LANGUAGE2DRIVE: DO VISION-LANGUAGE-ACTION MODELS REALLY FOLLOW THE INSTRUCTIONS IN AUTONOMOUS DRIVING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent Vision-Language-Action (VLA) models increasingly use natural-language instructions to condition autonomous driving behavior. Yet existing closed-loop benchmarks primarily measure route completion, safety, and rule compliance, leaving a fundamental question unresolved: does language actually determine which feasible behavior the vehicle executes? A trajectory may be safe and successful while still violating or even ignoring the passenger's instruction. To address this, we introduce Language2Drive, a benchmark designed to directly test this capability. Its key evaluation unit, the paired counterfactual unit (PCU), holds the initial driving scene fixed while changing only the instruction. This design enables us to distinguish isolated instruction success from counterfactual paired success across four instruction-following scopes. To test whether this distinction persists across evaluation regimes, we instantiate Language2Drive in complementary closed-loop CARLA simulation, reconstructed real-world scenes, and open-loop driving benchmarks, including our released CARLA-Language2Drive simulator with purpose-built maps. Evaluating representative driving VLAs across these settings reveals a striking mismatch between strong driving competence and weak language control: 93.2% of successful rollouts do not have a successful counterpart under the alternative feasible instruction. This failure is dominated by one-sided completion, where a model repeatedly executes one preferred behavior but fails when the same scene requires the counterfactual alternative. These results expose two fundamental limitations of current evaluation: **isolated success and driving scores, which are widely used by conventional driving benchmarks, can substantially overstate language-following capability, while current driving VLAs themselves lack reliable instruction following under matched counterfactual tests**. Effective evaluation must therefore measure not only whether a VLA can drive successfully, but whether changing the instruction reliably alters which feasible behavior it selects and successfully executes.

1 INTRODUCTION

Recent advances in Vision-Language-Action (VLA) models have opened a new paradigm for autonomous driving: natural-language instructions is able to condition driving behaviors (Zhou et al., 2026b;c; Shao et al., 2024; Tian et al., 2024; Hwang et al., 2024; Renz et al., 2025). By building upon internet-scale pretraining in large vision-language models, VLAs inherit rich world knowledge, open-world perception, scene understanding, and multimodal reasoning capabilities from foundation models. It promise an interface to express passenger intent without direct manual control, in which passengers can specify how to drive and what task to accomplish. Realizing this promise requires more than generating plausible motion: when several safe, legal, and physically feasible behaviors are available, the system must select and sustain the one requested by the passenger.

Existing evaluation does not fully establish this capability. Driving benchmarks emphasize route completion, safety, and traffic rule compliance (Dauner et al., 2024; Cao et al., 2025; Yu et al., 2025; Jia et al., 2024), while language-guided evaluations assess instructed route execution or instruction-specific predicted actions (Shao et al., 2024; Renz et al., 2025). These evaluations provide important evidence of driving competence and task completion. Yet a safe trajectory can still reach the wrong

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

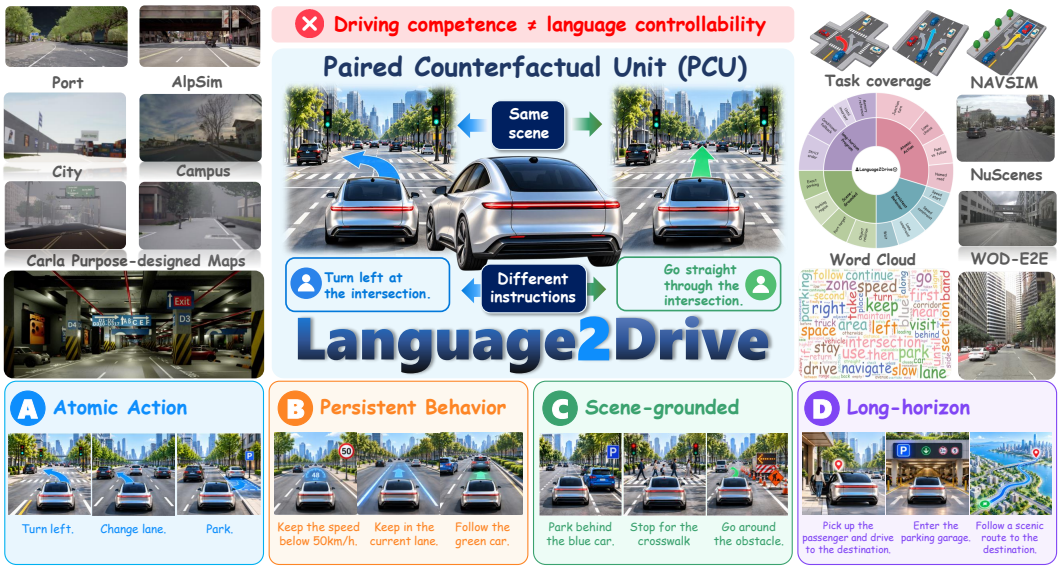


Figure 1: **Overview of Language2Drive.** Language2Drive evaluates autonomous-driving VLA instruction following across four scopes: (A) Atomic actions, (B) Persistent behaviors, (C) Scene-grounded instructions, and (D) Long-horizon programs. To support these task, we build purpose-designed CARLA maps includes cities, underground parking lot, campus and port. We also curate relevant scenarios from AlpSim, NAVSIM, nuScenes, and WOD-E2E. The Task-Coverage chart summarizes the diversity of driving behaviors, while the Word Cloud illustrates the breadth of natural-language expressions. We further introduce the Paired Counterfactual Unit (PCU), which varies the instruction while keeping the driving scene fixed, directly testing whether language changes the vehicle's behavior. Our benchmark highlights a key distinction: driving competence does not necessarily imply language controllability or reliable instruction following.

named parking space or violate a requested passing restriction. More fundamentally, even successful execution of an individual instruction does not establish that language reliably determines the selected behavior. The central question is therefore: Do VLAs really follow instructions, or can their apparent successes be explained by behaviors they would execute regardless of the request?

To address this, we introduce **Language2Drive**, a closed-loop benchmark for evaluating reliable instruction following in autonomous driving. The key challenge is that success on a single request does not establish language-conditioned control. At an intersection where turning left and going straight are both safe and feasible, for example, a policy that consistently turns left may succeed on the left-turn request without demonstrating that it can execute the alternative when asked. Conversely, comparing instructions across different scenes confounds language variation with environmental variation. Language2Drive therefore evaluates matched alternatives through paired counterfactual units (PCUs). Each PCU holds the decision context, non-language inputs, and simulation seed fixed while evaluating two mutually exclusive but feasible instructions in separate runs. This design distinguishes isolated success from the ability to execute either requested alternative under matched conditions. Because an instruction may admit many valid trajectories, each request is evaluated by machine-checkable spatial, temporal, and safety criteria rather than reference-trajectory matching. Strict Instruction Success Rate (SISR) measures individual completion, whereas Counterfactual Paired Success (CFPS) requires both alternatives in a PCU to succeed. Three meaning-preserving expressions further test whether paired success remains stable across linguistic variation.

Language2Drive organizes tasks by what language controls: atomic actions, persistent behavior, scene-grounded navigation, and long-horizon programs. To support these tasks, we release **CARLA-Language2Drive**, a CARLA extension that adds six purpose-designed maps covering underground parking, a port, Chinese urban roads, a U.S.-style city with detailed signage, a campus, and a highway. A complementary AlpaSim track is built from reconstructed real-world road scenes selected through targeted mining and manual review (NVIDIA et al., 2025; Wu et al., 2025). These environments provide alternative maneuvers, identifiable referents, and execution intervals needed to test language-conditioned behavior. We also extend paired evaluation based on the test sets developed from nuScenes, NAVSIM, and WOD-E2E, complementing closed-loop execution with open-loop tests on recorded driving contexts.

Our evaluation reveals that **driving quality and individual instruction success do not establish reliable language control**. Across principal CARLA systems, 93.2% of successful rollouts lack a successful opposite-instruction counterpart. Most observed successes are therefore one-sided: one request is completed, but its feasible alternative is not. In reconstructed scenes, Alpamayo 1.5 achieves 100% Safety and a reference-path DrivingScore of 97.17, yet **zero** CFPS. Moreover, no evaluated CARLA system completes both alternatives under all three tested language expressions. The individual–paired discrepancy also appears in open-loop evaluations across all three datasets. Together, these findings show why driving scores and isolated successes must be supplemented by tests of alternative selection, complete execution, and consistency across equivalent instructions.

