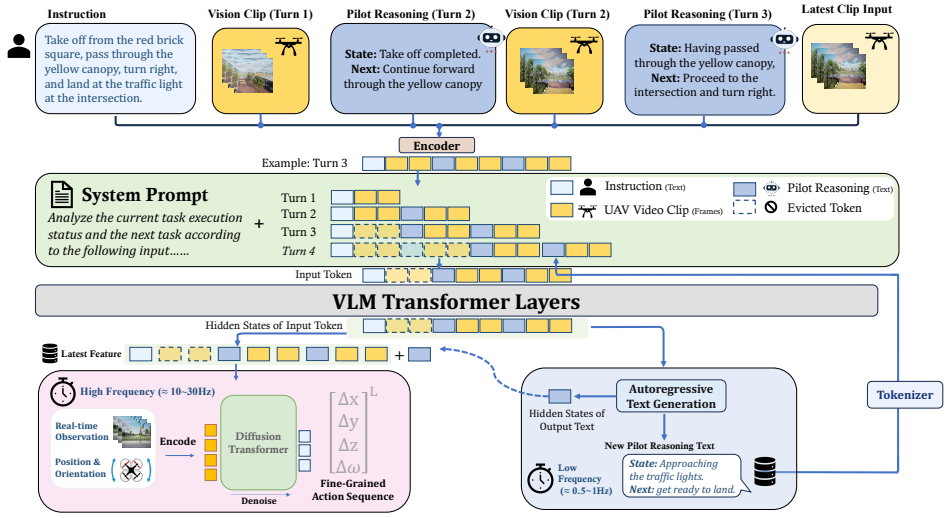


Figure 4: Overview of the FLIGHT VLA framework. A low-frequency Pilot VLM generates semantic features while a high-frequency action model combines these features with real-time visuals and UAV states to predict fine-grained trajectories. This dual-rate architecture decouples semantic planning from motion control, enabling long-horizon understanding and real-time control.

4.1 STREAMING PILOT VLM FOR ONLINE TASK REASONING

Streaming Pilot VLM is the slow reasoning module of the system inspired by Streaming VLM (Xu et al., 2025). It maintains task context over a continuous UAV FPV stream and produces "Pilot Reasoning" text about the current state and the near-future plan. As shown in Eq. 3, each VLM

update takes as input the task instruction I, the latest video-state window $(V_{\tau-W_i:\tau}, S_{\tau-W_i:\tau})$, and the historical reasoning memory $M_{\tau-W_M:\tau}$. This memory stores recent Pilot Reasoning texts, which compress long-horizon multimodal history into a compact textual representation. Consequently, W_M can be set relatively large, allowing the VLM to access a broader range of historical task memory without incurring the cost of retaining all past visual tokens.

The generated text (M_τ) summarizes the current flight state and predicts the next action. It is then interleaved with the corresponding visual frame (V) and current state (S), and written back to memory, as the format illustrated in the upper part of Fig. 4. We also extract the final-layer hidden states associated with the model input and the most recent generated reasoning tokens, yielding the semantic feature z_τ. This feature carries information about the instruction, historical context, current visual state, and next-step plan, and serves as the semantic condition for high-frequency action generation.

For training, we adopt an overlapped-chunk full-attention strategy. The long video stream is divided into consecutive temporally overlapping chunks. Within each chunk, historical Pilot Reasoning texts are concatenated with interleaved image-text sequences to construct an independent training instance, and full attention is applied over all tokens in the chunk. This preserves consistency between the training token layout and the inference-time streaming schedule. We supervise the VLM with Pilot Reasoning texts from the FLIGHT dataset, encouraging it to learn task-execution analysis and next-step action planning.

4.2 Diffusion Action Model for Fine-Grained Continuous Control

The Action Model serves as the fast execution module, generating continuous flight trajectories by combining the latest available VLM semantic features $z_{\tau(t)}$ with visual and state inputs refreshed at each high-frequency time step t. Here, $\tau(t) < t$ denotes the sampling timestamp associated with the latest available VLM features. An independent pretrained Vision Transformer encodes the current image as $\mathbf{f}_t = E_{\text{vit}}(V_t)$, capturing local visual changes between consecutive VLM reasoning steps. To mitigate global-coordinate drift and align the current state with historical semantic features, we encode the current UAV pose relative to the UAV coordinate frame at $\tau(t)$ as q_t. The resulting conditioning input is $c_t = \left[\mathbf{f}_t;\, q_t;\, z_{\tau(t)}\right]$.

