
DriveHierarchy: A Benchmark for Diagnosing VLM Driving Capabilities from Open-Loop Understanding to Closed-Loop Execution

Chengkai Xu^{1*} Jiaqi Liu^{2*} Yicheng Guo^{1*} Peng Hang¹ Jian Sun^{1†}

¹College of Transportation, Tongji University, Shanghai, China

²Department of Computer Science, UNC Chapel Hill, NC, USA

{xuchengkai, guo_yicheng, hangpeng, sunjian}@tongji.edu.cn
jqliu@cs.unc.edu

Abstract

Evaluating VLM-based autonomous driving remains difficult because driving competence is composite, where a capable system must ground traffic participants and hazards, integrate context across views and time, reason about future evolution, and act appropriately under closed-loop interaction. Existing benchmarks usually assess either open-loop understanding or closed-loop driving but provide limited structure for explaining how these abilities are organized, how they relate, and how they may inform model diagnosis and improvement. We present DRIVEHIERARCHY, a hierarchical benchmark that organizes VLM-based autonomous driving into four ranks, spanning perceptual grounding, contextual memory, mental reasoning, and closed-loop execution. To instantiate this hierarchy, we integrate multiple open-source autonomous-driving datasets into a unified open-loop benchmark with 76,798 question-answer pairs over 84,279 frames and develop a closed-loop simulation platform with interactive scenario construction on a real-world road network, from which 100 driving scenarios are curated for embodied evaluation. Experiments on 15 VLMs show that DRIVEHIERARCHY captures structured but non-redundant capability variation, relates open-loop understanding to closed-loop driving, and provides a practical basis for diagnosis and benchmark-guided optimization. DRIVEHIERARCHY therefore serves as a unified framework for evaluating and improving VLM-based autonomous driving systems. An anonymized project has been released on <https://github.com/PerfectXu88/DriveHierarchy>.

1 Introduction

End-to-end autonomous driving has emerged as a central paradigm for learning-based driving systems by directly mapping sensory observations to driving actions within a unified architecture[1, 2, 3]. Relative to traditional modular pipelines, this formulation offers a more integrated framework for coupling perception and decision-making, while avoiding hand-designed intermediate interfaces[4, 5]. Recent progress in foundation models has further accelerated this direction, with vision-language models (VLMs) attracting increasing attention for autonomous driving due to their strong multimodal representation and reasoning capabilities[2, 6, 7]. By unifying visual understanding with semantic inference, VLM-based driving systems provide a promising foundation for integrated scene interpretation and decision-making[8, 9]. However, their growing adoption also exposes a fundamental challenge: *how should the performance of such systems be evaluated?*

*Equal contributions.

†Corresponding author: sunjian@tongji.edu.cn.

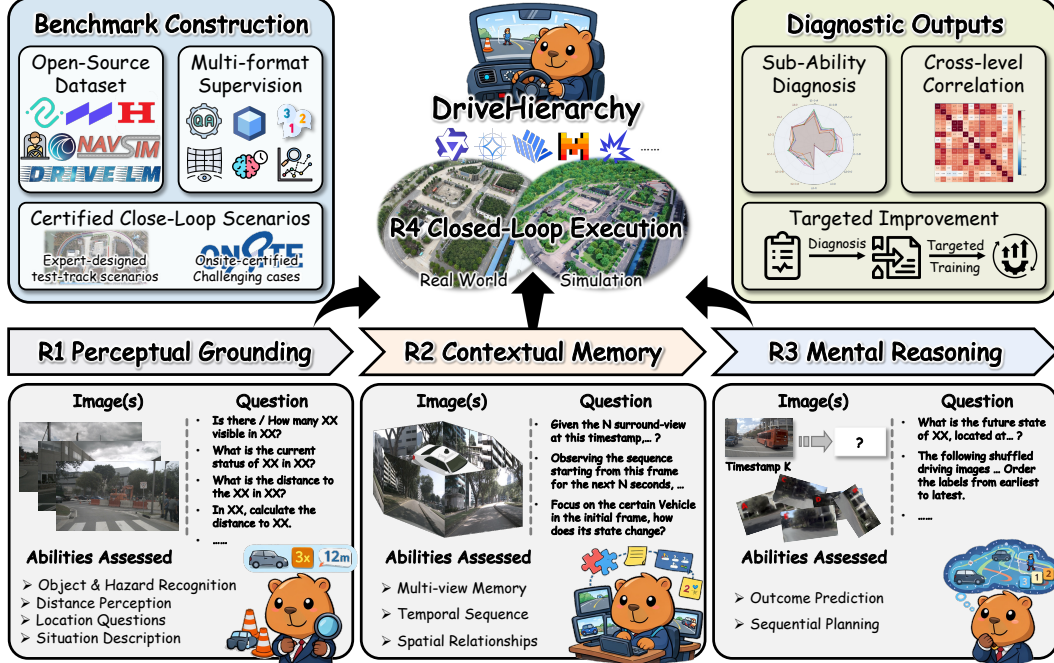


Figure 1: **Overview of DRIVEHIERARCHY**, which organizes VLM-based autonomous driving into four progressively structured ranks, spanning perceptual grounding, contextual memory, mental reasoning, and closed-loop execution.

Evaluating such systems is nontrivial because driving intelligence is not a monolithic ability[10, 11, 12]. An excellent driving system must ground traffic participants and hazards in the current scene, integrate information coherently across viewpoints and time, reason about how situations may evolve, and ultimately select appropriate actions under closed-loop interaction[6, 13]. Existing evaluation protocols can usually capture only part of this capability chain. Open-loop driving-language and question-answering benchmarks can probe perceptual and reasoning-related abilities, but they often entangle heterogeneous capabilities within task-level scores that are difficult to interpret diagnostically[14, 15, 16]. Closed-loop evaluations, in contrast, provide direct evidence of embodied driving performance, yet route-level outcomes such as success, collision, or completion rarely reveal which internal capability is responsible for the observed behavior. Consequently, current benchmarks are valuable for system comparison, but considerably less informative for explaining model behavior, isolating failure modes, or guiding targeted improvement.

To better address this limitation, we draw inspiration from cognitive science, which views complex embodied intelligence as arising from progressively organized levels of capability rather than from a single undifferentiated skill[17, 18, 19]. Based on this view, we propose **DRIVEHIERARCHY**, a hierarchical benchmark for VLM-based autonomous driving, as illustrated in Figure 1. The benchmark structures driving capability into four ranks, spanning **R1** Perceptual Grounding, **R2** Contextual Memory, **R3** Mental Reasoning, and **R4** Closed-loop Execution. By organizing evaluation along this hierarchy, the benchmark supports both fine-grained capability analysis and system-level assessment, enabling a more explicit examination of model strengths, weaknesses, capability relationships, and their connection to downstream driving behavior.

To instantiate this hierarchy, we integrate multiple public autonomous driving datasets to construct a capability-oriented corpus in both full-scale and streamlined forms, comprising 84,279 frames and 76,798 question-answer pairs in total. To support embodied evaluation beyond static analysis, we further develop a joint simulation platform for interactive scenario construction and create 100 functional driving scenarios for closed-loop testing. On this basis, we conduct a comprehensive study of 15 open-source VLMs and a targeted supervised fine-tuning (SFT) case study to examine the interactions among different capabilities and their influence on downstream driving performance. The results show that **DRIVEHIERARCHY** enables fine-grained diagnosis of model strengths and

weaknesses, reveals meaningful relationships across capability levels, and provides a useful basis for targeted analysis and intervention design.

Our contributions are as follows:

- A hierarchical capability benchmark for VLM-based autonomous driving is proposed, integrating multiple public autonomous driving datasets and decomposing driving intelligence into four progressively organized ranks from perceptual grounding to closed-loop execution.
- A joint simulation platform with interactive scenario construction is developed to support closed-loop evaluation under diverse functional driving scenarios.
- A comprehensive benchmarking of 15 open-source VLMs and targeted fine-tuning studies are conducted to support capability-level analysis and model improvement.

2 Related Work

VLM-Based End-to-End Autonomous Driving. End-to-end autonomous driving has increasingly evolved from purely sensor-to-control formulations toward unified architectures that incorporate semantic reasoning and language [1, 20, 21]. Recent advances in VLMs have further accelerated this trend by enabling driving systems to couple visual perception with instruction understanding, scene interpretation, and high-level decision support within a common multimodal framework [21, 22, 23]. This line of work has expanded from language-assisted driving agents to VLM-based end-to-end systems and further to Vision-Language-Action models that directly predict driving actions or control signals [24, 25, 26, 27, 8]. As the backbone of these approaches, VLMs offer stronger representational unification and greater interpretability, but they also introduce new challenges for evaluation, since perception, memory, reasoning, and decision-making are increasingly entangled within a single model [14, 28].

Autonomous Driving Benchmarks. Recent autonomous driving benchmarks have substantially advanced the evaluation of learning-based driving systems, yet remaining fragmented in both objective and granularity. Open-loop benchmarks, including driving-language, scene-understanding, and question-answering datasets, provide useful testbeds for perception, interaction understanding, and reasoning, but they typically organize evaluation around task-specific scores and therefore offer limited visibility into the internal capability structure of a model [29, 30, 14, 16, 12, 31]. Closed-loop benchmarks, in contrast, assess embodied performance through interactive simulation or scenario execution, making them more faithful to downstream driving behavior. However, they often reduce evaluation to route completion, collision statistics, or aggregate success rates, which are insufficient for diagnosing which underlying abilities contribute to failure [32, 33, 34, 35]. As a result, existing benchmarks are effective for model comparison, but less effective for explaining competence, isolating weaknesses, and relating open-loop ability to closed-loop driving outcomes.

Hierarchical Capability Decomposition for Embodied Intelligence. Cognitive science has long treated intelligence as hierarchically organized rather than flat [19], with Carroll’s three-stratum theory as a canonical example [17]. Reviews in neuroscience and psychometrics likewise argue that intelligence is a hierarchy in which partially separable abilities contribute to coherent higher-level performance [36]. This conceptual tradition has increasingly influenced benchmark design, where recent work in multimodal evaluation has moved beyond flattened aggregate scores toward fine-grained evaluations that separately probe abilities, providing a more informative alternative to flat aggregate metrics [37, 38, 39, 40]. Such work suggests that capability decomposition is essential for understanding not only whether a model succeeds, but also how its internal competencies are structured and where its failure modes originate [18, 41, 42]. However, in autonomous driving, this view remains insufficiently developed. Existing evaluations still tend to organize results around task families or outcome metrics, while providing limited structure for understanding how lower-level capabilities support higher-level reasoning and embodied driving behavior [33, 14].

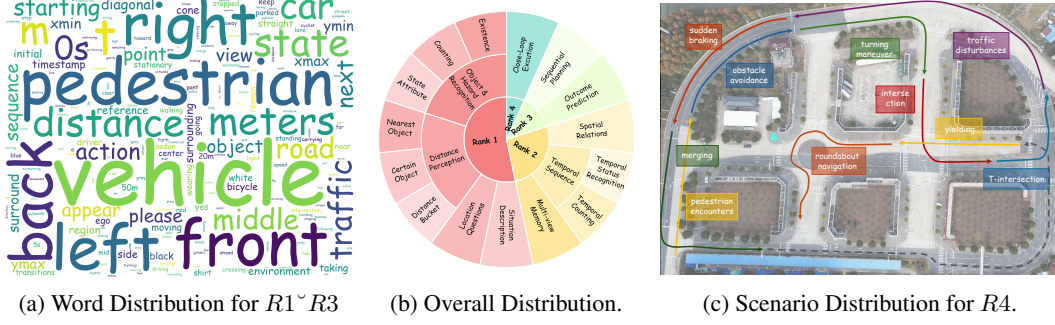


Figure 2: **Distribution of the DRIVEHIERARCHY benchmark**, which summarizes the data composition of the open-loop ranks and the scenario allocation of the closed-loop benchmark.

