HUGE-Bench: A Benchmark for High-Level UAV Vision-Language-Action Tasks

Jingyu Guo[1,2]*, Ziye Chen[1,2]*, Ziwen Li[3], Zhengqing Gao[3], Jiaxin Huang[3], Hanlue Zhang[3], Fengming Huang[4,5], Yu Yao[5], Tongliang Liu[3,5]†, Mingming Gong[1,3]†

[1]The University of Melbourne, Australia
[2]Melsy Tech, China
[3]MBZUAI, United Arab Emirates
[4]Navlyn, China
[5]The University of Sydney, Australia

*Equal contribution; †Corresponding authors.
guo27@student.unimelb.edu.au, tongliang.liu@sydney.edu.au, mingming.gong@unimelb.edu.au
Work done during internship at Melsy Tech.
https://jingyu198.github.io/HUGE_Bench

Abstract. We present **HUGE-Bench**, a benchmark for High-Level UAV Vision–Language–Action (HL-VLA) tasks that tests whether an agent can interpret concise language and execute complex, process-oriented trajectories with safety awareness. HUGE-Bench comprises 4 real-world digital twin scenes, 8 high-level tasks, and 2.56M meters of trajectories, and is built on an aligned 3D Gaussian Splatting (3DGS)–Mesh hybrid representation that combines photorealistic rendering with collision-capable geometry for scalable generation and collision-aware evaluation. We introduce process-oriented and collision-aware metrics to assess process fidelity and flight safety. Experiments on representative state-of-the-art VLA models reveal significant gaps in high-level semantic completion and safe execution, highlighting HUGE-Bench as a diagnostic testbed for high-level UAV autonomy.

Keywords: High-Level VLA · Benchmark · Data generation

1 Introduction

Unmanned aerial vehicles (UAVs) are increasingly deployed for inspection, search-and-rescue, infrastructure monitoring, and logistics, yet operating them in cluttered 3D environments remains labor-intensive and error-prone. Operators must translate high-level intent into dense waypoints and continuous low-level control while maintaining situational awareness and safety. Recent advances in vision-language-navigation (VLN) suggest a promising alternative: language-guided flight, where users express intent in natural language and an agent grounds it into

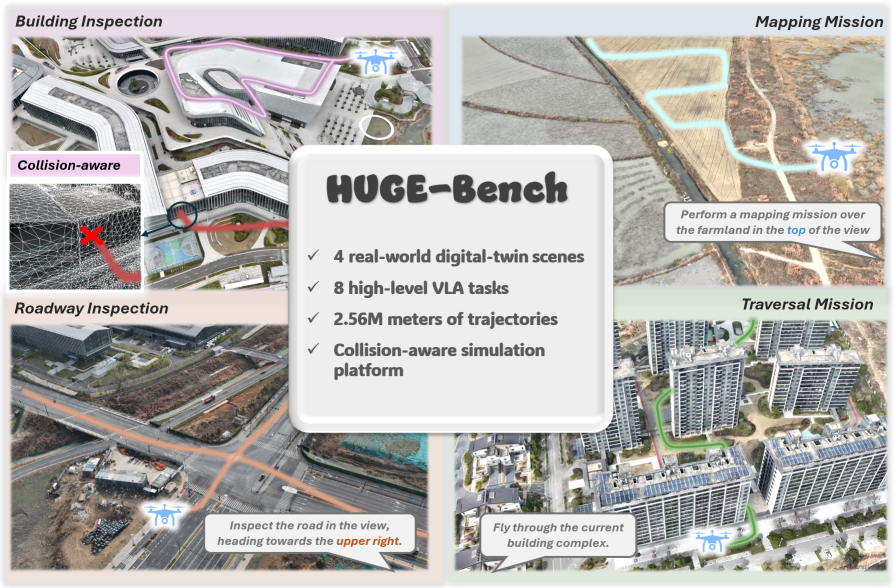

Fig. 1: Overview of HUGE-Bench. Our benchmark comprises four 3DGS–Mesh digital-twin environments reconstructed from real-world scenes, eight high-level UAV VLA tasks, and 2.56 million meters of trajectory data. In addition, we develop a collision-aware simulation platform that enables collision-free trajectory collection and facilitates safety evaluation.

perception and actions [2,7,17,20]. In the UAV domain, aerial VLN benchmarks have further catalyzed progress by standardizing tasks, datasets, and evaluation protocols [7, 20]. However, existing benchmarks still only partially reflect how UAVs are commanded and evaluated in real operations.

A central mismatch lies in instruction style and task abstraction. Most VLN settings emphasize long, step-wise route descriptions, and evaluation is largely goal-centric, e.g., Success Rate (SR) and Success weighted by Path Length (SPL) [2], sometimes complemented by normalized Dynamic Time Warping (nDTW) and Success weighted by nDTW (SDTW) for trajectory fidelity [10]. In contrast, UAV operators typically issue brief, high-level commands (e.g., "inspect the building on the left"), leaving the system to infer targets, decompose subtasks, and execute a multi-stage procedure safely. Such commands stress semantic grounding from aerial viewpoints, 3D spatial reasoning over direction/altitude/distance, and memory across stages. Yet most existing UAV benchmarks are still formulated as route-following VLN, which makes them less diagnostic for short, underspecified commands and process-oriented multi-stage behaviors (Table 1).

This gap is further amplified by environment representation and safety modeling. Benchmark validity depends on both photorealistic perception and phys-

Table 1: Comparison of UAV navigation and action benchmarks. "Source" indicates whether the environments are synthetic or reconstructed from real-world data. "3D Rep." denotes the 3D scene representation. "Subtask" indicates whether the benchmark provides explicit multi-stage task decomposition rather than single-step navigation. "Len. (m/w)" measures trajectory length normalized by instruction length. "Collision" indicates explicit collision-aware evaluation.