Our contributions are threefold. **First**, we introduce a paired evaluation framework that distinguishes individual instruction completion from successful execution of alternative requests under matched conditions. **Second**, we instantiate four language-control scopes in Language2Drive, contributing an extended CARLA simulator with six purpose-designed maps, a curated AlpaSim track, and complementary open-loop evaluations. **Third**, we reveal a substantial separation between driving quality and reliable instruction following, characterizing one-sided success, incomplete execution, and fragile paired success across paraphrases. These findings establish matched, instruction-specific execution as an essential complement to conventional driving evaluation.

2 LANGUAGE2DRIVE

2.1 OVERVIEW

We introduce Language2Drive, a closed-loop benchmark designed to evaluate whether vision-language-action models for autonomous driving reliably follow passenger instructions.

Task Definition

Autonomous-driving instruction following is the ability to select and execute the passenger-requested behavior among safe, legal, and physically feasible alternatives when navigation is absent or does not determine the choice. Without a passenger specifying each reaction, the system must still respond to traffic; when a request conflicts with safety or feasibility, it should perform an appropriate safe response.

2.2 PAIRED COUNTERFACTUAL UNITS

A single successful request can coincide with a system's default behavior. We therefore test the same decision context twice: for example, one run asks the vehicle to turn left, while the other to continue straight. Both requests are feasible, but a successful execution of one excludes the other. We call this matched test a Paired Counterfactual Unit (PCU). It tests whether the system can execute either alternative when asked.

Both experimental conditions are initialized with the same map, ego/actor states, history, non-language inputs, and simulation seed. The human language instruction changes; later observations evolve with each run's actions. Condition-specific destinations, future reference paths, and turn commands remain private to the evaluator. Each PCU has two instructions, each with an associated success criterion; meaning-preserving expressions test the same pair without changing their targets. Appendix A states the conditions under which an instruction-invariant system cannot succeed on both test conditions.

2.3 MACHINE-CHECKABLE SUCCESS CRITERIA

Each instruction is paired with a machine-checkable success criterion over the executed trajectory. It specifies the requested outcome, where the condition applies, and when execution is complete. Three examples illustrate the distinction between plausible motion and successful execution: *"Turn left at this intersection."* Enter the requested left exit, without satisfying the paired straight-exit criterion. *"Park in space B204."* Place the vehicle footprint inside B204 with a valid heading, sufficiently low speed, and the required dwell time. *"First visit B208, then go to F206."* Complete the two stages in that order; visiting both in reverse order does not count.

Persistent conditions apply to a reached, nonempty interval that the vehicle must complete. Stopping before the interval cannot satisfy a request to maintain a speed throughout it. The verifier checks

actual ego states and composes geometric, behavioral, and temporal predicates; exact operators and thresholds appear in Appendix C.4.

2.4 METRICS

Complete instruction execution is measured by two primary metrics: **SISR** for individual requests and **CFPS** for matched alternatives. Supplementary metrics characterize partial completion and consistency across equivalent instructions; driving quality is reported separately.

Individual and paired success. Let $s_{ir,j} \in \{0,1\}$ be the strict verdict for PCU i, expression r, and arm $j \in \{0,1\}$. Success requires the complete requested contract, rejection of its paired alternative, and all applicable safety and execution conditions. We average expressions within each PCU and then weight PCUs equally: $\langle z \rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{R_i}\sum_{r=0}^{R_i-1} z_{ir}$. Unsuccessful executions remain in the scheduled denominator. All instruction metrics below use a 0–100 scale. We report

$$\mathrm{SISR} = 100\left\langle \frac{s_{ir,0}+s_{ir,1}}{2}\right\rangle, \qquad \mathrm{CFPS} = 100\langle s_{ir,0}s_{ir,1}\rangle. \tag{1}$$

SISR credits each completed request; CFPS credits a matched pair only when both requests are completed. For example, always completing the left turn yields 50 SISR but zero CFPS on a balanced left/straight test. Their difference has an exact interpretation:

$$\mathrm{SISR} - \mathrm{CFPS} = \tfrac{1}{2}q_{\mathrm{one}}, \qquad q_{\mathrm{one}} = 100\langle |s_{ir,0} - s_{ir,1}|\rangle. \tag{2}$$

Here q_{one} is the percentage of paired instances with exactly one successful arm under the same averaging weights. The gap therefore quantifies one-sided completion. Conditions under which CFPS rules out instruction-invariant execution are given in Appendix B.9.

Partial completion. To characterize execution below strict success, we retain two diagnostic views. *Relaxed semantic proxies*, SemISR and Sem-CFPS, apply Equation 1 to predefined task-specific relaxed verdicts instead of strict verdicts. They measure satisfaction of weaker criteria, not additional strict successes. Graded completion uses $c_{ir,j} = C_{i,j}(\tau_{ir,j}) \in [0,1]$ to score how fully a trajectory completes its own request:

$$\mathrm{SCS} = 100\left\langle \frac{c_{ir,0}+c_{ir,1}}{2}\right\rangle, \qquad \mathrm{PCS} = 100\langle \min(c_{ir,0}, c_{ir,1})\rangle. \tag{3}$$

SCS measures average completion; PCS measures completion of the weaker alternative, so progress on one arm cannot compensate for an uncompleted counterpart. Their gap is $\mathrm{SCS} - \mathrm{PCS} = 50\langle |c_{ir,0} - c_{ir,1}|\rangle$. The relaxed and graded rules are distinct diagnostics rather than interchangeable success definitions, as detailed in Appendix B.2.

Summary reporting metric L2D. For compact within-track reporting, we first summarize language-following ability across three complementary families:

$$\begin{aligned} a_1 &= \tfrac{1}{2}(\mathrm{SISR} + \mathrm{CFPS}), \quad a_2 = \tfrac{1}{2}(\mathrm{SemISR} + \text{Sem-CFPS}), \quad a_3 = \tfrac{1}{2}(\mathrm{SCS} + \mathrm{PCS}), \\ L_{\mathrm{bal}} &= \left(\frac{\sqrt{a_1}+\sqrt{a_2}+\sqrt{a_3}}{3}\right)^2. \end{aligned} \tag{4}$$

Here, a_1, a_2, and a_3 summarize strict instruction following, semantic instruction following, and consistency, respectively. L_{bal} favors models that perform well across all three families rather than excelling in only one.

We then combine language following with track-specific driving quality:

$$\mathrm{L2D}_t = \frac{1 + 0.99L_{\mathrm{bal}}}{100}\begin{cases} \frac{1}{2}(\mathrm{PSM} + \mathrm{DrivingScore}), & t = \mathrm{CARLA}, \\ \frac{1}{2}(\mathrm{Safety} + \mathrm{DrivingScore}), & t = \mathrm{RealSim}. \end{cases} \tag{5}$$

Intuitively, the driving term measures whether the vehicle can drive safely and effectively, while L_{bal} measures whether its behavior is actually controlled by language. Therefore, high driving quality alone cannot produce a high L2D score: if the model largely ignores the instruction, its final score remains low.

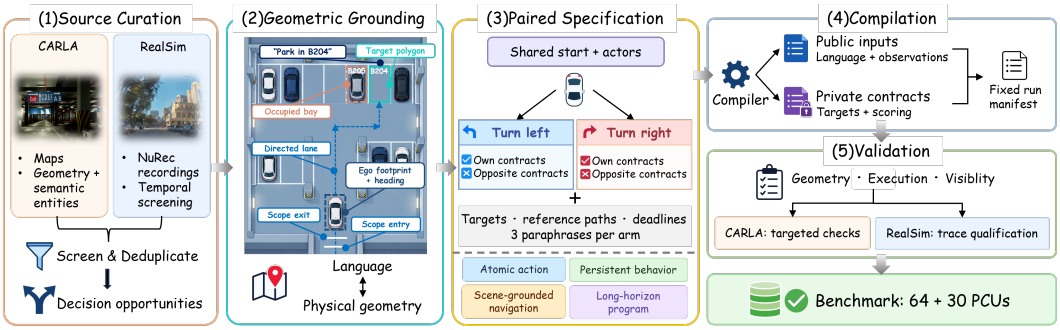


Figure 2: **Benchmark construction process.**The process to build the benchmark is as follows: (1) Source curation identifies decision opportunities; (2) Geometric grounding binds language to physical entities; (3) Paired specification fixes shared conditions and opposed outcomes; (4) Compilation separates public inputs from private criteria; (5) Validation checks the resulting tasks. The benchmark contains 64 CARLA and 30 RealSim PCUs.