3 DriveHierarchy

3.1 Overview and Design Principle

As shown in Fig. 1, DRIVEHIERARCHY organizes VLM-based autonomous driving into four progressively structured ranks: **R1** Perceptual Grounding, **R2** Contextual Memory, **R3** Mental Reasoning, and **R4** Closed-loop Execution. The first three ranks are instantiated as open-loop tasks that probe distinct aspects of scene understanding and cognitive processing, while the fourth rank evaluates whether these capabilities are associated with actual driving behavior under continuous interaction. This design provides fine-grained diagnosis at the local level and a structured basis for analyzing how lower-level capabilities relate to higher-level reasoning and embodied driving at the system level. To instantiate this hierarchy, we integrate multiple public driving datasets into a capability-oriented open-loop corpus and complement it with a controllable closed-loop benchmark built on an interactive co-simulation platform. The refined version used for all reported $R1$ - $R3$ results contains 14,000 question-answer pairs over 31,945 frames, together with 100 closed-loop scenarios for $R4$. Dataset provenance and benchmark composition details are provided in Appendix B.

3.2 Open-Loop Benchmark Construction

R1: Perceptual Grounding. $R1$ contains 8 task types and evaluates whether a model can build a reliable single-frame representation of the current scene. We group these tasks into four capability families. The first family, **object and hazard recognition**, includes existence judgment, target counting, and state identification. These tasks are derived from annotated driving scenes by converting object categories and state labels into binary, numeric, or multiple-choice question-answer pairs. The second family, **distance perception**, contains nearest-object distance estimation, referred-object distance estimation, and distance-bucket counting. These tasks are constructed from ego-object geometry and calibrated image coordinates, allowing the benchmark to test both direct metric estimation and language-guided spatial reference. The third family, **visual grounding**, asks the model to localize an object described in natural language and is supervised by annotated bounding boxes. The fourth family, **situation description**, evaluates whether the model can summarize the current traffic state in natural language from image evidence.

R2: Contextual Memory. $R2$ extends the benchmark from single-frame understanding to structured scene representation across views, time, and object relations, containing 4 task types and evaluates whether a model can preserve memory when information is distributed across views, time, and entities. We instantiate this rank with three capability families. **Multi-view memory** is constructed from synchronized six-camera observations at the same timestamp and asks the model to infer the number of unique surrounding targets after resolving cross-view overlap. To avoid reducing the task to repeated single-view counting, we retain only samples that require genuine cross-view integration. **Temporal memory** is instantiated in two forms. The first uses fixed short clips to count unique targets appearing over time, thereby testing memory-based aggregation across frames. The second tracks a target object across a temporal window and asks the model to identify whether its state changes, which directly probes dynamic state-transition understanding. **Spatial relation reasoning** constructs

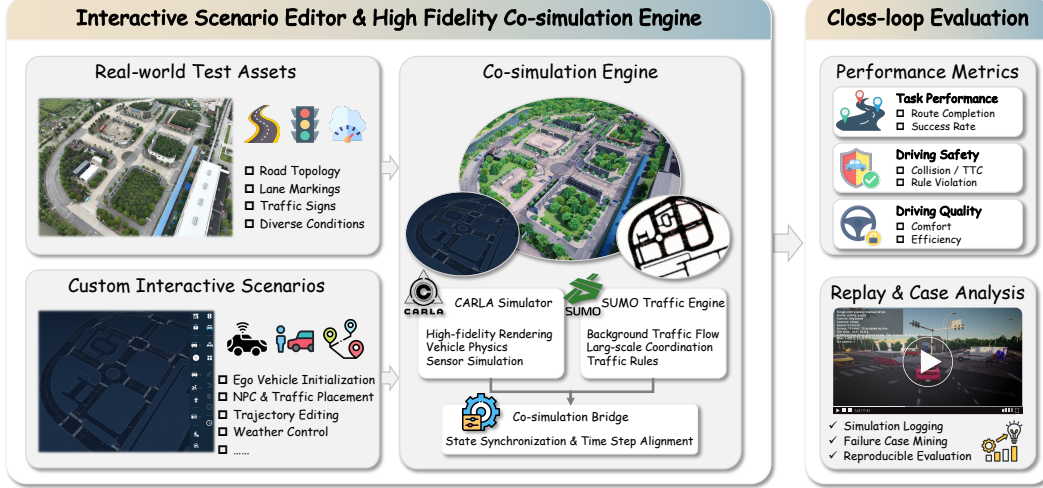


Figure 3: **Closed-loop evaluation platform for R_4** , which combines simulation with interactive scenario construction on a real-road layout, enabling controllable evaluation under diverse traffic interactions.

pairwise queries between visible entities and requires the model to infer relative positional relations such as front, rear, left, or right.

R_3 : Mental Reasoning. R_3 contains 2 task types, marking the transition from contextual understanding to future-oriented reasoning. The first task family, *outcome prediction*, asks the model to infer the likely future behavior of a target traffic participant from the current scene context. These questions are derived from future-aware driving annotations and are posed as multiple-choice predictions over plausible downstream actions or states. The second family, *sequence planning*, presents a set of candidate observations and requires the model to recover their correct temporal order. At this rank, the benchmark tests whether the model can go beyond describing the current scene and instead reason about how the scene may evolve.

Task Formulation and Evaluation. Across R_1 - R_3 , each open-loop sample is formulated as a tuple (x_i, q_i, y_i) , where x_i denotes the visual input, q_i the task-specific natural-language query, and y_i the ground-truth target. Depending on the task, x_i may be a single image, a synchronized multi-view observation set, or a short temporal clip. Given a model prediction $\hat{y}_i$, we assign each sample a bounded score $s_i \in [0, 100]$. This formulation provides a unified evaluation interface while preserving the distinct supervision structure of different capability types. Distinct evaluation methods are designed for different question types, with further details regarding scoring and implementation discussed in Appendix D.

3.3 Closed-loop Benchmark Construction

Platform and Scenario Design. R_4 evaluates whether the capabilities measured in open loop transfer to actual driving behavior under continuous interaction. As shown in Fig. 3, we construct a CARLA-SUMO co-simulation platform on a real-world road network that preserves road topology, lane structure, and traffic semantics. CARLA [43] provides the ego-centric simulation environment and multimodal observations, while SUMO[44] generates and controls the background traffic flow, allowing the platform to remain both physically expressive and behaviorally scalable. On top of this backbone, we develop an interactive scenario editing framework that supports configurable placement and control of the ego vehicle, surrounding vehicles, pedestrians, and other dynamic elements together with their trajectories, interactions, and traffic-control conditions. Using this framework, we curate 100 closed-loop scenarios covering representative urban interaction patterns, including *pedestrian encounters*, *obstacle avoidance*, *turning maneuvers*, *intersections*, *traffic disturbances*, *sudden braking*, *merging*, *yielding*, and *roundabout navigation*.

Table 1: **Open-loop benchmark results on DriveHierarchy for 15 VLMs.** The overall score is computed using the frozen calibration weights shown in the header. Gray and green cells mark the second-best and best results, respectively.

Models	Avg.Score	R1 Perceptual Grounding						R2 Contextual Memory				R3 Mental Reasoning			
		R1_1_A	R1_1_B	R1_1_C	R1_2_A	R1_2_B	R1_2_C	R1_3	R1_4	R2_1	R2_2_A	R2_2_B	R2_3	R3_1	R3_2
		0.030	0.027	0.056	0.035	0.036	0.029	0.056	0.031	0.064	0.079	0.067	0.090	0.233	0.167
Generalist															
Qwen3-VL(2B)[45]	43.91	68.80	29.54	39.10	25.79	38.97	35.46	0.99	50.53	33.78	41.36	49.90	39.70	59.20	46.58
InternVL3_5(2B)[46]	39.28	62.00	30.97	45.40	7.62	9.92	33.41	0.07	36.35	30.71	35.94	44.90	35.20	54.70	45.45
Qwen3-VL(8B)[45]	45.09	70.00	33.32	51.20	46.70	46.76	35.74	1.05	50.23	32.87	41.25	53.60	37.70	56.90	45.84
InternVL3_5(8B)[46]	43.49	62.70	27.06	49.40	42.47	38.83	32.62	0.90	44.23	33.91	29.58	49.70	44.50	57.50	45.70
ZwZ(8B)[47]	45.09	70.30	37.41	50.60	43.92	44.60	39.29	1.00	48.91	35.63	41.98	53.30	41.10	54.60	46.11
MiniCPM-V-4_5(9B)[48]	45.50	66.10	32.24	45.70	56.53	59.60	33.67	1.06	49.87	41.95	39.46	50.90	42.40	55.00	45.16
Gemma-3-it(12B)[49]	46.27	63.00	34.65	50.00	49.55	53.97	31.49	2.22	47.11	45.40	42.94	49.10	37.20	56.40	50.28
Pixtral(12b)[50]	42.31	56.90	25.02	39.80	45.27	51.55	21.46	1.12	41.06	17.43	36.28	46.10	49.70	57.40	44.28
Gemma-3-it(27B)[49]	49.53	68.00	37.16	51.80	63.29	65.81	36.17	5.66	49.54	49.31	44.49	50.00	51.70	57.00	48.77
Qwen3-VL(32B)[45]	49.64	73.80	35.61	52.40	45.76	49.12	36.34	1.08	56.83	46.56	48.47	54.10	46.50	62.60	48.42
InternVL3_5(38B)[46]	46.61	64.70	31.18	50.80	43.05	42.11	34.12	1.01	48.08	29.45	41.58	52.90	49.30	60.90	48.43
Qwen2.5-VL(72B)[51]	54.47	68.90	35.55	49.90	59.41	70.67	36.38	67.26	50.09	43.49	35.47	56.80	62.00	62.40	48.97
InternVL3(78B)[52]	46.93	68.60	38.85	53.70	23.91	20.33	41.82	3.77	49.86	43.48	41.18	51.70	48.60	61.70	48.05
Driving-Specialized															
DA-DriveLM(4B)[53]	38.31	66.80	26.13	61.00	34.99	23.18	26.80	0.03	33.23	14.88	26.11	49.00	26.50	52.70	44.06
ReasonDrive(7B)[54]	43.13	50.20	21.20	45.10	4.51	16.28	29.28	63.24	10.68	3.65	9.94	47.60	28.40	78.90	47.62

Scoring Design. For each scenario, the evaluated model receives multimodal driving inputs, including front-view imagery, bird’s-eye-view context, and navigation cues, and must output low-level control signals for vehicle actuation. The closed-loop score jointly measures route completion, safety, and efficiency. Let S_{route} , S_{safety} , and S_{eff} denote the corresponding terms. The final scene score is defined as

$$S_{R4} = S_{\text{route}} \cdot S_{\text{safety}} \cdot S_{\text{eff}}. \quad (1)$$

This multiplicative design prevents route progress alone from masking unsafe or inefficient behavior. The route term measures normalized progress toward the destination, the safety term aggregates infraction penalties and lane-keeping quality, and the efficiency term evaluates whether the realized trajectory is unnecessarily long or slow relative to the route reference. The detailed formulation, event penalties, and implementation-specific thresholds are provided in Appendix E.

4 Experiments

4.1 Experimental Setup

We evaluate DRIVEHIERARCHY on 15 VLMs, covering both general-purpose VLMs and driving-specialized models. The open-loop benchmark spans the three capability ranks $R1$ - $R3$, while $R4$ evaluates the same models in closed-loop driving scenarios. To test whether the benchmark can guide targeted model improvement, we further perform SFT on Qwen3-VL-8B-Instruct by selecting the task groups on which its gap to the corresponding state-of-the-art result is the largest.

4.2 Open-Loop Benchmark Results

Overall Capability Coverage. Table 1 shows an overall ordering on the open-loop benchmark. Qwen2.5-VL(72B) achieves the best weighted score of 54.47, followed by Qwen3-VL(32B) and Gemma-3-it(27B) with 49.64 and 49.53, respectively. This advantage is not explained by scale alone. The leading models are competitive across all three ranks rather than peaking on a single subset of tasks, which aligns with the characteristic of autonomous driving that requires comprehensive capabilities. By contrast, driving-specialized VLMs remain substantially weaker in overall coverage, indicating that current domain-specific adaptation more often strengthens local task families than it produces robust hierarchical competence, which will be further highlighted in the subsequent closed-loop testing. Meanwhile, different models peak on different sub-capabilities, and the identity of the strongest model changes across tasks, which is particularly evident in driving-specific VLMs.