Benchmark	Source	3D Rep.	Subtask	Len. (m/w)	Collision	Type
AerialVLN [20]	Syn.	Mesh	×	7.97	×	VLN
CityNav [19]	Real	P. Cloud	×	20.92	×	VLN
TravelUAV [31]	Syn.	Mesh	×	2.45	✓	VLN
Openfly [7]	Syn., Real	Mesh, 3DGS	×	1.68	×	VLN
UAV-Flow [30]	Syn., Real	Mesh	×	< 4	×	VLA
HUGE-Bench	**Real**	**3DGS-Mesh**	✓	**32.07**	✓	**VLA**

ically executable collision checking. Mesh-based simulators support physics and collision queries (e.g., high-fidelity UAV simulation in AirSim) [27], but may introduce a perception gap for vision–language grounding. Conversely, neural rendering such as 3D Gaussian Splatting (3DGS) provides real-time, photorealistic novel-view rendering [15], but native 3DGS is appearance-first and does not directly yield collision-ready geometry. Recent efforts toward physically executable 3DGS highlight the need for hybrid designs that pair 3DGS rendering with mesh-based collision bodies [21], which is particularly crucial for safety-critical UAV evaluation. Moreover, current evaluation protocols under-measure process correctness. Goal-only metrics can miss important failure modes in high-level missions: a policy may reach an endpoint while skipping required procedures, drifting from the intended process, or colliding en route. Many UAV missions are inherently process-oriented (inspection coverage, orbiting with clearance, mapping completeness), and thus require metrics that assess procedural completion beyond endpoint success.

To address these limitations, we introduce **HUGE-Bench**, a benchmark for High-Level UAV Vision Language Action (HL-VLA) control. HUGE-Bench contains 4 real-world 3DGS-Mesh scenes, 8 high-level tasks, and 2.56M meters of trajectories, together with collision-aware simulation. Unlike route-following VLN, HUGE-Bench targets the operational regime where a short command implies structured multi-stage behaviors. Our task suite spans landing, road/building inspection, area mapping, orbiting (altitude/radius control), multi-turn spiral-down, and obstacle-aware region traversal (Fig. 1). **HUGE-Bench** is enabled by a real-to-sim pipeline that couples photorealistic rendering with physics-grounded collision evaluation (Fig. 2). We reconstruct an aligned 3DGS–Mesh digital twin: 3DGS supports realistic perception, while the mesh provides collision queries and depth. This design supports scalable trajectory generation while preserving an explicit safety channel (collision outcomes), enabling evaluation of both semantic grounding under short commands and safe execution.

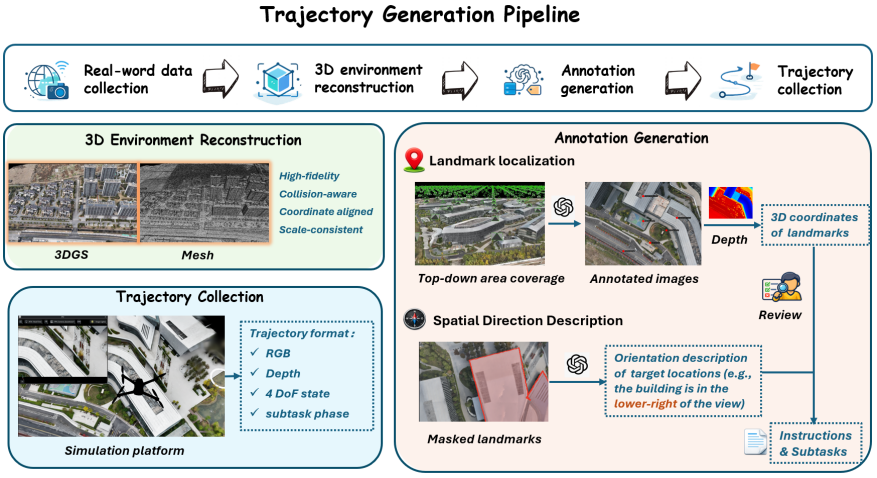

Fig. 2: Trajectory generation pipeline of HUGE-Bench. Starting from real-world data collection, we reconstruct 3D environments using 3DGS and mesh representations. Landmarks are localized and annotated to generate spatial descriptions and task instructions. Trajectories are collected in the simulation platform with multimodal observations, including RGB, depth, 4-DoF state, and subtask phases.

We propose evaluation tailored to high-level UAV behaviors. We report Average Trajectory Coverage Rate (Avg. TCR), normalized Dynamic Time Warping (nDTW), and Normalized Task Progress (NTP) to assess process completion, trajectory alignment, and task progress. For safety-critical traversal, we additionally measure Collision Rate (CR) and Collision-aware Success weighted by Path Length (CSPL) to evaluate collision frequency, goal reaching, and path efficiency. Using this protocol, we benchmark representative VLA and VLM models (OpenVLA, FastVLM, π_0, and $\pi_{0.5}$) [11,12,16,29]. Results reveal substantial gaps in process completion and safe execution under short instructions, positioning HUGE-Bench as a diagnostic and safety-relevant testbed for high-level UAV VLA.