Driving quality and summary reporting. DrivingScore measures reference-path advancement. Safety is characterized by Predictive Safety Margin (PSM) on independent CARLA routes and collision/off-road-free Safety in RealSim. Neither establishes instruction satisfaction. L2D summarizes the aggregated language and driving measurements within each track, with driving-only credit capped at one point. Its definition and properties appear in Appendix B.6; SISR and CFPS remain the primary measures of complete execution.

2.5 Four language-control scopes

We organize instructions by what language controls and how long its requirements remain active. Figure 1 summarizes their task families and vocabulary.

A: Atomic action. Language selects a local maneuver or setpoint, such as a junction turn, lane choice, speed/start action, or wait-then-park maneuver. Success requires completing the selected maneuver under its applicable dwell and progress conditions while rejecting the paired alternative. This scope concerns one local decision, even when execution takes several seconds.

B: Persistent behavior. Language imposes a speed, lane, or actor-related requirement over an annotated interval, including passing versus following. Paired requests share the interval or actor but prescribe different behaviors. Verification requires complete traversal and compliance from scope entry to exit; briefly satisfying the condition or stopping before the interval is insufficient.

C: Scene-grounded navigation. Language selects a named road, parking region, port target, or exact parking space. The requested entity must be grounded to its physical target: directed exit gates for roads and polygons for regions or spaces. Exact parking additionally checks footprint, heading, speed, and dwell. Reaching a nearby target does not satisfy the requested identity.

D: Long-horizon program. Language specifies obligations that depend on task state: ordered visits, conditional fallbacks, invariants until an event, or reuse of a remembered reference. The verifier tracks stages, triggers, and bound entities, rejecting incorrect ordering or incomplete execution. Program structure, rather than duration alone, distinguishes this scope from persistent behavior.

Two CARLA diagnostics complement the feasible-request core. Foundation Driving Qualification (FDQ) tests traffic reactions without passenger-specified instructions. Safety Qualification uses requests requiring a safe fallback, refusal, or constrained alternative when literal compliance conflicts with safety or feasibility. An open-loop extension developed from nuScenes, NAVSIM, and WOD-E2E evaluates predicted trajectories; Section 3.5 describes its construction.

2.6 Benchmark construction process

Our construction process turns a physical decision opportunity into a matched, executable test (Figure 2). Each stage addresses a separate validity requirement: meaningful alternatives, unambiguous grounding, controlled inputs, or verifiable execution.

Source curation. CARLA-Language2Drive extends CARLA (Dosovitskiy et al., 2017) with six purpose-designed environment types: underground parking, a port, Chinese urban roads, a US city with detailed signage, a campus, and a highway. We select decision contexts that permit opposed feasible behaviors. **RealSim** mines reconstructed PhysicalAI-NuRec scenes executed through AlpaSim (NVIDIA et al., 2025; Wu et al., 2025); temporal screening requires both pre-instruction context and enough recording time for the requested maneuver.

Geometric grounding. Directed lane graphs, exit gates, named regions, parking polygons, and physical actors bind linguistic references to measurable entities. Parking uses the ego footprint and heading, while persistent tasks specify explicit entry/exit geometry and relevant actors. These bindings distinguish executing a request from merely approaching its target and prevent an empty or unreached interval from counting as compliance.

Paired specification. Each PCU fixes the initial scene and actor setup, then assigns mutually exclusive requests with corresponding success predicates, private reference paths, and deadlines. CARLA has three meaning-preserving expressions per arm; RealSim uses one.

Public/private compilation. Compilation separates public language, observations, ego state, and neutral context from private targets and scoring information. Consequently, the evaluator can check the requested alternative without revealing its future path or destination through auxiliary inputs.

Task validation. Static checks verify entity bindings, directed reachability, footprint constraints, and matched non-language initialization. All PCUs are each validated via a full-scale evaluator that accounts for dynamic execution—incorporating deadlines, occupancy, dynamic participants, and temporal requirements. Human reviewers further verify designated parking spaces, road names, and the operational ranges of "next road segments," making sure the model under test is able to determine the target and the scope of constraints based on actual camera feeds, historical data, and shared context. Every PCU has successful privileged executions of both alternatives under three seeds, giving witnesses that satisfy the requested criterion and reject its counterpart.

3 EXPERIMENTS

3.1 EVALUATION PROTOCOL

We distinguish three outcomes: driving successfully, completing an individual instruction, and completing both requested alternatives under matched conditions. CARLA evaluates 64 PCUs with two arms and three expressions, yielding 384 core rollouts and 192 paired instances per system as reported in Table 1. RealSim evaluates 30 PCUs from 23 reconstructed scenes with two arms and one expression, yielding 60 rollouts and 30 pairs per system as reported in Table 2. Released weights remain frozen, and each evaluated system includes its input adapter and controller. Appendix D specifies checkpoint identities, sensors, action interfaces, and execution settings.FDQ and Safety-Q provide complementary closed-loop diagnostics; open-loop evaluations test predicted trajectories on three recorded-driving datasets.

3.2 CAN SUCCESS ON ONE INSTRUCTION ESTABLISH LANGUAGE CONTROL?

Most successful requests have no successful counterpart. Across the principal CARLA systems, 93.2% of successes are one-sided: one request succeeds while its feasible alternative fails. LMDrive achieves the highest SISR, 12.24%, yet completes no pair. Alpamayo 1.5 and SimLingo each complete only 1.04% CFPS, leaving 27 of 31 and 26 of 30 individual successes unpaired. These are genuine individual completions, but they are rarely accompanied by successful execution of the alternative request. The discrepancy also appears in reconstructed scenes: only one of 240 evaluated system–PCU pairs succeeds strictly, achieved by SimLingo, while Alpamayo2-Super reaches 13.33% SISR with zero CFPS.

Successful pairs demonstrate the capability that isolated successes leave untested. Alpamayo 1.5 completes both passing and following on CARLA B-027/p1, and SimLingo satisfies both requested speed bands on RealSim P22. These positive examples show that the evaluated systems can complete matched alternatives in some contexts; the aggregate results show how rarely they do so across the benchmark. Paired evaluation therefore distinguishes individual completion from demonstrated access to both requested behaviors. Appendix F gives the count-level decomposition.

Table 1: **CARLA-Language2Drive main results.** 384 core rollouts and 192 paired instances per system (64 PCUs, three expressions). PSM uses 24 additional routes. All values are on a 0–100 scale; higher is better. Strict individual and paired success are shown first; L2D follows the component scores.

System	Training data	Strict success		Relaxed proxy		Graded		Driving		Summary
		SISR	CFPS	SemISR	Sem-CFPS	SCS	PCS	PSM	Drive Score	L2D
Alpamayo 1.5	NVIDIA AV	8.07	**1.04**	42.19	30.73	**26.18**	**12.50**	70.10	62.69	**12.20**
SimLingo	Action Dream	7.81	**1.04**	**48.70**	**35.94**	23.51	11.94	62.30	**65.04**	12.15
LMDrive	CARLA-64K	**12.24**	0.00	29.17	18.23	21.45	9.26	**88.84**	50.12	10.39
OneVL	NAVSIM	0.52	0.00	25.00	3.65	16.69	7.05	77.72	50.69	4.87
Qwen-Drive-1.0-4B	WOD-E2E	1.82	0.00	15.62	3.12	17.05	7.09	76.68	43.67	4.31
Impromptu-VLA	Impromptu	0.26	0.00	19.79	6.25	14.48	4.61	77.43	42.02	3.87
Diagnostic configuration: high-level bridge										
Senna VLM bridge	DriveX	12.24	0.00	27.60	3.65	25.93	8.12	80.85	68.38	9.89

Table 2: **RealSim main results.** Scoring v0.2 on 30 PCUs from 23 reconstructed scenes: 60 rollouts and 30 pairs per system. All values are on a 0–100 scale; higher is better. Strict individual and paired success are shown first; L2D follows the component scores.

