IndoorUAV: Benchmarking Vision-Language UAV Navigation in Continuous Indoor Environments

Xu Liu[1*], **Yu Liu**[1*], **Hanshuo Qiu**[1], **Yang Qirong**[1], **Zhouhui Lian**[1, 2†]

[1]Wangxuan Institute of Computer Technology, Peking University, Beijing, China
[2]State Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China
2301213301@stu.pku.edu.cn, yuliu_@hust.edu.cn, qiuhanshuo@nudt.edu.cn,
2400013185@stu.pku.edu.cn, lianzhouhui@pku.edu.cn

Abstract

Vision-Language Navigation (VLN) enables agents to navigate in complex environments by following natural language instructions grounded in visual observations. Although most existing work has focused on ground-based robots or outdoor Unmanned Aerial Vehicles (UAVs), indoor UAV-based VLN remains underexplored, despite its relevance to real-world applications such as inspection, delivery, and search-and-rescue in confined spaces. To bridge this gap, we introduce **IndoorUAV**, a novel benchmark and method specifically tailored for VLN with indoor UAVs. We begin by curating over 1,000 diverse and structurally rich 3D indoor scenes from the Habitat simulator. Within these environments, we simulate realistic UAV flight dynamics to collect diverse 3D navigation trajectories manually, further enriched through data augmentation techniques. Furthermore, we design an automated annotation pipeline to generate natural language instructions of varying granularity for each trajectory. This process yields over 16,000 high-quality trajectories, comprising the **IndoorUAV-VLN** subset, which focuses on long-horizon VLN. To support short-horizon planning, we segment long trajectories into sub-trajectories by selecting semantically salient keyframes and regenerating concise instructions, forming the **IndoorUAV-VLA** subset. Finally, we introduce **IndoorUAV-Agent**, a novel navigation model designed for our benchmark, leveraging task decomposition and multimodal reasoning. We hope IndoorUAV serves as a valuable resource to advance research on vision-language embodied AI in the indoor aerial navigation domain.

Datasets — https://www.modelscope.cn/datasets/valyentine/Indoor_UAV

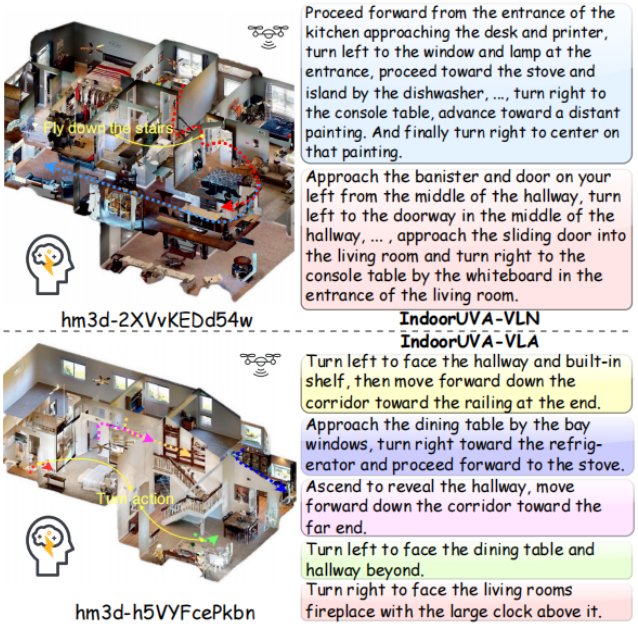

Figure 1: Illustration of IndoorUAV-VLN (upper) and IndoorUAV-VLA (lower) datasets. Long-horizon VLN tasks typically involve complex instructions and longer trajectory lengths, while VLA tasks focus on fine-grained maneuver execution, consisting of 1-3 executable actions.

Introduction

Vision-Language Navigation (VLN) is a foundational task in embodied AI that aims to enable autonomous agents to follow natural language instructions and navigate complex environments. During the past few years, a wide range of benchmarks and models have been developed for VLN tasks, primarily in two main settings: (1) indoor environments with ground-based agents such as wheeled robots or quadrupeds (Anderson et al. 2018; Krantz et al. 2020; Han et al. 2025; Zhang et al. 2025), and (2) outdoor environments with aerial platforms such as Unmanned Aerial Vehicles (UAVs) (Fan et al. 2023; Liu et al. 2023; Wang et al. 2025). These two lines of work have significantly advanced the capabilities of VLN. However, they overlook a critical and practically relevant domain: UAV-based vision-language navigation in complex, cluttered, and structured 3D continuous indoor environments.

This new setting introduces a unique set of challenges that diverge significantly from both ground-based indoor VLN and outdoor aerial VLN. In indoor environments, ground agents are typically confined to 2D navigation on a planar surface, limiting their ability to reason about or interact with

*These authors contributed equally.
†Corresponding author

the full 3D structure of the space. As a result, widely-used benchmarks such as R2R (Anderson et al. 2018), SOON (Zhu et al. 2021), and RxR (Ku et al. 2020) are inherently ill-suited for aerial navigation tasks that demand vertical reasoning, free-form 3D maneuvering, and fine-grained spatial understanding. Although recent work has started to explore UAV-based VLN in outdoor environments, these scenarios are typically open and sparse, lacking dense obstacles, narrow corridors, and high-precision maneuvering requirements that define indoor spaces, making it difficult to transfer such models to generalize such models to real-world indoor applications.