In summary, our contributions are three-fold:

- **HL-VLA task formulation.** We introduce a new benchmark setting for HL-VLA tasks, where UAVs must interpret short, potentially ambiguous commands and execute multi-stage semantic behaviors.
- **Real-to-sim benchmark.** We build HUGE-Bench, a UAV benchmark constructed from real-world scenes and an aligned 3DGS–Mesh digital twin, enabling large-scale trajectory generation and realistic safety-aware evaluation.
- **Process-oriented and safety evaluation.** We propose process-oriented and collision-aware metrics that evaluate UAV execution along three dimensions: process fidelity and safety.

2 Related Work

2.1 Aerial Vision–Language Navigation and VLA

Vision–Language Navigation (VLN) studies instruction-conditioned navigation, typically evaluated with goal-centric success/efficiency metrics (e.g., SR/SPL) and trajectory-faithfulness scores (e.g., nDTW/SDTW) [2, 10, 17]. Large-scale VLN datasets and variants (e.g., R4R, RxR, CVDN, REVERIE) further expose long-horizon and grounded language challenges [13, 18, 24, 28]. Methodologically, VLN has progressed from imitation/RL baselines to pretraining and transformer-based policies with longer memory and stronger grounding [4, 5, 8, 9].

Aerial VLN inherits this protocol but must handle continuous 3D flight dynamics, larger scale, and stronger safety constraints. AerialVLN proposes Look-ahead Guidance (LAG) to mitigate supervision mismatch between shortest paths and language-described routes [20]. OpenFly scales aerial VLN via automated trajectory–instruction generation across multiple engines and introduces a keyframe-aware agent with history and token merging [7]. Recent VLM/VLA-based navigation explores video-conditioned action prediction and cross-embodiment generalization [6, 32, 33]. Despite these advances, the dominant formulation remains route-following with detailed instructions. In contrast, we study high-level UAV Vision Language Action tasks under short, underspecified commands, where success requires process-oriented multi-stage execution and safety-aware behavior.

2.2 UAV VLN/VLA Benchmarks

UAV benchmarks vary widely in data sources, scene representations, annotation granularity, and safety protocols. As summarized in Table 1, AerialVLN [20], CityNav [19], and OpenFly [7] are primarily VLN-style benchmarks without intermediate subtask annotations and with limited collision-centric evaluation, while TravelUAV [31] emphasizes more realistic UAV simulation and assistant-guided object search. UAV-Flow shifts to language-conditioned fine-grained control with real-world episodes, complementing VLN with a VLA-style control regime [30]. Overall, existing resources underrepresent the regime of short language driving long, multi-stage flights and rarely standardize collision-aware evaluation. HUGE-Bench is designed to occupy this missing region with (i) high-level task definitions, (ii) scalable trajectory generation, and (iii) explicit collision-aware evaluation enabled by our digital twin.

2.3 Environment Representations and Evaluation

Safety-relevant benchmarking depends on both photorealistic perception and physically executable collision checking. UAV simulation platforms such as AirSim support physics-based UAV dynamics and collision queries [27], while embodied AI simulators and datasets (e.g., Habitat, Matterport3D, HM3D) provide

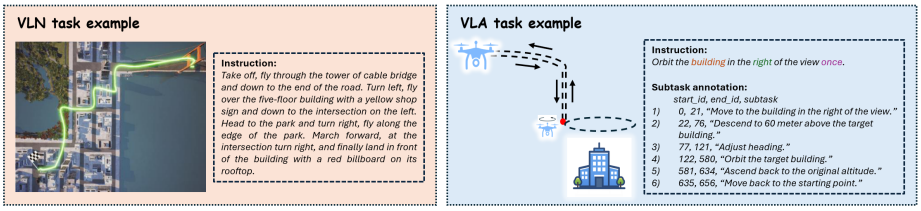

Fig. 3: Difference between previous VLN tasks and HL-VLA tasks. Left: a VLN task example [20] where the UAV follows a long-horizon navigation instruction through multiple landmarks. Right: a HL-VLA task example with a high-level instruction decomposed into temporally aligned subtasks for structured trajectory execution.

efficient rendering and large-scale reconstructed geometry [3, 25, 26]. Neural rendering, especially 3D Gaussian Splatting (3DGS), enables high-quality real-time novel-view synthesis [15], but is not collision-ready by default; recent work therefore advocates hybrid designs that pair 3DGS rendering with mesh-based collision bodies for physically executable evaluation [21]. On the metrics side, classic VLN emphasizes SR/SPL and trajectory similarity (nDTW/SDTW), and complementary measures such as instruction-fidelity/coverage have also been proposed [2, 10, 13]. Motivated by the need to evaluate process correctness and safety in high-level UAV behaviors, HUGE-Bench adopts an aligned 3DGS–Mesh digital twin and reports process-oriented and collision-aware metrics tailored to multi-stage trajectories rather than goal-only navigation.

3 HUGE-Bench

We introduce HUGE-Bench, a benchmark for high-level Vision–Language–Action control of UAVs. HUGE-Bench includes (i) a task suite with representative high-level UAV behaviors, (ii) a scalable trajectory and annotation generation pipeline grounded in real-world capture, (iii) collision-aware simulation and data generation platform based on an aligned 3DGS–Mesh digital twin, and (iv) process-oriented evaluation metrics that jointly measure process fidelity and safety. An overview of the pipeline is shown in Fig. 2.

3.1 High-level VLA Tasks for UAVs

We consider a UAV VLA policy that, given the current observation (e.g., RGB or RGBD) and a natural-language instruction, predicts the next-step action target, such as velocity commands or the next state along a trajectory:

$$\mathbf{a}_t = \pi_\theta(\mathbf{o}_t, \mathbf{x}, \mathbf{s}_t)\,, \tag{1}$$

where $\mathbf{o}_t$ denotes the UAV observation at time t, $\mathbf{x}$ is the instruction, $\mathbf{s}_t$ is the current state, and $\mathbf{a}_t$ is the action target (e.g., velocity $\mathbf{v}_t$ or the next state $\mathbf{s}_{t+1}$) predicted by the policy π_θ with parameters θ.