For action generation, we adopt a trajectory generator based on a Diffusion Transformer. Given the condition c_t, the model outputs an action sequence over the future horizon of H steps $A_t = f_\theta(c_t)$. Conditioned on the latest available semantic features, the action model continues generating action sequences while the VLM performs its next reasoning update, thereby avoiding control gaps caused by the inference latency of the model.

4.3 Joint Training with Latency-Aware Alignment

The training loss consists of both the VLM loss and the action flow matching loss:

$$\mathcal{L} = \lambda_{\text{vlm}}\mathcal{L}_{\text{vlm}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}}, \tag{5}$$

where λ_{vlm} and λ_{act} control the weights of the two supervision signals, respectively. Due to their different convergence rates, we first jointly train the VLM and Action Model for semantic adaptation, then freeze the VLM to prevent overfitting.

In real-world deployment, the inference latency of the VLM is not fixed. If training always assumes that the Action Model receives VLM features from the current time step, the model may suffer from distribution shift when exposed to stale semantic features during inference. To address this issue, we explicitly simulate asynchronous latency during training data construction. For each action training segment, we sample VLM features from several historical temporal offsets. As illustrated in Fig. 5, VLM inputs are constructed using video windows with 1-second and 3-second delays for selected action-model training chunks, and 0-second and 2-second delays for the remaining chunks, excluding the initial and final chunks. The relative pose encoding is also computed using the associated VLM reasoning time as the local reference frame.

This training data organization strategy enables the Action Model to adaptively handle VLM semantic features originating from different historical time steps, and to learn to compensate for their temporal lag using current visual feedback and relative state information. Consequently, during

deployment, even when the VLM reasoning interval fluctuates, the Action Model can still stably generate control sequences that remain consistent with the current flight state.

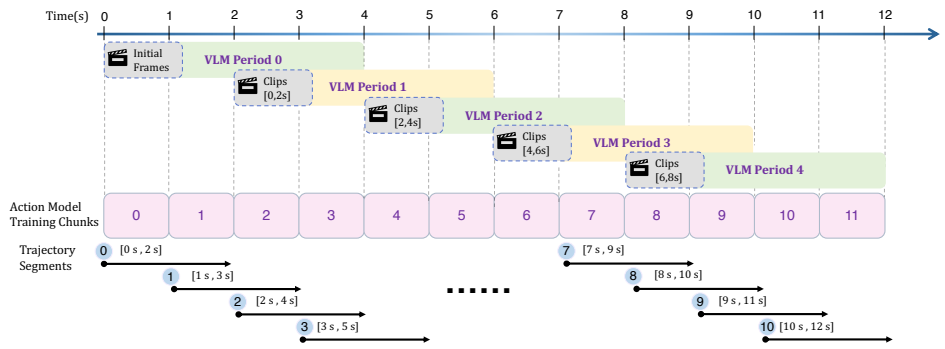


Figure 5: Asynchronous Training Strategy for FLIGHT VLA. We simulate asynchronous inference during training by conditioning high-frequency action chunks on temporally offset features from VLM, which processes 2-second FPV video. At $t = 7$ s, for example, the model predicts the 7–9 s trajectory using Period 3 or Period 2 features, corresponding to 1- or 3-second delays.

5 Experiments

5.1 Experimental Setup

Evaluation Metrics. We use specific metrics for Long-Horizon Flow and Fine-Grained VLN task. *Long-Horizon Flow.* Success requires reaching the destination and correctly completing every instructed subtask. Success rate (**SR**) measures this overall success, while Stage Success Rate (**SSR**) measures the proportion of correctly completed subtasks within a trajectory. Normalized Dynamic Time Warping (**NDTW**) measures the similarity between predicted and reference trajectories.

Fine-Grained VLN. Navigation Error (**NE**) measures the mean final distance to the target, while Success Rate (**SR**) and Oracle Success Rate (**OSR**) measure the fractions of episodes reaching within 15 m of the target at termination and at any point, respectively. Since standard SR considers only final arrival without assessing compliance with sequential instruction constraints, we introduce Instruction Adherence Success Rate (**IASR**), which counts a trial as successful only if all instructed subtasks (e.g., takeoff, obstacle traversal, and landing) are completed in order.

Comparison Baselines. *VLA Models.* (1) **OpenVLA** (Kim et al., 2024) is designed for generalizable manipulation. Following OpenVLA-UAV (Wang et al., 2025a), we encode UAV states and instructions as a unified text prompt and discretize UAV motion into 256 output tokens. (2) **Memory VLA** (Shi et al., 2025) stores historical information during inference. We adapt it to predict 4D relative pose commands from the current FPV image and instruction.

