WAM-BENCH: A CLOSED-LOOP BENCHMARK FOR OPEN-ENDED INSTRUCTION FOLLOWING IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

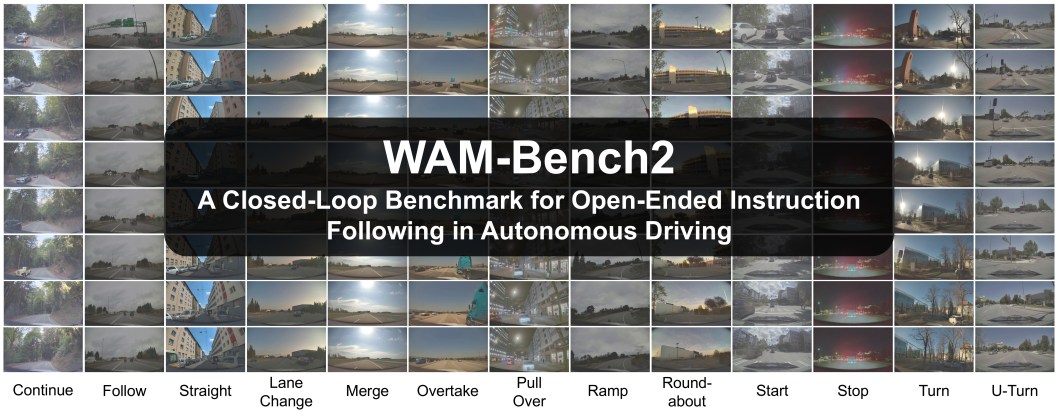


Figure 1: WAM-Bench is the first real-world, closed-loop autonomous driving benchmark for open-ended instruction following.

ABSTRACT

Recently, Vision-Language-Action (VLA) models have shown great potential in advancing end-to-end autonomous driving (E2E-AD). However, existing E2E-AD benchmarks typically provide only two predefined instructions (go straight and turn), leaving the ability of VLA models to follow human instructions largely unexplored. To address this limitation, we introduce WAM-Bench, a real-world closed-loop benchmark for evaluating open-ended instruction following in E2E-AD. It contains 37 hours of driving videos paired with human-reviewed open-ended instructions and covers 13 action classes. We further establish a comprehensive evaluation framework with representative VLA baselines, and propose an instruction-following metric (IF) for closed-loop evaluation. Extensive experiments across diverse settings, including the standard test split (with various action classes), long-horizon execution, changed instructions, and OOD instructions, reveal that open-ended instruction following remains challenging for current VLA models. The code is available at -.

1 INTRODUCTION

Vision-Language-Action (VLA) models Zhou et al. (2025a); Han et al. (2025); Zhou et al. (2025b); Xu et al. (2025c); Wang et al. (2026a) have recently attracted widespread attention in end-to-end autonomous driving (E2E-AD) due to their powerful generalization capabilities. E2E-AD aims to take driving observations, ego states, and linguistic instructions as input and generate future trajectories.

However, existing E2E-AD benchmarks Karnchanachari et al. (2024); Dauner et al. (2024); Xu et al. (2025b) mainly evaluate predefined instructions (e.g., go straight and turn), where language inputs only specify a limited set of actions. Consequently, current VLA models are often reduced to Vision-Action (VA) models rather than truly instruction-driven systems. For example, DiffusionDrive Liao et al. (2025) and RecogDrive Li et al. (2025) represents predefined instructions as one-hot encoded

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



Figure 2: An example of instruction-driven autonomous driving, where the ego-vehicle trajectory is predicted by the trained baseline model (Alpamayo-1.5 NVIDIA et al. (2025b)).

Dataset	Open-Source	Real	Closed-Loop	Open-Ended Instruction	Action Class	Action Subclass	Duration (h)
Intention-Drive Zheng et al. (2025)	✗	✓	✗	✓	5	8	5.5
InstructScene Zuo et al. (2026)	✗	✓	✗	✓	5	6	40
Bench2Drive Jia et al. (2024)	✓	✗	✓	✗	4	6	-
nuScenes Caesar et al. (2020)	✓	✓	✗	✗	2	3	5.5
NAVSIM Dauner et al. (2024)	✓	✓	✗	✗	2	3	40
WOD-E2E Xu et al. (2025b)	✓	✓	✗	✗	2	3	12
PAI-AV NVIDIA Corporation (2025)	✓	✓	✗	✗	-	-	1700
PAI-AV-nuRec Wu et al. (2025)	✓	✓	✓	✗	-	-	9
WAM-Bench (Ours)	✓	✓	✓	✓	13	21	37

Table 1: Comparison with existing datasets. WAM-Bench is a real-world, closed-loop autonomous driving benchmark for evaluating open-ended instruction following. It comprises 37 hours of driving data with 13 high-level action classes and 21 fine-grained action subclasses.

action classes, which cannot express free-form instructions such as "pull over in front of the first vehicle on the right side." (see segment 2 in Figure 2).

To address the above issues, we introduce WAM-Bench, the first closed-loop E2E-AD benchmark for open-ended instruction following. It contains 37 hours of real-world driving videos paired with human-annotated open-ended instructions. Following driving maneuver definitions in navigation systems Google (2026), we categorize instructions into 13 action classes (Figure 1) and 21 action subclasses, enabling systematic analysis of instruction-following capabilities. Furthermore, we propose a straightforward instruction-following metric (IF) to evaluate whether the generated trajectory satisfies the given instruction. We establish a comprehensive benchmark evaluation framework with representative VLA baselines (such as Qwen3.5 Qwen Team (2026a), Alpamayo-1.5 NVIDIA (2026)) We further conduct extensive experiments under diverse settings, including different action classes, long-horizon execution, changed instructions, and out-of-distribution (OOD) instructions, to comprehensively investigate the challenges of open-ended instruction following.

Our contributions are summarized as follows:

- To our best knowledge, we propose the first real-world closed-loop benchmark for evaluating open-ended instruction following in autonomous driving, WAM-Bench, consisting of 37 hours of driving data and 13 action classes.
- We develop a comprehensive benchmark framework with representative VLA baselines, incorporating the proposed instruction-following metric and existing closed-loop planning metrics for evaluation.
- We perform systematic evaluations across various scenarios, including action classes, long-horizon execution, changed instructions, and OOD instructions, providing insights into the limitations of current VLA models for instruction-driven driving.

2 Related Work

With the rapid development of open-source foundation models Qwen Team (2026b); Chen et al. (2026); DeepSeek-AI et al. (2025), VLA models Xu et al. (2025a); Zhu et al. (2026); He et al. (2026) have achieved promising performance in E2E-AD. VLA models aim to integrate visual perception, language reasoning, and action generation within a unified framework. Despite this progress, existing E2E-AD benchmarks Ettinger et al. (2021); Arai et al. (2025); Chi et al. (2025); Cao et al. (2025) typically provide only a limited set of predefined instructions (e.g., go straight and turn), as summarized in Table 1. Recent results on benchmarks such as NAVSIM Dauner et al. (2024) even show that advanced VLA models Xu et al. (2026); Wang et al. (2026b) can surpass human

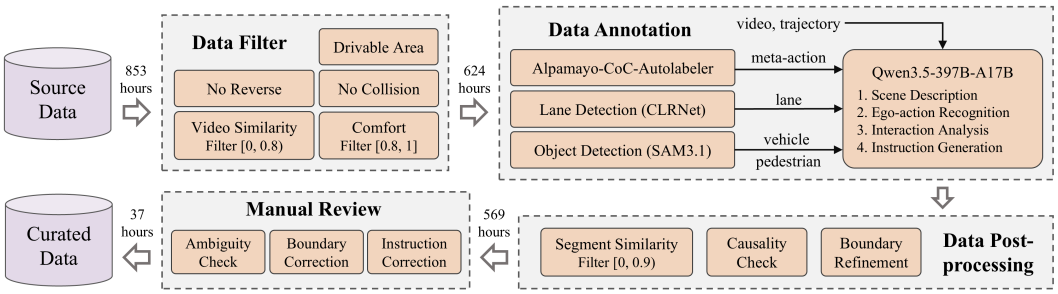


Figure 3: The data processing pipeline. Starting from source data, we perform data filtering, annotation, post-processing, and manual review to construct curated driving segments paired with open-ended instructions.

performance, suggesting that existing evaluation paradigms have become insufficient for assessing more challenging instruction-driven autonomous driving capabilities. In addition, the ability of VLA models to understand and follow free-form human instructions remains largely unexplored. Several recent works Zheng et al. (2025); Zuo et al. (2026) have started to investigate instruction-driven E2E-AD, but they mainly rely on closed-source and open-loop settings, making it difficult to evaluate whether models can truly execute instructions in closed-loop driving. To address these limitations, we introduce WAM-Bench, a real-world closed-loop benchmark for evaluating open-ended instruction following in E2E-AD.