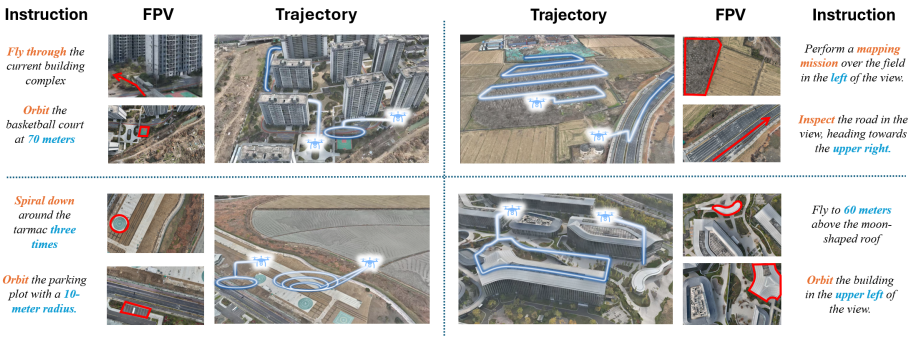

Fig. 4: Examples of HL-VLA tasks in HUGE-Bench. Each example shows a natural-language instruction, the corresponding first-person view (FPV) observations highlighting target regions, and the resulting multi-stage flight trajectory.

High-level instructions in our setting are short and potentially ambiguous, reflecting realistic human interaction patterns (e.g., "inspect the building on the left"). Unlike step-by-step low-level commands, a high-level instruction typically implies a sequence of latent subtasks. For example, "inspect the building on the left" requires the UAV to: (1) identify the referred building from an aerial viewpoint, (2) approach it, (3) descend to an inspection altitude, (4) orbit while maintaining safe clearance, and (5) return to the initial altitude and position. Therefore, HL-VLA for UAVs stresses: (1) Language grounding and subtask decomposition under short and underspecified instructions, (2) Semantic visual understanding from aerial perspectives, and (3) 3D spatial reasoning (direction, altitude, distance, orientation). A comparison between conventional UAV VLN and our proposed HL-VLA task is illustrated in Fig. 3

Based on these requirements, we define eight representative tasks (see Fig. 4):

1. Target Landing: orient toward a target, reach its overhead position, descend to a specified altitude, and hover (referred to as **Landing**).
2. Road Inspection: reach the road, descend to inspection altitude, align with the required direction, perform inspection, then ascend back to the start altitude and stop (referred to as **Inspection-R**).
3. Adaptive Building Inspection: approach the building boundary, descend to inspection altitude, select an orbit direction, circumnavigate along the boundary with adaptive clearance, then return to the start pose (referred to as **Inspection-B**).
4. Area Mapping: move to the boundary of a target region, descend to a specified altitude, perform coverage-style mapping, and return to the start altitude (referred to as **Mapping**).
5. Orbiting at Different Heights: reach the target overhead location at a specified altitude, enter a circular track, loiter, then exit and return (referred to as **Orbit-H**).

6. Orbiting with Different Radius: similar to 5. but with an instruction-conditioned orbit radius (referred to as **Orbit-R**).
7. Multi-turn Spiral Down: execute a spiral loiter pattern (multi-turn) conditioned on the instruction (referred to as **Spiral Down**).
8. Region Traversal: traverse a designated area while detecting obstacles and executing avoidance behaviors (referred to as **Traversal**).

Collectively, these tasks probe target recognition, directional grounding, depth and altitude awareness, and multi-stage trajectory completion in a unified HL-VLA setting.

3.2 Trajectory Data Generation

HUGE-Bench trajectories are generated through a real-to-sim pipeline that couples photorealistic perception with physics-grounded collision awareness. The pipeline consists of: (1) real-world data capture, (2) aligned 3DGS–mesh digital twin reconstruction, (3) annotation generation, and (4) simulator-based trajectory collection.

Real-world Data Capture. We collect data using a DJI M400 UAV equipped with a Zenmuse L2 payload across four representative outdoor scenes: office buildings, dense urban blocks, swamp and farmland, and construction roads. The total covered area is approximately 6.45 km^2. Flights are performed at $\sim$ 85 m above ground, recording GPS from GNSS, UAV pose, camera pose, and high-resolution RGB images (5280×3956). To improve reconstruction quality at near-ground altitudes, we additionally capture supplementary low-altitude views.

3D Environment Reconstruction. For each scene, we reconstruct both a 3DGS model and a triangle mesh. This joint 3DGS–Mesh digital twin representation is central to HUGE-Bench: 3DGS provides high-fidelity renderings for realistic perception inputs. The mesh provides collision-enabled geometry and supports physics-based queries, as well as depth via ray casting.

Landmark Localization. We place a grid of top-down cameras at $\sim$ 60 m altitude and render the 3DGS scene to obtain global 2D map views, while rendering the mesh to obtain aligned depth. We then use an LLM to localize landmarks on the rendered maps and output normalized image coordinates. After manual review to remove ambiguous or incorrect labels, we back-project pixel coordinates using camera intrinsics and depth to obtain pixel-level 3D landmark positions. For semantically extended regions (e.g., buildings, roads, fields), we additionally provide human-annotated masks.