VLN Models. (1) **CMA+LAG** (Liu et al., 2023) extends Cross-Modal Attention (CMA) (Anderson et al., 2018), which fuses language instructions with RGB-D observations for navigation, using look-ahead targets from ground-truth trajectories for supervision. (2) **NaVid** (Zhang et al., 2024), the first video-based embodied large model for VLN, maps navigation videos and language instructions to robot actions through generated reasoning text.

5.2 Results

Long-horizon Flow Task. As shown in Table 1a, the proposed method outperforms the baseline models across all evaluation metrics. OpenVLA does not explicitly account for the execution requirements of long-horizon tasks in its architectural design and lacks an effective contextual memory mechanism, limiting its ability to continuously track the execution state of long-sequence tasks. Although Memory VLA introduces a memory module to alleviate this issue, it struggles to retain sufficient historical context in semantically complex tasks, leading to substantial performance drops with two or more subtasks. In contrast, the proposed **FLIGHT VLA** architecture fully leverages the advantages of streaming video understanding models and supports an ultra-long context window. It successfully completes test instances containing up to 7 subtasks.

Method	Control	History Storage	SR↑ (%)	SSR↑ (%)	NDTW↑ (%)
OpenVLA-UAV	Discrete	×	10.50	30.87	18.02
Memory VLA	Continuous	✓	18.50	38.69	16.58
FLIGHT VLA	**Continuous**	✓	**59.00**	**78.97**	**45.78**

(a) Results on Long-horizon Flow.

Method	Control	SR↑ (%)	OSR↑ (%)	NE↓ (m)	IASR↑ (%)
CMA+LAG	Discrete	6.5	22.0	52.56	4.5
NaVid	Discrete	13.0	25.5	46.69	5.0
FLIGHT VLA	**Continuous**	**13.5**	**37.0**	**45.52**	**11.0**

(b) Results on Fine-grained VLN task.

Table 1: Comparison with baseline methods on the two FLIGHT tasks.

Fine-grained VLN Task. We evaluate models on the more challenging **Fine-grained VLN** task, which features substantially longer trajectories (Table 1b). CMA+LAG struggles to interpret complex instructions and multimodal inputs. NaVid relies on historical images for autoregressive action generation, facing a context-length bottleneck and struggling with precise landing. Its single-step discrete control also forces the UAV to pause between actions for inference and transmission. Our **FLIGHT VLA** overcomes this context bottleneck, decomposes multi-stage tasks, and follows fine-grained constraints for accurate takeoff, navigation, and landing. Its slow-fast architecture supports efficient real-time prediction through continuous trajectory prediction. Fig. 6 presents successful navigation examples for Long-horizon Flow and Fine-grained VLN task.

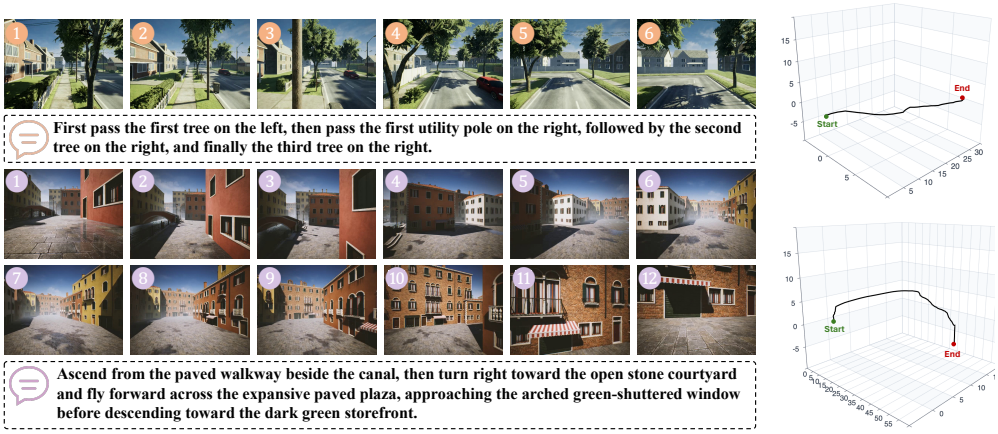


Figure 6: Visualization results of our method in Long-Horizon Flow and Fine-Grained VLN task.