To bridge this gap, we introduce IndoorUAV benchmark specifically designed for aerial VLN in 3D indoor environments. Our goal is to facilitate research on navigation agents that must interpret natural language instructions and execute flight paths in fully three-dimensional, cluttered indoor spaces with realistic constraints. Unlike previous work, IndoorUAV emphasizes both high-level instruction grounding and low-level aerial control in a unified setting. It serves as a testbed for developing embodied agents capable of complex spatial reasoning, obstacle avoidance, and motion planning—capabilities that are critical for real-world applications such as indoor inspection, search and rescue, and autonomous delivery.

To construct IndoorUAV, we first curate over 1,000 high-quality 3D indoor environments from the Habitat simulator (Puig et al. 2023; Szot et al. 2021; Savva et al. 2019). Within these environments, we simulate plausible UAV flight dynamics to collect rich 3D trajectory data manually. To increase data diversity, we apply a variety of trajectory augmentation strategies, including trajectory reverse and sub-trajectory recombination. Then an automatic annotation pipeline is developed to generate natural language instructions at two levels of granularity. This process yields IndoorUAV-VLN, a long-horizon navigation dataset containing over 16,000 instruction-trajectory pairs. This dataset challenges agents to interpret complex, multi-step language instructions and navigate accordingly in 3D space.

Furthermore, to promote fine-grained spatial understanding and short-term planning navigation, we derive IndoorUAV-VLA from IndoorUAV-VLN by segmenting long trajectories into short sub-trajectories via keyframe selection. We then regenerate concise instructions corresponding each sub-trajectory. Each instruction in IndoorUAV-VLA maps to only 1 to 3 UAV actions, offering a complementary benchmark that emphasizes low-level flight control.

Finally, we propose IndoorUAV-Agent, leveraging a LLM to decompose long-horizon natural language instructions into shorter sub-instructions, which are then sequentially executed by a VLA model. This hierarchical design enables the agent to handle complex multi-step tasks while maintaining fine-grained control over UAV motion.

Experimental results highlight the significant challenges posed by IndoorUAV, revealing a substantial performance gap between current state-of-the-art models and the demands of real-world indoor UAV navigation. These findings underscore the need for further research on grounded language understanding, 3D spatial reasoning, and fine-grained motion control in aerial settings.

In summary, our contributions are as follows:

- We introduce IndoorUAV, the first large-scale benchmark specifically targeting UAV-based VLN in 3D indoor environments.
- We develop an automated data collection and annotation pipeline that generates realistic UAV flight trajectories and multi-granularity natural language instructions.
- We propose IndoorUAV-Agent, a strong baseline model tailored to the unique challenges of indoor aerial VLN across different navigation settings.

Related Work

Ground-based Vision-Language Navigation Datasets

Early VLN datasets focus on ground-based navigation. The upper part of Table 1 presents recent ground-based datasets. Room-to-Room (R2R) (Anderson et al. 2018) is the first VLN dataset to enable navigation in discrete environments by using panoramic images. Follow-up datasets expanded this setting. For example, Room-Across-Room (RxR) (Ku et al. 2020) extended R2R to multiple languages and longer trajectories. TouchDown (Chen et al. 2019) leverages Google Street View to address the challenges of outdoor long-range navigation. CVDN (Thomason et al. 2020) introduces the navigation from dialog history task. All the above datasets are graph-based, with predefined navigable points. VLN-CE (Krantz et al. 2020) lifts the VLN task to continuous 3D environments by removing the nav-graph. Recent work like LHPR-VLN (Song et al. 2025) emphasizes long-term planning and decision consistency across consecutive subtasks, while OctoNav (Gao et al. 2025a) focuses on generalist embodied navigation.

Aerial Vision-Language Navigation Datasets

Aerial VLN has recently been gaining significant momentum. Unlike ground-based navigation, Aerial VLN involves 3D trajectories, typically has more degrees of freedom, and primarily operates in outdoor environments. The lower part of Table 1 presents the primary Aerial VLN datasets. AVDN (Fan et al. 2023) comprises 3K continuous drone trajectories in a photorealistic simulator paired with asynchronous human–human dialogs, supporting dialog-guided aerial navigation tasks. OpenFly (Gao et al. 2025b) delivers a comprehensive aerial VLN platform with 100K automatically generated trajectories across 18 diverse scenes, combining multiple rendering engines. UAV-Flow (Wang et al. 2025) is the first real-world benchmark for fine-grained, language-conditioned UAV control, with large-scale real and simulated datasets focusing on short-horizon navigation.