Capability Structure Across Model Families. Figure 4 clarifies the family-level trends that are partially hidden in the table. Within the Qwen series, scaling improves both average performance and profile balance, yielding progressively more complete coverage from grounding to predictive

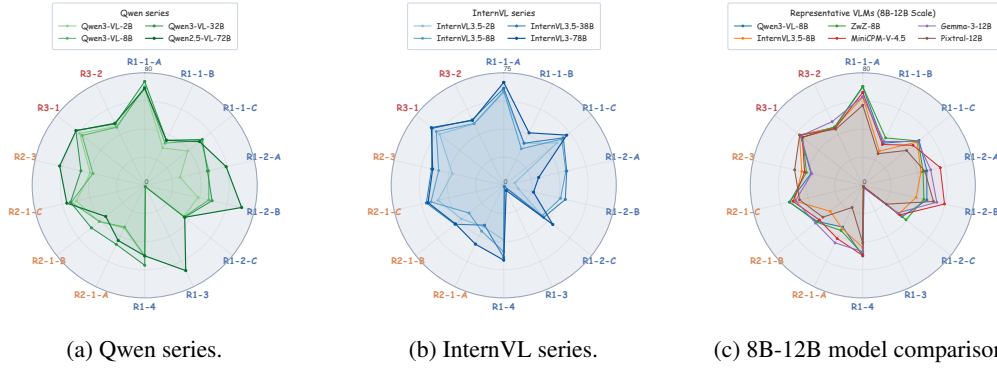


Figure 4: **Representative capability profiles revealed by DriveHierarchy.** The radar plots highlight both within-family scaling trends and architecture-specific trade-offs among similarly sized models.

Table 2: **Closed-loop benchmark results on $R4$ across 10 scenario families.** Higher scores indicate better driving performance under the composed route, safety, and efficiency metric.

Models	Sce 1	Sce 2	Sce 3	Sce 4	Sce 5	Sce 6	Sce 7	Sce 8	Sce 9	Sce 10	Avg.Score	Rank
Generalist												
Qwen3-VL(2B)	1.176	0.062	0.000	0.686	0.114	0.299	0.518	0.620	0.207	0.084	0.377	14
InternVL3_5(2B)	1.813	1.411	0.121	1.443	0.635	0.377	1.964	0.903	1.644	1.872	1.218	7
Qwen3-VL(8B)	1.278	1.726	0.384	1.128	0.574	0.271	1.379	0.394	0.661	1.227	0.902	9
InternVL3_5(8B)	0.001	0.246	0.960	0.939	0.848	0.642	1.285	0.662	0.412	1.025	0.702	13
ZwZ(8B)	1.147	1.034	0.415	1.312	0.471	0.164	0.773	0.508	0.948	1.322	0.809	10
MiniCPM-V-4_5(9B)	10.885	8.949	0.099	4.931	0.439	0.000	17.892	2.281	3.592	4.980	5.405	3
Gemma-3-it(12B)	1.427	0.825	0.553	0.926	0.625	0.431	0.794	0.364	0.697	0.548	0.719	11
Pixtral(12b)	0.940	1.758	0.642	1.103	0.107	0.158	0.323	0.206	0.874	0.932	0.704	12
Gemma-3-it(27B)	27.950	10.269	0.212	16.599	2.276	8.492	43.936	2.992	9.406	4.054	12.619	1
Qwen3-VL(32B)	5.380	3.550	0.191	2.605	2.290	9.165	7.690	2.457	0.487	0.702	3.452	5
InternVL3_5(38B)	0.001	1.720	0.285	0.501	0.245	0.111	6.939	0.035	0.369	0.912	1.112	8
Qwen2.5-VL(72B)	21.056	10.123	0.052	14.513	2.516	9.325	13.508	2.478	16.237	3.849	9.366	2
InternVL3(78B)	8.145	2.468	0.063	2.317	0.524	0.202	6.132	2.077	6.500	2.988	3.142	6
Driving-Specialized												
DA-DriveLM(4B)	0.001	0.000	0.023	0.008	0.008	0.005	0.041	0.012	0.024	0.044	0.017	15
ReasonDrive(7B)	1.814	1.912	0.388	2.030	0.948	0.188	22.411	1.747	8.590	1.983	4.201	4

reasoning. InternVL improves more unevenly, with gains concentrated in recognition and parts of $R2$ but weaker balance on geometric grounding and localization. The 8B-12B comparison is especially informative because it removes large-scale parameter advantages and still exposes markedly different capability profiles, separating balanced models from systems with narrower strengths in distance perception, spatial relations, or temporal reasoning. Among similarly sized models, MiniCPM-V-4.5(9B) is strong on distance perception but weaker on temporal reasoning, Gemma-3-it(12B) is the most balanced and leads on $R3_2$, Pixtral(12B) is relatively strong on spatial relations but weak on multi-view memory, and Qwen3-VL(8B) offers one of the best overall trade-offs at this scale. The open-loop benchmark therefore measures structured capability variation, rather than merely reproducing a generic VLM ranking.

4.3 Capability Correlation and Weight Allocation

Empirical Correlation Structure. Figure 5a shows that DRIVEHIERARCHY exhibits structured, but non-collapsed, capability dependencies. At the subtask level, semantically adjacent abilities form clear local clusters: nearest-object and referred-object distance estimation are aligned ($\rho = 0.961$), and counting-oriented recognition is strongly coupled with distance-bucket estimation ($\rho = 0.875$). In contrast, state recognition and outcome prediction ($\rho = 0.004$), as well as localization and state recognition ($\rho = 0.011$), are nearly independent, which captures both shared structure and genuine capability separation, rather than reducing to either a set of unrelated tasks or a single latent score. The rank-level correlations reinforce this interpretation: $R1$ and $R2$ are strongly associated ($\rho = 0.843$), whereas their relations to $R3$ are substantially weaker ($\rho = 0.346$ and $\rho = 0.425$), indicating that higher-order reasoning builds on lower-level competence without being reducible to it.

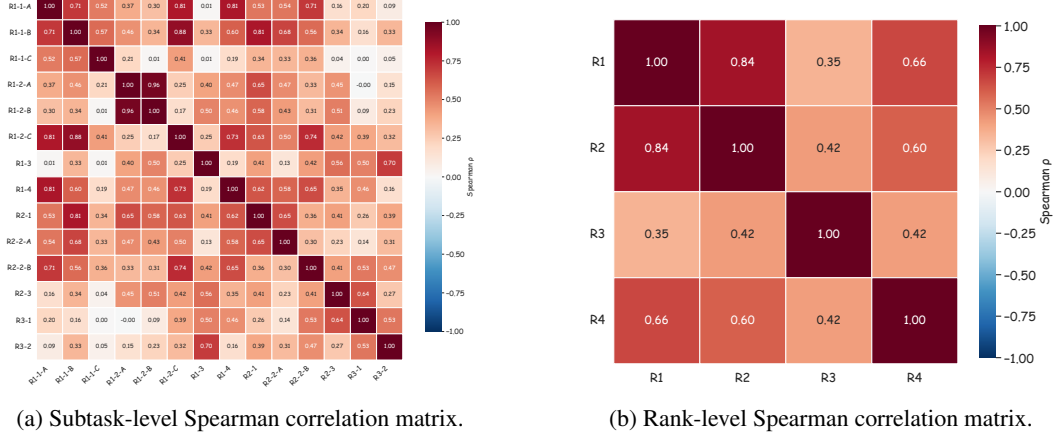


Figure 5: **Correlation structure of DriveHierarchy across subtasks and ranks**, which shows that the benchmark contains both locally coupled abilities and globally progressive capability levels.

Correlation-Informed Weight Allocation. For the released benchmark, we adopt a correlation-aware weighting scheme, with the full methodology and implementation details given in Appendix F. We first fix the rank budgets to $R1:R2:R3 = 3:3:4$, and then allocate each budget according to subtask uniqueness derived from the correlation matrix. These correlations are estimated once during an initial calibration phase after the first benchmark-testing round, and the resulting subtask weights are then frozen for all subsequent evaluations. Highly redundant subtasks therefore receive smaller fixed weights, while subtasks carrying more independent signal receive larger fixed shares. This logic is visible in Table 1: within $R1$, the less redundant $R1_1_C$ and $R1_3$ are weighted more heavily than the tightly coupled counting and distance variants, and within $R2$, the relational task $R2_3$ receives the largest share. The resulting composite score is therefore designed to preserve distinct capability dimensions rather than repeatedly rewarding the same underlying competence. Meanwhile, task-level and rank-level profiles remain the primary diagnostic outputs, especially when comparing models whose weighted averages are close.

4.4 From Open-Loop Capability to Closed-loop Driving

Table 2 reports the Closed-loop results on $R4$, showing that Closed-loop driving is meaningfully related to open-loop capability, but is not determined by it. At the model level, the strongest $R4$ systems, such as Gemma-3-it(27B) and Qwen2.5-VL(72B), also belong to the strongest open-loop models, indicating that embodied driving performance remains grounded in the same capability spectrum measured by $R1$ - $R3$. This interpretation is consistent with the rank-level correlations shown in Figure 5b, where the Spearman coefficients with $R4$ are 0.664 for $R1$, 0.596 for $R2$, and 0.418 for $R3$. These values establish that stronger open-loop capability is generally associated with stronger closed-loop driving, while also showing that the association weakens as one moves from grounding and scene integration to the current form of higher-level reasoning. This indicates that open-loop evaluation is behaviorally valid but not behaviorally sufficient. For current VLM-based driving systems, lower- and mid-level abilities remain more predictive of closed-loop success than the present $R3$ tasks, suggesting that robust grounding and scene integration are still the dominant bottlenecks for embodied performance. From the perspective of training optimization, $R1$ - $R3$ provide a principled basis for selecting supervision targets to improve closed-loop driving.

4.5 Benchmark-Guided SFT

To evaluate whether DriveHierarchy can guide model improvement, we adopt Qwen3-VL(8B) as the base model and derive SFT targets from the task groups with the largest deficits relative to task-wise SOTA performance[55]. Rather than emphasizing the expected gains on directly supervised tasks, we focus on transfer beyond the fine-tuning scope, asking whether capability-targeted supervision leads to measurable improvements on non-target question types and downstream closed-loop driving. The first row of Table 3 reports this deficit profile solely as a reference for target selection.

Table 3: **Open-loop transfer results of benchmark-guided SFT on Qwen3-VL(8B)**. The first row reports the baseline deficit relative to the task-wise SOTA result and is used only for target selection.

Models	Avg.Score	R1 Perceptual Grounding							R2 Contextual Memory					R3 Mental Reasoning	
		R1_1_A	R1_1_B	R1_1_C	R1_2_A	R1_2_B	R1_2_C	R1_3	R1_4	R2_1	R2_2_A	R2_2_B	R2_3	R3_1	R3_2
		0.03	0.027	0.056	0.035	0.036	0.029	0.056	0.031	0.064	0.079	0.067	0.09	0.233	0.167
Qwen3-VL(8B)	-0.17	-0.05	-0.14	-0.16	-0.26	-0.34	-0.15	-0.98	-0.12	-0.33	-0.15	-0.06	-0.39	-0.28	-0.09
Baseline*	45.10	70.00	33.32	51.20	46.70	46.76	35.74	1.05	50.23	32.87	41.25	53.60	37.70	56.90	45.84
B+R1_2_A	+2.66	+7.70	+12.78	-1.20	+46.54	+14.71	+5.86	0.00	+1.06	+0.17	+5.49	-1.00	-1.20	-2.30	+0.37
B+R1_2_B	+3.57	+5.50	+10.91	+1.80	+35.78	+47.82	+4.91	+0.01	-2.09	+1.99	+3.03	+1.10	-1.30	-1.80	+0.32
B+R1_3	+2.77	-3.70	+1.50	-2.40	-27.97	-3.23	+3.56	+79.45	+1.07	-4.99	-2.58	-1.10	-3.60	+1.50	+0.37
B+R1_2_A+ R1_2_B+R1_3	+2.14	+4.90	+7.35	-4.60	+45.49	+46.73	+3.91	+0.01	-4.40	-7.12	+2.13	+0.80	-9.30	-0.80	+0.35
B+R2_1	+4.05	+0.30	+7.02	+0.90	+1.39	-0.72	+6.74	+0.01	-5.41	+43.71	+5.67	0.00	+2.30	+0.50	+1.10
B+R2_3	+0.53	+4.70	+2.21	-0.60	+11.17	+3.14	+1.30	+0.01	-8.93	-10.01	-3.32	-7.20	+61.20	-17.70	+0.59
B+R2_1+R2_3	+0.42	+3.00	+2.21	+0.20	+8.85	+4.31	-0.51	0.00	-8.06	-0.28	-2.69	-9.50	+61.10	-19.90	+0.40
B+R3_1	+7.12	-5.60	-1.33	-5.00	-3.26	-6.67	+3.06	-0.01	-5.70	-9.05	-4.13	-1.00	-5.40	+40.50	+0.45
B+ALL	+20.88	+6.70	+12.35	-6.70	+44.33	+46.69	+3.14	+79.74	-2.73	-0.87	+1.65	-0.50	+57.40	+33.20	+0.46

Table 4: **Closed-loop transfer of benchmark-guided supervised fine-tuning on R4**. Each intervention is defined by capability groups selected from DriveHierarchy.