6 Conclusion

In this work, we introduced FLIGHT, a benchmark for long-horizon, fine-grained, instruction-guided UAV navigation and reasoning tasks. FLIGHT addresses the limitations of prior UAV benchmarks by integrating fine-grained control with long-horizon language navigation, encompassing both long-horizon flow and fine-grained VLN tasks. To support robust model training and evaluation, we also introduced Pilot Reasoning, a textual reasoning signal providing explicit supervision for task understanding and planning. We further proposed the FLIGHT VLA, which combines a streaming VLM backbone with a diffusion-based action model, organized under an asynchronous system. This design enables real-time, continuous control while preserving long-range semantic understanding and reasoning capabilities. Experimental results across both Long-Horizon Flow and Fine-Grained VLN tasks demonstrate that our approach outperforms baseline VLA and VLN models, validating the efficacy of streaming reasoning and the integration of explicit pilot supervision.

AI USE STATEMENT

Large Language Models (LLMs) were used to aid in the writing and polishing of the manuscript. Specifically, we used an LLM to assist in refining the language, improving readability, and ensuring clarity in various sections of the paper. The model helped with tasks such as sentence rephrasing, grammar checking, and enhancing the overall flow of the text. It is important to note that the LLM was not involved in the ideation, research methodology, or experimental design. All research concepts, ideas, and analyses were developed and conducted by the authors. The contributions of the LLM were solely focused on improving the linguistic quality of the paper, with no involvement in the scientific content or data analysis. The authors take full responsibility for the content of the manuscript, including any text generated or polished by the LLM. We have ensured that the LLM-generated text adheres to ethical guidelines and does not contribute to plagiarism or scientific misconduct.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton {Van Den Hengel}. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pp. 3674–3683, 2018. doi: 10.1109/CVPR.2018.00387.

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. *CoRR*, abs/2502.18041, 2025.

Yue Hu, Avery Xi, Qixin Xiao, Seth Isaacson, Henry X. Liu, Ram Vasudevan, and Maani Ghaffari. LongNav-R1: Horizon-adaptive multi-turn RL for long-horizon VLA navigation. *arXiv preprint arXiv:2602.12351*, 2026.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

Xu Liu, Yu Liu, Hanshuo Qiu, Yang Qirong, and Zhouhui Lian. IndoorUAV: Benchmarking vision-language UAV navigation in continuous indoor environments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(28):23864–23872, Mar. 2026. doi: 10.1609/aaai.v40i28.39562.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. MemoryVLA: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2508.19236*, 2025.

Marcel Torne, Karl Pertsch, Homer Walke, Kyle Vedder, Suraj Nair, Brian Ichter, Allen Z. Ren, Haohuan Wang, Jiaming Tang, Kyle Stachowicz, Karan Dhabalia, Michael Equi, Quan Vuong, Jost Tobias Springenberg, Sergey Levine, Chelsea Finn, and Danny Driess. Mem: Multi-scale embodied memory for vision language action models. *arXiv preprint arXiv:2603.03596*, 2026.

Shaoan Wang, Yuanfei Luo, Xingyu Chen, Aocheng Luo, Dongyue Li, Chang Liu, Sheng Chen, Yangang Zhang, and Junzhi Yu. VLingNav: Embodied navigation with adaptive reasoning and visual-assisted linguistic memory. *arXiv pre-print*, 2026.

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 *Advances in Neural Information Processing Systems*, pp. 1234–1245, 2025a.

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 *International Conference on Learning Representations*, volume 2025, pp. 7292–7310, 2025b.

Ziqin Wang, Jinyu Chen, Xiangyi Zheng, Qinan Liao, Linjiang Huang, and Si Liu. 'hi airstar, guide me to the badminton court.'. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 13477–13479, New York, NY, USA, 2025c. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3754491.

Junjie Wen, Minjie Zhu, Jiaming Liu, Zhiyuan Liu, Yicun Yang, Linfeng Zhang, Shanghang Zhang, Yichen Zhu, and Yi Xu. DVLA: Diffusion vision-language-action model with multimodal chain-of-thought. *arXiv preprint arXiv:2509.25681*, 2025.

Ruyi Xu, Guangxuan Xiao, Yukang Chen, Liuning He, Kelly Peng, Yao Lu, and Song Han. StreamingVLM: Real-time understanding for infinite video streams. *arXiv preprint arXiv:2510.09608*, 2025.