System	Strict success		Relaxed proxy		Graded		Driving		Summary
	SISR	CFPS	SemISR	Sem-CFPS	SCS	PCS	Safety	Driving Score	L2D
Alpamayo2-Super	**13.33**	0.00	**18.33**	0.00	**21.43**	0.00	93.33	97.10	**9.22**
Alpamayo 1.5	11.67	0.00	15.00	0.00	18.12	1.11	**100.00**	**97.17**	8.38
SimLingo	6.67	**3.33**	13.33	**6.67**	16.67	**10.00**	68.33	83.09	7.58
AutoVLA	1.67	0.00	8.33	**6.67**	9.73	6.67	41.67	76.94	3.36
Qwen-Drive-1.0-4B	0.00	0.00	1.67	0.00	4.44	0.00	43.33	69.16	0.92
Impromptu-VLA	0.00	0.00	1.67	0.00	3.71	0.00	30.00	68.85	0.78
OneVL	0.00	0.00	0.00	0.00	0.00	0.00	43.33	43.82	0.44
ReCogDrive	0.00	0.00	0.00	0.00	0.00	0.00	33.33	43.57	0.38

3.3 Do driving scores establish instruction following?

Strong driving performance can coexist with zero paired success. In RealSim, Alpamayo 1.5 achieves 100% Safety and 97.17 DrivingScore but zero strict or relaxed-proxy paired success; On P11, both Alpamayo 1.5 arms attain 100 DrivingScore, although the requested lane change fails (Figure 3). Conversely, SimLingo completes both P22 instructions with DrivingScores of 97.82 and 85.78: full instruction execution need not maximize reference-path advancement. These results indicate that High scores on driving performance therefore do not certify instruction-specific completion.

Avoiding danger and fulfilling a safe alternative are also distinct outcomes. In the six exploratory Safety-Q cases, LMDrive passes safety in five cases but completes only two safe-refusal responses. Across the six principal systems, all three strict Safety-Q successes are refusals; none of the 24 fallback or constrained-alternative trials succeeds. These diagnostics evaluate responses to conflicting requests separately from the feasible-request core, distinguishing avoidance of a prohibited action from execution of a useful replacement. Further details are provided in Appendix G, particularly Section G.4.

3.4 How reliable is control across wording and task structure?

Successful pairs do not persist across all tested language expressions. No CARLA system completes both arms of a PCU under all three meaning-preserving expressions. A low overall outcome-change rate can conceal fragile successes: SimLingo changes on only 9 of 128 arms, but those nine account for 60% of the 15 arms it ever solves. Alpamayo 1.5's corresponding conditional rate is 75%. On A-010, SimLingo's arm-outcome patterns are `110` and `101`, so paired success occurs only for expression 0. LMDrive, by contrast, completes 14 arms under every expression but never

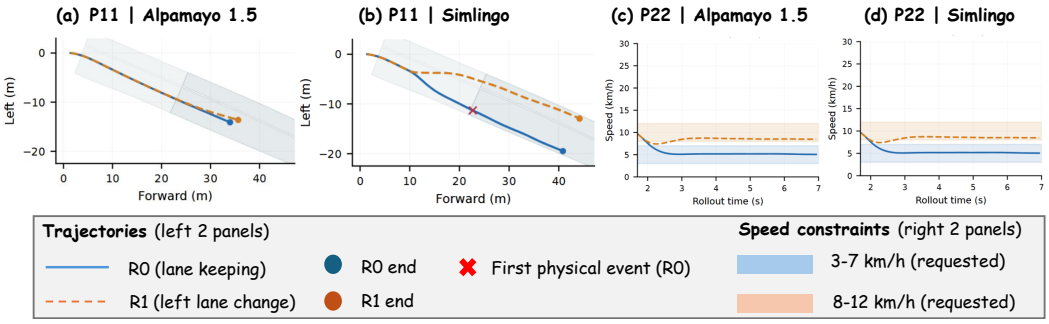


Figure 3: **Advancement and behavioral response do not establish complete instruction execution.** (a,b) P11 requests lane keeping (R0, blue) or a left lane change (R1, orange dashed). Alpamayo 1.5 reaches 100 DrivingScore in both arms but succeeds only on R0. P22 requests 3–7 or 8–12 km/h. Shading marks the requested speed bands, *not* their active time windows. Curves cover the full post-instruction interval; verification applies from entry into the annotated segment to exit and requires complete traversal and compliance at all active samples. SimLingo passes both criteria; Alpamayo 1.5 passes neither. Figure 9 gives the segment geometry.

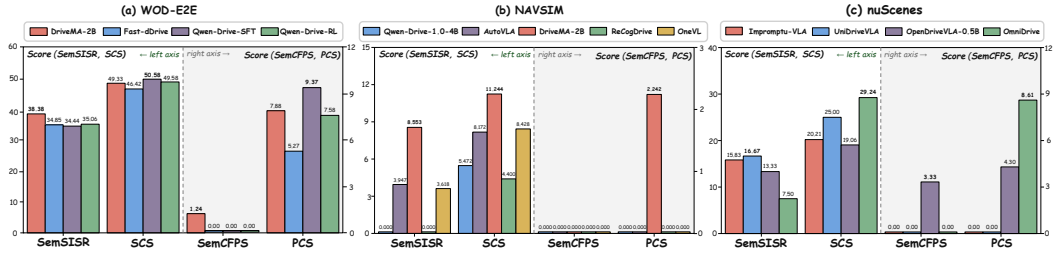


Figure 4: **Open-loop instruction following track on three datasets.** Predicted trajectories are scored for exclusive binary semantic success (SemSISR/SemCFPS) and graded completion (SCS/PCS). All values use a 0–100 scale.

completes a pair. Consistency on one request therefore does not establish consistent access to its alternative under the tested wordings.

Observed long-horizon program success is confined to maintaining an invariant until an event. Across CARLA tracks, all seven strict successes in scope Long-horizon program occur in this family: six from LMDrive and one from SimLingo. Ordered visits, conditional fallbacks, and reference-return programs yield no strict success in 504 rollouts. The scope-level breakdown thus distinguishes limited success on one temporal structure from the broader ability to execute instruction-defined programs. Detailed results and cases appear in Appendices F and I.

3.5 Does the single–paired discrepancy extend to predicted trajectories?

The discrepancy also appears in open-loop settings without controller execution. We evaluate paired predicted trajectories on our test set developed from nuScenes, NAVSIM, and WOD-E2E (Caesar et al., 2020; Dauner et al., 2024; Xu et al., 2026a), holding the recorded history context fixed while changing the instruction. Open-loop binary success requires satisfying the requested criterion and excluding its alternative. Figure 4 reports within-dataset comparisons, with protocol and sample coverage in Appendix E.2. On WOD-E2E, DriveMA-2B attains 38.38 SemSISR but 1.24 SemCFPS; All five NAVSIM systems have zero SemCFPS, with DriveMA-2B reaching the highest SCS/PCS of 11.244/2.242. On nuScenes, OpenDriveVLA-0.5B reaches 3.33 SemCFPS, while all other reported systems score zero. A separate WOD-E2E geometry diagnostic characterizes weak trajectory differentiation. Across 441 pairs, the two rated references differ by 3.962 m mean ADE, whereas the four systems' paired predictions differ by only 0.178–0.355 m on average. These distances describe output sensitivity, not semantic correctness, and use a different cohort from the semantic-success analysis. Without pooling scores or rankings across datasets, the open-loop results extend the central observation: individual completion is much more prevalent than successful completion of both requested alternatives.

4 Related Work

VLA Models for Autonomous Driving. LMDrive and SimLingo connect visual observations and language to closed-loop driving, with SimLingo explicitly aligning language and ac-

Table 3: **Comparison of driving evaluation protocols.** S: synthetic worlds; R: real-log-derived contexts, not on-road tests. CL: sequential closed-loop execution. Matched: instruction variants share a scene or route context. Alt.: variants have distinct scored behavioral targets. Behavior check: instruction-specific acceptance beyond route-level metrics. Para.: explicit paraphrase evaluation. Paired success: both opposed, feasible requests must succeed.