3 Dataset

3.1 Source Data

The source data consist of two datasets: the PAI-AV NVIDIA Corporation (2025) training set (853 hours) and PAI-AV-nuRec Wu et al. (2025) (9 hours). Both provide front-view camera videos (30 Hz), ego states (10 Hz), such as trajectory, velocity and acceleration. PAI-AV-nuRec is a subset of PAI-AV that includes HD maps, enabling closed-loop evaluation. In contrast, PAI-AV lacks HD maps and is therefore unsuitable for closed-loop evaluation. Accordingly, we process PAI-AV and PAI-AV-nuRec to construct the training and test data for WAM-Bench, respectively.

3.2 Data Processing

Data Filter. Inspired by NAVSIM Dauner et al. (2024), we first discard cases with a comfort score below 0.8. For PAI-AV-nuRec, we remove cases in which the ego vehicle leaves the drivable area or is involved in a collision. Then, we exclude reversing cases, since both the source videos and the closed-loop renderer provide only front-view observations, making reverse maneuvers insufficiently observable. Finally, we observe that some videos contain highly similar or duplicated scenes. To prevent test data leakage, we use Qwen3-VL-Embedding-8B Li et al. (2026) to measure video similarity between PAI-AV and PAI-AV-nuRec, and remove training videos with a similarity of 0.9 or higher to any test video.

Data Annotation. We leverage CoC-Autolabeler NVIDIA Corporation (2026) to derive meta-actions (e.g., go straight and steer left/right) from ego states based on kinematic parameters. The detailed parameter settings for meta-actions can be found in Alpamayo-R1 NVIDIA et al. (2025b). Then, consecutive frames with the same meta-action are grouped into candidate video segments.

We apply CLRNet Zheng et al. (2022) to detect lane markings and SAM 3.1 Carion et al. (2025) to identify traffic participants (such as vehicles and pedestrians) and static objects. The video, ego states, and other generated annotations (meta-actions, candidate segments, lane information, and detected objects) are then fed into Qwen3.5-397B-A17B Qwen Team (2026a). The model sequentially describes the scene, recognizes the ego action, and analyzes interactions with surrounding traffic participants. Finally, it consolidates these annotations into a concise driving instruction.

Data Post-processing. We use Qwen3.5-397B-A17B to merge short segments and refine their temporal boundaries, ensuring that each final segment is at least 5 seconds long. Segments that cannot be merged are discarded. For the training data, we then perform a causality check (Qwen3.5-397B-A17B) to ensure that the target location or traffic participant referenced in the instruction is observable

	Action	Action Subclass	Duration (h)	Segments	Instruction Example
	Continue	-	3	1.2k	Continue along the road.
	Follow	-	3.1	1.2k	Follow the vehicle ahead.
	Go Straight	-	3.2	1.4k	Go straight through the intersection.
	Lane Change	Lane Change Left/Right	2.7	1.1k	Change lane left.
	Merge	Merge Left/Right	3.1	1.2k	Merge left onto the main road.
	Overtake	Overtake Left/Right	2.8	0.9k	Overtake the vehicle ahead on the left.
	Pull Over	-	0.5	0.2k	Pull over and stop behind the white vehicle.
	Ramp	Enter/Exit/Along Ramp	3.1	1.2k	Enter the ramp and continue along it.
	Roundabout	Enter/Exit/Along Roundabout	3.2	1.3k	Enter the roundabout and take the second exit.
	Start Moving	-	3.1	1.2k	Start moving and continue along the road.
	Stop	-	3	1.3k	Slow down and then stop behind the truck.
	Turn	Turn Left/Right	3.7	1.5k	Turn right at the intersection.
	U-Turn	-	2.5	0.9k	Perform a U-turn at the intersection.
All	**13 Classes**	**21 Subclasses**	**37**	**14.4k**	

Table 2: Statistics of WAM-Bench, including action classes, action subclasses, duration, and number of video segments. The dataset contains 13 action classes, 21 action subclasses, 37 hours of driving data, and 14.4k video segments. "-" indicates that the action class itself serves as the subclass.

in the historical frames. Finally, we utilize Qwen3-VL-Embedding-8B to compute video-segment similarity and remove training segments with a similarity of 0.9 or higher to any test segment.

Manual Review. We manually review the annotations to correct inaccurate instructions and video segment boundaries. We further verify instruction causality and ambiguity to ensure that the referenced targets and intended actions are clearly specified.

For systematic evaluation and analysis, inspired by the maneuver taxonomy used in Google Maps Navigation Google (2026), we annotate each instruction with its corresponding action class and subclass during manual review. We remove action classes with less than 0.1 hours of data, as well as those that do not appear in the test data. To mitigate action-class imbalance, we further subsample the curated segments to keep the duration of different action classes approximately balanced. Finally, we obtain a 37-hour dataset with approximately balanced coverage across action classes.

3.3 Curated Data

The curated dataset (WAM-Bench) contains driving videos (37 hours, 2 Hz, 1920×1080) and ego states (10 Hz), with each video segment paired with an open-ended, free-form language instruction. For systematic evaluation and analysis, each instruction is manually assigned to one of 13 action classes and 21 action subclasses.

Table 2 summarizes the total duration of each action class and provides representative instruction examples. For action classes with less than 3 hours of data, the reported duration reflects the maximum amount that can be mined from the source data after data processing. More detailed dataset statistics and analyses are presented in Appendix A.

3.4 Data Split

We utilize the training data (36 hours) in WAM-Bench as the *train* split. The remaining test data (1 hour) are divided into four splits: *test*, *long-horizon*, *changed instructions*, and *OOD instructions*, all of which are used for closed-loop evaluation.

Train Split. The train split contains 36 hours of driving videos and is converted into 100k training samples using a sliding window with a 0.5-second stride.

Test Split. The test split evaluates VLA models across different action classes. It contains 233 video segments in total, as shown in Table 4. Only 8 pull-over and 5 U-turn segments can be mined from the entire test data. For the remaining action, we sample 20 segments per class to maintain the balanced action distribution.

Long-Horizon Split. The long-horizon split evaluates the instruction following over extended time horizons. It contains 63 video segments with durations longer than 15 seconds. The split consists of two types of scenarios: segments directly selected from the test split with a single instruction, and consecutive segments concatenated from the test split containing multiple instructions.

Changed Instructions Split. The changed-instruction split evaluates the model's ability to adapt to modified instructions within the same scene. It contains 40 video segments, where the original instructions are manually changed and the corresponding trajectories are collected using an external steering wheel.

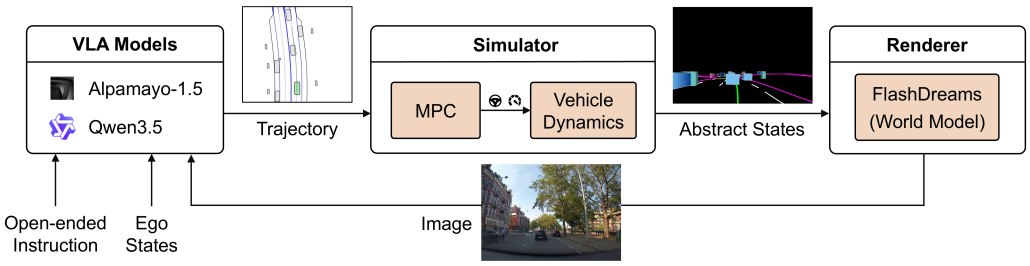


Figure 4: The framework of closed-loop evaluation in WAM-Bench. VLA predicts a future ego trajectory from an open-ended instruction, historical ego states, and front-view images. The simulator converts the trajectory into abstract states, and the renderer generates the next front-view image. The rendered image is fed back to the VLA every 0.5 seconds, forming a closed loop.

OOD Instructions Split. The out-of-distribution (OOD) instruction split evaluates generalization to unseen actions. We mine 10 video segments from the test data, including slight turn (i.e., bear left/right) and pull into, which are not included in the predefined instruction classes.

4 Benchmark

4.1 Closed-Loop Framework

We build the closed-loop evaluation framework upon the existing simulator (AlpaSim NVIDIA et al. (2025a)) and renderer (FlashDreams FlashDreams Contributors (2026)), as illustrated in Fig. 4. In addition, we provide Qwen3.5 series and Alpamayo-1.5 as VLA baselines.

At each step, the VLA model takes an open-ended instruction, ego states from the past 1.6 seconds (10 Hz), and four front-view images (2 Hz), and predicts the ego trajectory for the next 5 seconds (10 Hz). The simulator uses Model Predictive Control (MPC) Mayne et al. (2000) to convert the predicted trajectory into front-wheel steering angles and longitudinal accelerations, which are integrated by the vehicle dynamics model Rajamani (2012) to update the ego pose. The ego pose, surrounding-vehicle poses replayed from recorded data, and map information are then converted into abstract states.

The renderer uses the world model (OmniDreams Basant et al. (2026) trained on PAI-AV dataset) to generate the next front-view image from the abstract states and historical images. The rendered image is fed back to the VLA model, and the process repeats every 0.5 seconds, forming the closed-loop.