Vision-Language Navigation Methods

Early VLN agents typically employ LSTM (Graves 2012) to process language instructions and visual observations, such as Seq2Seq (Anderson et al. 2018) and CMA (Krantz et al. 2020). To more effectively encode historical navigation information, graph-based methods (Zhu et al. 2021;

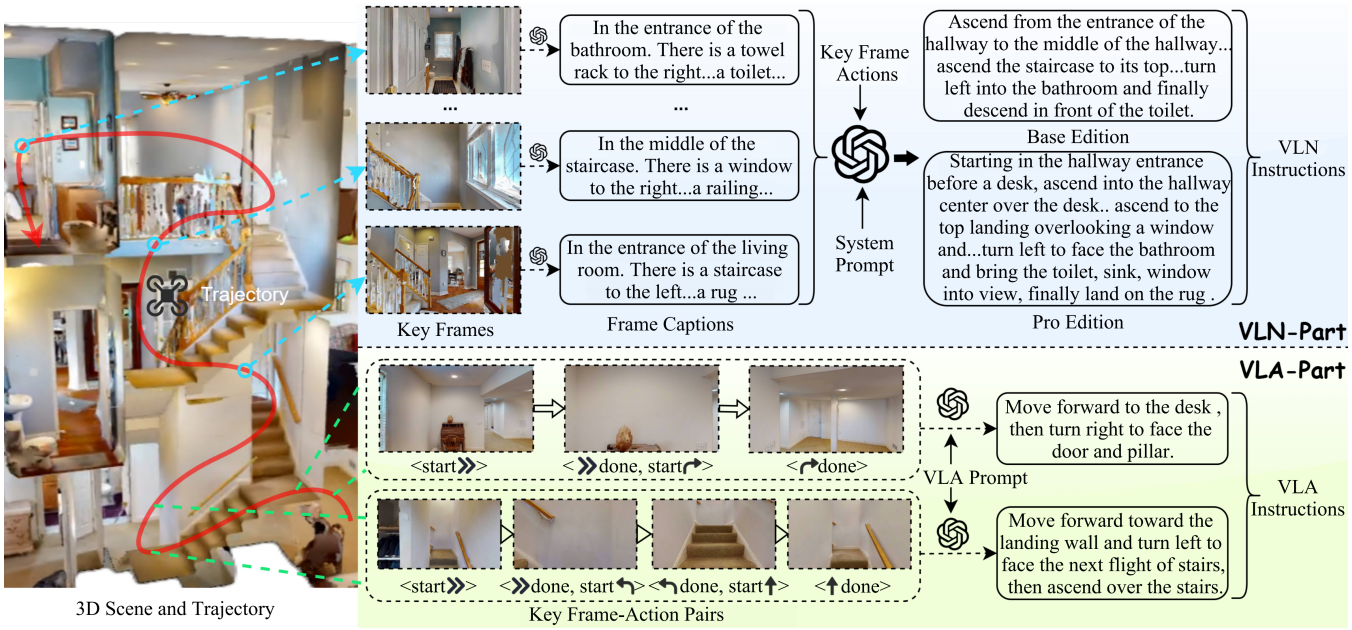

Figure 2: Overview of the IndoorUAV data collection and instruction generation pipeline.

Deng, Narasimhan, and Russakovsky 2020; Wang et al. 2021; Chen et al. 2022) build structured memory in the form of topological or semantic maps. Recent work explore leveraging LLM as planners. NavGPTv2 (Zhou et al. 2024) incorporates a Vision Language Model (VLM) and a graph-based policy network for effective action prediction. UniNaVid (Zhang et al. 2025) leverages a video-based VLM to unify different paradigms of navigation tasks and propose an online token merge strategy to efficiently process video streams. NavAgent (Liu et al. 2024) undertakes UAV navigation tasks by synthesizing multi-scale environmental information. SkyVLN (Li et al. 2025) incorporates an NMPC module for dynamic obstacle avoidance during the process of UAV navigation.

IndoorUAV Benchmark

To advance research in vision-language navigation for indoor aerial agents, we introduce the IndoorUAV benchmark—a large-scale, high-quality dataset specifically designed for simulating realistic UAV-based navigation in cluttered 3D indoor environments. IndoorUAV aims to address the unique challenges of indoor aerial VLN, including unconstrained 3D movement, fine-grained spatial understanding, and real-world-style language instruction. IndoorUAV benchmark consists of over 50000 high-resolution UAV trajectories paired with richly annotated natural language instructions. The benchmark contains two subsets: IndoorUAV-VLN, which focuses on long-horizon, goal-directed navigation through multi-step natural language commands; IndoorUAV-VLA, which targets short-horizon, low-level action planning. This section details the environment sources, data collection methodology, and the composition of the dataset.

Environment Source

We construct the IndoorUAV benchmark within high-fidelity simulated 3D indoor environments drawn from four widely-used 3D indoor space datasets in the embodied AI community: Matterport3D (MP3D), Gibson, HM3D, and Replica. These environments encompass a diverse range of residential, office, and public spaces, offering rich variability in layout, appearance, and object distribution. From these sources, we manually select and curate over 1,000 high-quality scenes that support rich 3D exploration and navigation. All selected environments are compatible with the Habitat simulator, enabling efficient data generation with realistic physics and photorealistic rendering.

Data Collection

To enable unrestricted 3D UAV movement, we first remove the built-in navigation mesh (navmesh) constraints from the environmints, allowing agents to move freely in three dimensions of space. We define a 4 degree-of-freedom (4 DoF) UAV action space, consisting of (1) forward movement, (2) vertical translation (up/down), (3) lateral motion (left/right), and (4) yaw rotation (see Figure 3(a)). This configuration closely mirrors realistic UAV flight dynamics and supports the generation of expressive and complex flight trajectories.