	Execution setting		Instruction evaluation					
Benchmark / protocol	World	CL	Language input	Matched context	Alt. targets	Behavior check	Para. test	**Paired success**
nuPlan 2021	R	✓	–	–	–	–	–	–
Bench2Drive 2024	S	✓	–	–	–	–	–	–
NAVSIM v2 2025	R	–	–	–	–	–	–	–
LangAuto 2024	S	✓	✓	–	–	–	–	–
Action Dreaming 2025	S	–	✓	✓	✓	✓	–	–
Drive4C 2025	S	✓	✓	–	–	✓	–	–
ICR-Drive 2026	S	✓	✓	✓	–	–	✓	–
Language2Drive	**S+R**	✓	✓	✓	✓	✓	✓	✓

tions (Shao et al., 2024; Renz et al., 2025). Recent developments include Qwen-Drive's unified perception–understanding–planning framework (Zhou et al., 2026a), Alpamayo 1.5's user-command-conditioned trajectory generation (Wang et al., 2025b), and Drive My Way's adaptation to persistent preferences and immediate style instructions (Wang et al., 2026). These methods motivate a separate evaluation question: can language reliably select and sustain the requested behavior among feasible alternatives?

Language-Conditioned Driving Evaluation. nuPlan, Bench2Drive, and NAVSIM/v2 assess planning quality and safety (Caesar et al., 2021; Jia et al., 2024; Dauner et al., 2024; Cao et al., 2025). LangAuto, and Drive4C extend evaluation to language-guided navigation, capability-specific tasks, and open-ended instruction realization, respectively (Shao et al., 2024; Sohn et al., 2025). Action Dreaming evaluates alternative instruction-conditioned predictions within shared scenes (Renz et al., 2025), whereas ICR-Drive measures driving-score changes under matched instruction perturbations (Hamid et al., 2026). Complementing these protocols, Language2Drive requires *both mutually exclusive, feasible requests* to satisfy their own spatial–temporal contracts in matched closed-loop runs, separating isolated completion from successful execution of both alternatives (Table 3).

5 Conclusion

We introduce Language2Drive, a benchmark that makes reliable passenger control through language an explicit evaluation objective for autonomous-driving vision-language-action models. Its paired counterfactual framework combines matched initial conditions with machine-checkable spatial and temporal criteria, distinguishing individual instruction success from the ability to execute both requested alternatives. We instantiate this framework through an extended CARLA simulator with purpose-designed maps and curated reconstructed scenes in AlpaSim, complemented by open-loop evaluations developed from nuScenes, NAVSIM, and WOD-E2E. Our results reveal a substantial gap between driving performance and reliable instruction following. Across the six principal CARLA systems, 93.2% of successful rollouts lack a successful paired counterpart, and strict paired success reaches at most 1.04%. Success also remains fragile across paraphrases, while ordered, conditional, and reference-return programs expose persistent execution difficulties. These findings establish that producing a plausible driving trajectory is insufficient evidence of instruction following. By evaluating which alternative is selected, whether its requirements are fully executed, and whether success persists across equivalent expressions, Language2Drive provides a concrete basis for developing driving systems that passengers can direct consistently, precisely, and safely through language.

AI USE STATEMENT

AI tools assisted manuscript drafting and editing, mathematical and numerical checks, bibliography verification, and figure preparation. They were not used to execute the driving models or simulator experiments. Experimental measurements and their interpretations are the authors' responsibility.

ETHICS AND REPRODUCIBILITY

The benchmark concerns simulated vehicle behavior. Safety-priority prompts are evaluated within simulation and are not instructions for deployment on public roads. A simulated success is not a certification of real-world driving safety. Distribution of reconstructed assets, maps, and model weights follows their respective rights and licenses. Public model inputs are separated from private scoring geometry and privileged reference trajectories. Reproducibility requires the task manifest, exact adapter/checkpoint identity, run and attempt provenance, and scoring version; the appendix specifies these interfaces and the scope of the reported results. Third-party model weights and private credentials are not part of the manuscript package.

REFERENCES

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 *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 11618–11628. IEEE, 2020.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, et al. Impromptu vla: Open weights and open data for driving vision-language-action models. *Advances in Neural Information Processing Systems*, 38, 2026.

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.

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.

Kaiser Hamid, Can Cui, and Nade Liang. Icr-drive: Instruction counterfactual robustness for end-to-end language-driven autonomous driving. *arXiv preprint arXiv:2604.05378*, 2026.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024.

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.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *International Journal of Computer Vision*, 134(9):407, 2026.

Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026.

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, et al. Xiaomi onevl: One-step latent reasoning and planning with vision-language explanation. *arXiv preprint arXiv:2604.18486*, 2026.

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

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11993–12003. IEEE, 2025.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15120–15130. IEEE, 2024.

Tin Stribor Sohn, Maximilian Dillitzer, Johannes Bach, Jason J Corso, Tim Brühl, Robin Schwager, Tim Dieter Eberhardt, and Eric Sax. Drive4c: A closed-loop benchmark on what foundation models really need to be capable of for language-guided autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3859–3869. IEEE, 2025.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22442–22452. IEEE, 2025a.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025b.

Zehao Wang, Huaide Jiang, Shuaiwu Dong, Yuping Wang, Hang Qiu, and Jiachen Li. Drive my way: Preference alignment of vision-language-action model for personalized driving. *arXiv preprint arXiv:2603.25740*, 2026.

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

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pages 157518–157556, 2026.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, et al. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3709–3718, 2026a.

Yifang Xu, Jiahao Cui, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Feipeng Cai, Neng Zhang, Yaoyi Li, Jia Cai, et al. Wam-flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24918–24928, 2026b.

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.

Kewei Zhang, Jin Wang, Sensen Gao, Chengyue Wu, Yulong Cao, Songyang Han, Boris Ivanovic, Langechuan Liu, Marco Pavone, Song Han, et al. Fast-ddrive: Efficient block-diffusion vlm for autonomous driving. *arXiv preprint arXiv:2605.23163*, 2026.

Weicheng Zheng, Yixin Huang, Qiao Sun, Derun Li, and Hang Zhao. Drivema: Driving vision-language-action models with verifiable meta-actions. *arXiv preprint arXiv:2605.31271*, 2026.

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, et al. Qwen-drive-1.0: An initial step towards a vision-language foundation model for autonomous driving. *arXiv preprint arXiv:2609.00111*, 2026a.

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, 2026b.

Zewei Zhou, Tianhui Cai, Seth 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. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026c.

A FORMAL PROPERTIES

A.1 WHAT A PAIRED TEST RULES OUT

For PCU i, define the strict acceptance set for request j by

$$\mathcal{A}_{i,j} = \{\tau : O_{i,j}(\tau)(1 - O_{i,1-j}(\tau))S_{i,j}(\tau)P_{i,j}(\tau)T_{i,j}(\tau)H_{i,j}(\tau) = 1\}.$$

The two sets are disjoint because each requires the paired-alternative criterion to fail. This remains true even if a permissive trajectory can satisfy both raw target predicates.

Proposition 1 (Instruction-invariant execution)**.** *With identical non-language initialization and fully coupled execution randomness, an instruction-invariant system has zero strict CFPS and at most 50 SISR on balanced, mutually exclusive arm acceptances.*

Proof. An instruction-invariant system receives the same effective inputs at initialization. The same control law and coupled randomness give identical actions and simulator transitions. Induction over time therefore gives the same executed trace τ under both requests. If both arms succeeded, τ would belong to the disjoint sets $\mathcal{A}_{i,0}$ and $\mathcal{A}_{i,1}$, a contradiction. At most one arm can succeed, so every pair contributes zero to CFPS and at most one half to SISR. A default behavior satisfying exactly one request in every pair attains the SISR bound. □

Coupling is a substantive assumption. An instruction-independent policy that independently chooses two valid behaviors with probabilities q and $1 - q$ in the two runs has expected paired success $q(1 - q)$, up to 25%, despite ignoring language. Coupling that behavior-selection randomness instead produces the same choice in both runs and zero CFPS. A shared seed label does not guarantee identical latent randomness across asynchronous workers or decoding implementations. The empirical scores establish execution of matched alternatives under the reported system; instruction-ablated repetitions would additionally test causal language dependence.