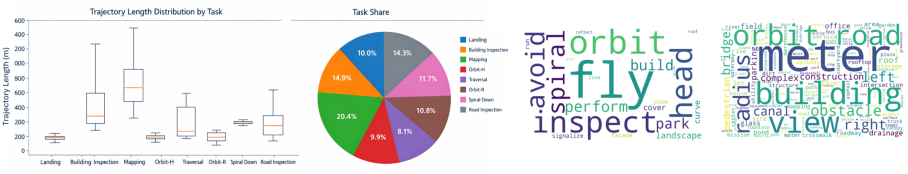

Fig. 5: Dataset statistics of HUGE-Bench. We analyze trajectory length distributions across tasks (left), the proportion of different task types (middle), and the vocabulary distribution in instructions through verb and landmark word clouds (right).

Spatial Direction Descriptions. Many HL tasks cannot be reliably specified with only generic nouns like "building" or "road," especially when multiple visually similar candidates appear in view or when a route direction must be specified (e.g., road inspection "along the right side"). To address this, before trajectory collection we select the intended target, sample random initial views in its vicinity, and use the target mask to prompt an LLM to produce human-like spatial referring expressions (e.g., "upper-left," "center," "along the right side of the frame"). This yields instructions combining directional phrases with generic categories (e.g., "the building in the upper-left"), enabling more precise language grounding under high-level commands.

Simulator-based Trajectory Collection Trajectory collection is conducted in Isaac Sim [22], which offers GPU-accelerated rendering and PhysX-based rigid-body simulation. First, we import the aligned 3DGS–Mesh digital twin into the simulator. Then we generate a sequence of 3D waypoints using task-specific rules. During execution, the simulator records synchronized multimodal streams including RGB, depth, pose, flight states, and collision signals, enabling scalable data generation and standardized evaluation. For data sampling, we record frames at approximately 1 m spatial intervals along the trajectory; during turns, we additionally sample at every 5° heading change to better capture viewpoint transitions. For the obstacle-avoidance traversal task, we generate paths using an RRT-based planner [14]. Because dense triangle meshes can significantly increase planning and collision-query cost, we perform mesh cleaning floater removal and mesh decimation to reduce complexity. Additional implementation details are provided in the Appendix.

3.3 Data Analysis

Fig. 5 summarizes the dataset statistics. Trajectory lengths vary substantially across tasks: Mapping exhibits the longest and most diverse trajectories, while Landing, Orbit-H, Orbit-R, and Spiral Down are shorter and more concentrated. The task distribution is relatively balanced, with Mapping as the largest subset (20.4%) and Traversal the smallest (8.1%). The two word clouds summarize the dominant semantics in HL-VLA instructions. The left cloud is dominated by

scene entities and landmarks (e.g., building, road, tree, obstacle). The right cloud highlights action-oriented verbs (e.g., fly, orbit, avoid, inspect, spiral), suggesting that the tasks emphasize high-level motion primitives and inspection behaviors beyond simply reaching a goal.

3.4 Dataset Splits

We partition our dataset into three splits: train, test-seen, and test-unseen. This design aims to evaluate a model's ability to generalize across both outdoor environments and natural-language instructions.

In our benchmark, we define test-seen as trajectories whose target landmark appears in the training set. Although the landmark itself is seen during training, the initial observation in the first frame is randomized, including the agent's altitude, horizontal distance to the target, and the relative direction with respect to the landmark. This split evaluates whether the model can robustly reach a familiar target under novel initial viewpoints and spatial configurations.

To further assess generalization to new scenarios and language, we construct a test-unseen split from two complementary sources: (1) Landmark-unseen: we hold out a subset of landmarks that never appear in training. (2) Instruction-unseen: we apply instruction-level generalization by randomly diversifying the language. Specifically, we use an LLM to generate paraphrases that replace the verb or the textual description of the target landmark while preserving the intended task semantics. In total, our dataset contains: 5,330 trajectories in train, 593 trajectories in test-seen, and 294 trajectories in test-unseen. This split measures the model's ability to handle novel landmarks and linguistic variations beyond the training distribution.

3.5 Evaluation Metrics

Classic goal-oriented VLN benchmarks mainly emphasize terminal success. In contrast, our HL-VLA suite requires evaluating process completion, trajectory alignment and safety-aware execution. For process-oriented tasks such as inspection, mapping, and orbiting, we use **Trajectory Coverage Rate (TCR)** to measure how well a predicted trajectory covers the intended ground-truth process. Given a ground-truth trajectory $T^{gt} = \{p_i\}_{i=1}^{N}$ and a predicted trajectory T^{pred} treated as a polyline, we compute:

$$d_i = \min_{q \in T^{pred}} \|p_i - q\|_2. \tag{2}$$

TCR is then defined as:

$$\mathrm{TCR} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(d_i < \delta), \tag{3}$$

where δ is a distance tolerance. We report TCR@K under multiple thresholds, where $K \in \{1\,\mathrm{m}, 2\,\mathrm{m}, 5\,\mathrm{m}\}$. We further report **Avg. TCR**, which averages TCR scores over the evaluated tasks to summarize overall process completion.

Table 2: Overall comparison on the test set. We report both process-oriented metrics and safety metrics across different baseline models. Gray values indicate limited comparative significance due to low performance, e.g., spinning in place.