4.2 Metrics

We evaluate VLA models from three aspects: trajectory error, planning quality, and instruction following.

Trajectory Error. We utilize Average Displacement Error (ADE) to measure trajectory error. ADE is widely adopted in previous benchmarks such as nuScenes Caesar et al. (2020), nuPlan Karnchanachari et al. (2024), and WOD-E2E Xu et al. (2025b).

Planning Quality. Following NAVSIM Dauner et al. (2024) and Alpamayo-R1 NVIDIA et al. (2025b), we adopt Drivable Area Compliance (DAC), No Collision (NC), Comfort (CF), and Ego Progress (EP) to evaluate planning quality. DAC, NC, and CF are binary metrics indicating whether the ego vehicle stays within the drivable area, avoids collisions, and satisfies kinematic comfort constraints, respectively. EP measures driving progress by projecting the final ego position onto the ground-truth trajectory and computing the corresponding arc-length ratio.

Instruction Following. To explicitly evaluate whether the executed trajectory follows the given instruction, we propose an Instruction Following (IF) metric. Let $\mathcal{B}_{\text{end}}^{\text{GT}}$ denote the ego vehicle BBox (bounding box) at the endpoint of GT trajectory, and $\mathcal{B}_t$ denote the predicted ego BBox at time step t. IF@ϵm is defined as

$$\text{IF@}\epsilon\text{m} = \mathbb{I}\left[\min_t d(\mathcal{B}_t, \mathcal{B}_{\text{end}}^{\text{GT}}) \leq \epsilon\right], \tag{1}$$

where $d(\cdot,\cdot)$ denotes the polygon distance between two BBoxes, and $\mathbb{I}[\cdot]$ is the indicator function. A sample is considered successfully followed if the predicted ego BBox reaches the ϵ-meter neighborhood of the GT endpoint BBox during closed-loop execution. For actions involving stationary states (such as stop and pull over), we use the polygon distance between the final predicted BBox and the

Method	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
	0m	1m	2m	DAC	NC	CF	EP	ADE
Human	100	100	100	100	100	100	100	0.0
Constant Velocity	19.7	26.6	28.8	52.9	72.1	100	76.7	11.7
Predefined Instruction								
RecogDrive-8B Li et al. (2025)	10.8	15.3	23.5	30.9	67.8	84	72.4	13.3
DiffusionDrive Liao et al. (2025)	18.9	21.5	26.2	41.3	81.5	95.7	84.6	7.7
Open-ended Instruction								
Qwen3.5-2B Qwen Team (2026a)	31.8	41.2	47.6	70.7	81.1	97.9	91.4	4.8
Qwen3.5-4B Qwen Team (2026a)	40.3	47.6	53.6	66.7	85.8	92.3	92.7	5.0
Alpamayo-1.5 NVIDIA et al. (2025b)	62.2	69.1	76	91.1	90.6	95.3	95	4.1

Table 3: Quantitative comparison of planning models on the WAM-Bench test split. *Predefined Instruction* denotes methods that use one-hot encoded action classes as instructions. The best results for each metric are highlighted.

GT endpoint BBox, which better reflects the achievement of the target state. We report IF@ϵm with $\epsilon \in \{0, 1, 2\}$ meters. IF@0m indicates that the predicted ego BBox overlaps with or contacts the GT endpoint BBox, while larger thresholds allow larger deviations from the target position.

5 Experiments

5.1 Experimental Setups

For training, the model takes 1.6-second historical observations as input, including four front-view images at timestamps $[-1.5, -1.0, -0.5, 0]$s and 16 ego states (10 Hz). The future 5-second ego trajectory (10 Hz) is used as supervision. For evaluation, we follow the metrics in Section 4.2. In addition, DAC is excluded for pull-over and pull-into actions, as the GT ego vehicle leave the drivable area to reach the designated stopping location. EP is excluded for stop actions, as GT progress is undefined when the vehicle remains stationary for the entire trajectory.

We benchmark representative methods, including AR-based (Qwen3.5 series Qwen Team (2026a)), diffusion-based (DiffusionDrive Liao et al. (2025)), and AR-diffusion hybrid (RecogDrive Li et al. (2025), Alpamayo-1.5 NVIDIA et al. (2025b)). For Alpamayo-1.5, we train the VLM and action decoder for 5 and 3 epochs on 16 H200 GPUs, respectively, for a total of 15 hours. Evaluation uses 8 H200 GPUs, taking 1 hour on the test split and 50 minutes in total on the other three splits.

5.2 Benchmark Evaluation

In this section, we first report the quantitative results on the WAM-Bench test split. Then, we analyze the impact of action classes and traffic density on baseline model (Alpamayo-1.5).

Test Split. Table 3 presents the quantitative comparison on the WAM-Bench test split. Human denotes the GT trajectory. Constant Velocity propagates the ego trajectory using the initial velocity. The results show that DiffusionDrive and RecogDrive achieve limited performance, as they only support *Predefined Instruction* in the form of one-hot encoded action subclasses. Such representations are insufficient for open-ended instruction following.

In contrast, Qwen3.5 and Alpamayo-1.5 achieve better performance, as they support *Open-ended Instruction*. Among them, Alpamayo-1.5 achieves the best performance on most metrics. It obtains IF scores of 62.2, 69.1, and 76.0 at 0m, 1m, and 2m thresholds, respectively, achieving a clear advantage in instruction following. Moreover, it achieves the highest DAC, NC, and EP scores of 91.1, 90.6, and 95.0, respectively, demonstrating effective instruction-driven autonomous driving.

Action Class. Table 4 reports the detailed performance of Alpamayo-1.5 across different action classes on the test split. Given the large number of action classes, we focus on representative classes with clear performance trends and failure patterns.

For basic actions such as *Continue*, *Follow*, and *Stop*, the model achieves strong instruction-following performance, with IF@0m $\geq$ 80 and IF@2m $\geq$ 90. However, *Stop* obtains an NC score of 85. Inspection of the failure cases shows that the *Stop* samples with NC$=0$ are consistently associated with insufficient deceleration, leading to collisions with the vehicle ahead, as shown in Fig. 5(a).

For *Pull Over*, *Overtake*, *Lane Change*, *Ramp*, *Go Straight*, *Merge*, *Roundabout*, and *Start Moving*, Alpamayo-1.5 achieves moderate instruction-following performance, with IF@0m $\geq$ 50. Among them, *Merge* has a relatively low DAC score of 70. Inspection of the failure cases shows that most

Action	Num	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
		0m	1m	2m	DAC	NC	CF	EP	ADE
Stop	20	85	90	90	95	85	95	-	1.3
Continue	20	80	90	90	95	100	100	98.8	3.9
Follow	20	80	85	90	100	100	100	97.8	2.3
Pull Over	8	75	75	87.5	-	100	100	99.2	2.1
Overtake	20	70	75	85	100	100	95	99.1	5.5
Lane Change	20	60	75	85	95	95	95	99.2	3.6
Ramp	20	60	65	80	100	95	90	96.3	4.5
Go Straight	20	55	60	70	95	95	100	95.4	3.1
Merge	20	55	65	70	70	100	90	94.9	3.9
Roundabout	20	55	55	70	75	90	100	95.0	5.0
Start Moving	20	55	60	65	95	85	95	91.6	2.6
Turn	20	35	50	50	80	50	90	85.5	7.7
U-Turn	5	20	20	20	100	80	80	75.5	14.4
All	233	62.2	69.1	76	91.1	90.6	95.3	95	4.1

Table 4: Performance across different action classes on the WAM-Bench test split, sorted by IF@0m in descending order. The best and worst results for each metric are highlighted.

Traffic Density	Num	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
		0m	1m	2m	DAC	NC	CF	EP	ADE
Low (0)	56	48.2	60.7	67.9	87.5	100	91.1	94.8	4.5
Medium (1–5)	76	59.2	64.5	73.7	90.4	96.1	97.4	97.3	4.7
High (≥6)	101	72.3	77.2	82.2	93.8	81.2	96.0	93.2	3.4
All	233	62.2	69.1	76.0	91.1	90.6	95.3	95.0	4.1

Table 5: Performance across different traffic densities on the WAM-Bench test split.

Merge samples with DAC= 0 occur when the model drives into the non-drivable gore area between the ramp and the main road during merging, as shown in Fig. 5(b).

In contrast, *Turn* and *U-Turn* show substantially lower instruction-following performance, with IF@0m below 50. Both actions require accurate intersection understanding and large heading changes. For *Turn*, the low IF and NC are mainly associated with challenging interactions with surrounding traffic at intersections; Fig. 5(c) shows a failed left-turn case. *U-Turn* achieves the lowest IF score among all action classes. Inspection of the failure cases shows that the model often continues straight through the intersection instead of executing the U-turn, as shown in Fig. 5(d).