Cheng Yin, Yankai Lin, Wang Xu, Sikyuen Tam, Xiangrui Zeng, Zhiyuan Liu, and Zhouping Yin. DeepThinkVLA: Enhancing reasoning capability of vision-language-action models. *arXiv preprint arXiv:2511.15669*, 2025.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. *Robotics: Science and Systems*, 2024.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025.

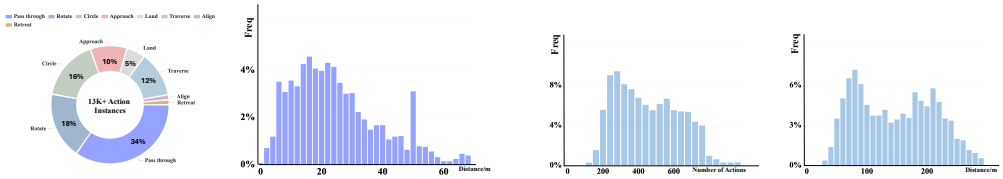
Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. *arXiv preprint arXiv:2503.22020*, 2025.

Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. Unrealzoo: Enriching photo-realistic virtual worlds for embodied AI. *arXiv preprint arXiv:2412.20977*, 2024.

Linqing Zhong, Yi Liu, Yifei Wei, Ziyu Xiong, Maoqing Yao, Si Liu, and Guanghui Ren. ACoT-VLA: Action chain-of-thought for vision-language-action models. *arXiv preprint arXiv:2601.11404*, 2026.

Zhongyi Zhou, Yichen Zhu, Junjie Wen, Chaomin Shen, and Yi Xu. Vision-language-action model with open-world embodied reasoning from pretrained knowledge. *arXiv preprint arXiv:2505.21906*, 2025.

A Data Analysis



(a) Statistic of Long-Horizon Flow Task Dataset (b) Statistic of Fine-Grained VLN Task Dataset

Figure 7: Dataset statistics for FLIGHT. (a) Long-Horizon Flow statistics, including sub-task categories and trajectory lengths. (b) Fine-Grained VLN trajectory profiles, including the number of actions per trajectory and total trajectory distance.

The Long-horizon Flow dataset with 4,098 trajectories inherits eight action categories from UAV-Flow (Wang et al., 2025a), with the category distribution shown in the left panel of Fig. 7(a). It contains 13,815 action instances in total, with each trajectory averaging 3.37 action instances. The average destination distance is 17.8 m, and the average trajectory length is 32.8 m; their distributions are shown in the right panel of Fig. 7(a). Some trajectories have near-zero displacement because the dataset includes rotation-centric actions, and algorithmically generated orbiting trajectories produce a few high-frequency displacement values. Overall, however, the unique trajectory lengths remain broadly distributed.

The Fine-grained VLN dataset with 6,689 trajectories features linguistically diverse instructions with dense spatial and action-level grounding. We sample continuous actions at 10 Hz, resulting in an average of 475 actions per trajectory and an average trajectory length of 154.5 m. Fig. 7 (b) reports the detailed distributions of action counts and trajectory lengths.

B Navigation-oriented video QA benchmark

In addition, to evaluate the video reasoning capability of vision-language models (VLMs) on UAV FPV video data, FLIGHT introduces the **Pilot Reasoning task**. Given a UAV video (V) and a corresponding textual instruction (I), the VLM is required to infer the UAV's current state and its subsequent task plan. The model then generates this analysis in natural language, denoted as $\mathcal{T}_{\text{reason}}$. Formally, the Pilot Reasoning task is formulated as a conditional text generation process:

$$\mathcal{T}_{\text{reason}} = \pi_{\text{vlm}}(V, I), \tag{6}$$

where π_{vlm} represents the vision-language model that maps the multimodal inputs to the textual analysis space.

This task effectively evaluates the instruction understanding and planning capabilities of VLMs in UAV navigation scenarios, thereby providing a benchmark for assessing foundation models for navigation agents. Performance is measured by GPT-5–evaluated semantic consistency between predicted and ground-truth reasoning.

As shown in Table 2, streaming-reasoning training substantially improves navigation understanding and reasoning, providing semantic guidance for instruction-aligned trajectory generation.

Table 2: Reasoning accuracy of representative VLMs on VLN video-text sequences.