Methods	Process-oriented Metrics			Safety Metrics	
	Avg. TCR ↑	nDTW ↑	NTP ↑	CR ↓	CSPL ↑
OpenVLA	0.112	0.011	0.122	0.003	0.032
MemoryVLA	0.234	0.077	0.549	0.072	0.226
FastVLM	0.285	0.136	0.633	0.060	0.391
π_0	0.558	0.443	**0.653**	**0.011**	0.802
Depth-aware $\pi_{0.5}$	0.577	0.459	0.617	0.013	0.792
$\pi_{0.5}$	**0.581**	**0.467**	0.618	0.018	**0.805**

Since TCR measures spatial coverage but is less sensitive to temporal ordering, we also report **normalized Dynamic Time Warping (nDTW)** [10]. nDTW penalizes reversed or re-ordered trajectories, and therefore complements TCR by measuring direction-aware trajectory alignment.

We further report **Normalized Task Progress (NTP)** to measure how much of the intended high-level task is completed. Unlike terminal success, which only evaluates whether the final goal is reached, NTP provides a normalized progress score in $[0, 1]$ based on the completion of annotated task stages. This allows us to evaluate partial task completion even when the full trajectory is not successfully completed.

For safety-critical traversal, we compute **Collision Rate (CR)** as the fraction of episodes in which at least one collision occurs, following common practice in collision-aware simulation benchmarks [23]. To jointly reflect goal reaching, collision-free execution, and path efficiency, we further report **Collision-aware SPL (CSPL)** based on Success weighted by Path Length (SPL) [1]:

$$\text{CSPL} = S \cdot (1 - \mathbb{I}(C > 0)) \cdot \frac{L}{\max(P, L)}, \tag{4}$$

where S indicates terminal success, L is the reference path length, and P is the predicted path length. Here, $\mathbb{I}(C > 0)$ is an episode-level collision indicator that equals 1 if any collision occurs and 0 otherwise.

Overall, these metrics capture trajectory alignment and task progress (TCR, nDTW and NTP), and safety-aware efficiency (CR and CSPL), enabling a faithful evaluation of HL-VLA behaviors.

4 Experiments

4.1 Baselines

The HL-VLA task requires strong vision–language understanding, reliable spatial perception, and memory over multi-stage execution. We therefore evaluate

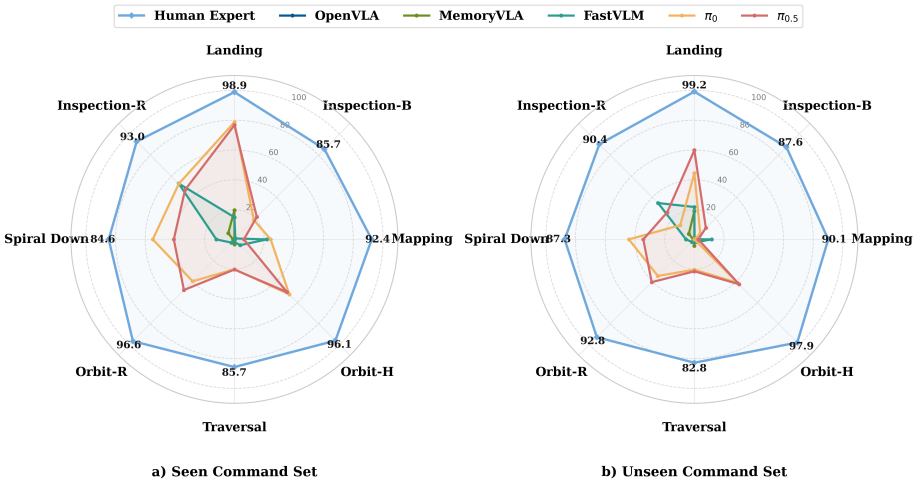

Fig. 6: Per-task nDTW comparison on seen and unseen sets.

representative state-of-the-art VLA models on HUGE-Bench, including OpenVLA [16], π_0 [12], $\pi_{0.5}$ [11], and FastVLM [29]. These baselines cover a trade-off between general-purpose capability and efficiency. For FastVLM, we use FastVLM to efficiently encode visual and language inputs into hidden states, and append π_0's action expert, initialized from π_0, to predict actions.

We additionally evaluate MemoryVLA, an OpenVLA-based baseline with implicit memory, to examine whether memory improves long-horizon HL-VLA execution. To study multimodal spatial perception, we also include a depth-aware $\pi_{0.5}$ baseline, which encodes depth maps as additional visual observations.

Following prior work [30], we provide the model with visual observations from two time steps: the first frame of the trajectory and the current frame. This two-frame input is intended to help the model infer the agent's relative position and progress in 3D space. To ensure a fair comparison, all baselines are evaluated under the same input format and protocol on HUGE-Bench. Additional details on the models, training procedures, and implementation settings are provided in the Appendix.

4.2 Results

Table 2 summarizes the overall performance on the test set. Overall, the π-based policies achieve the strongest performance among the evaluated baselines. $\pi_{0.5}$ obtains the best Avg. TCR, nDTW, and CSPL, suggesting that richer visual pretraining can improve overall trajectory quality and process alignment. π_0 achieves the best NTP and the lowest CR, indicating stronger stage-wise progress and safer execution in our evaluation. FastVLM remains competitive on process-oriented metrics, while OpenVLA performs substantially worse. MemoryVLA

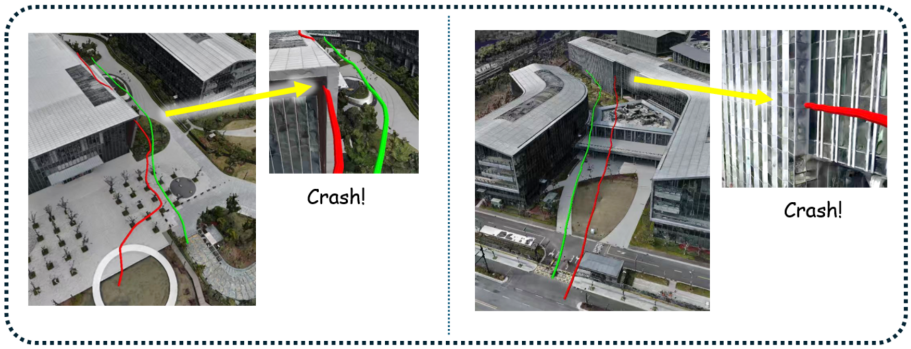

Fig. 7: A visualization of model predictions for Traversal tasks. The green trajectories indicate collision-free results, and the red trajectories denote trajectories that result in collisions.

improves over OpenVLA, confirming the benefit of memory for long-horizon HL-VLA tasks. This supports the need for future memory-augmented VLA models that can better track task progress across multi-stage aerial behaviors.