IndoorUAV-VLN Collection For the Vision-and-Language Navigation (VLN) subset of IndoorUAV, we first sample a large number of start and goal locations within each scene to define diverse navigation tasks covering various spatial scales and complexity levels. Experienced operators then remotely control the UAV within the simulator to manually navigate from predefined start points to target goals. This manual piloting captures natural, smooth flight paths that reflect realistic human UAV operation in

Dataset	N_{traj}	N_{vocab}	Path Len.	Intr Len.	Action Space	N_{scenes}	Environment
R2R (Anderson et al. 2018)	7189	3.1K	10.0	29	graph-based	90	Matterport3D
RxR (Ku et al. 2020)	13992	7.0K	14.9	129	graph-based	90	Matterport3D
CVDN (Thomason et al. 2020)	7415	4.4K	25.0	34	graph-based	83	Matterport3D
TouchDown (Chen et al. 2019)	9326	5.0K	313.9	90	graph-based	-	Google Street View
VLN-CE (Krantz et al. 2020)	4475	4.3K	11.1	19	2 DoF	90	Matterport3D
LHPR-VLN (Song et al. 2025)	3260	0.5K	-	18.17	2 DoF	216	Habitat
OctoNav (Gao et al. 2025a)	45k	-	-	-	2 DoF	438	Mp3D, Gibson, HM3D, ProcTHOR
AVDN (Fan et al. 2023)	6269	3.3K	144.7	89	3 DoF	-	xView
AerialVLN (Liu et al. 2023)	8446	4.5K	661.8	83	4 DoF	25	AirSim + UE
CityNav (Lee et al. 2024)	32637	6.6K	545	26	4 DoF	34	SensatUrban
OpenUAV (Wang et al. 2024)	12149	10.8K	255	104	6 DoF	22	AirSim + UE
OpenFly (Gao et al. 2025b)	100K	15.6K	99.1	59	4 DoF	18	AirSim, GTA5, 3D GS, GE
UAVFlow (Wang et al. 2025)	40801	0.3K	10.7	9	6 DoF	-	Real world, UnrealCV
IndoorUAV-VLN	16040	3.9K	21.6	112	4 DoF	1075	Mp3D, Gibson, HM3D, Replica
IndoorUAV-VLA	34925	2.2K	2.2	14.5	4 DoF	1075	Mp3D, Gibson, HM3D, Replica

Table 1: Comparison between existing VLN (Vision-and-Language Navigation) datasets and our IndoorUAV. Above the middle dividing line lies the ground-based datasets, while below is the aerial VLN datasets. N_{traj}: the number of total trajectories. N_{vocab}: vocabulary size. Path Len: the average length of trajectories, measured in meters. Intr Len: the average length of instructions. N_{scenes}: the number of used scenes.

complex indoor settings. To augment the dataset without sacrificing quality, we reverse the direction of each trajectory (by swapping start and goal points) and adjust the orientation sequence accordingly. This simple yet effective technique doubles the dataset size while maintaining trajectory realism.

To generate language instructions for each trajectory, we employ a multi-step pipeline built on top of a GPT-4-based instruction generation module (Figure 2). For each trajectory, we extract keyframes based on significant motion changes, such as sharp turns (over 45°), vertical climbs (over 1m), or long linear flights. For each keyframe, we label its corresponding action type (e.g., `fly_up`, `turn_right`) based on the preceding motion segment. Next, we prompt GPT-4o to describe each keyframe image by: (1) describing the location (e.g., kitchen); (2) describing nearby objects and structures using a coarse-to-fine scale (near/mid/far and left-/center/right). These image-grounded descriptions are concatenated and passed to GPT-4 again to generate the full navigation instruction. Finally, each trajectory is paired with two levels of instruction: (1) detailed, long-form instruction capturing all spatial and semantic nuances; (2) relatively shorter instruction focusing only on coarse goal descriptions. IndoorUAV-VLN focuses on long-horizon navigation, requiring the agent to understand and execute multi-step, semantically rich commands in cluttered 3D space.

IndoorUAV-VLA Collection To enable research on low-level control and fine-grained action prediction, we construct a complementary dataset derived from IndoorUAV-VLN. In this subset, we segment each long trajectory into multiple short sub-trajectories, each covering only 1–3 key actions (e.g., "fly forward past the cabinet," "descend near the table"). For each segment, we regenerate concise navigation instructions that focus on local spatial goals and immediate surroundings by directly prompt the GPT-4o with images. As shown in the Table 1, IndoorUAV-VLA contains 34,925 short trajectories, but the average instruction length is only 14.5 words.

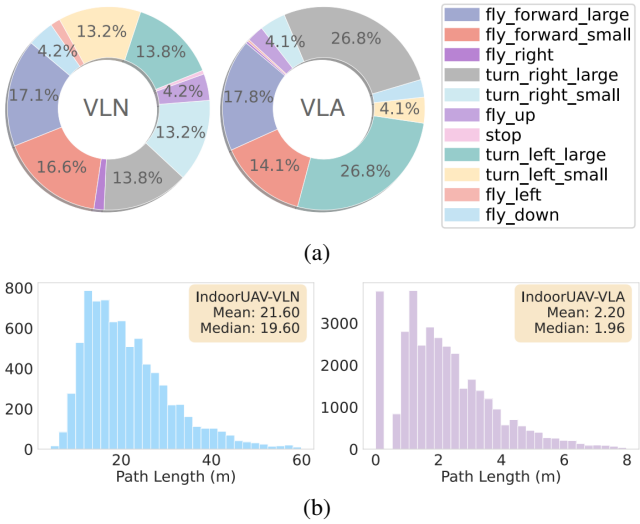

Figure 3: Statistical analysis of the IndoorUAV benchmark. (a) Action distributions. (b) trajectory length distributions.