A.2 INDIVIDUAL–PAIRED IDENTITIES AND BOUNDS

For $x, y \in \{0, 1\}$,

$$\frac{x + y}{2} = xy + \frac{|x - y|}{2}.$$

Averaging with the same PCU/expression weights proves Equation 2. For $x, y \in [0, 1]$,

$$\min(x, y) = \frac{x + y - |x - y|}{2},$$

which proves the SCS–PCS imbalance identity in Section 2.4. Since

$$xy \le \frac{x + y}{2}, \quad xy \ge x + y - 1 \quad (x, y \in \{0, 1\}),$$

$$\min(x, y) \le \frac{x + y}{2}, \quad \min(x, y) \ge x + y - 1 \quad (x, y \in [0, 1]),$$

each individual/paired metric family (U, V) obeys

$$\max(0, 2U - 100) \le V \le U. \tag{6}$$

This applies to $(\mathrm{SISR}, \mathrm{CFPS})$, $(\mathrm{SemISR}, \text{Sem-CFPS})$, and $(\mathrm{SCS}, \mathrm{PCS})$ when the instances and weights agree. Strict success implies the closed-loop relaxed semantic verdict, hence $\mathrm{SISR} \le \mathrm{SemISR}$ and $\mathrm{CFPS} \le \text{Sem-CFPS}$. The graded map is a separate rule, so no analogous ordering with SemISR is assumed. PCS reduces to the corresponding binary paired rate only when its completion scores are that same binary success indicator.

A.3 ALL-EXPRESSION PAIRED SUCCESS

CARLA fixes three expressions per arm and one seed. Let $b_{ir} = s_{ir,0}s_{ir,1}$. We report

$$W = \frac{100}{N} \sum_i \min_{r \in \{0,1,2\}} b_{ir}.$$

Both arms must succeed under every tested expression. This is not the minimum of three dataset-average CFPS values. It satisfies $W \le \mathrm{CFPS}$ for the uniform expression weighting used here. A zero W is compatible with occasional successful pairs; RealSim's one-expression evaluation does not estimate wording robustness.

A.4 PROPERTIES OF THE L2D AGGREGATION

Let $a = (a_1, a_2, a_3) \in [0, 100]^3$, $M = (a_1 + a_2 + a_3)/3$, and $L(a) = \left(\sum_{k=1}^{3} \sqrt{a_k}/3\right)^2$. For a fixed driving-credit cap $b \in (0, 100)$, define

$$F_b(a, d) = d \left[b + (100 - b) \frac{L(a)}{100} \right], \qquad d \in [0, 1].$$

The reported score fixes $b = 1$.

Proposition 2 (Balance without a family-level zero veto)**.** *For every $a \in [0, 100]^3$,*

$$L(a) = M - \frac{1}{9} \sum_{k<h} (\sqrt{a_k} - \sqrt{a_h})^2, \qquad \frac{M}{3} \le L(a) \le M.$$

Thus $L(a) = 0$ if and only if all three families are zero. For a fixed M, the maximum is attained precisely when all families are equal.

Proof. Expand $(\sum_k \sqrt{a_k})^2$ to obtain the identity. Its nonnegative cross terms give $L(a) \ge \sum_k a_k/9 = M/3$. Cauchy–Schwarz gives $(\sum_k \sqrt{a_k})^2 \le 3 \sum_k a_k$, hence $L(a) \le M$. Equality in the upper bound requires equal square roots, proving the balance statement. □

Proposition 3 (Monotonicity and weaker-family priority)**.** *L is continuous and strictly increasing in each family coordinate. For positive a_i, a_j,*

$$\frac{\partial L}{\partial a_i} = \frac{\sum_k \sqrt{a_k}}{9\sqrt{a_i}}, \qquad \frac{\partial L / \partial a_i}{\partial L / \partial a_j} = \sqrt{\frac{a_j}{a_i}}.$$

Thus an equal marginal increase receives greater weight in the weaker family. At fixed $d > 0$, F_b preserves strict coordinatewise monotonicity.

Proof. Each square root is continuous and strictly increasing, and the final square is strictly increasing on nonnegative inputs. Differentiation gives the displayed expressions. The multiplier $(100 - b)d/100$ is positive when $d > 0$. The continuity and monotonicity statements also hold at zero, without requiring a finite derivative there. □

Proposition 4 (Bounded driving-only credit and exact separation)**.** *The composite satisfies $0 \le F_b \le 100$, $F_b(a, 0) = 0$, and $F_b((100, 100, 100), 1) = 100$. If $a = 0$, then $F_b(0, d) = bd \le b$; distinct driving scores remain distinguishable. Moreover,*

$$F_b - bd = (100 - b) \frac{L(a)}{100} d, \qquad F_b \le \min \left\{ 100d,\ b + \left(1 - \frac{b}{100}\right) L(a) \right\}.$$

In particular, $F_b > b$ implies nonzero language completion credit.

Proof. By the preceding bounds, $0 \le L(a)/100 \le 1$. Consequently the bracket in F_b lies in $[b, 100]$. Substitution proves the boundary statements and decomposition. The two upper bounds follow respectively from the bracket being at most 100 and from $d \le 1$. □

Proposition 5 (Limited perturbation of the multiplicative score)**.** *Let $T = L(a)d$ be the score without driving-only credit. Then*

$$F_b = \left(1 - \frac{b}{100}\right) T + bd, \qquad 0 \le F_b - T = bd \left(1 - \frac{L(a)}{100}\right) \le b.$$

Therefore any pairwise score gap greater than b under T cannot be reversed by this change. Also $|F_b - F_{b'}| \le |b - b'|$ for fixed a, d.

Proof. The identities follow by expansion. For systems A, B, write $F_{b,A} = T_A + e_A$ and $F_{b,B} = T_B + e_B$, with $e_A, e_B \in [0, b]$. If $T_A - T_B > b$, then $F_{b,A} - F_{b,B} \ge T_A - T_B - b > 0$. Finally $\partial F_b / \partial b = d(1 - L(a)/100) \in [0, 1]$ proves the cap-sensitivity bound. □

Proposition 6 (Unique fusion under explicit endpoint and linearity conditions)**.** *Let $G(\ell, d)$ obey $G(\ell, d) = dG(\ell, 1)$, let $G(\ell, 1)$ be affine in ℓ, and require $G(0, 1) = b$ and $G(1, 1) = 100$. Then necessarily $G(\ell, d) = d[b + (100 - b)\ell]$.*

Proof. Write $G(\ell, 1) = u + v\ell$. The endpoint conditions imply $u = b$ and $v = 100 - b$. The first condition then determines $G(\ell, d)$ for every d. □

These results establish the specified range, balance, partial-credit, and cap properties. They do not identify a universally optimal b, prove causal use of language, or replace the individual and paired success metrics.

A.5 Complementarity and fixed evaluation scale

Write $\ell = L(a)/100$. Differentiating $F_b = d[b + (100 - b)\ell]$ gives

$$\frac{\partial F_b}{\partial \ell} = (100 - b)d, \qquad \frac{\partial F_b}{\partial d} = b + (100 - b)\ell, \qquad \frac{\partial^2 F_b}{\partial \ell \partial d} = 100 - b > 0.$$

Language improvement contributes more when driving quality is higher, and driving improvement contributes more when language completion is higher. Driving affects both terms; the one-point cap is not a claim that driving has only 1% weight. The arithmetic driving component is not a hard veto on each safety metric, and L2D is not a rollout-level joint success probability. PSM is measured on independent routes.

For fixed tasks, scorer, $q = 1/2$, and $b = 1$, a system's score depends only on its own component metrics. Adding other systems does not change it. Nonlinear aggregation is applied *after* the population metrics are computed; averaging a per-PCU nonlinear score would define a different metric. The proofs establish the chosen scoring properties, not statistical significance, causal language use, or a universal ordering of driving systems.

B METRIC DEFINITIONS AND PROPERTIES

B.1 EVALUATION UNITS AND STRICT VERDICTS

For PCU i, expression r, and arm j, let $\tau_{ir,j}$ denote the executed trajectory. The two arms share their initial non-language context but are executed separately under their respective instructions. Let $O_{i,j}(\tau)$ indicate complete satisfaction of the requested task criterion. Using the verifier's safety, progress, timing, and stability gates, the strict verdict is

$$\begin{aligned} s_{ir,j} = O_{i,j}(\tau_{ir,j})\big[1 - O_{i,1-j}(\tau_{ir,j})\big] \\ \times S_{i,j}(\tau_{ir,j})P_{i,j}(\tau_{ir,j})T_{i,j}(\tau_{ir,j})H_{i,j}(\tau_{ir,j}). \end{aligned} \tag{7}$$

Each factor is binary, and the gates use the applicable track-specific rules. Alternative rejection makes the two strict acceptance sets disjoint. Persistent requirements apply to a reached, nonempty interval that must be completed; stopping before that interval does not satisfy the requirement. The task specifications retain the exact spatial tolerances, deadlines, dwell requirements, and event-ordering rules.