To evaluate safety behavior, we further study collision-aware metrics enabled by our 3DGS–Mesh digital twin. As shown in Table 2, π_0 achieves the lowest Collision Rate, while $\pi_{0.5}$ obtains the best collision-aware path quality measured by CSPL. The depth-aware $\pi_{0.5}$ baseline reduces CR from 0.018 to 0.013 compared with $\pi_{0.5}$, but does not consistently improve trajectory alignment or overall task progress. These results suggest that richer visual pretraining and explicit depth can improve different aspects of behavior, but neither alone guarantees safe and reliable 3D execution.

Fig. 6 provides a per-task nDTW comparison on seen and unseen splits. Across most tasks, the π-based policies maintain stronger trajectory alignment than OpenVLA and FastVLM, especially on landing and orbiting tasks where landmark recognition and localization are central. The seen–unseen gap is task-dependent: models generally transfer better on structured landmark-centric behaviors, while inspection, mapping, spiral-down, and traversal remain more challenging because they require sustained geometric tracking, long-horizon progress estimation, and safe 3D execution. These results show that HUGE-Bench covers a broad difficulty spectrum rather than only measuring endpoint reaching. To contextualize model performance, we further include expert-pilot evaluation as a human upper bound, and implementation details are provided in the supplementary material.

5 Conclusion

We introduce HUGE-Bench, a benchmark for evaluating high-level UAV vision–language–action capabilities in long-horizon missions. HUGE-Bench targets a

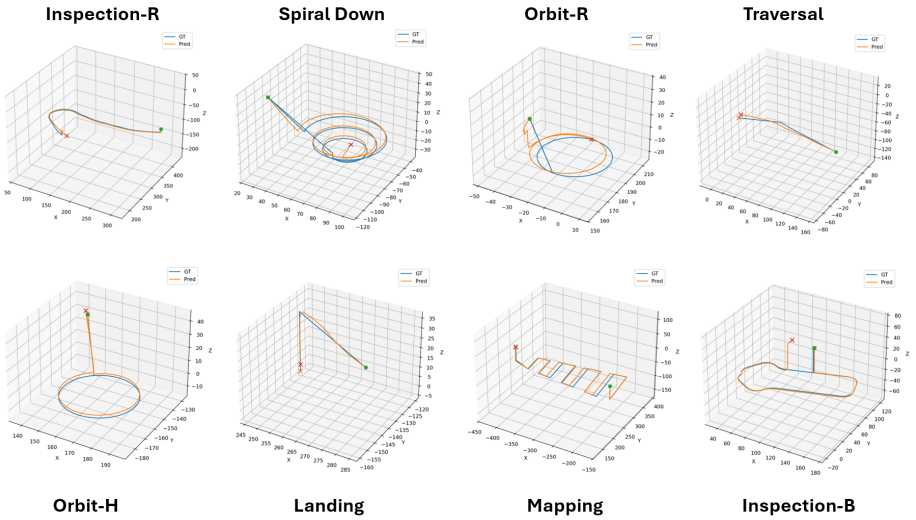

Fig. 8: Visualization of successful results from π_0 across different tasks.

practical operating regime in which a UAV must execute multi-stage tasks from brief, high-intent instructions while maintaining semantic correctness and safety. Unlike conventional UAV-VLN settings that primarily emphasize reaching a destination, our benchmark reflects a more realistic paradigm: short instruction → implicit subtask decomposition → stage-wise execution. To enable fine-grained diagnosis, we incorporate explicit stage annotations and process-oriented metrics to quantify execution progress and stage completion.

To support scalable and cost-effective evaluation, we build an end-to-end data and assessment pipeline that combines real UAV capture with an aligned 3DGS–Mesh digital twin. This hybrid representation leverages the high-fidelity rendering of 3DGS and the collision-aware geometry of meshes, enabling physically actionable simulation with collision-sensitive assessment as well as large-scale data generation. HUGE-Bench further releases richly annotated episodes, including RGB, depth, pose, destination semantics, and subtask stage labels. We also propose a unified evaluation protocol that jointly characterizes stage completion, trajectory efficiency, and collision safety. Experiments with representative state-of-the-art VLA models reveal substantial gaps in high-level semantic stage completion and safe execution, highlighting the value of HUGE-Bench for diagnosing limitations and advancing high-level UAV VLA systems.

Limitations The current benchmark primarily focuses on static environments. Incorporating diverse and realistic dynamics, e.g., moving obstacles, changing illumination and weather, or non-stationary objects, remains an important direction to better reflect real-world deployment conditions. Furthermore, while our digital-twin pipeline improves physical plausibility and enables collision-

aware evaluation, bridging the gap from complex long-horizon high-level VLA behaviors in simulation to robust real-world execution is still challenging. Future work should investigate stronger domain generalization, online adaptation, and hardware-in-the-loop validation to facilitate reliable real-world deployment.