Representative successful examples (with IF@0m = DAC = NC = 1) for the above four actions are shown in Fig. 6. Detailed visualizations with evaluation metrics, BEV trajectories, and front-view images are provided in Appendix C. More closed-loop visualization videos are provided in the **supplementary material**

Traffic Density. Table 5 analyzes the impact of traffic density, categorized by surrounding vehicles within 20 m: Low (0), Medium (1–5), and High (≥6). We observe that instruction following improves as traffic density increases. Specifically, IF@0m increases from 48.2 in low-density scenes to 72.3 in high-density scenes, suggesting that richer surrounding contexts may provide more informative cues for instruction execution. Similarly, DAC improves from 87.5 to 93.8 with increasing traffic density. However, NC decreases from 100 to 81.2, indicating that denser traffic introduces more potential interactions and collision risks.

5.3 Generalization Evaluation

Long-Horizon Split. The top part of Table 6 reports the results on the WAM-Bench long-horizon split. Compared with the test split (Table 3), all VLA models show a clear performance degradation over longer closed-loop horizons, indicating that long-horizon instruction following remains challenging and deserves further attention in future research. For Alpamayo-1.5, IF@0m decrease from 62.2 to 31.7. Nevertheless, it achieves the best overall performance, showing relatively stronger long-horizon instruction execution capability. Fig. 7 shows two successful examples: (a) the ego vehicle follows the vehicle ahead through the roundabout; (b) it follows a truck and then overtakes it.

Split	Method	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
		0m	1m	2m	DAC	NC	CF	EP	ADE
Long-Horizon	Constant Velocity	4.8	6.3	7.9	36.1	58.7	100	68.1	35.4
	Qwen3.5-2B	11.1	12.7	15.9	27.9	55.6	93.7	84.6	18.2
	Qwen3.5-4B	15.9	20.6	28.6	31.1	73.0	88.9	85.4	16.2
	Alpamayo-1.5	31.7	36.5	47.6	83.6	81.0	93.7	89.3	13.6
Changed Instructions	Constant Velocity	10.0	17.5	25.0	44.7	72.5	100	75.0	11.3
	Qwen3.5-2B	32.5	42.5	62.5	52.6	65.0	97.5	91.2	9.6
	Qwen3.5-4B	27.5	35.0	52.5	52.6	72.5	95.0	91.1	8.6
	Alpamayo-1.5	40.0	42.5	52.5	78.9	77.5	90.0	83.1	12.2
OOD Instructions	Constant Velocity	10.0	10.0	20.0	44.4	70.0	100	70.1	9.0
	Qwen3.5-2B	40.0	40.0	40.0	66.7	80.0	100	86.3	2.9
	Qwen3.5-4B	20.0	20.0	20.0	66.7	60.0	100	87.6	3.4
	Alpamayo-1.5	40.0	60.0	60.0	77.8	90.0	100	91.4	2.7

Table 6: Quantitative comparison on different data splits of WAM-Bench.

Action	Num	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
		0m	1m	2m	DAC	NC	CF	EP	ADE
Pull Into	1	0	0	0	-	100	100	57.8	9.9
Slight Turn	9	44.4	66.7	66.7	77.8	88.9	100	95.2	1.9
All	10	40	60	60	77.8	90	100	91.4	2.7

Table 7: Performance across different action classes on the WAM-Bench OOD instructions split.



Figure 5: Representative failure cases of Alpamayo-1.5 on the WAM-Bench test split. The **red** texts denote the input open-ended instructions, while the **orange** texts and boxes indicate the corresponding failure reasons and regions.



Figure 6: Representative successful instruction-following cases on the WAM-Bench test split.

Changed Instructions Split. The middle part of Table 6 reports the results on the changed-instructions split. Compared with the test split, Alpamayo-1.5 shows a clear drop in instruction following, with IF@0m decreasing from 62.2 to 40.0, indicating that adapting the driving trajectory to changed instructions remains challenging. Fig. 8 shows qualitative examples in the same scene,

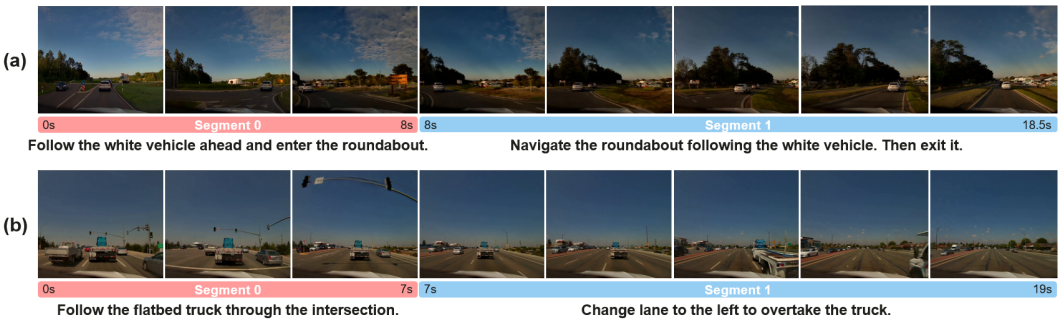


Figure 7: Visualization of successful long-horizon instruction-following cases.

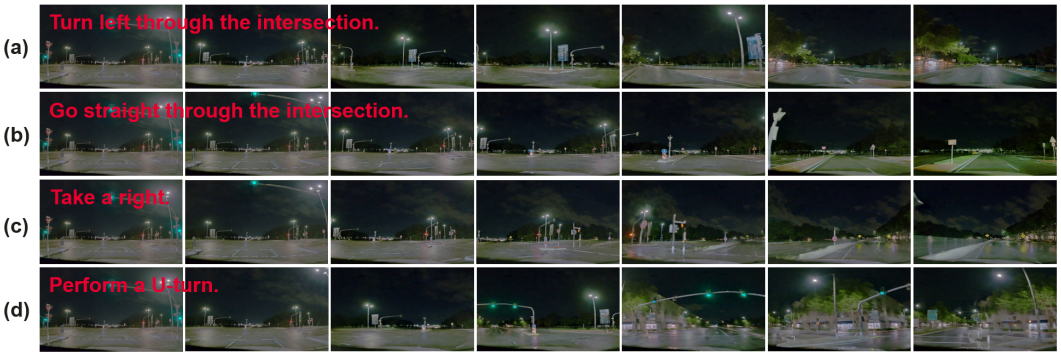


Figure 8: Visualization of instruction adaptation in the WAM-Bench changed-instructions split. Under the same driving scene, Alpamayo-1.5 generates different trajectories according to different open-ended instructions, including turn left, go straight, turn right, and U-turn.



Figure 9: Visualization on the WAM-Bench OOD instructions split. (a)(b) Successful slight-turn cases (IF@0m=1). (c)(d) A failure case of pull-into: Alpamayo-1.5 predicts a left turn in (c), while the ground-truth trajectory stops between two vehicles in (d).

where the model successfully executes *turn left*, *go straight*, *turn right*, and *U-turn* under different instructions, demonstrating its ability to adapt the driving trajectory to instruction changes.

OOD Instructions Split. The bottom part of Table 6 reports the results on the OOD instructions split. Alpamayo-1.5 achieves the best performance across all metrics, with an IF@0m score of 40.0, indicating a certain degree of generalization to unseen instruction patterns. Table 7 further reports the results for *Slight Turn* (bear left/right) and *Pull Into*. Fig. 9(a) and (b) show two successful *Slight Turn* cases with IF@0m$=1$. For *Pull Into*, Fig. 9(c) shows the failure case where the model predicts a left turn, whereas the GT trajectory in Fig. 9(d) stops between two vehicles.

6 Conclusion

In this work, we introduce WAM-Bench, a closed-loop benchmark for open-ended instruction-driven autonomous driving. We further propose the IF metric to measure instruction-following capability. Through extensive experiments, we show that open-ended instruction following remains challenging for current VLA models. Our analysis reveals that performance varies significantly across different action classes, especially for maneuvers involving large heading changes and interactions with surrounding vehicles. Furthermore, long-horizon execution, instruction changes, and OOD instructions expose the limitations of current models in maintaining consistent instruction execution. More discussions can be found in **appendix**. More visualization results can be found in the **appendix** and **supplementary materials**.

AI USE STATEMENT

In this work, we use GPT-5.6 for language polishing of the manuscript, GLM-5.3 for code assistance, and Qwen3.5 for data annotation assistance. All AI-assisted outputs were reviewed and verified by the authors, and we take full responsibility for the final content of this work.

ETHICS STATEMENT

This study does not involve human-subject experiments or the collection of new sensitive personal information. All data are obtained from publicly available autonomous-driving datasets.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our work. We release the dataset, source code, and model weights to the open-source community to facilitate the development of open-ended instruction-following autonomous driving.

REFERENCES

Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *WACV*, pp. 1933–1943. IEEE, 2025.