Model	**Parameters**	**State Acc.** (%)	**Action Acc.** (%)
Gemini-3-flash	–	25.8	39.1
Qwen 3.6 plus	–	2.9	10.6
Qwen 2.5 VL	3B	5.0	6.9
Streaming Pilot Reasoning VLM	3B	**64.2**	**68.1**

C PROMPT ENGINEERING IN DATA ANNOTATION AND PILOT REASONING VLM DESIGN

Task Decomposition Prompt

You are an expert drone flight instructor. Your task is to generate a precise, step-by-step flight log based on the provided sequence of timestamped frames extracted from the FPV flight and the provided Flight Telemetry Data.
1. APPROVED ACTION VOCABULARY (STRICT)
You must ONLY use action verbs from the following list to describe the drone's movements.
* **Basic Movement:** Fly forward, Move forward, Head towards, Advance, Proceed straight, Go straight, Keep straight, Maintain heading, Continue, Keep flying, Resume, Ascend, Rise, Climb, Gain altitude, Take off, Descend, Lower, Drop down, Lose altitude, Land.
* **Turning (Use ONLY for significant directional changes):** Turn right, Turn left, Make a right, Make a left, Yaw right, Yaw left, Bear right, Bear left. * **Complex Maneuvers (Telemetry Only):** Perform a U-turn, Circle once, Circle twice, Orbit, Ascend, Descend.
* **Orientation:** Follow, Trace, Fly parallel to, Approach, Head for, Close in on, Align with, Face, Orient towards, Line up with.
* **Interaction:** Pass, Pass by, Go past, Fly past, Leave behind, Pass between, Go through the gap of, Thread through, Continue past, Overtake, Go beyond, Fly over, Pass over, Soar over, Clear, Fly through, Traverse, Enter, Fly under, Pass beneath, Go under.

2. MANDATORY TELEMETRY INJECTION (HIGHEST PRIORITY)
The following complex maneuvers were physically recorded by sensors. The logs now contain two time ranges (in seconds): the **ACTION TIME** (where the turn or vertical move occurred) and the **CONTEXT** time (which includes the action time plus {context seconds} seconds before and after, for visual reference).
CRITICAL INSTRUCTION: You MUST visually locate these moments and insert the action **EXACTLY** as written in the logs. **You MUST use the wider CONTEXT time range to identify the surrounding landmarks and add a contextual description of the location to the action.**
turn telemetry str
CRITICAL RULES FOR TELEMETRY:
1. **NO SIMPLIFICATION/MODIFICATION:** Use the action phrase **EXACTLY** as written (e.g., "Circle once Right", "Ascend while Turning Left").
2. **MUST ADD LOCATION:** The final step output must combine the action and a precise description of the surroundings (e.g., "perform a U-turn Left at the edge of the blue-roofed market.").
3. **TIMESTAMP REQUIREMENT:** You MUST append the specific '[ACTION TIME]' from the log to the end of the step in the format '[start s - end s]'.
4. **LOCATION EXAMPLE (REQUIRED FORMAT):** * Log: '- [ACTION TIME: 4.00s - 6.00s] [CONTEXT: 2.00s - 8.00s] MANDATORY ACTION: perform a U-turn Left.' * Output: 'perform a U-turn Left near the tiered stone fountain. [4.00s-6.00s]'

3. MOVEMENT SENSITIVITY & FILTERING
* **IGNORE MINOR ADJUSTMENTS:** Do **NOT** report small yaw corrections, camera jitters, or slight steering (< 30 degrees). Treat these as **"Fly forward"**.
* **MAJOR VISUAL TURNS:** Only report a visual turn NOT in the logs if it is a **major, obvious turn**.

4. OBJECT DIVERSITY & SALIENCY
- **Prioritize Prominent Landmarks:** Focus on the most visually distinct objects (e.g., large buildings, unique statues, colorful signs).
- **Avoid Repetition:** Shift focus to **different** nearby objects in consecutive steps.
- **Dynamic Attention:** Point out new features in each step rather than mentioning the same object repeatedly.

5. STRICT SENTENCE STRUCTURE & TIMING
Except for the mandatory telemetry actions, every step must strictly follow this pattern:
[Action Verb] + [Preposition] + [Visual Description] + [Object] [Start_Time s - End_Time s]
* **TIMING RULE (OVERLAP ALLOWED):** Actions often happen simultaneously. **It is permitted for timestamps to overlap.** For example, Step 2 may start before Step 1 ends if the actions are concurrent. * **Correct Examples:**
* "Fly over the white fountain. [1.2s-3.5s]"
* "Pass between the blue market stalls. [3.0s-6.0s]" (Note: Starts before previous step ends)

* "Turn right towards the brick building. [6.0s-8.2s]" (Only if major turn)

6. NEGATIVE CONSTRAINTS (FORBIDDEN)
- **NO TEXT RECOGNITION:** Do NOT output specific text characters (e.g., "UFoods", "Stop"). Describe visually (e.g., "the red brick building").
- **NO Bold Text**: Do not use asterisks (**).
- **NO Meta-Comments**: Start **directly** with "1.".
- **NO Degree Numbers**: Use qualitative terms for angles.