Models	Scce 1	Scce 2	Scce 3	Scce 4	Scce 5	Scce 6	Scce 7	Scce 8	Scce 9	Scce 10	Avg.Score
Baseline	1.278	1.726	0.384	1.128	0.574	0.271	1.379	0.394	0.661	1.227	0.902
B+R1_2_A+ R1_2_B+R1_3	9.436	4.788	0.816	3.199	2.418	8.468	5.048	1.924	9.958	3.379	4.943
B+R2_1+R2_3	15.181	3.279	0.207	4.095	1.778	8.579	4.299	2.176	9.924	2.192	5.171
B+R3_1	2.453	3.512	0.122	2.657	1.762	17.400	4.176	2.499	9.477	1.892	4.595
B+ALL	22.585	3.146	0.734	15.888	2.144	3.694	11.855	2.340	16.717	1.105	8.021

Single-Capability Adaptation and Cross-Ability Transfer. Table 3 shows that benchmark-guided SFT produces structured transfer rather than merely improving the directly supervised task. While in-scope gains are expected by construction, the more informative signal is the response of non-target question types outside the selected supervision group. For example, after supervising *R1_2_B*, 10 of the 13 non-target abilities increase; the strongest spillovers are *R1_2_A* (+35.78), *R1_1_B* (+10.91), and *R2_2_A* (+3.03), whereas the largest non-target decrease is only -2.09 . Similar transfer patterns also appear under other interventions. Even the more localized interventions do not produce a rank-wide collapse. For example, supervising *R3_1* still raises *R1_2_C* by +3.06 and *R3_2* by +0.45, while supervising *R1_3* raises *R1_1_B* and *R3_1* by +1.50 each. Taken together, these results support the interpretation that the capabilities are sufficiently differentiated for targeted intervention that optimizing one does not induce systematic degradation elsewhere, yet sufficiently coupled that targeted supervision often produces positive cross-ability spillover, which can provide guidance for training optimization, showing that the hierarchy can be used to probe and compare cross-capability transfer rather than merely to report in-scope task repair.

From Open-Loop Improvement to Closed-loop Gain. These benchmark-guided interventions transfer to embodied driving. Importantly, *R4* is never used as a supervision target in these experiments and all reported closed-loop changes therefore reflect transfer from open-loop capability supervision. Table 4 shows that every selected intervention yields a substantial gain on *R4* over the baseline score of 0.902, and that the magnitude of the gain depends on which capability group is optimized. Strengthening the selected *R2* capabilities raises the closed-loop score to 5.171 (+4.269), slightly exceeding the gain from the selected *R1* group, which reaches 4.943 (+4.041), while direct supervision on *R3_1* improves *R4* to 4.595 (+3.693). Joint optimization over the benchmark-identified weak set produces the largest effect, increasing the closed-loop score to 8.021 (+7.119). These results suggest that the open-loop capability deficits identified by DriveHierarchy are actionable optimization targets for improving closed-loop driving rather than merely descriptive diagnostics. These results indicate that the hierarchy can help select open-loop supervision targets whose effects propagate to downstream closed-loop behavior, rather than treating *R4* as an opaque end metric.

5 Conclusion

This paper presented DRIVEHIERARCHY, a hierarchical benchmark for VLM-based autonomous driving that organizes evaluation from perceptual grounding to closed-loop execution and shifts

driving evaluation from flat model comparison toward capability-structured diagnosis. Its central value lies not only in measuring performance, but also in making the internal organization of driving ability explicit and relating open-loop competence to downstream behavior, with a targeted SFT case study further illustrating how such diagnoses may inform model intervention. We hope this benchmark can serve as a practical framework for understanding and advancing VLM-based autonomous driving systems and for studying how open-loop capabilities relate to closed-loop performance.

Acknowledgments and Disclosure of Funding

This work was supported by National Natural Science Foundation of China (52125208).

References

- [1] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024.
- [2] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer, 2024.
- [3] Chengkai Xu, Yiming Cui, Jiaqi Liu, Cheng Qin, Geyuan Zhang, Xinwei Dong, Shiyu Fang, Yicheng Guo, Peng Hang, and Jian Sun. A survey on end-to-end autonomous driving training from the perspectives of data, strategy, and platform. *Authorea Preprints*, 2025.
- [4] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8795–8801. IEEE, 2025.
- [5] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21983–21994, 2023.
- [6] Sicong Jiang, Zilin Huang, Kangan Qian, Ziang Luo, Tianze Zhu, Yang Zhong, Yihong Tang, Menglin Kong, Yunlong Wang, Siwen Jiao, et al. A survey on vision-language-action models for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4524–4536, 2025.
- [7] Chengkai Xu, Jiaqi Liu, Shiyu Fang, Yiming Cui, Dong Chen, Peng Hang, and Jian Sun. Tell-drive: Enhancing autonomous driving with teacher llm-guided deep reinforcement learning. *arXiv preprint arXiv:2502.01387*, 2025.
- [8] 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. *arXiv preprint arXiv:2506.13757*, 2025.
- [9] Wenhui Huang, Songyan Zhang, Qihang Huang, Zhidong Wang, Zhiqi Mao, Collister Chua, Zhan Chen, Long Chen, and Chen Lv. Automot: A unified vision-language-action model with asynchronous mixture-of-transformers for end-to-end autonomous driving. *arXiv preprint arXiv:2603.14851*, 2026.
- [10] Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025.
- [11] Chengkai Xu, Jiaqi Liu, Yicheng Guo, Peng Hang, and Jian Sun. A knowledge-driven diffusion policy for end-to-end autonomous driving based on expert routing. *arXiv preprint arXiv:2509.04853*, 2025.
- [12] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pages 292–308. Springer, 2024.
- [13] Jingyuan Zhao, Yuyan Wu, Rui Deng, Susu Xu, Jinpeng Gao, and Andrew Burke. A survey of autonomous driving from a deep learning perspective. *ACM Computing Surveys*, 57(10):1–60, 2025.
- [14] Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are vlms ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6585–6597, 2025.
- [15] Ana-Maria Marcu, Long Chen, Jan Hünemann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pages 252–269. Springer, 2024.
- [16] Xianhui Meng, Yuchen Zhang, Zhijian Huang, Zheng Lu, Ziling Ji, Yaoyao Yin, Hongyuan Zhang, Guangfeng Jiang, Yandan Lin, Long Chen, et al. Is your vlm for autonomous driving safety-ready? a comprehensive benchmark for evaluating external and in-cabin risks. *arXiv preprint arXiv:2511.14592*, 2025.

- [17] JB Carroll. A theory of cognitive abilities: The three-stratum theory. *Human cognitive abilities: A survey of factor-analytic studies*, pages 631–655, 1993.
- [18] Yuxi Xiao, Longfei Li, Shen Yan, Xinhang Liu, Sida Peng, Yunchao Wei, Xiaowei Zhou, and Bingyi Kang. Spatialtree: How spatial abilities branch out in mlms. In *The First Workshop on Efficient Spatial Reasoning*.
- [19] Jamie P Eggers and Sarah Kaplan. Cognition and capabilities: A multi-level perspective. *Academy of Management annals*, 7(1):295–340, 2013.
- [20] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12037–12047, 2025.
- [21] Shuo Xing, Chengyuan Qian, Yuping Wang, Hongyuan Hua, Kexin Tian, Yang Zhou, and Zhengzhong Tu. Openemima: Open-source multimodal model for end-to-end autonomous driving. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1001–1009, 2025.
- [22] Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, pages 403–420. Springer, 2024.
- [23] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pages 256–274. Springer, 2024.
- [24] Jiajun Cao, Qizhe Zhang, Peidong Jia, Xuhui Zhao, Bo Lan, Xiaolan Zhang, Xiaobao Wei, Sixiang Chen, Liyun Li, Xianming Liu, et al. Fastdrivevla: Efficient end-to-end driving via plug-and-play reconstruction-based token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 2571–2579, 2026.
- [25] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 13782–13790, 2026.
- [26] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11993–12003, 2025.
- [27] Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkan Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 24823–24834, 2025.
- [28] Yue Li, Meng Tian, Zhenyu Lin, Jiangtong Zhu, Dechang Zhu, Haiqiang Liu, Yueyi Zhang, Zhiwei Xiong, and Xinhai Zhao. Fine-grained evaluation of large vision-language models in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9431–9442, 2025.
- [29] Xu Cao, Tong Zhou, Yunsheng Ma, Wenqian Ye, Can Cui, Kun Tang, Zhipeng Cao, Kaizhao Liang, Ziran Wang, James M Rehg, et al. Maplm: A real-world large-scale vision-language benchmark for map and traffic scene understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21819–21830, 2024.
- [30] Christian Fruhwirth-Reisinger, Dušan Malić, Wei Lin, David Schinagl, Samuel Schuster, and Horst Possegger. Stsbench: A spatio-temporal scenario benchmark for multi-modal large language models in autonomous driving. *arXiv preprint arXiv:2506.06218*, 2025.
- [31] Ruiyang Hao, Bowen Jing, Haibao Yu, and Zaiqing Nie. Styledrive: Towards driving-style aware benchmarking of end-to-end autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 4627–4635, 2026.
- [32] Haibao Yu, Wenxian Yang, Ruiyang Hao, Chuanye Wang, Jiaru Zhong, Ping Luo, and Zaiqing Nie. Drivee2e: Closed-loop benchmark for end-to-end autonomous driving through real-to-simulation. *arXiv preprint arXiv:2509.23922*, 2025.
- [33] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

- [34] Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. *Advances in Neural Information Processing Systems*, 35:25667–25682, 2022.
- [35] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.
- [36] Ian J Deary, Simon R Cox, and W David Hill. Genetic variation, brain, and intelligence differences. *Molecular psychiatry*, 27(1):335–353, 2022.
- [37] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li E Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024.
- [38] Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.
- [39] Harsh Agrawal, Eldon Schoop, Xinlei Pan, Anuj Mahajan, Ari Seff, Di Feng, Ruijia Cheng, Andres Romero Mier Y Teran, Esteban Gomez, Abhishek Sundararajan, et al. Uinavbench: A framework for comprehensive evaluation of interactive digital agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23353–23363, 2025.
- [40] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospacial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15768–15780, 2025.
- [41] Qineng Wang, Wenlong Huang, Yu Zhou, Hang Yin, Tianwei Bao, Jianwen Lyu, Weiyu Liu, Ruohan Zhang, Jiajun Wu, Li Fei-Fei, et al. Enact: Evaluating embodied cognition with world modeling of egocentric interaction. *arXiv preprint arXiv:2511.20937*, 2025.
- [42] Pingyue Zhang, Zihan Huang, Yue Wang, Jieyu Zhang, Letian Xue, Zihan Wang, Qineng Wang, Keshigeyan Chandrasegaran, Ruohan Zhang, Yejin Choi, et al. Theory of space: Can foundation models construct spatial beliefs through active exploration? In *The Fourteenth International Conference on Learning Representations*, 2026.
- [43] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.
- [44] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187, 2002.
- [45] Qwen Team. Qwen3 technical report, 2025.
- [46] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.
- [47] Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, Zhuosheng Zhang, and Weiran Huang. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*, 2026.
- [48] Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, Wenhao Hu, Bingxiang He, Jie Zhou, Jie Cai, Ji Qi, Zonghao Guo, Chi Chen, Guoyang Zeng, Yuxuan Li, Ganqu Cui, Ning Ding, Xu Han, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe, 2025.
- [49] Gemma Team. Gemma 3. 2025.
- [50] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. Pixtral 12b. *arXiv preprint arXiv:2410.07073*, 2024.