Aarti Basant, Amlan Kar, Despoina Paschalidou, Fangyin Wei, Francesco Ferroni, Guillermo Garcia Cobo, Haithem Turki, Huan Ling, Jaewoo Seo, James Lucas, et al. Nvidia omnidreams: Real-time generative world model for closed-loop autonomous vehicle simulation. *arXiv preprint arXiv:2606.03159*, 2026.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving, May 2020.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-Simulation for Autonomous Driving, June 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Baoyou Chen, Hanchen Xia, Peng Tu, Haojun Shi, Shan Mu, Weihao Yuan, and Siyu Zhu. Bard: Bridging autoregressive and diffusion vision-language models via highly efficient progressive block merging and stage-wise distillation. *arXiv preprint arXiv:2604.16514*, 2026.

Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, Leichen Wang, Xingtao Hu, Hao Sun, Hang Zhao, and Hao Zhao. Impromptu VLA: Open Weights and Open Data for Driving Vision-Language-Action Models, May 2025.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, and Marco Pavone. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *NeurIPS*, 37:28706–28719, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, January 2025.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, and Yin Zhou. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *ICCV*, pp. 9710–9719, 2021.

FlashDreams Contributors. Flashdreams: High-performance inference and serving for interactive autoregressive video and world models. https://github.com/NVIDIA/flashdreams, 2026.

Google. Google maps platform routes api: Maneuver. https://developers.google.com/maps/documentation/routes/reference/rest/v2/TopLevel/computeRoutes, 2026.

Jianhua Han, Meng Tian, Jiangtong Zhu, Fan He, Huixin Zhang, Sitong Guo, Dechang Zhu, Hao Tang, Pei Xu, Yuze Guo, Minzhe Niu, Haojie Zhu, Qichao Dong, Xuechao Yan, Siyuan Dong, Lu Hou, Qingqiu Huang, Xiaosong Jia, and Hang Xu. Percept-WAM: Perception-Enhanced World-Awareness-Action Model for Robust End-to-End Autonomous Driving, November 2025.

Zhuolin He, Jing Li, Guanghao Li, Xiaolei Chen, Jiacheng Tang, Siyang Zhang, Zhounan Jin, Feipeng Cai, Bin Li, Jian Pu, et al. Dynamicvggt: Learning dynamic point maps for 4d scene reconstruction in autonomous driving. *arXiv preprint arXiv:2603.08254*, 2026.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *NeurIPS*, 37:819–844, 2024.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, and Yiluan Guo. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *ICRA*, pp. 629–636. IEEE, 2024.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Chen Keqin, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A Reinforced Cognitive Framework for End-to-End Autonomous Driving, June 2025.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. DiffusionDrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, 2025.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Autom.*, 36:789–814, 2000. URL https://api.semanticscholar.org/CorpusID:8363289.

NVIDIA. Alpamayo 1.5. https://github.com/NVlabs/alpamayo1.5, 2026. GitHub repository.

NVIDIA, Yulong Cao, Riccardo de Lutio, Sanja Fidler, Guillermo Garcia Cobo, Zan Gojcic, Maximilian Igl, Boris Ivanovic, Peter Karkus, Janick Martinez Esturo, Marco Pavone, Aaron Smith, Ellie Tanimura, Michal Tyszkiewicz, Michael Watson, Qi Wu, and Le Zhang. Alpasim: A modular, lightweight, and data-driven research simulator for autonomous driving, October 2025a. URL https://github.com/NVlabs/alpasim.

NVIDIA, Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, Dongran Liu, Ming-Yu Liu, Langechuan Liu, Zhijian Liu, Jason Lu, Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Shida Shen, Yunfei Shi, Sarah Tariq, Ran Tian, Tilman Wekel, Xinshuo Weng, Tianjun Xiao, Eric

Yang, Xiaodong Yang, Yurong You, Xiaohui Zeng, Wenyuan Zhang, Boris Ivanovic, and Marco Pavone. Alpamayo-R1: Bridging Reasoning and Action Prediction for Generalizable Autonomous Driving in the Long Tail, October 2025b.

NVIDIA Corporation. PhysicalAI-Autonomous-Vehicles, October 2025. Dataset hosted on Hugging Face.

NVIDIA Corporation. Chain-of-causation (coc) autolabeling pipeline. `https://github.com/NVlabs/alpamayo-coc-autolabeler`, 2026. GitHub repository.

Qwen Team. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5, February 2026a.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL `https://qwen.ai/blog?id=qwen3.8`.

Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, 2 edition, 2012. doi: 10.1007/978-1-4614-1433-9.

Huimin Wang, Yue Wang, Bihao Cui, Pengxiang Li, Ben Lu, Mingqian Wang, Tong Wang, Chuan Tang, Teng Zhang, and Kun Zhan. Reflectdrive-2: Reinforcement-learning-aligned self-editing for discrete diffusion driving. *arXiv preprint arXiv:2605.04647*, 2026a.

Xiyang Wang, Xinlin Wang, Tingguang Zhou, Gong Chen, Xingtai Gui, Zhi Xu, Xiaolei Wu, Feiyang Tan, Hangning Zhou, and Mu Yang. Chainflow-vla: Causal flow planning with vision-language models, 2026b. URL `https://arxiv.org/abs/2605.23270`.

Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26036–26046. IEEE, 2025.

Mingwang Xu, Jiahao Cui, Feipeng Cai, Hanlin Shang, Zhihao Zhu, Shan Luan, Yifang Xu, Neng Zhang, Yaoyi Li, Jia Cai, and Siyu Zhu. WAM-Diff: A Masked Diffusion VLA Framework with MoE and Online Reinforcement Learning for Autonomous Driving, December 2025a.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. WOD-E2E: Waymo Open Dataset for End-to-End Driving in Challenging Long-tail Scenarios, November 2025b.

Yifang Xu, Jiahao Cui, Feipeng Cai, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Neng Zhang, Yaoyi Li, Jia Cai, and Siyu Zhu. WAM-Flow: Parallel Coarse-to-Fine Motion Planning via Discrete Flow Matching for Autonomous Driving, December 2025c.

Yihong Xu, Eloi Zablocki, Yuan Yin, Elias Ramzi, Ellington Kirby, Alexandre Boulch, and Matthieu Cord. Test-time trajectory optimization for autonomous driving. *arXiv preprint arXiv:2606.07170*, 2026.

Huan Zheng, Yucheng Zhou, Tianyi Yan, Jiayi Su, Hongjun Chen, Dubing Chen, Xingtai Gui, Wencheng Han, Runzhou Tao, Zhongying Qiu, et al. From human intention to action prediction: Intention-driven end-to-end autonomous driving. *arXiv preprint arXiv:2512.12302*, 2025.

Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 888–897. IEEE, 2022.

Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C. Knoll. OpenDriveVLA: Towards end-to-end autonomous driving with large vision language action model. *arXiv preprint arXiv:2503.23463*, 2025a.

Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A Vision-Language-Action Model for End-to-End Autonomous Driving with Adaptive Reasoning and Reinforcement Fine-Tuning, June 2025b.

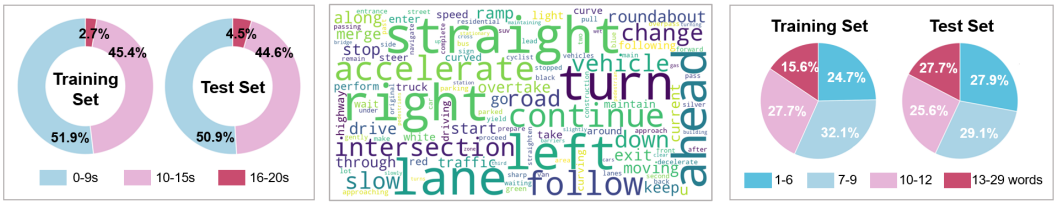
Zhihao Zhu, Hanlin Shang, Mingwang Xu, Feipeng Cai, Zhuolin He, Yaoyi Li, Jianhua Han, Hang Xu, and Siyu Zhu. Wam-diff2: Hierarchical ar-to-diffusion distillation for highly efficient autonomous driving vla. *arXiv preprint arXiv:2608.01035*, 2026.

Sicheng Zuo, Yuxuan Li, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Vega: Learning to drive with natural language instructions. *arXiv preprint arXiv:2603.25741*, 2026. URL https://arxiv.org/abs/2603.25741.

APPENDIX

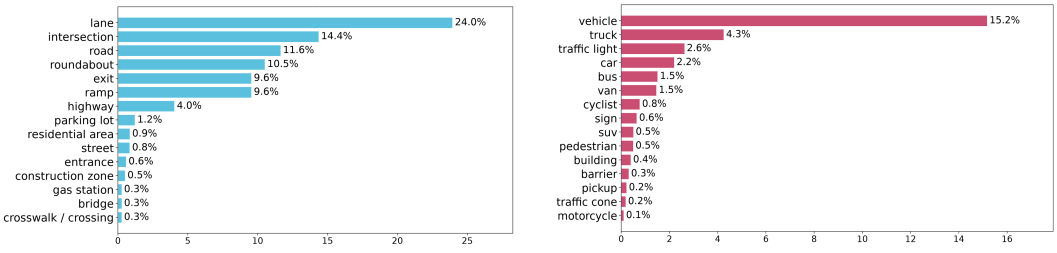
A DATASET ANALYSIS

To better understand the characteristics of WAM-Bench, we follow previous works Zheng et al. (2025); Zuo et al. (2026) and analyze the dataset from four perspectives: temporal diversity, semantic richness, linguistic complexity, and action diversity.