All language scores use the expression-then-PCU averaging operator defined in Section 2.4. The scheduled pair set is fixed within each protocol, and unsuccessful executions remain in its denominator. Equal PCU weighting prevents a task with more expressions from receiving greater weight. It is equivalent to pooling all paired instances only when each PCU has the same number of expressions. The evaluated CARLA protocol has $N = 64$ and $R_i = 3$; RealSim has $N = 30$ and $R_i = 1$. Duplicate retries and smoke tests do not create additional evaluation instances.

B.2 RELAXED PROXIES AND GRADED COMPLETION

Relaxed semantic proxies. Let $\widetilde{O}_{i,j}$ denote a predefined task-specific relaxation of the requested criterion. The relaxed verdict and its aggregate rates are

$$\begin{aligned} m_{ir,j} &= \mathbf{1}\Big\{O_{i,j}(\tau_{ir,j}) = 1 \text{ or } \widetilde{O}_{i,j}(\tau_{ir,j}) = 1\Big\}, \\ \mathrm{SemISR} = 100\left\langle \frac{m_{ir,0} + m_{ir,1}}{2} \right\rangle, &\qquad \text{Sem-CFPS} = 100\langle m_{ir,0} m_{ir,1} \rangle. \end{aligned} \tag{8}$$

These are deterministic task rules, not free-form language-model judgments. CARLA proxies can relax target geometry, heading, or stopped dwell; sequence and reference proxies retain ordered event checks. RealSim uses its own task-specific rules: some criteria relax timeliness or dwell, whereas scoped speed tasks retain complete entry-to-exit traversal and compliance at every active sample. The proxies do not impose the same universal alternative-rejection and execution-gate conjunction as Equation 7. Passing both relaxed criteria therefore need not demonstrate exclusive, complete execution of both requests. A relaxed verdict neither terminates a rollout nor reduces its opportunity to reach strict success.

Graded completion. The task-specific map $C_{i,j}$ assigns $c_{ir,j} \in [0, 1]$ to the execution of its own request. Equation 3 averages the two arms for SCS and takes the weaker arm for PCS. These scores preserve partial completion without interpreting it as a binary success probability. The graded map is separate from the relaxed predicate: neither $\mathrm{SCS} \leq \mathrm{SemISR}$ nor the reverse is assumed. PCS reduces to CFPS only when the per-arm graded scores equal the strict binary verdicts themselves. Task-specific maps and tolerances remain part of the evaluator specification.

B.3 INDIVIDUAL–PAIRED IDENTITIES AND INTERPRETATION

For binary x, y,

$$\frac{x + y}{2} = xy + \frac{|x - y|}{2}. \tag{9}$$

Applying the same averaging weights to both sides proves Equation 2. The SISR–CFPS difference is thus exactly half the percentage of one-sided paired instances. For graded scores $x, y \in [0, 1]$,

$$\min(x, y) = \frac{x + y - |x - y|}{2}, \tag{10}$$

which yields

$$\mathrm{SCS} - \mathrm{PCS} = 50\langle |c_{ir,0} - c_{ir,1}| \rangle. \tag{11}$$

This difference quantifies completion imbalance between alternatives, not a particular internal failure mechanism.

Bounds. For each individual–paired family (U, V),

$$\max(0, 2U - 100) \le V \le U. \tag{12}$$

This follows from $xy \le (x + y)/2$ and $xy \ge x + y - 1$ for binary verdicts, and the analogous inequalities for $\min(x, y)$ on $[0, 1]^2$. It applies to $(\mathrm{SISR}, \mathrm{CFPS})$, $(\mathrm{SemISR}, \text{Sem-CFPS})$, and $(\mathrm{SCS}, \mathrm{PCS})$ when their instances and weights agree. Strict success implies the relaxed verdict, giving $\mathrm{SISR} \le \mathrm{SemISR}$ and $\mathrm{CFPS} \le$ Sem-CFPS. These inequalities do not extend automatically to the separately defined graded scores.

The denominator of one-sided success. The percentage of *pairs* with one successful arm, q_{one}, differs from the percentage of *successful arms* lacking a successful counterpart. With equal instance weights, let n_2 and n_1 count pairs with two and one successful arms. There are $2n_2 + n_1$ successful arms, of which n_1 are unpaired. More generally, under the same metric weights and $\mathrm{SISR} > 0$, the latter percentage is

$$100 \frac{q_{\mathrm{one}}}{2\,\mathrm{SISR}} = 100 \left(1 - \frac{\mathrm{CFPS}}{\mathrm{SISR}}\right). \tag{13}$$

This is an interpretation of the existing success decomposition, not an additional primary metric. It is undefined when no instruction succeeds. For pooled counts, the numerator and denominator must be pooled before taking the ratio; averaging per-system percentages generally gives a different quantity.

B.4 Consistency across expressions

Outcome changes among successful arms. For arm (i, j), define

$$\begin{aligned} f_{i,j} &= \mathbf{1}\left\{0 < \sum_{r=0}^{2} s_{ir,j} < 3\right\}, \\ a_{i,j} &= \mathbf{1}\left\{\sum_{r=0}^{2} s_{ir,j} > 0\right\}. \end{aligned} \tag{14}$$

We report $100\,\frac{\sum_{i,j} f_{i,j}}{\sum_{i,j} a_{i,j}}$ as Flip/any. The percentage is not applicable when no arm succeeds. Conditioning on at least one success separates fragile successes from consistently unsuccessful arms. These statistics describe execution under the tested wordings. With one evaluation seed, they do not isolate wording effects from residual execution variability.

B.5 Track-specific advancement and safety

CARLA core advancement. For the set $\mathcal{V}$ of valid core traces, let $\pi_v(t) \in [0, 1]$ be normalized projection onto the task-specific private reference path. Then

$$\mathrm{DriveScore} = \frac{100}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \max_t \pi_v(t). \tag{15}$$

The maximum records the furthest attained reference progress; it does not enforce target exclusivity, ordering, dwell, or the complete instruction contract. This valid-trace denominator differs conceptually from the scheduled denominator for language scores. All seven displayed CARLA configurations have 384 valid core traces.

RealSim advancement and safety. Let λ_v be the length of the frozen arm-specific reference path truncated at the task deadline T_v. For projected arc length $a_v(t)$ and distance $e_v(t)$ to that path, define

$$g_v = \mathrm{clip}\left(\frac{\max\big(\{0\} \cup \{a_v(t) : t \le T_v,\ e_v(t) \le 5\,\mathrm{m}\}\big)}{\lambda_v}, 0, 1\right),$$
$$\mathrm{DrivingScore} = \frac{100}{60}\sum_{v=1}^{60} g_v. \tag{16}$$

Failed or unavailable slots contribute no advancement credit rather than being removed from the fixed denominator. Safety separately records collision/off-road-free execution under scoring v0.2. The strict Progress gate is not this continuous score: RealSim uses a 10 m threshold over the available rollout, including the pre-instruction segment, while CARLA's default gate requires at least 0.05 evaluator progress. CARLA's strict safety gate also excludes red-light events; RealSim's requested speed constraints remain language criteria.

Independent driving routes. FDQ evaluates 24 independent routes, one per ability group. Its strict pass requires a Completed/Perfect status and no recorded infractions. The accompanying continuous score is

$$\mathrm{DS}_v = \mathrm{RC}_v\mathrm{IP}_v, \qquad \mathrm{DS} = \frac{1}{24}\sum_{v=1}^{24}\mathrm{RC}_v\mathrm{IP}_v. \tag{17}$$

RC and DS use a 0–100 scale; the infraction penalty IP uses 0–1. Multiplying mean RC by mean IP does not in general recover mean DS. These routes supply complementary driving diagnostics, not additional instruction PCUs.