Dataset Analysis

In this section, we conduct a comprehensive analysis of IndoorUAV from multiple perspectives, including scene diversity, trajectory complexity, and instruction granularity.

As shown in Table 1, IndoorUAV is constructed from a diverse collection of 1,075 photorealistic indoor scenes, significantly surpassing all previous VLN datasets in terms of scene diversity. In total, the dataset comprises 50965 navigation trajectories, each representing a 4-DoF UAV trajectory that supports horizontal transla-

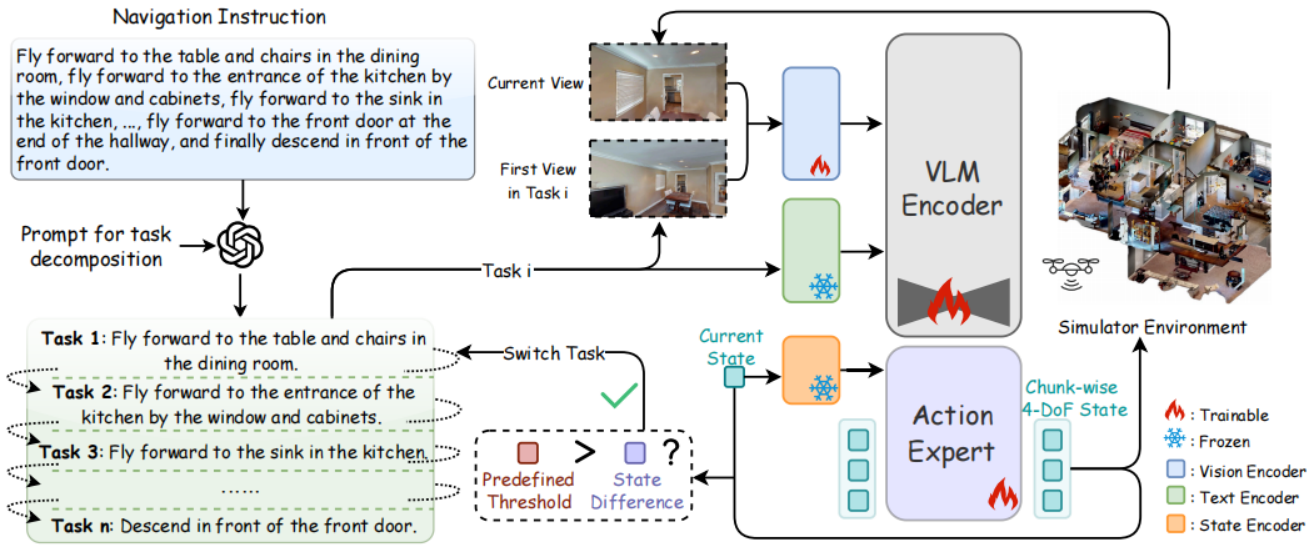

Figure 4: For the long-horizon VLN task, we first use GPT-4o to decompose the long instruction into n shorter VLA-style instructions as subtasks, each containing 1 to 3 actions. We then process each subtask sequentially using a VLA model based on the π_0 architecture.

tion, vertical movement, and yaw rotation. Figure 3(a) presents the overall distribution of discrete actions across the two dataset parts. The most frequent actions are `fly_forward`, `turn_left` and `turn_right`, which are essential for navigating tight indoor spaces. To improve balance across action types and enable finer control, we define dual-scale versions of key movements: `fly_forward_small` moves the UAV forward by 0.15 meters, while `fly_forward_large` performs a 0.9 meters step; `turn_left_small` and `turn_right_small` rotate the UAV by **3**°, while `turn_left_large` and `turn_right_large` perform turns of **15**°. This fine-grained action space is particularly suited for indoor aerial navigation, where small adjustments are critical for avoiding obstacles and maintaining stable flight.

Figure 3(b) illustrates the distribution of trajectory lengths across the two subsets of IndoorUAV. To provide a more systematic characterization of task complexity, we annotate each trajectory with a difficulty level. Specifically, in IndoorUAV-VLN, trajectories containing fewer than 120 actions are categorized as easy, those with 120–200 actions as medium, and those exceeding 200 actions as hard. For IndoorUAV-VLA, difficulty levels are defined based on action types, with 1 type denoting easy, 2 types denoting medium, and 3 types denoting hard.

IndoorUAV-Agent

Task Definition

Formally, an instruction I describes a navigation goal composed of multiple semantic steps (e.g., "Exit the kitchen, fly down the hallway, and enter the second room on the right."). The agent starts at an initial pose $s_0 = (x_0, y_0, z_0, \theta_0)$ and must predict a sequence of poses that lead to the successful completion of the task. We divide the overall problem into two categories: (1) Short-horizon VLA tasks, where the instruction corresponds to 1–3 primitive actions and can be directly translated into a continuous low-level trajectory. (2) Long-horizon VLN tasks, where the instruction is compositional and must be decomposed into a sequence of simpler navigation goals.