(a) Distribution of Segment Length (b) Word Cloud of Open-ended Instructions (c) Distribution of Instruction Length

Figure 10: Statistical analysis of WAM-Bench. (a) Distribution of video segment length, illustrating the temporal diversity of driving scenarios. (b) Word cloud of open-ended instructions, demonstrating semantic richness. (c) Distribution of instruction length, reflecting the linguistic complexity of open-ended instructions.



(a) Location Distribution in Instructions (b) Object Distribution in Instructions

Figure 11: Statistical analysis of semantic elements in open-ended instructions. (a) Location distribution (top-15). (b) Object distribution (top-15).

Temporal Diversity. Fig. 10(a) presents the distribution of video segment lengths in WAM-Bench. The dataset covers a wide range of temporal horizons, including both short-term and long-horizon driving scenarios. Such temporal diversity requires VLA models to not only react to immediate changes but also maintain consistent instruction execution over extended durations.

Semantic Richness. Unlike existing benchmarks Dauner et al. (2024); Xu et al. (2025b) that mainly focus on predefined instructions (e.g., go straight and turn), WAM-Bench contains diverse open-ended instructions. Fig. 10(b) reports the word cloud of instructions, showing that they cover diverse semantic elements, including driving actions (e.g., lane change, merge, and overtake), locations (e.g., intersection, roundabout, and highway), and target objects (e.g., vehicle and traffic light).

To further analyze the semantic elements in instructions, we report the top-15 referenced locations and objects in Fig. 11. The location distribution demonstrates diverse spatial references (such as exit, parking lot, gas station), while the object distribution covers various traffic entities (including truck, van, cyclist, and suv).

Table 9 provides representative examples of open-ended instructions across different action classes and subclasses, demonstrating diverse linguistic expressions and rich semantic elements.

Linguistic Complexity. Fig. 10(c) shows the distribution of instruction lengths measured by the number of words. Compared with traditional datasets dominated by short predefined instructions, WAM-Bench contains a substantial proportion of long-form instructions. Such instructions require VLA models to extract multiple semantic elements (including actions, locations, and objects), and translate them into appropriate driving trajectory.

Velocity (mph)	Num	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
		0m	1m	2m	DAC	NC	CF	EP	ADE
Low ($\leq$ 30)	151	62.3	68.9	74.8	87.4	86.1	94.7	93.2	4.0
Medium (30–55)	48	66.7	72.9	79.2	95.8	97.9	93.8	97.4	4.3
High ($>$ 55)	34	55.9	64.7	76.5	100	100	100	98.9	4.2
All	233	62.2	69.1	76.0	91.1	90.6	95.3	95.0	4.1

Table 8: Performance across different ego velocities on the WAM-Bench test split.

Action Diversity. Existing autonomous driving benchmarks usually contain a limited set of action classes and subclasses. Table 10 and Table 11 compare the action coverage of WAM-Bench with existing datasets. WAM-Bench provides 13 action classes and 21 action subclasses, covering diverse driving actions, including lane change, merge, overtake, ramp, roundabout, and U-turn. Such a diverse action space enables evaluation of instruction-driven autonomous driving beyond conventional navigation commands.

B Benchmark Analysis

In this section, we add some analysis results of Alpamayo-1.5 in the WAM-Bench test split.

Ego Velocity. Following typical U.S. speed-limit ranges, we divide ego velocity into Low ($\leq$ 30), Medium (30–55), and High ($>$ 55) levels. As shown in Table 8, IF@1m and IF@2m vary moderately across different velocity ranges, indicating that Alpamayo-1.5 maintains relatively stable instruction-following performance across different ego velocities. In contrast, DAC and NC increase from 87.4 and 86.1 in low-speed scenes to 100 and 100 in high-speed scenes. This may be because low-speed scenes are more often urban, with narrower roads and denser traffic interactions.

Action Subclass. Table 12 provides a fine-grained breakdown of Alpamayo-1.5 across different action subclasses. Overall, the trends are consistent with the action-class analysis, and no additional consistent failure patterns are observed. Since some subclasses contain relatively few samples, these results are mainly used for fine-grained characterization rather than direct comparison.

C Visualization

We provide additional qualitative visualizations of WAM-Bench in Fig. 12–Fig. 16. Each visualization consists of three parts. The upper-left part shows the BEV visualization of the driving scene, where the green bounding boxes indicate the predicted positions of surrounding vehicles and the yellow line denotes the predicted future trajectory of the ego vehicle. The light-green bounding boxes and green trajectory represent the GT vehicle states and ego trajectory, respectively. The upper-right part reports the corresponding evaluation metrics, including instruction following and planning metrics. The bottom part presents the rendered front-view images used as model inputs.

Fig. 12 presents representative instruction-following cases on the WAM-Bench test split. These examples demonstrate that the model can execute diverse open-ended instructions under different driving actions.

Fig. 13 shows representative long-horizon instruction-following cases on the WAM-Bench long-horizon split. The model maintains consistent instruction execution over multiple consecutive segments.

Fig. 14 visualizes instruction adaptation on the WAM-Bench changed-instructions split. Under the same driving scene, the model generates different trajectories according to different open-ended instructions, demonstrating the capability to adapt driving behaviors to instruction changes.

Fig. 15 provides a representative example from the WAM-Bench OOD-instructions split. The visualization demonstrates the model's ability to generalize to unseen instruction patterns.

Fig. 16 presents representative failure cases on the WAM-Bench test split. These examples reveal challenging scenarios where the model fails to satisfy the given instructions or generate feasible trajectories.

D Data Annotation Platform

To ensure the quality of open-ended instructions, we develop a manual annotation platform for data verification and refinement. As shown in Fig. 17, annotators review the generated instructions together with the corresponding driving videos and trajectories. The platform enables annotators to correct instruction descriptions, adjust segment boundaries, and assign action labels. This manual verification process ensures the alignment between instructions, driving scenarios, and ego-vehicle behaviors.

E Discussion

WAM-Bench is the first closed-loop benchmark for open-ended instruction following in autonomous driving, aiming to facilitate the development of instruction-driven autonomous driving systems. Nevertheless, several limitations remain.

Data Scale and Diversity. Due to the limitation of the original test data (PAI-AV-nuRec), some action classes contain relatively few samples (such as U-turn, slight turn, and pull into). In future work, we will collect additional real-world human driving data to expand the dataset with more diverse driving actions and richer open-ended instructions.

World Model Rendering. Limited by the generation capability of the video world model (OmniDreams), our current benchmark only supports front-view closed-loop simulation. Future work will explore extending the world model to multi-view generation, enabling more comprehensive closed-loop evaluation.

Instruction Following Metric. Although we introduce an instruction-following metric in this work, there remains substantial room for further improvement. Future research will investigate more advanced evaluation approaches, such as VLM-based instruction-following scorers, to better measure the semantic alignment between instructions and driving trajectory.

Finally, we will continue maintaining and expanding the WAM-Bench series in the open-source community to promote future research on instruction-driven autonomous driving.