- [51] Qwen Team. Qwen2.5: A party of foundation models, September 2024.
- [52] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.
- [53] Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. Mini-internvl: A flexible-transfer pocket multimodal model with 5% parameters and 90% performance. *arXiv preprint arXiv:2410.16261*, 2024.
- [54] Amirhosein Chahe and Lifeng Zhou. Reasondrive: Efficient visual question answering for autonomous vehicles with reasoning-enhanced small vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3870–3879, 2025.
- [55] Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. Swift:a scalable lightweight infrastructure for fine-tuning, 2024.
- [56] Yiming Cui, Shiyu Fang, Jiarui Zhang, Yan Huang, Chengkai Xu, Bing Zhu, Hao Zhang, Peng Hang, and Jian Sun. Vp-autotest: A virtual-physical fusion autonomous driving testing platform. *arXiv preprint arXiv:2512.07507*, 2025.
- [57] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [58] Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 629–636. IEEE, 2024.
- [59] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [60] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.
- [61] Xiaosong Jia, Yuqian Shao, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive-vl: Benchmarks for closed-loop autonomous driving with vision-language models. *arXiv preprint arXiv:2604.01259*, 2026.

A Limitations, Broader impacts and Future work

Limitations. DRIVEHIERARCHY still has several important limitations. Firstly, although *R4* is constructed on a real-road-layout co-simulation platform, it remains a simulator-based closed-loop benchmark and therefore cannot be interpreted as a direct proxy for real-world deployment performance. Secondly, the current hierarchy is broad but not exhaustive. For example, higher-level reasoning is represented by a limited number of task families, leaving some capabilities under-covered. Thirdly, the composite open-loop score depends on a correlation-aware weighting scheme that is more principled than uniform averaging but still reflects a modeling choice rather than a uniquely determined ground truth. Fourth, given that repeated evaluations are computationally expensive, the current experiments report only deterministic single-run scores. Consequently, minor differences between models should be interpreted with caution. Finally, the benchmark-guided optimization results, while informative, are established on a single base model and should therefore be interpreted as evidence of practical utility rather than as a universal characterization of capability transfer.

Broader Impacts. The primary positive impact of DRIVEHIERARCHY is to make evaluation of VLM-based autonomous driving more diagnostic, interpretable, and actionable by exposing the internal capability structure that underlies downstream driving behavior. This may help researchers move beyond flat success metrics, identify concrete failure modes, and design more targeted optimization strategies. At the same time, the benchmark should be used with appropriate caution. Strong performance on DRIVEHIERARCHY, including strong *R4* results, does not imply readiness for real-world autonomous driving deployment, and benchmark-oriented optimization may encourage overfitting to the task distributions, annotations, or scenario designs represented here. More broadly, because the benchmark is built from public datasets and simulation infrastructure, any omissions, biases, or simplifications in those resources may propagate into model development and evaluation. DRIVEHIERARCHY should therefore be treated as a research instrument for capability analysis and controlled comparison, rather than as a certification standard for safety-critical deployment.

Future Work. Several works are going to be conducted following this work. At the platform level, the current interactive scene editor for *R4* shown in Fig 3 remains at an early stage, although it already supports integration with hardware components and real test vehicles[56]; an important next step is therefore to mature this infrastructure and optimize it specifically for end-to-end autonomous driving. Building on this foundation, we plan to refine the virtual-real fusion testing system that couples controllable simulation with physical testing and larger-scale replay- or vehicle-based validation, thereby strengthening the connection between benchmark performance and real-world driving behavior. At the methodology level, the weighting and aggregation scheme can be further strengthened through robustness analysis, alternative composite-score constructions, and uncertainty estimates over repeated scenario executions; the generality of benchmark-guided intervention should also be examined across more model families, training recipes, and intervention strategies. More broadly, DRIVEHIERARCHY is positioned as a dynamic diagnostic and analytical framework, rather than merely a static leaderboard. We plan to maintain and extend it over time so that it supports sustained analysis of capability structure instead of becoming a narrow optimization target.

B Benchmark Construction Details

The open-loop component of DRIVEHIERARCHY is constructed by integrating multiple public autonomous driving datasets, including NuScenes[57], NuPlan[58], NAVSIM[59], DriveLM[23], DriveBench[14], LingoQA[15], and DRAMA[60], and converting them into structured multimodal evaluation items aligned with the capability hierarchy. The resulting corpus contains 76,798 question-answer pairs over 84,279 frames, spanning recognition, grounding, memory, reasoning, and planning. From this corpus, we further derive a refined evaluation set containing 14,000 question-answer pairs over 31,945 frames to enable more compact and controlled benchmarking. The closed-loop component is built on the interactive CARLA-SUMO co-simulation platform and contains 100 curated driving scenarios.

Artifact Availability. For anonymous review, the project release at `anonymous.4open.science/r/DriveHierarchy-2FD5` separates redistributable benchmark assets from upstream resources that must be obtained under their original licenses. The release includes benchmark metadata,

Table 5: **Comparison with representative autonomous-driving benchmarks.** “Hier.” indicates whether the benchmark explicitly organizes evaluation into a capability hierarchy. “Diag.” indicates whether it exposes capability-level diagnostic outputs rather than only task or route scores.

Benchmark	Open QA	Closed loop	VLM iface.	Hier.	Diag.	Primary focus
DriveLM [23]	✓	✗	✓	✗	△	Language-grounded driving QA
LingoQA [15]	✓	✗	✓	✗	✗	Language-grounded driving QA
DriveBench [14]	✓	✗	✓	✗	△	The Visual Robustness of VLMs
NAVSIM [35]	✗	△	✗	✗	✗	Planning-oriented simulation
SafeBench [34]	✗	✓	✗	✗	△	Safety-critical scenarios
Bench2Drive [33]	✗	✓	✗	✗	△	Closed-loop driving agents
Bench2Drive-VL [61]	✓	✓	✓	✗	△	Closed-loop VLM driving agents
DRIVEHIERARCHY	✓	✓	✓	✓	✓	Capability-structured diagnosis

Table 6: **Construction-quality controls for the refined open-loop set.** The released manifest stores per-item source identifiers and answer labels so that users can audit source proportions and answer distributions for each task.

Task group	Filtering and balance controls	Validation signal
Recognition/counting	visibility projection checks, category vocabulary normalization, scene-level quotas	nonnegative labels, valid category IDs, duplicate-item checks
Distance/grounding	calibrated ego-object geometry, projected-coordinate validity, invalid-box removal	finite distances, in-frame boxes, IoU-compatible box format
Situation description	LingoQA-context filtering for current-scene and ego-action semantics	learned-judge compatibility and prompt/answer schema checks
Multi-view/temporal memory	overlap trigger, fixed clip windows, instance-ID deduplication, transition balancing	valid synchronized views, valid temporal order, instance continuity checks
Relation/reasoning/planning	minimum visible-object constraints, randomized choices, nontrivial motion clips	answer-choice consistency, source-scene sanity checks, order-label validation

task annotations permitted for redistribution, prompt templates, scoring code, closed-loop scenario configurations, reproduction scripts, and component-level logs for route, safety, efficiency, and termination status. For large or license-restricted resources, the release provides inspectable samples and reconstruction scripts rather than redistributing raw upstream data.

Reproduction Contract. Re-running the full benchmark requires obtaining the upstream datasets and model checkpoints listed in Table 11; the release provides reconstruction scripts, split manifests, prompt templates, parser/scoring code, and scenario configurations. The package pins the exact software environment in environment files, including Python, CUDA, vLLM, SWIFT, CARLA, SUMO, and benchmark-code versions, and provides commands for (i) rebuilding the refined *R1-R3* manifests, (ii) recomputing open-loop scores, (iii) replaying *R4* scenarios, and (iv) aggregating route, safety, efficiency, termination, and parse-status logs. The released annotations and code use the project license specified in the repository, while raw upstream assets remain governed by their original licenses and terms.

C Experiment details and reproducibility

C.1 Model Configurations

Table 7 lists the exact checkpoints used in the open-loop evaluation. Unless otherwise stated, all models are loaded from their official open-source releases, and the short names reported in the main paper are aliases for the full checkpoints shown here. The reported GPU counts correspond to the tensor-parallel configuration used during inference on NVIDIA A800 GPUs.

C.2 Inference and Evaluation Details

Each sample is instantiated as a multimodal chat input composed of a fixed system prompt and a task-specific user prompt. Depending on the task, the visual input may consist of a single image, a synchronized multi-view set, or a short temporal sequence. To keep evaluation deterministic and parsing-stable across heterogeneous task types, the output format is constrained at the task level:

Table 7: **Model configurations used in the open-loop evaluation.**

Models	Provider	Full name	Inference Hardware
Qwen3-VL(2B)	Qwen	Qwen3-VL-2B-Instruct	1 × A800
InternVL3_5(2B)	OpenGVLab	InternVL3_5-2B-Instruct	1 × A800
Qwen3-VL(8B)	Qwen	Qwen3-VL-8B-Instruct	1 × A800
InternVL3_5(8B)	OpenGVLab	InternVL3_5-8B-Instruct	1 × A800
ZwZ(8B)	inclusionAI	ZwZ-8B	1 × A800
MiniCPM-V-4_5(9B)	openbmb	MiniCPM-V-4_5	1 × A800
Gemma-3-it(12B)	google	Gemma-3-12b-it	1 × A800
Pixtral(12B)	mistralai	Pixtral-12B-2409	1 × A800
Gemma3-it(27B)	google	Gemma-3-27b-it	2 × A800
Qwen3-VL(32B)	Qwen	Qwen3-VL-32B-Instruct	2 × A800
InternVL3_5(38B)	OpenGVLab	InternVL3_5-38B-Instruct	2 × A800
Qwen2.5-VL(72B)	Qwen	Qwen2.5-VL-72B-Instruct	4 × A800
InternVL3(78B)	OpenGVLab	InternVL3-78B	4 × A800
DA-DriveLM(4B)	OpenGVLab	Mini-InternVL2-4B-DA-DriveLM	1 × A800
ReasonDrive(7B)	ac4462	Qwen2.5-VL-7B-DriveLM	1 × A800

binary tasks require a yes/no response, numeric tasks require a bare number, multiple-choice tasks require only the selected option label, grounding tasks require a bounding box, and sequence-ordering tasks require an ordered list of labels. Multi-image inputs are presented in a fixed order, and temporal-order tasks are explicitly indexed so that the model receives a consistent visual-textual interface across all $R1$ - $R3$ evaluations.

Inference is conducted under a unified deterministic decoding protocol. The principal settings are summarized in Table 8.

Table 8: **Open-loop inference settings.**

Symbol	Meaning	Value
T	Decoding temperature	0.0
p	Nucleus sampling threshold	1.0
N_{gen}	Maximum generated tokens	256
λ_{rep}	Repetition control coefficient	1.0
λ_{pre}	Presence penalty coefficient	0.0
B	Number of samples per inference batch	2
s	Random seed	0
η_{mem}	Fraction of GPU memory reserved for serving	0.90
L_{ctx}	Maximum model context length	16384

Predictions are stored in per-model result files, and partially completed runs are resumed by skipping previously processed evaluation items. Samples with missing visual inputs, malformed records, or failed generations are logged, excluded from scoring, and re-run when evaluation resumes from a breakpoint. This protocol ensures that the open-loop evaluation remains deterministic, restartable, and auditable while preserving a uniform interface across all evaluated models.

C.3 Closed-Loop Evaluation Setting

The reported $R4$ results are obtained under a unified closed-loop runtime protocol in which each model is deployed as an online VLM service and evaluated in a fresh CARLA-SUMO co-simulation instance for every scenario. For fair comparison, the scenario loop, simulator configuration, observation interface, and termination rules are fixed across models, and only the model-specific serving configuration is changed. Several key parameters are summarized in Table 9. At runtime, the controller queries the VLM service through an HTTP endpoint, receives action decisions at a fixed control frequency, and applies them directly to the ego vehicle while SUMO governs the surrounding

traffic flow. Each scenario is evaluated independently to ensure that simulator state, traffic evolution, and control history do not carry over across runs.