Model Architecture

For short-horizon VLA tasks, we directly leverage a fine-tuned π_0 model to predict a horizon of h future robot states, including the 3D coordinates (x, y, z) and yaw angle θ at each step. This model takes as input the current egocentric visual observation and a short natural language instruction, and outputs continuous low-level controls in the form of a predicted trajectory. The π_0 architecture, with its language-conditioned visual encoder and trajectory decoder, proves effective for such short and precise navigation tasks that typically involve only one or two atomic actions.

In contrast, long-horizon VLN tasks often involve complex instructions spanning multiple semantic steps (e.g., "Exit the room, go down the hallway, and turn left at the second door"), where current end-to-end continuous control models struggle to generalize. To address this, we propose a task decomposition pipeline (see Figure 4), which first utilizes GPT-4o to split a long-horizon instruction into n shorter, VLA-style sub-instructions, each describing a simple goal that can typically be completed in 1–3 actions. These sub-instructions are then processed sequentially, where for each subtask we invoke the π_0-based VLA model to predict and execute the corresponding trajectory segment. Importantly, for the i-th subtask, we use the predicted final-state observation (image) from the $(i-1)$-th subtask as the first reference frame for the current model input. This design ensures temporal continuity across subtasks and

IndoorUAV-VLA	Full		Easy		Medium		Hard	
	SR/%↑	NDTW/%↑	SR/%↑	NDTW/%↑	SR/%↑	NDTW/%↑	SR/%↑	NDTW/%↑
GPT-4o	11.69	9.2	30.30	12.30	4.00	6.57	1.96	4.84
Seq2Seq*	1.33	2.74	1.6	2.63	1.2	2.74	1.03	3.03
CMA*	0.99	1.88	1.28	1.75	0.75	1.94	1.03	2.01
OpenVLA*	7.81	2.42	22.52	2.89	1.19	1.12	0.0	0.12
π_0-FAST*	8.62	4.71	18.09	8.83	5.26	2.93	1.14	2.68
NaVid*	15.82	5.28	25.31	13.1	18.21	3.21	2.31	1.72
π_0*	27.16	9.44	46.58	14.52	21.64	7.64	7.55	4.27

Table 2: Comparison results on the IndoorUAV-VLA test split. * indicates models fine-tuned on the dataset.

IndoorUAV-VLN	TEST SEEN				TEST UNSEEN			
	NE/m↓	SR/%↑	OSR/%↑	NDTW/%↑	NE/m↓	SR/%↑	OSR/%↑	NDTW/%↑
GPT-4o	7.96	3.69	4.67	6.96	8.52	0.56	2.78	6.82
Seq2Seq*	10.18	0.89	2.23	1.41	11.6	0.41	1.85	1.23
CMA*	11.6	1.56	8.48	1.39	12.15	1.64	10.47	0.81
π_0*	8.35	2.92	11.5	11.87	8.81	2.83	10.02	11.69
NaVid*	21.83	0.75	14.70	1.36	19.40	0.84	16.21	2.32
OpenFly-Agent*	8.17	4.12	10.96	10.63	8.83	2.58	9.45	10.14
Ours	6.62	7.29	12.83	17.19	7.27	5.06	13.49	15.65

Table 3: Comparison results on the IndoorUAV-VLN test split. * indicates models fine-tuned on the dataset.

mitigates error accumulation by aligning each instruction with an updated visual context. By interleaving natural language understanding and low-level motion planning in this modular fashion, our approach improves interpretability, robustness, and success rate on complex VLN tasks.

Experiments

Evaluation Metrics

Following the previous work, we utilize four metrics for evaluation. (1) **Success Rate (SR)**: The definition of SR varies depending on the task type. For *VLA* tasks, a trajectory is considered successful if the predicted final position is within 0.5 meters of the target and the yaw difference is less than $\frac{\pi}{4}$. For *VLN* tasks, where high-level navigation goals are given, a prediction is considered successful if the final position is within 2 meters of the target. (2) **Normalized Dynamic Time Warping (NDTW)** (Ilharco et al. 2019): While some predicted trajectories may be semantically correct, they may follow irregular or suboptimal paths. To account for such discrepancies, we additionally compute NDTW to quantitatively assess the alignment between predicted and reference trajectories. For VLA tasks, we calculate both the three-dimensional coordinate NDTW and yaw angle NDTW for each trajectory, and weighted them respectively based on the path length and cumulative rotation angle of the trajectory as the final NDTW. And for VLN tasks, we only calculate the three-dimensional coordinate NDTW. (3) **Navigation Error (NE)**: This metric measures the distance between the final position of the predicted trajectory and the target position. (4) **Oracle Success Rate (OSR)**: OSR evaluates whether any point along the predicted trajectory satisfies the success condition defined in SR.

Baseline Models

We benchmark various representative models spanning VLA and VLN. The VLA models directly output continuous robot motion controls informed by visual observations and language instructions. In contrast, VLN models predict high-level navigation steps from visual inputs and language, which are then executed as motion.

VLA Models (1) π_0 (Black et al. 2024) and π_0-FAST (Pertsch et al. 2025): build on a pretrained VLM and use diffusion-style decoding to generate smooth continuous control sequences at up to 50 Hz. (2) **OpenVLA** (Kim et al. 2024): An open-source discrete-token VLA model trained on 970K real-robot episodes, which achieves broad generalization and strong zero-shot and fine-tuned performance.