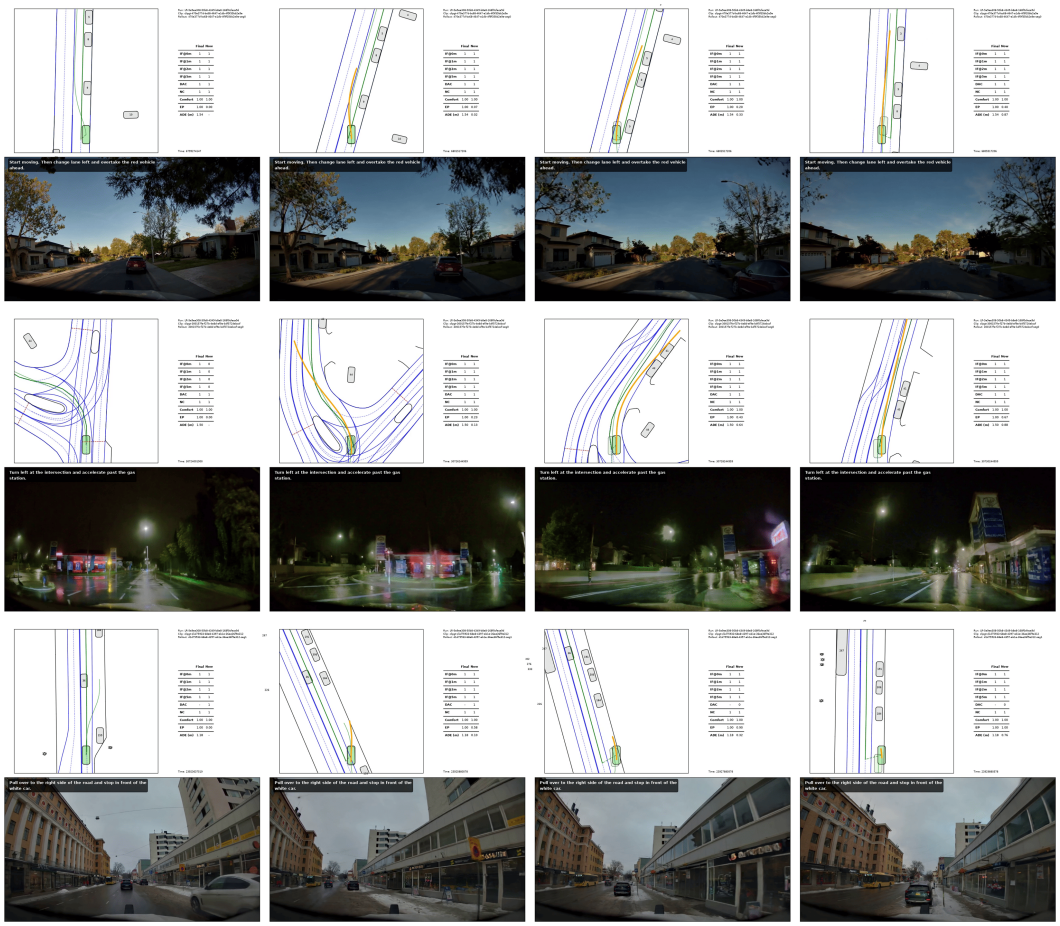


Figure 12: Visualization of three cases on the WAM-Bench test split.

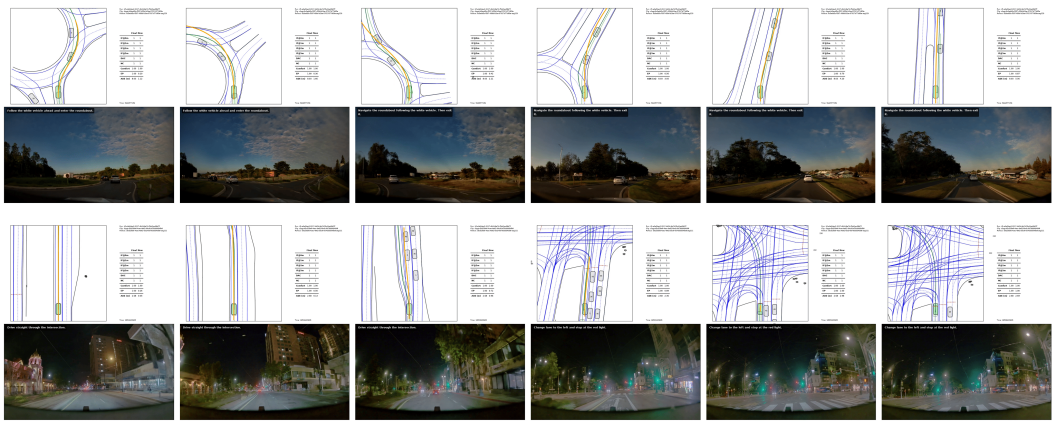


Figure 13: Visualization of two cases on the WAM-Bench long-horizon split.

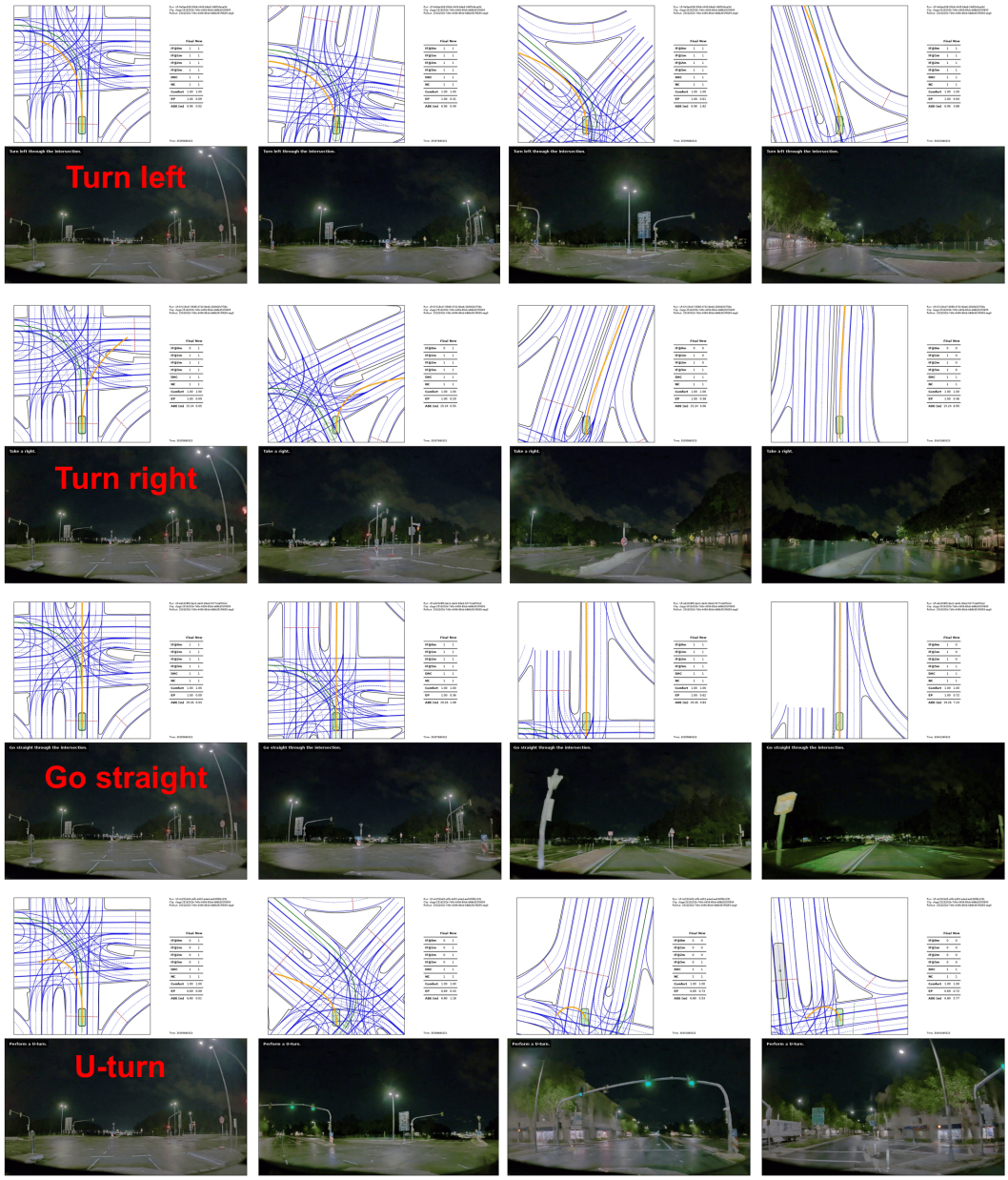


Figure 14: Visualization on the WAM-Bench changed-instructions split.

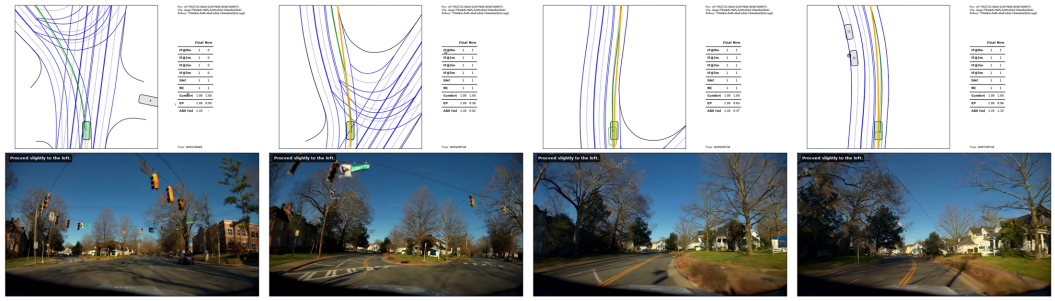


Figure 15: Visualization of one case on the WAM-Bench OOD-instructions split.