References

1. Anderson, P., Chang, A.X., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., Zamir, A.: On evaluation of embodied navigation agents (2018)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3674–3683 (2018)
3. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. In: International Conference on 3D Vision (3DV) (2017)
4. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
5. Chen, S., Li, Q., Qiu, H., Chen, Y., Wang, W., Xie, W., et al.: 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 (CVPR) (2022)
6. Cheng, A.C., Ji, Y., Yang, Z., Gongye, Z., Zou, X., Kautz, J., Biyik, E., Yin, H., Liu, S., Wang, X.: Navila: Legged robot vision-language-action model for navigation (2024)
7. Gao, Y., Li, C., You, Z., Liu, J., Li, Z., Chen, P., Chen, Q., Tang, Z., Wang, L., Yang, P., Tang, Y., Tang, Y., Liang, S., Zhu, S., Xiong, Z., Su, Y., Ye, X., Li, J., Ding, Y., Wang, D., Wang, Z., Zhao, B., Li, X.: Openfly: A versatile toolchain and large-scale benchmark for aerial vision-language navigation (2025)
8. Gu, J., Zhang, H., Zhao, H., Guo, J., et al.: Vision-and-language navigation: A survey of tasks, methods, and future directions. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL) (2022)
9. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: Towards learning a generic agent for vision-and-language navigation via pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
10. Ilharco, G., Jain, V., Ku, A., Ie, E., Baldridge, J.: General evaluation for instruction conditioned navigation using dynamic time warping (2019)
11. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization (2025). https://doi.org/10.48550/arXiv.2504.16054
12. Intelligence, P., 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., Zhilinsky, U.: π_0: A vision-language-action flow model for general robot control (2024). https://doi.org/10.48550/arXiv.2410.24164
13. Jain, V., Magalhaes, G., Ku, A., Vaswani, A., Ie, E., Baldridge, J.: Stay on the path: Instruction fidelity in vision-and-language navigation (2019)

14. Karaman, S., Walter, M.R., Perez, A., Frazzoli, E., Teller, S.: Anytime motion planning using the rrt. In: 2011 IEEE international conference on robotics and automation. pp. 1478–1483. ieee (2011)
15. Kerbl, B., Kopanas, G., Leimkuhler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023)
16. Kim, M., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An open-source vision-language-action model (2024)
17. Krantz, J., Wijmans, E., Majumdar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: Computer Vision – ECCV 2020. pp. 104–120 (2020)
18. Ku, A., Anderson, P., Patel, R., Ie, E., Baldridge, J.: Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020)
19. Lee, J., Miyanishi, T., Kurita, S., Sakamoto, K., Azuma, D., Matsuo, Y., Inoue, N.: Citynav: Language-goal aerial navigation dataset with geographic information (2024)
20. Liu, S., Zhang, H., Qi, Y., Wang, P., Zhang, Y., Wu, Q.: Aerialvln: Vision-and-language navigation for uavs. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15338–15348 (2023). https://doi.org/10.1109/ICCV51070.2023.01411
21. Miao, B., Wei, R., Ge, Z., Sun, X., Gao, S., Zhu, J., Wang, R., Tang, S., Xiao, J., Tang, R., Li, J.: Sage-3d: Towards physically executable 3d gaussian for embodied navigation (2025)
22. NVIDIA: NVIDIA Isaac Sim. https://docs.isaacsim.omniverse.nvidia.com/latest/index.html (2025), version 5.1.0. GitHub: https://github.com/isaac-sim/IsaacSim. Accessed: 2026-03-06
23. Puig, X., Undersander, E., Szot, A., Dallaire Cote, M., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y., et al.: Habitat 3.0: A co-habitat for humans, avatars and robots. In: International Conference on Learning Representations (ICLR) (2024)
24. Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W.Y., Shen, C., van den Hengel, A.: Reverie: Remote embodied visual referring expression in real indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9982–9991 (2020)
25. Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J., Undersander, E., Galuba, W., Westbury, A., Chang, A., Savva, M., Zhao, Y., Batra, D.: Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai (2021), https://openreview.net/forum?id=-v4OuqNs5P, neurIPS Datasets and Benchmarks Track
26. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A platform for embodied ai research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9339–9347 (2019)
27. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics (2017)
28. Thomason, J., Murray, M., Cakmak, M., Zettlemoyer, L.: Vision-and-dialog navigation. In: Proceedings of the 37th International Conference on Machine Learning (ICML) (2020)

29. Vasu, P.K.A., Faghri, F., Li, C., Koc, C., True, N., Antony, A., Santhanam, G., Gabriel, J., Grasch, P., Tuzel, O., Pouransari, H.: Fastvlm: Efficient vision encoding for vision language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19769–19780 (June 2025)
30. Wang, X., Yang, D., Liao, Y., Zheng, W., Wu, W., Dai, B., Li, H., Liu, S.: Uav-flow colosseo: A real-world benchmark for flying-on-a-word uav imitation learning (2025)
31. Wang, X., Yang, D., Wang, Z., Kwan, H., Chen, J., Wu, W., Li, H., Liao, Y., Liu, S.: Towards realistic uav vision-language navigation: Platform, benchmark, and methodology (2024)
32. Zhang, J., Wang, K., Wang, S., Li, M., Liu, H., Wei, S., Wang, Z., Zhang, Z., Wang, H.: Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. In: Robotics: Science and Systems (RSS) (2025)
33. Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., Wang, H.: Navid: Video-based vlm plans the next step for vision-and-language navigation. In: Robotics: Science and Systems (RSS) (2024)