7. OUTPUT FORMAT EXAMPLE
1. Ascend from the paved area. [0.00s-3.50s]
2. Fly forward over the circular fountain. [2.50s-5.10s]
3. Proceed straight through the open plaza. [4.80s-8.00s]
4. Pass between the covered stalls. [4.60s-10.20s]
5. Approach the large tree. [10.00s-12.50s]
6. perform a U-turn Left near the tiered stone fountain. [12.50s-15.00s]
7. Circle once Right above the market area. [12.00s-19.00s]
8. Fly along the pedestrian street. [18.50s-22.10s] ...
Generate the Numbered List now:

Instruction Annotation Prompt

You are an expert drone flight narrator.
INPUT DATA
1. **Visuals**: The sequence of timestamped frames provided.
2. **Step-by-Step Flight Log**: {step_list_text}
TASK
Convert the Step-by-Step Flight Log into a **Single, Coherent Global Mission Instruction** (one continuous narrative).
STRICT REQUIREMENTS
1. **Preserve All Details**: You MUST include **EVERY** object and **EVERY** descriptive adjective mentioned in the flight log. Do not generalize or omit details (e.g., "blue and white striped awnings" must NOT be shortened to "awnings").
2. **Handle Simultaneity (Critical)**: Watch the video carefully to identify actions from the list that happen at the same time.
- *Example*: If the drone flies under tree branches and passes a lamp post at the same moment, connect them using **"while"**, **"as"**, or **"simultaneously"** (e.g., "Pass under the green tree branches **while** passing by the black lamp post...").
3. **Flow**: Connect all steps into a fluent, natural narrative using varied transition words.
4. **Constraint**: Do not add new information not present in the list, but do not lose any information from the list.
5. **Timestamp Removal**: Do NOT include the timestamps [0.0s-0.0s] in this narrative summary. Keep it as pure text.
OUTPUT FORMAT
Output ONLY the global instruction text.

Pilot Reasoning Annotation Prompt

You are a professional drone flight assistant. Analyze the following inputs: Previous Modal inference results(History),three sequentially sampled frames from the video, Overall Mission List, Current Mission, and Human Pilot's Decision for next mission. Assume no knowledge of the human pilot's reasoning or declared next decision. Based on the visual context from the video, provide a single, concise paragraph analyzing why the pilot chooses the next mission. Your analysis must ONLY address: Completion Status: Has the current mission been completed based on the visual evidence?
ACTION PLAN:
- If YES: How to execute the next step based on the pilot's decision.
- If NO: How to proceed to finish the current mission.
Constraint: Direct output only. Do not include introductory or concluding remarks.
OUTPUT RULES:
- Keep your output strictly under 80 English words. Output English only.
- Do not include the word "pilot's" in the output.
- Output **only** the reasoning process. Ground all reasoning strictly in observable visual context;

do not assume intent or external knowledge beyond what is shown.
- Concise and clear.
- No commentary, no additional formatting
- NO precise timestamps like 00:06.53 or 7.33s
- DO NOT include task numbers in the output.

Pilot Reasoning VLM Prompt

You are a professional drone flight assistant. Analyze the following inputs: Previous Modal inference results(History), the video frames, overall mission instruction. Based on the visual context from the video, provide a single, concise paragraph analyzing which action or mission should be executed next. **Constraints:** 1. **Format:**
‘Status: [State]. Next: [Action].
‘ 2. **Length:** Total output must be **under 30 words**. 3.Direct output only. Do not include introductory or concluding remarks. The overall mission instruction is: {Instruction}. Following is the modal inference history: /n {History with frames, states and previous Pilot Reasoning Text}

Critic GPT Prompt

State score prompt: "You are evaluating whether a predicted instruction sentence matches the ground-truth " "instruction sentence semantically." "Focus on whether the key meaning is consistent, including:" "- important objects, locations, direction, actions and mission progress" "Treat paraphrases or minor wording differences as equivalent." "Treat contradictions, missing crucial actions, wrong objects/locations, or clearly " "different mission state as not equivalent." "Return JSON only with this exact schema:" {"equivalent": true/false, "reason": "short explanation", "confidence": 0.0}f"Ground-truth Status:{gt text}" f"Predicted Status:pred text"