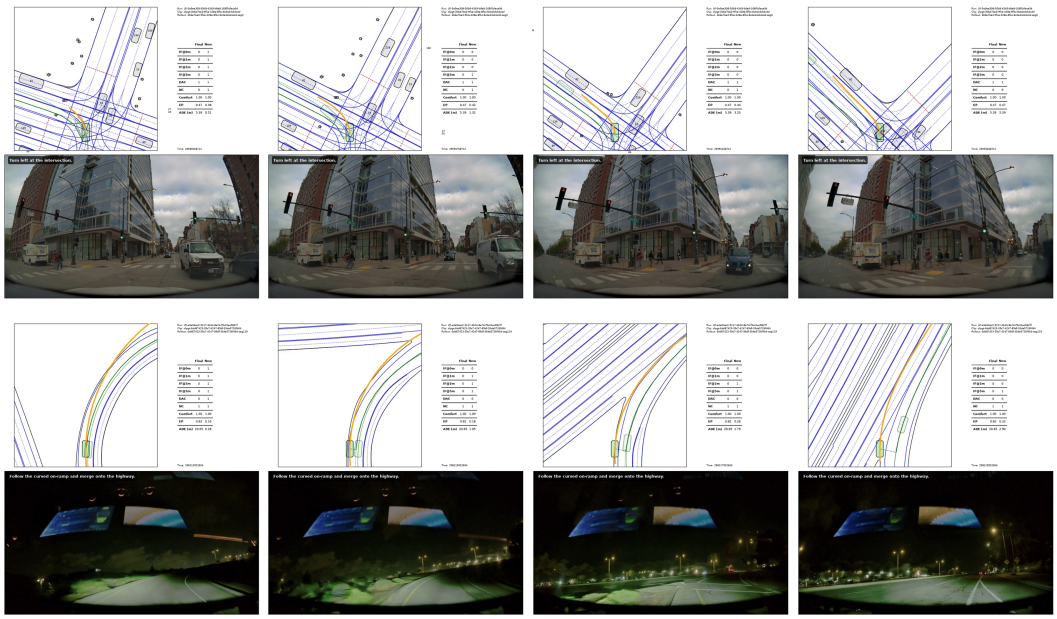


Figure 16: Visualization of two failure cases on the WAM-Bench test split.

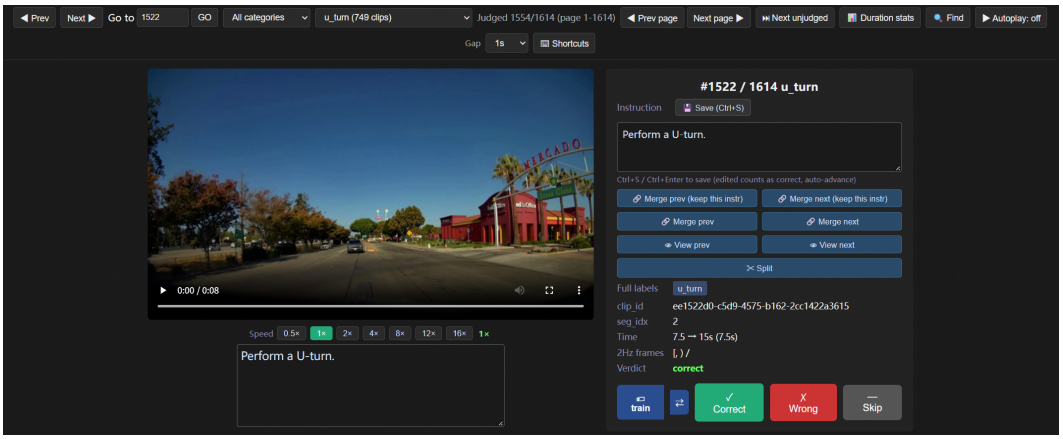


Figure 17: Data annotation interface for manual review. The platform enables annotators to refine open-ended instructions, adjust segment boundaries, and assign action labels based on driving videos and trajectories.

Action	Action Subclass	Instruction Examples
Continue	-	Continue along the road. Keep going along the current road. Follow the road ahead.
Follow	-	Follow the vehicle ahead. Continue following the car in front. Maintain distance behind the leading vehicle.
Go Straight	-	Go straight through the intersection. Continue straight ahead. Proceed forward along the road.
Lane Change	Lane Change Left/Right	Change lane left. Move to the right lane. Change two lanes left into the turning lane.
Merge	Merge Left/Right	Merge onto the main road. Merge right onto the highway. Merge left into the adjacent lane.
Overtake	Overtake Left/Right	Overtake the vehicle ahead on the left. Pass the slow vehicle on the right. Change lane left to overtake the truck.
Pull Over	-	Pull over and stop behind the white vehicle. Pull over to the right side. Pull over at the entrance of the glass building.
Ramp	Enter/Exit/Along Ramp	Enter the ramp and continue along it. Take the exit ramp from the highway. Follow the ramp to merge onto the main road.
Roundabout	Enter/Exit/Along Roundabout	Enter the roundabout and take the second exit. Continue around the roundabout. Exit the roundabout at the left exit.
Start Moving	-	Start moving and continue along the road. Begin driving forward. Start moving after the traffic light turns green.
Stop	-	Slow down and then stop behind the truck. Stop at the intersection. Stop for the red traffic light.
Turn	Turn Left/Right	Turn right at the intersection. Make a left turn at the junction. Turn left onto the side road.
U-Turn	-	Perform a U-turn at the intersection. Make a U-turn and continue straight. Navigate the roundabout to perform a U-turn.

Table 9: Examples of open-ended instructions in WAM-Bench across different action classes and action subclasses. "-" indicates that the action class itself serves as the subclass.

Dataset	Action Class	Annotation
Bench2Drive	4	Follow, Go Straight, Lane Change, Turn
Bench2Drive-Speed	5	Follow, Go Straight, Lane Change, Overtake, Turn
nuScenes	2	Go Straight, Turn
NAVSIM	2	Go Straight, Turn
PAI-AV	0	–
PAI-AV-nuRec	0	–
Intention-Drive	5	Go Straight, Lane Change, Stop, Turn, U-Turn
InstructScene	5	Follow, Go Straight, Pull Over, Stop, Turn
WAM-Bench	13	Continue, Follow, Go Straight, Lane Change, Merge Overtake, Pull Over, Ramp, Roundabout Start, Stop, Turn, U-Turn

Table 10: Statistics of action classes across different datasets.

Dataset	Action Subclass	Annotation
Bench2Drive	6	Change Lane Left/Right, Follow, Go Straight, Turn Left/Right
Bench2Drive-Speed	7	Change Lane Left/Right, Follow, Go Straight, Overtake, Turn Left/Right
nuScenes	3	Go Straight, Turn Left/Right
NAVSIM	3	Go Straight, Turn Left/Right
PAI-AV	0	–
PAI-AV-nuRec	0	–
Intention-Drive	8	Change Lane Left/Right, Go Straight, Stop, Turn Left/Right, U-Turn Left/Right
InstructScene	6	Follow, Go Straight, Pull Over, Stop, Turn Left/Right
WAM-Bench	21	Continue, Enter/Exit/Along Ramp Enter/Exit/Along Roundabout, Follow, Go Straight Lane Change Left/Right, Merge Left/Right Overtake Left/Right, Pull Over, Start Moving, Stop Turn Left/Right, U-Turn

Table 11: Statistics of action subclasses across different datasets. Paired left/right actions are counted as separate action classes.

Action Subclass	Num	Instruction (%) ↑			Planning (%) ↑				Error (m) ↓
		0m	1m	2m	DAC	NC	CF	EP	ADE
Stop	20	85	90	90	95	85	95	-	1.3
Continue	20	80	90	90	95	100	100	98.8	3.9
Follow	20	80	85	90	100	100	100	97.8	2.3
Overtake Left	9	77.8	88.9	88.9	100	100	88.9	99.6	5.8
Pull Over	8	75	75	87.5	-	100	100	99.2	2.1
Enter Ramp	4	75	75	75	100	100	100	98.5	4.6
Exit Ramp	9	66.7	66.7	77.8	100	88.9	77.8	94.5	5.8
Overtake Right	11	63.6	63.6	81.8	100	100	100	98.7	5.3
Lane Change Left	11	63.6	81.8	90.9	90.9	100	100	99.2	3.5
Merge Right	10	60	70	70	90	100	90	93.2	4.2
Lane Change Right	9	55.6	66.7	77.8	100	88.9	88.9	99.1	3.7
Go Straight	20	55	60	70	95	95	100	95.4	3.1
Start Moving	20	55	60	65	95	85	95	91.6	2.6
Exit Roundabout	10	50	50	60	80	90	100	93.0	7.9
Merge Left	10	50	60	70	50	100	90	96.6	3.6
Along Ramp	7	42.9	57.1	85.7	100	100	100	97.1	2.7
Turn Right	8	37.5	62.5	62.5	87.5	50	87.5	79.7	9.1
Turn Left	12	33.3	41.7	41.7	75	50	91.7	89.4	6.9
Along Roundabout	3	33.3	33.3	100	66.7	100	100	98.2	1.3
U-Turn	5	20	20	20	100	80	80	75.4	14.4
All	233	62.2	69.1	76	91.1	90.6	95.3	95	4.1

Table 12: Fine-grained performance of Alpamayo-1.5 across different action subclasses on the WAM-Bench test split, sorted by IF@0m in descending order.