Table 9: **Closed-loop runtime settings used for the *R4* benchmark.**

Setting	Meaning	Value
Hardware	Evaluation node configuration for CARLA	$1 \times A800$
Control rate	Frequency of VLM action execution	4 Hz
Simulation step	Fixed simulator step length	0.05 s
Conversation window	Number of retained dialogue turns	1
image size	Front-view image resolution sent to the model	800×450
Off-road grace time	Allowed off-road duration before termination	1.5 s

To ensure that no mutual interference occurs during scene execution, the CARLA server is restarted for every scenario, the simulator and VLM service will also be checked whether they are healthy before rollout, and results are written to a scenario-specific output directory. The controller runs with an optional chase-view ego camera, a BEV spectator view, and explicit route visualization, while the scripted traffic actors are managed by the synchronized CARLA-SUMO backend. A run terminates when the ego vehicle reaches the destination or encounters a benchmark-defined stopping condition, including collision, persistent off-road behavior, blocked driving, or scenario timeout. Completed scenarios are skipped automatically during resumed evaluation, whereas scenarios that produce control traces without valid VLM inference cause the batch to terminate immediately, which keeps the *R4* evaluation restartable, while preventing partially failed runs or stale simulator state from contaminating the reported results.

C.4 Model-to-Control Interface for *R4*

The *R4* interface converts a general VLM into an executable closed-loop driving policy without introducing model-specific controllers. At each decision step, the runner captures the ego front-view image and a bird’s-eye-view route image from CARLA. It also extracts structured state from the simulator, including ego speed, the remaining route, the distance to destination, a coarse navigation command, and short-horizon future route points expressed in ego coordinates. The default online chain uses the action-decision question, while an optional waypoint-prediction question can be enabled for analyses that require waypoint-style intermediate outputs.

The textual query is instantiated by prepending a live driving condition to the fixed prompt below, where variables are filled from the current CARLA state and route sampler.

Dynamic condition prefix. The ego vehicle is driving at the speed of v_t m/s, and it wants to [navigation command]. Remaining route distance to destination is approximately d_t meters. The local future route in ego coordinates is given as (`forward_m`, `right_m`): [future route points].

Action-decision prompt. Your primary objective is to reach the destination in the minimum time while remaining safe and legal. Hard constraints: do not collide with vehicles, pedestrians, or static obstacles; do not leave the drivable route; obey traffic lights, stop signs, and right-of-way rules. Driving policy: do not crawl or wait without a concrete safety or rule-based reason; if the route ahead is clear, actively increase speed toward the a safe speed; use steering proactively to follow the future path and begin steering before turns instead of defaulting to zero steering; slow down when required by route geometry, traffic rules, obstacles, or collision risk. For the steer value, a negative value indicates left, while a positive value indicates right. When making the decision, consider the ego speed, the future route geometry, drivable space, nearby vehicles, pedestrians, traffic lights, traffic signs, and any immediate collision or off-road risk. Output only the final action decision without explanation in the exact format: `steer: <float in [-1.0, 1.0]>; target_speed_mps: <float in [0.0, v_max]>`.

The model response is parsed as a compact action decision consisting of steering intent $\hat{s}_t$ and target speed $\hat{v}_t$. The bridge clips $\hat{s}_t$ to $[-1, 1]$, clips $\hat{v}_t$ to $[0, v_{max}]$ m/s, rate-limits steering by 0.12 per control update, and converts the target-speed error $e_t = \hat{v}_t - v_t$ into a CARLA command

$$u_t = (\delta_t, \tau_t, b_t):$$

$$\delta_t = \text{clip}(\hat{s}_t, \delta_{t-1} - 0.12, \delta_{t-1} + 0.12) \quad (2)$$

$$\tau_t = \mathbf{1}[e_t \geq 0] \min(0.85, 0.25 + 0.22e_t) \quad (3)$$

$$b_t = \mathbf{1}[e_t < 0] \min(0.85, 0.30|e_t|) \quad (4)$$

If the response cannot be parsed, the previous valid control is retained and the failure is recorded. The resulting command is applied to the ego vehicle at the fixed $R4$ control rate, with the last command held between model queries. This interface isolates the capability under evaluation from simulator-specific implementation details. All models receive the same visual inputs, route context, query protocol, parsing rules, and actuation bridge; only the model backend differs.

C.5 Benchmark-Guided SFT Details

All benchmark-guided SFT experiments are conducted on the base model Qwen/Qwen3-VL-8B-Instruct[45] using LoRA-based SFT, launched through the swift-sft[55] pipeline. A validation split of 5% is held out from each training dataset. Across all runs, the vision tower and aligner are frozen, LoRA adapters are attached to all linear layers, and the same optimization recipe is used to ensure the differences in performance can be attributed to the selected capability supervision rather than to changing training conditions. The analytical focus of this study is transfer beyond the supervised scope: in the open-loop analysis, we pay primary attention to changes on non-target question types, and in the closed-loop analysis, we test whether open-loop supervision transfers to $R4$. No $R4$ scenario is used for training or fine-tuning in any reported experiment. The main parameters used during training are shown in Table 10.

Table 10: Training configuration for benchmark-guided SFT.

Symbol	Meaning	Value
r	LoRA rank	8
α	LoRA scaling factor	32
d_{train}	Training dtype	bfloat16
B_{train}	Per-device training batch size	1
B_{eval}	Per-device evaluation batch size	1
G	Gradient accumulation steps	8
η	Learning rate	1×10^{-4}
ρ_{warm}	Warmup ratio	0.05
L_{max}	Maximum training sequence length	16384
N_{img}	Maximum visual token budget per sample	1024
S_{save}	Checkpoint interval	50
S_{eval}	Validation interval	50
$\mathcal{H}$	Training hardware	4×A800

All A800 GPUs used in this work are 80GB variants. Compute requirements vary across models due to differences in model size, tensor-parallel configuration, and serving throughput. For the reported benchmark runs, open-loop evaluation requires approximately 24 hours per model on average, with a total cost of roughly 48 to 144 A800 GPU-hours. Closed-loop evaluation over the 100 scenarios requires approximately 20 hours of wall-clock time per model on average, with a total cost of roughly 36 to 72 A800 GPU-hours.

D Open-Loop Evaluation Metrics Designs

For binary judgment and multiple-choice tasks, we use exact-match scoring,

$$s_i = 100 \cdot \mathbf{1}[\hat{y}_i = y_i]. \quad (5)$$

This protocol is used for existence recognition, state identification, spatial relation reasoning, and outcome prediction. In implementation, yes/no answers are normalized to canonical binary labels, while

multiple-choice responses are first matched by option letter and then, when needed, by normalized answer text. Unparseable outputs receive score 0 and are logged as parse failures.

For counting tasks, we first extract the valid numeric token from the response, round it to the nearest non-negative integer, and define the normalized error

$$E_i^{\text{cnt}} = \frac{|\hat{y}_i - y_i|}{\max(y_i, c)}, \quad (6)$$

where $c > 0$ prevents the denominator from becoming unstable for small counts. We fix $c = 1$, $\tau_{\text{cnt}} = 0.05$, and choose σ_{cnt} from the half-credit target $E_{50}^{\text{cnt}} = 0.20$, yielding

$$\sigma_{\text{cnt}} = \frac{0.20 - 0.05}{\ln 2}. \quad (7)$$

For distance estimation, we extract the first valid numeric token, clip it to a non-negative scalar, and adopt a scale-invariant log-ratio error

$$E_i^{\text{dist}} = \left| \log \frac{\hat{y}_i + \epsilon}{y_i + \epsilon} \right|, \quad (8)$$

where $\epsilon > 0$ avoids numerical singularity near zero distance. We fix $\epsilon = 0.1$, $\tau_{\text{dist}} = 0$, and set the half-credit target to $E_{50}^{\text{dist}} = \ln 2$, which gives $\sigma_{\text{dist}} = 1.0$. Both errors are converted into scores through the same bounded decay function

$$G(E; \tau, \sigma) = \begin{cases} 100, & E \leq \tau, \\ 100 \exp\left(-\frac{E - \tau}{\sigma}\right), & E > \tau, \end{cases} \quad (9)$$

yielding $s_i = G(E_i^{\text{cnt}}; \tau_{\text{cnt}}, \sigma_{\text{cnt}})$ for counting and $s_i = G(E_i^{\text{dist}}; \tau_{\text{dist}}, \sigma_{\text{dist}})$ for distance estimation. This design assigns full credit to exact or near-exact predictions while penalizing progressively larger numeric deviations in a smooth and bounded manner.

For localization, the model predicts a bounding box $\hat{B}_i$ and is scored by intersection-over-union with the ground-truth box B_i :

$$s_i = 100 \cdot \text{IoU}(B_i, \hat{B}_i). \quad (10)$$

For situation description, we follow the official protocol[15]. Given the prompt, we run the released text-classification model and take the classifier confidence returned in the score field as the semantic-correctness probability $p_i \in [0, 1]$. The sample score is then

$$s_i = 100 p_i. \quad (11)$$

The evaluation script also reports a diagnostic pass flag at threshold 0.5, but this flag is not used in the benchmark score.

For sequence planning, let π_i and $\hat{\pi}_i$ denote the ground-truth and predicted temporal orders. If I_i is the number of pairwise inversions between them and $\binom{n_i}{2}$ is the total number of ordered pairs, the score is defined as

$$s_i = 100 \left(1 - \frac{I_i}{\binom{n_i}{2}} \right), \quad (12)$$

which rewards partial order consistency rather than requiring exact full-sequence recovery.

We aggregate scores hierarchically. For a task type t with N_t evaluation items, the task score is

$$S_t = \frac{1}{N_t} \sum_{i=1}^{N_t} s_i^{(t)}. \quad (13)$$

For a rank r containing task set $\mathcal{T}_r$, the rank score is the arithmetic mean over task scores:

$$S_r = \frac{1}{|\mathcal{T}_r|} \sum_{t \in \mathcal{T}_r} S_t. \quad (14)$$

For the headline open-loop score used in Table 1, we report a fixed weighted sum over the 14 task scores,

$$S_{\text{open}} = \sum_{t=1}^{14} w_t S_t, \quad \sum_{t=1}^{14} w_t = 1, \quad (15)$$

where the released task weights w_t are the frozen calibration weights shown in the table header and defined in Appendix F. This aggregation preserves interpretability at the task and rank levels while avoiding repeated over-rewarding of highly redundant subtasks.

E Closed-Loop Scoring Details

Inspired by Bench2Drive[33], the closed-loop score for $R4$ is implemented as follows. Let $S_{\text{route}} \in [0, 100]$, $S_{\text{safety}} \in [0, 1]$, and $S_{\text{eff}} \in [0, 1]$ denote the route-completion, safety, and efficiency terms, respectively. The final scene score is defined as

$$S_{R4} = S_{\text{route}} \cdot S_{\text{safety}} \cdot S_{\text{eff}}. \quad (16)$$

This multiplicative design prevents successful route completion from masking unsafe or inefficient behavior.

The route term measures normalized progress toward the designated destination. In implementation, the ego trajectory is projected onto a reference route polyline, and route progress is updated only when the projected lateral deviation does not exceed 6 m. If the ego vehicle reaches the goal region, the route score is set to 100; otherwise it is proportional to the completed fraction of the reference route:

$$S_{\text{route}} = \begin{cases} 100, & \text{destination reached,} \\ 100 \cdot \frac{d_{\text{comp}}}{d_{\text{route}}}, & \text{otherwise,} \end{cases} \quad (17)$$

where d_{comp} is the maximum completed route distance and d_{route} is the reference route length. Destination arrival is triggered when the ego vehicle enters the scenario-defined goal radius, which is set to 8 m by default.

The safety term combines event penalties with lane-keeping quality:

$$S_{\text{safety}} = \left(\prod_{e \in \mathcal{E}} p_e \right) \cdot (1 - r_{\text{offroad}}), \quad (18)$$

where $\mathcal{E}$ is the set of recorded safety-critical events, $p_e \in [0, 1]$ is the penalty multiplier for event e , and r_{offroad} is the proportion of time spent outside the valid route region. In the current implementation, pedestrian collisions incur zero score for the corresponding event term, vehicle collisions and static-object collisions incur multiplicative penalties of 0.2 and 0.3, respectively, and scenario timeout and blocked timeout incur penalties of 0.7. Off-road timeout, manual interruption, and system error are treated as terminal failure events in the result record, with their corresponding penalty multipliers defined directly in the implementation.