Action score prompt: "You are evaluating whether a predicted instruction sentence matches the ground-truth " "instruction sentence semantically." "Focus on whether the key meaning is consistent, including:" "- important objects, locations, direction, actions and mission progress" "Treat paraphrases or minor wording differences as equivalent." "Treat contradictions, missing crucial actions, wrong objects/locations, or clearly " "different mission state as not equivalent." "Return JSON only with this exact schema:" ’{"equivalent": true/false, "reason": "short explanation", "confidence": 0.0}’ f"Ground-truth Status:{gt text}" f"Predicted Status:{pred text}")

D EXAMPLES INSTRUCTIONS FOR FLIGHT TASKS

This section provides instruction examples for the tasks defined in FLIGHT and a quantitative comparison with Aerial VLN.

Table 3: Quantitative comparison between Aerial VLN and FLIGHT-fine-grained VLN. Despite having shorter average trajectories, FLIGHT-fine-grained VLN incorporates denser language grounding and finer action supervision.

Metric	Aerial VLN (Liu et al., 2023)	FLIGHT-FG VLN
Average instruction length (words)	83	71
Average trajectory length (m)	661	154
Verbs per 100 m	2.17	6.23
Nouns per 100 m	3.25	12.26
Adjectives per 100 m	0.96	6.55
Average action steps per 100 m*	30.86	295.45
Control formulation	Discrete action classification	Continuous action sequence

* In Aerial VLN, each action step denotes one discrete action choice. In FLIGHT-FG VLN, each action step denotes one element in the fine-grained continuous action sequence.

Table 4: Comparison between conventional VLN instructions and the fine-grained VLN instructions in FLIGHT.

Source	Instruction example	Task property
Conventional VLN	Start along the street, turn left at the intersection, fly across the bridge, and finally land.	Coarse route-level navigation with sparse grounding cues.
Fine-grained VLN	Start along the street, pass the blue umbrella, turn left at the intersection, fly across the bridge, and finally land beside the second tree on the left.	Fine-grained navigation with dense landmark grounding and precise terminal localization.

Table 5: Comparison between single-stage UAV-Flow instructions and the multi-stage compositional instructions in FLIGHT.

UAV-Flow	Long-horizon Flow
Circle around the person in a clockwise direction.	(1) First circle around the person in a clockwise direction, and then circle a second target in a different direction. (2) Circle around the person **three times** clockwise.
Approach the person from the left side	Pass the tree in front from the right, then approach the person from the left side, and finally land.
Pass through the tree from the right side	Pass through the first tree on the right, then pass the second tree on the left, and finally land beside the third tree.

E Training Details

This section provides the complete training configuration for the training of FLIGHT VLA model.

E.1 Model Architecture

Base Vision-Language Model. We initialize our model from Qwen2.5-VL-3B-Instruct and train it with low-rank adaptation (LoRA).

Action Prediction Head. We adopt a DiT-B (Diffusion Transformer) based action head. The action head configuration is summarized in Table 6.

Table 6: Action head configuration.

Parameter	Value
Action dimension	4
State dimension	4
Hidden dimension	1024
Future action window size	7
Repeated diffusion steps	2

E.2 Training Configuration

During training, videos are sampled at 2 FPS. The visual window lengths for Long-Horizon Flow and Fine-Grained VLN are set to 12 seconds and 8 seconds per training chunk, respectively. For text processing, we employ 512 text sink tokens alongside a sliding window of 512 tokens. To balance the optimization rates—since the VLM converges faster than the DiT action model—the joint co-training loss is defined as:

$$\mathcal{L} = 0.1\mathcal{L}_{\text{vlm}} + \mathcal{L}_{\text{act}}, \tag{7}$$

Table 7 and 8 summarize the detailed hyperparameters used for training FLIGHT VLA on the Long-Horizon Flow and Fine-Grained VLN datasets.

Table 7: Training hyperparameters of Long-horizon Flow.

Hyperparameter	**Value**
Learning rate	5×10^{-5}
Batch size per device	8
Gradient accumulation steps	2
Effective batch size	128 ($8 \times 2 \times 8$ GPUs)
GPU	$8\times$ A800 80G

Table 8: Training hyperparameters of Fine-Grained VLN.

Hyperparameter	**Value**
Learning rate	1×10^{-4}
Batch size per device	8
Gradient accumulation steps	2
Effective batch size	128 ($8 \times 2 \times 8$ GPUs)
GPU	$8\times$ A100 80G