At each time step t, the agent receives the initial observation o_1, current observation o_t, and current state s_t. The state is defined as a tuple of 3D coordinates and a yaw angle, i.e., $s_t = (x_t, y_t, z_t, \theta_t)$. The VLA model encodes these inputs to predict a sequence of future states for the next h time steps:

$$S_{t+1:t+h+1} = \text{Model}_{\text{VLA}}(\mathcal{O}_1, \mathcal{O}_t, \mathcal{I}, s_t) \quad (1)$$

VLN Models (1) **Seq2Seq** and **CMA** (Krantz et al. 2020): Traditional LSTM-style models that encode visual inputs and instructions to regress continuous waypoint sequences using a recurrent policy. (2) **NaVid** (Zhang et al. 2024): A video-based VLM trained with a large amount of data , which is designed for ground-based VLN. (3) **OpenFly-Agent** (Gao et al. 2025b): an aerial navigation model that builds upon the OpenVLA.

Figure 5: Visualization on both IndoorUAV-VLA and IndoorUAV-VLN. The upper two are VLA tasks, with medium/hard (2-3 executable actions) difficulty, respectively. The lower is a VLN task where the markers *S* and *T* in the trajectory plot indicate the start position and target position of the trajectory, respectively.

At each time step t, the agent receives a navigation instruction $\mathcal{I}$, along with a sequence of past observations $\mathcal{O}_{1:t} = \{o_1, o_2, \ldots, o_t\}$. The VLN model encodes both modalities and produces a navigation action $a_t \in \mathcal{A}$, where $\mathcal{A}$ denotes a discrete action space. Formally:

$$a_t = \text{Model}_{\text{VLN}}(\mathcal{O}_{1:t}, \mathcal{I}) \tag{2}$$

Quantitative results

IndoorUAV-VLA. Table 3 summarizes the performance of various models on the IndoorUAV-VLA task. Among all methods, fine-tuned π_0 achieves the best overall results, attaining 27.16% Success Rate (SR) and 9.44% NDTW across the full test split. On easy trajectories, it achieves an impressive 46.58% SR, significantly outperforming other methods, and remains competitive on medium (21.64%) and hard (7.55%) tasks. Traditional VLN-style regression models fail to generalize to the low-level action space, with SR below 3%. NaVid shows improved performance but still falls short of the results achieved by π_0.

IndoorUAV-VLN. Table 3 presents results on the IndoorUAV-VLN task. Compared with the VLA setting, the challenges of long-range goal-directed planning amplify performance differences among models. Our IndoorUAV-Agent method achieves the best performance across all metrics, attaining 7.29% and 5.06% SR in seen and unseen environments respectively, while also achieving the highest NDTW (17.19% / 15.65%). Compared to the π_0 baseline, which uses the same low-level policy but without any task decomposition, our method improves SR by +4.37% in the seen split and +2.23% in the unseen split. These gains demonstrate the effectiveness of integrating instruction decomposition into the planning process.

While NaVid yields relatively high OSR scores (14.70% / 16.21%), its SR remains low (0.75% / 0.84%). Upon inspection, we find that NaVid often fails to predict the `stop` action. As a result, the agent tends to overshoot the goal region, leading to failures under the SR criterion, even when parts of the path are aligned with the instruction (thus yielding a higher OSR).

Qualitative results

Figure 5 presents several qualitative results of our IndoorUAV-Agent model on both VLA and VLN tasks. The top two examples illustrate short-range navigation scenarios, where the agent executes 2–3 actions to reach the target. The bottom example showcases a long-horizon VLN task. By decomposing the original instruction into a sequence of shorter sub-instructions, our model is able to successfully complete the long-range navigation.

Conclusion

We present IndoorUAV, the first large-scale benchmark for aerial VLN in indoor environments. Built with 1000+ environments, IndoorUAV features diverse indoor layouts and simulates UAV-specific viewpoints and motion dynamics, enabling research across both VLN and VLA paradigms. We further propose IndoorUAV-Agent, a modular framework designed to tackle the challenges of long-horizon aerial navigation through task decomposition and multi-granularity instruction understanding.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant No.: 62372015), Key Laboratory of

Intelligent Press Media Technology, and State Key Laboratory of General Artificial Intelligence.

References

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and Van Den Hengel, A. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3674–3683.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; Jakubczak, S.; Jones, T.; Ke, L.; Levine, S.; Li-Bell, A.; Mothukuri, M.; Nair, S.; Pertsch, K.; Shi, L. X.; Tanner, J.; Vuong, Q.; Walling, A.; Wang, H.; and Zhilinsky, U. 2024. π_0: A Vision-Language-Action Flow Model for General Robot Control. arXiv:2410.24164.

Chen, H.; Suhr, A.; Misra, D.; Snavely, N.; and Artzi, Y. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12538–12547.

Chen, S.; Guhur, P.-L.; Tapaswi, M.; Schmid, C.; and Laptev, I. 2022. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16537–16547.

Deng, Z.; Narasimhan, K.; and Russakovsky, O. 2020. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 33: 20660–20672.

Fan, Y.; Chen, W.; Jiang, T.; Zhou, C.; Zhang, Y.; and Wang, X. 2023. Aerial vision-and-dialog navigation. In *Findings of the Association for Computational Linguistics: ACL 2023*, 3043–3061.