The efficiency term evaluates whether the realized trajectory is unnecessarily long or slow relative to the route reference. Distance efficiency and time efficiency are computed as

$$\eta_{\text{dist}} = \min \left(1, \frac{\alpha d_{\text{route}}}{d_{\text{travel}}} \right), \quad \eta_{\text{time}} = \min \left(1, \frac{\beta t_{\text{ref}}}{t_{\text{sim}}} \right), \quad (19)$$

where d_{travel} is the actual traveled distance, t_{sim} is the scenario duration, t_{ref} is the reference duration, and α, β are grace factors. The implementation sets $\alpha = 1.10$ and $\beta = 1.50$. The final efficiency term is the weighted sum

$$S_{\text{eff}} = w_d \eta_{\text{dist}} + w_t \eta_{\text{time}}, \quad (20)$$

with $w_d = 0.7$ and $w_t = 0.3$.

In addition to the composed score, the evaluator records route, safety, and efficiency terms separately, together with the event list, traveled distance, off-road ratio, lane-keeping penalty, reference duration, and scenario termination status. This design preserves interpretability at both the overall and component levels.

F Weight Allocation Details

The objective of weight allocation is to make the composite open-loop score reflect distinct capability dimensions rather than repeated measurements of highly correlated subtasks. To keep the released metric stable, we estimate the correlation structure once during an initial calibration phase after the first benchmark-testing round, derive the weights from that calibration pool, and then freeze them for all subsequent evaluations. We therefore fix the rank-level budgets to

$$B_{R1} = 0.3, \quad B_{R2} = 0.3, \quad B_{R3} = 0.4, \quad (21)$$

and distribute each budget according to correlation-adjusted subtask uniqueness.

Let ρ_{ij} denote the Spearman correlation between subtasks i and j . For a subtask i in rank $r(i)$, we define its redundancy load as

$$L_i = 1 + \lambda_{\text{same}} \sum_{\substack{j \neq i \\ r(j)=r(i)}} [\max(\rho_{ij}, 0)]^\gamma + \lambda_{\text{cross}} \sum_{\substack{j \neq i \\ r(j) \neq r(i)}} [\max(\rho_{ij}, 0)]^\gamma, \quad (22)$$

where λ_{same} and λ_{cross} control the penalty strength for within-rank and cross-rank redundancy, respectively, and γ controls how strongly the penalty concentrates on highly correlated pairs. In our implementation, $\lambda_{\text{same}} = 1.0$, $\lambda_{\text{cross}} = 0.35$, and $\gamma = 2.0$. Negative correlations are clipped to zero so that anticorrelation is not treated as additional uniqueness.

The raw uniqueness of subtask i is then defined as the inverse redundancy load,

$$u_i = \frac{1}{L_i}, \quad (23)$$

and normalized within each rank:

$$\pi_i = \frac{u_i}{\sum_{k \in \mathcal{T}_{r(i)}} u_k}, \quad (24)$$

where $\mathcal{T}_{r(i)}$ is the set of subtasks in the same rank as i . The final subtask weight is

$$w_i = B_{r(i)} \cdot \pi_i. \quad (25)$$

As a result, two nearly identical subtasks with similar correlation profiles tend to split a shared effective weight, whereas a less redundant subtask within the same rank will receive a larger allocation, which can guarantee that subtask weights sum to the preassigned rank budget while naturally down-weighting subtasks whose information is already extensively covered by correlated neighbors, as reflected in the weights reported in Table 1.

G Examples for QAs in DriveHierarchy



R1-1-A Existence

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only 'yes' or 'no'.

Prompt: Is there any **Pedestrian** visible in CAM_FRONT?

Answer: Yes.



R1-1-A Existence

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only 'yes' or 'no'.

Prompt: Is there any **Traffic Cone** visible in CAM_FRONT?

Answer: No.



R1-1-B Counting

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (integer).

Prompt: How many **Vehicles** are visible in CAM_FRONT?

Answer: 4.



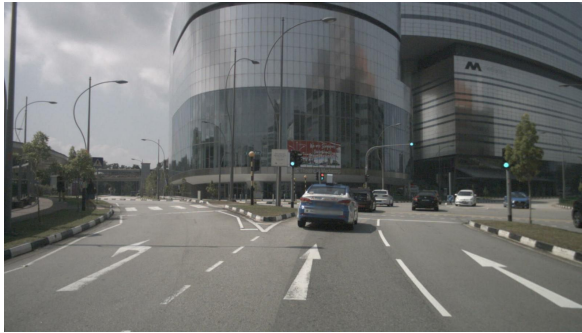
R1-1-C State Attribute

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Choose the correct option and output only the option letter (e.g., A/B/C).

Prompt: What is the current status of the nearest Pedestrian in CAM_FRONT?
["A. moving.", "B. standing.", "C. sitting or lying."]

Answer: C.

Figure 6: Representative examples from Rank 1-1 Questions.



R1-2-A Nearest Object

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (decimal).

Prompt: What is the distance to the nearest Vehicle in CAM_FRONT? (Image resolution: 1600x900). Answer in meters.

Answer: 17.77 m.



R1-2-A Nearest Object

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (decimal).

Prompt: What is the distance to the nearest Bicycle in CAM_FRONT? (Image resolution: 1600x900). Answer in meters.

Answer: 32.00 m.

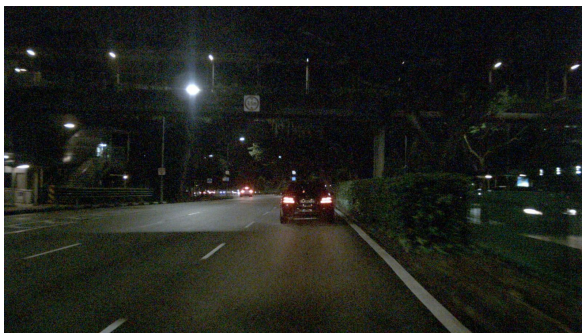


R1-2-B Certain Object

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (decimal).

Prompt: Calculate the distance to the Vehicle located at pixel coordinates [105, 475] (Image resolution: 1600x900). Answer in meters.

Answer: 40.94 m



R1-2-C Distance Bucket

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (integer).

Prompt: How many Vehicles are within the Mid range (20-50m) in camera?

Answer: 2.

Figure 7: Representative examples from Rank 1-2 Questions.



R1-3 Location Questions

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Output only the bounding box as [x1, y1, x2, y2] in pixels.

Prompt: Please locate the object described as: "There is a man wearing blue pants and holding a phone, walking on the left side of the road, towards the ego car.". Image resolution: 2704x1520. Find the bounding box and output its diagonal points in the format [xmin, ymin, xmax, ymax].

Answer: [264, 709, 525, 1469]



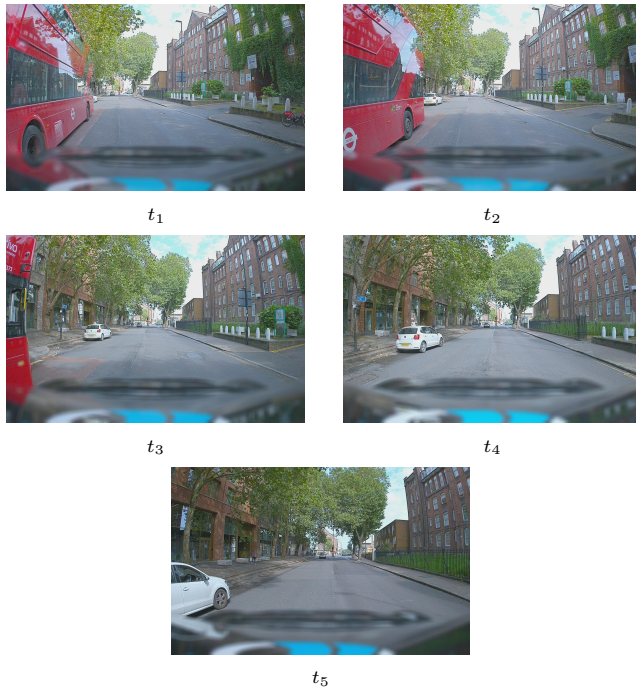
R1-3 Location Questions

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Output only the bounding box as [x1, y1, x2, y2] in pixels.

Prompt: Please locate the object described as: "There is a traffic stop sign, in front of the ego car.". Image resolution: 2704x1520. Find the bounding box and output its diagonal points in the format [xmin, ymin, xmax, ymax].

Answer: [660, 263, 771, 360]

Figure 8: Representative examples from Rank 1-3 Questions.



R1-4 Situation Description

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Output only the final answer.

Prompt: What actions are you taking currently?

Answer: I am trying to safely overtake a stopped bus on its right side and moving to the left of the road to be correctly positioned.

Figure 9: Representative example from Rank 1-4 Questions.



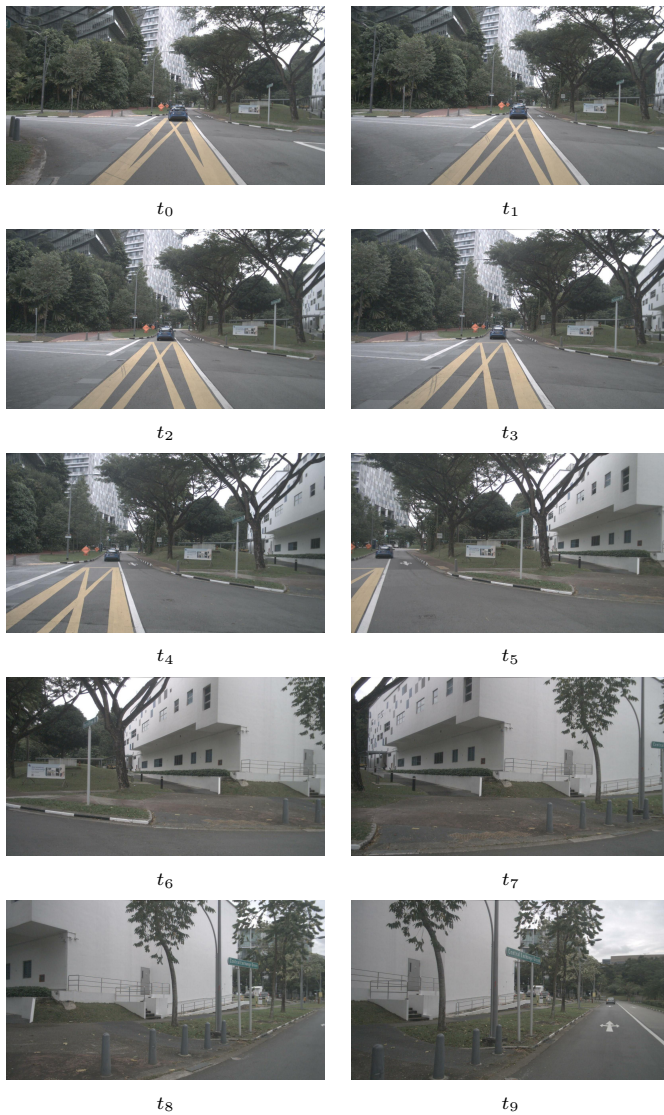
R2-1 Multi-view Memory

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (integer).

Prompt: Given the 6 surround-view images (Front, Front-Left, Front-Right, Back, Back-Left, Back-Right) at this timestamp, how many unique vehicles are visible in the surrounding environment?

Answer: 10.

Figure 10: Representative example from Rank 2-1 Questions.



R2-2-A Temporal Counting

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Answer with only the numeric result (integer).

Prompt: Observing the sequence starting from this frame for the next 5.0 seconds (including the current frame), how many unique vehicles appear in CAM_FRONT?

Answer: 7.

Figure 11: Representative example from Rank 2-2-A Questions.



R2-2-B Temporal Status Recognition

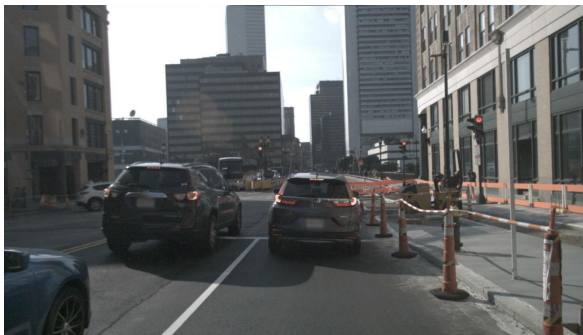
Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Choose the correct option and output only the option letter (e.g., A/B/C/D).

Prompt: Focus on the Vehicle at pixel coordinates [1599, 714] in the initial frame (Image resolution: 1600x900), located in the lower-right region. Observing the clip from $t=0s$ to $t=3.5s$, how does its state change?

["A. It transitions from parked to moving.", "B. It transitions from stopped to moving.", "C. No Change (It remains stopped).", "D. No Change (It remains parked)."]

Answer: B. It transitions from stopped to moving.

Figure 12: Representative example from Rank 2-2-B Questions.



R2-3 Spatial Relations

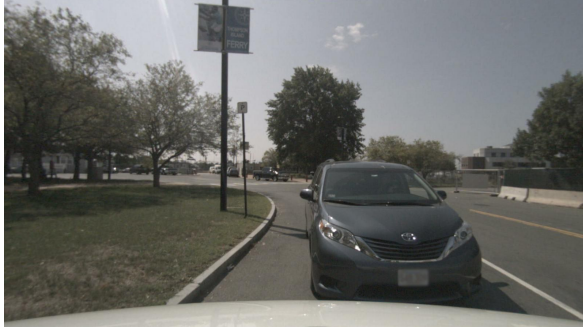
Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Choose the correct option and output only the option letter (e.g., A/B/C/D).

Prompt: In CAM_FRONT, what is the position of the Vehicle located at pixel coordinates [860, 549] relative to the Vehicle located at pixel coordinates [500, 536] (Image resolution: 1600x900. Using the car as a reference frame)?

["A. Left.", "B. Rear-Left.", "C. Rear-Right.", "D. Front-Right."]

Answer: D. Front-Right.

Figure 13: Representative example from Rank 2-3 Questions.



R3-1 Outcome Prediction

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Choose the correct option and output only the option letter (e.g., A/B/C/D).

Prompt: In CAM_BACK, what is the future state of the Black car, located at [755.0, 400.0, 1404.0, 900.0]?
 ["A. Slightly offset to the right while maneuvering.", "B. Decelerate gradually without braking.", "C. Back up.", "D. Accelerate and go ahead."]

Answer: B. Decelerate gradually without braking.

Figure 14: Representative example from Rank 3-1 Questions.



F_A



F_B



F_C



F_D



F_E

R3-2 Sequential Planning

Sys. Prompt: You are a mature and cautious driver with many years of driving experience. You now need assistance answering questions related to the driving process. Output only the ordered list of option labels in the format ["A", "B", ...].

Prompt: Please determine the correct order of the images. Option labels: A, B, C, D, E.

Answer: ["E", "D", "B", "A", "C"]

Figure 15: Representative example from Rank 3-2 Questions.

H Description of scenarios in *R4*

The current version of DRIVEHIERARCHY features 10 categories of typical scenarios, all of which are constructed using the scenario editor illustrated in Fig 16. The specific details of these 10 scenario categories are outlined below. As noted in Section A, future scenarios will be further expanded upon this foundation.

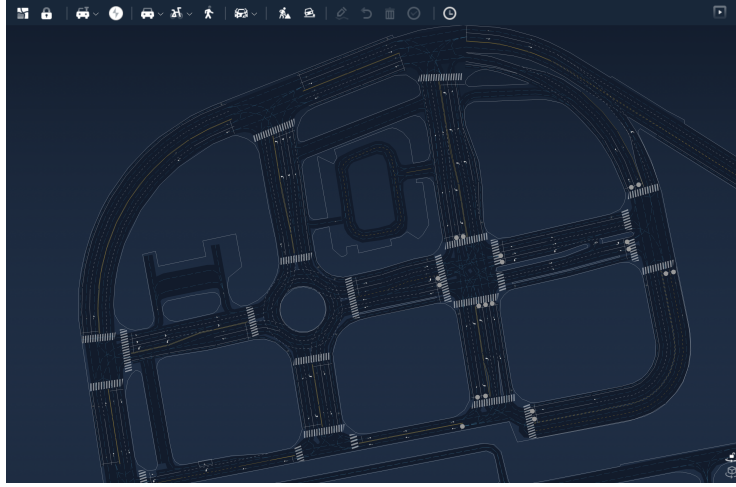


Figure 16: Visualization of part of the Scene Editor.



R4-1 Pedestrian encounters.

Sce. Description: The ego vehicle encounters pedestrians crossing ahead and must detect the crossing intent, decelerate in time, yield safely when required, and resume driving only after the conflict region is clear.



R4-2 Obstacle avoidance.

Sce. Description: The ego vehicle faces static obstacles along its lane and must select a safe avoidance maneuver while preserving route continuity and preventing unnecessary collision risk.

Figure 17: Representative Scenarios from Rank 4-1 to Rank 4-2.



R4-3 Turning maneuver.

Sce. Description: The ego vehicle performs a turning operation under constrained road geometry and surrounding traffic, requiring coordinated speed control, path tracking, and conflict-aware maneuver execution.



R4-4 Intersection.

Sce. Description: The ego vehicle traverses an intersection with non-fixed entry and exit directions, and must resolve multi-agent interactions, respect traffic rules, and choose an appropriate passage strategy under potentially competing movements.



R4-5 T-intersection.

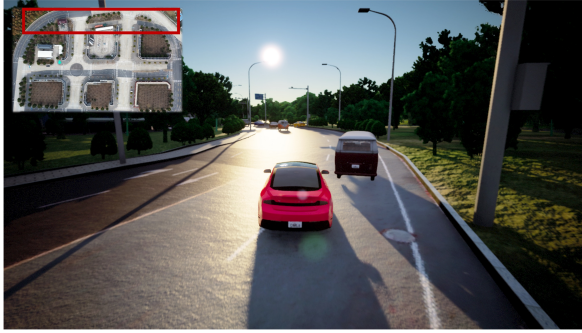
Sce. Description: The ego vehicle traverses a T-intersection and must infer right-of-way, negotiate crossing or joining traffic, and complete the maneuver without blocking or unsafe encroachment.



R4-6 Traffic disturbances.

Sce. Description: The ego vehicle operates in disturbed traffic conditions involving irregular surrounding motion and must remain stable under interference, preserving both safety margins and route progress.

Figure 18: Representative Scenarios from Rank 4-3 to Rank 4-6.



R4-7 Sudden braking.

Sce. Description: The ego vehicle encounters abrupt deceleration of the leading vehicle and must react promptly with safe longitudinal control while avoiding rear-end collision and excessive instability.



R4-8 Merging.

Sce. Description: The ego vehicle performs lane joining and turning in the presence of neighboring traffic, requiring accurate gap assessment, timing control, and smooth trajectory adjustment.



R4-9 Yielding.

Sce. Description: The ego vehicle approaches a yield situation and must identify priority conflicts, slow or stop when necessary, and proceed only when the available gap satisfies safe interaction constraints.



R4-10 Roundabout navigation.

Sce. Description: The ego vehicle navigates a roundabout under continuous multi-agent interaction with non-fixed entry and exit directions, and must maintain lane discipline, merge at an appropriate time, and exit without disrupting surrounding flow.

Figure 19: Representative Scenarios from Rank 4-7 to Rank 4-10.

I Public Resources Used

We build DRIVEHIERARCHY using public datasets, simulators, model checkpoints, and training/evaluation frameworks. We do not claim ownership of these resources, and all original data, codebases, and checkpoints remain the property of their respective authors or organizations under their corresponding licenses or terms of use. We release only the benchmark assets, annotations, metadata, and code that are permitted for redistribution, and users should obtain the original materials from the official sources when required. In addition, part of the closed-loop scoring implementation is adapted from Bench2Drive to fit our benchmark design and evaluation.

Table 11: **Public resources used in this work.** Exact checkpoint names for evaluated models are listed in Table 7.

Resource	Type	License / Terms	Official Link
NuScenes	Dataset	CC BY-NC-SA 4.0	Data
nuPlan	Dataset	Apache License 2.0	Data
NAVSIM	Dataset	Apache License 2.0	Code
DriveLM	Dataset	Apache License 2.0	Code
DriveBench	Dataset / Benchmark	Apache License 2.0	Project
LingoQA	Dataset	Non-commercial use	Project
DRAMA	Dataset	Non-commercial academic use	Data
CARLA	Simulator	MIT	Project
SUMO	Simulator	EPL-2.0	Project
Bench2Drive	Code / Benchmark	CC-BY-NC-ND	Code
Bench2Drive-VL	Code / Benchmark	CC-BY-NC-ND	Code
Qwen family	Model checkpoints	Apache License 2.0	Model
InternVL family	Model checkpoints	Apache License 2.0	Model
MiniCPM-V	Model checkpoint	Apache License 2.0	Model
Gemma	Model checkpoints	Google Gemma 27 Terms of Use	Model
Pixtral	Model checkpoint	Apache License 2.0	Model
ZwZ	Model checkpoint	Apache License 2.0	Model
DA-DriveLM	Model checkpoint	MIT	Model
ReasonDrive	Model checkpoint	Apache License 2.0	Model
swift-sft	Training framework	Apache License 2.0	Code
vLLM	Software	Apache License 2.0	Code

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly state the paper’s scope and contributions, including the four-rank hierarchy, benchmark construction, and evaluation on 15 VLMs (See Abstract and Section 1). These claims are supported by the benchmark design and empirical results.

Guidelines:

- The answer [\[N/A\]](#) means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [\[No\]](#) or [\[N/A\]](#) answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The paper includes a dedicated *Limitations* discussion that covers the simulator-based nature of *R4*, incomplete capability coverage, the modeling dependence of the aggregate score, and the limited generality of the optimization study, which bound the scope of the conclusions.

Guidelines:

- The answer [\[N/A\]](#) means that the paper has no limitation while the answer [\[No\]](#) means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results or proofs; it focuses on benchmark construction and empirical evaluation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper specifies benchmark construction, evaluated checkpoints, inference settings, scoring rules, closed-loop runtime protocol, and SFT hyperparameters (See Appendix B and Appendix C). These details are sufficient to reproduce the main reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

- (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: An anonymized repository accompanying the submission provides the benchmark assets, code, and reproduction instructions, while the paper documents the public data sources, checkpoints, and licenses used in this work (anonymous.4open.science/r/DriveHierarchy-2FD5). Redistributable materials are released directly, and non-redistributable resources are linked to their official sources.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://neurips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The paper reports benchmark construction, evaluated checkpoints, decoding settings, scoring protocols, closed-loop runtime settings, and the main SFT configuration, including the validation split and key hyperparameters (See Appendix B and Appendix C). These details are sufficient to interpret the reported results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: While the paper reports comprehensive evaluations across multiple open-loop and close-loop benchmarks, it does not include error bars, confidence intervals, or statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper reports the GPU configuration used for open-loop evaluation, closed-loop evaluation, and SFT training, specifies that the A800 GPUs are 80GB variants, and provides approximate wall-clock time and GPU-hours for the main benchmark runs. Runtime varies across models, but the reported ranges make the compute requirements of the experiments explicit.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The work uses public datasets, simulators, and model checkpoints under their respective licenses and terms of use and discusses responsible use and release scope in the paper (See Appendix A and Appendix I).

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper includes a dedicated *Broader Impacts* discussion that addresses both positive impacts, such as more diagnostic evaluation and targeted improvement, and negative risks, including overfitting to benchmark distributions, propagation of upstream dataset biases, and the gap between benchmark performance and real-world deployment (See Appendix A).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk generative models or scraped datasets. It releases only benchmark assets, annotations, metadata, and code that are permitted for redistribution, while non-redistributable resources remain available from their official sources under their original licenses and terms.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: The paper credits the public datasets, simulators, frameworks, and model checkpoints used in this work, and explicitly lists their licenses or terms of use together with official sources in Appendix I. Only redistributable materials are released by the authors.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: The paper introduces new benchmark assets, annotations, metadata, and code, and the anonymized release is accompanied by documentation describing the benchmark composition, data fields, usage constraints, and reproduction procedure. The released materials are limited to assets permitted for redistribution.

Guidelines:

- The answer [\[N/A\]](#) means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[N/A\]](#)

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper explicitly studies VLMs/LLMs as the core research object and describes their role in benchmark evaluation, closed-loop testing, and benchmark-guided SFT, including the evaluated model checkpoints and training settings.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.