Gao, C.; Jin, L.; Peng, X.; Zhang, J.; Deng, Y.; Li, A.; Wang, H.; and Liu, S. 2025a. OctoNav: Towards Generalist Embodied Navigation. *arXiv preprint arXiv:2506.09839*.

Gao, Y.; Li, C.; You, Z.; Liu, J.; Li, Z.; Chen, P.; Chen, Q.; Tang, Z.; Wang, L.; Yang, P.; et al. 2025b. OpenFly: A versatile toolchain and large-scale benchmark for aerial vision-language navigation. *arXiv e-prints*, arXiv–2502.

Graves, A. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.

Han, M.; Ma, L.; Zhumakhanova, K.; Radionova, E.; Zhang, J.; Chang, X.; Liang, X.; and Laptev, I. 2025. Roomtour3d: Geometry-aware video-instruction tuning for embodied navigation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 27586–27596.

Ilharco, G.; Jain, V.; Ku, A.; Ie, E.; and Baldridge, J. 2019. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; et al. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

Krantz, J.; Wijmans, E.; Majumdar, A.; Batra, D.; and Lee, S. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, 104–120. Springer.

Ku, A.; Anderson, P.; Patel, R.; Ie, E.; and Baldridge, J. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*.

Lee, J.; Miyanishi, T.; Kurita, S.; Sakamoto, K.; Azuma, D.; Matsuo, Y.; and Inoue, N. 2024. Citynav: Language-goal aerial navigation dataset with geographic information. *arXiv preprint arXiv:2406.14240*.

Li, T.; Huai, T.; Li, Z.; Gao, Y.; Li, H.; and Zheng, X. 2025. SkyVLN: Vision-and-Language Navigation and NMPC Control for UAVs in Urban Environments. *arXiv preprint arXiv:2507.06564*.

Liu, S.; Zhang, H.; Qi, Y.; Wang, P.; Zhang, Y.; and Wu, Q. 2023. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15384–15394.

Liu, Y.; Yao, F.; Yue, Y.; Xu, G.; Sun, X.; and Fu, K. 2024. Navagent: Multi-scale urban street view fusion for uav embodied vision-and-language navigation. *arXiv preprint arXiv:2411.08579*.

Pertsch, K.; Stachowicz, K.; Ichter, B.; Driess, D.; Nair, S.; Vuong, Q.; Mees, O.; Finn, C.; and Levine, S. 2025. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*.

Puig, X.; Undersander, E.; Szot, A.; Cote, M. D.; Partsey, R.; Yang, J.; Desai, R.; Clegg, A. W.; Hlavac, M.; Min, T.; Gervet, T.; Vondrus, V.; Berges, V.-P.; Turner, J.; Maksymets, O.; Kira, Z.; Kalakrishnan, M.; Malik, J.; Chaplot, D. S.; Jain, U.; Batra, D.; Rai, A.; and Mottaghi, R. 2023. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; Parikh, D.; and Batra, D. 2019. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Song, X.; Chen, W.; Liu, Y.; Chen, W.; Li, G.; and Lin, L. 2025. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12078–12088.

Szot, A.; Clegg, A.; Undersander, E.; Wijmans, E.; Zhao, Y.; Turner, J.; Maestre, N.; Mukadam, M.; Chaplot, D.; Maksymets, O.; Gokaslan, A.; Vondrus, V.; Dharur, S.; Meier, F.; Galuba, W.; Chang, A.; Kira, Z.; Koltun, V.; Malik, J.; Savva, M.; and Batra, D. 2021. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Thomason, J.; Murray, M.; Cakmak, M.; and Zettlemoyer, L. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, 394–406. PMLR.

Wang, H.; Wang, W.; Liang, W.; Xiong, C.; and Shen, J. 2021. Structured scene memory for vision-language naviga-

tion. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 8455–8464.

Wang, X.; Yang, D.; Liao, Y.; Zheng, W.; Dai, B.; Li, H.; Liu, S.; et al. 2025. UAV-Flow Colosseo: A Real-World Benchmark for Flying-on-a-Word UAV Imitation Learning. *arXiv preprint arXiv:2505.15725*.

Wang, X.; Yang, D.; Wang, Z.; Kwan, H.; Chen, J.; Wu, W.; Li, H.; Liao, Y.; and Liu, S. 2024. Towards realistic uav vision-language navigation: Platform, benchmark, and methodology. *arXiv preprint arXiv:2410.07087*.

Zhang, J.; Wang, K.; Wang, S.; Li, M.; Liu, H.; Wei, S.; Wang, Z.; Zhang, Z.; and Wang, H. 2025. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. *Robotics: Science and Systems*.

Zhang, J.; Wang, K.; Xu, R.; Zhou, G.; Hong, Y.; Fang, X.; Wu, Q.; Zhang, Z.; and Wang, H. 2024. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. *Robotics: Science and Systems*.

Zhou, G.; Hong, Y.; Wang, Z.; Wang, X. E.; and Wu, Q. 2024. Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In *European Conference on Computer Vision*, 260–278. Springer.

Zhu, F.; Liang, X.; Zhu, Y.; Yu, Q.; Chang, X.; and Liang, X. 2021. Soon: Scenario oriented object navigation with graph-based exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12689–12699.