B.6 The L2D summary and its properties

Definition. L2D combines already aggregated component metrics for compact within-track reporting. Define the three language-family scores and their balanced aggregate:

$$\begin{aligned} a_1 &= \tfrac{1}{2}(\mathrm{SISR} + \mathrm{CFPS}),\\ a_2 &= \tfrac{1}{2}(\mathrm{SemISR} + \text{Sem-CFPS}),\\ a_3 &= \tfrac{1}{2}(\mathrm{SCS} + \mathrm{PCS}),\\ L_{\mathrm{bal}} &= \left(\frac{\sqrt{a_1} + \sqrt{a_2} + \sqrt{a_3}}{3}\right)^2. \end{aligned} \tag{18}$$

All a_k and L_{bal} lie in $[0, 100]$. The track-specific driving component is

$$D_t = \begin{cases} \tfrac{1}{2}(\mathrm{PSM} + \mathrm{DriveScore}), & t = \mathrm{CARLA},\\ \tfrac{1}{2}(\mathrm{Safety} + \mathrm{DrivingScore}), & t = \mathrm{RealSim}, \end{cases} \tag{19}$$

and the reported summary is

$$\mathrm{L2D}_t = \frac{D_t}{100}\big(1 + 0.99 L_{\mathrm{bal}}\big). \tag{20}$$

The square-root mean retains partial credit while favoring balance across language families. With zero credit in all three language families, L2D is at most one point. This is a summary of distinct measurements, not a substitute for strict success or a rollout-level joint-success probability.

Balance and monotonicity. Write $L(\mathbf{a}) = L_{\mathrm{bal}}$ and $M = (a_1 + a_2 + a_3)/3$. Expanding the square yields

$$L(\mathbf{a}) = M - \frac{1}{9}\sum_{k<h}(\sqrt{a_k} - \sqrt{a_h})^2, \qquad \frac{M}{3} \le L(\mathbf{a}) \le M. \tag{21}$$

Nonnegative cross terms give the lower bound; Cauchy–Schwarz gives the upper bound. At fixed M, equality in the upper bound holds when all families are equal. Also, $L(\mathbf{a}) = 0$ if and only if all

three family scores are zero, so a single zero family does not veto all partial credit. The aggregate is continuous and strictly increasing in every family score. For positive a_i, a_j,

$$\frac{\partial L}{\partial a_i} = \frac{\sum_k \sqrt{a_k}}{9\sqrt{a_i}}, \qquad \frac{\partial L/\partial a_i}{\partial L/\partial a_j} = \sqrt{\frac{a_j}{a_i}}. \tag{22}$$

Thus an equal marginal improvement receives greater weight in the weaker family. Monotonicity also holds at zero, without requiring a finite derivative there.

Bounded driving-only credit. For $d = D_t/100$ and a general cap $\beta \in (0, 100)$, define

$$F_\beta(\mathbf{a}, d) = d\left[\beta + (100 - \beta)\frac{L(\mathbf{a})}{100}\right]. \tag{23}$$

Equation 20 fixes $\beta = 1$. The bounds on L imply $0 \le F_\beta \le 100$, $F_\beta(\mathbf{a}, 0) = 0$, and $F_\beta((100, 100, 100), 1) = 100$. Moreover,

$$\begin{aligned} F_\beta(\mathbf{0}, d) &= \beta d \le \beta, \\ F_\beta - \beta d &= (100 - \beta)\frac{L(\mathbf{a})}{100} d, \\ F_\beta &\le \min\left\{100d,\ \beta + \left(1 - \frac{\beta}{100}\right) L(\mathbf{a})\right\}. \end{aligned} \tag{24}$$

A score above the driving-only cap therefore requires nonzero language-family credit, which can include partial rather than strict completion.

Cap sensitivity and the specified fusion rule. Relative to the multiplicative score $T = L(\mathbf{a})d$,

$$F_\beta = \left(1 - \frac{\beta}{100}\right) T + \beta d, \qquad 0 \le F_\beta - T = \beta d\left(1 - \frac{L(\mathbf{a})}{100}\right) \le \beta. \tag{25}$$

Consequently, a pairwise gap greater than β under T cannot be reversed by this bounded addition. At fixed $\mathbf{a}, d$, $|F_\beta - F_{\beta'}| \le |\beta - \beta'|$. Writing $\ell = L(\mathbf{a})/100$, the form $G(\ell, d) = d[\beta + (100 - \beta)\ell]$ is uniquely determined by proportionality in d, affinity in ℓ at $d = 1$, and endpoints $G(0, 1) = \beta$ and $G(1, 1) = 100$. These are explicit design conditions, not a claim that the cap or fusion rule is universally optimal.

Interpretation and aggregation order. The cross-partial derivative is $\partial^2 F_\beta/(\partial\ell\,\partial d) = 100 - \beta > 0$: improvements in either component contribute more when the other is larger. The one-point cap is therefore not a claim that driving has only one-percent weight. The arithmetic driving component is not a hard veto on every safety metric, and CARLA PSM is measured on separate routes. Nonlinear aggregation is applied *after* computing population component scores; averaging nonlinear per-PCU summaries would define a different metric. For fixed tasks, scorer, and constants, adding other evaluated systems does not change a system's score. The reported tables compute L2D from displayed two-decimal components using high-precision intermediate arithmetic and final display rounding. CARLA and RealSim retain separate rankings. The properties above neither establish statistical separation between nearby scores nor identify causal use of language.

B.7 Open-loop semantic success

Open-loop evaluation checks predicted trajectories without controller execution. For a binary task criterion $E_{i,j}$, define exclusive semantic success as

$$B_{ir,j} = E_{i,j}(\tau_{ir,j})\big[1 - E_{i,1-j}(\tau_{ir,j})\big]. \tag{26}$$

The corresponding metrics are

$$\mathrm{SemSISR} = 100\left\langle \frac{B_{ir,0} + B_{ir,1}}{2} \right\rangle, \qquad \mathrm{SemCFPS} = 100\langle B_{ir,0} B_{ir,1}\rangle. \tag{27}$$

The names remain distinct from closed-loop SemISR and Sem-CFPS because their acceptance rules differ: both open-loop arms explicitly reject their alternative. These scores evaluate semantic properties of predictions, not realized closed-loop safety or execution.

SCS and PCS use Equation 3 with the open-loop task maps. The graded map assigns one when all binary task constraints are met; otherwise, native tolerances or acceptance intervals determine linear decay, with conjunctions taking the minimum. Lane tasks use footprint overlap and endpoint progress, while stopping tasks combine target distance and stopping speed. Ordered-event tasks score zero when neither event occurs, 0.5 when one occurs, and use the event interval when both occur. Other tasks retain their speed, distance, and relative-position conditions. Invalid or unparseable predictions receive zero completion credit. Scores and sample coverage are reported within each dataset without a pooled cross-dataset rank.

B.8 Additional driving and behavioral diagnostics

Supplementary FDQ measurements. The auxiliary per-route statistics use the evaluator's event counts, reference curve, normalized exceedances, and recovery quantities:

$$\mathrm{SI}_v = \exp[-(3N_{\mathrm{ped},v} + 2N_{\mathrm{veh},v} + N_{\mathrm{layout},v})], \tag{28}$$

$$\mathrm{GPE}_v = r_v \exp\left[-0.5 \max\left(0, \frac{T_v}{T_{\mathrm{ref}}(r_v)} - 1\right)\right], \quad r_v = \mathrm{RC}_v/100, \tag{29}$$

$$\mathrm{PSM}_v = \frac{\sum_t \Delta t \ \mathrm{clip}(\mathrm{TTC}_{\min,v}(t)/3, 0, 1)}{\sum_t \Delta t}, \tag{30}$$

$$\mathrm{MDQ}_v = \exp\left[-\frac{1}{6}\sum_{q=1}^{6} E_{v,q}\right], \tag{31}$$

$$\mathrm{RLS}_v = \frac{1}{|\mathcal{J}_v|}\sum_{j\in\mathcal{J}_v} u_j \exp(-t_{\mathrm{recover},j}/5). \tag{32}$$

These auxiliary scores are retained on their original 0–1 scales with equal-route means; the maintable PSM is $100\,(24^{-1}\sum_{v=1}^{24}\mathrm{PSM}_v)$. Infinite TTC maps to one after clipping, whereas missing TTC is not treated as infinity. PSM uses the recorded constant-velocity oriented-box TTC model with a 75 m observation radius and 0.2 m clearance. GPE is zero at $r_v = 0$, and T_{ref} is a privileged-reference curve rather than a mathematical optimum. RLS is unavailable when no recovery opportunity exists; its reported mean is accompanied by coverage.

MDQ uses a seven-point centered mean and finite differences for longitudinal and lateral acceleration, yaw rate, yaw acceleration, longitudinal jerk, and jerk magnitude. The acceleration ranges are $[-4.05, 2.40]$ and $[-4.89, 4.89]\,\mathrm{m/s^2}$ for longitudinal and lateral motion, respectively. Yaw rate and yaw acceleration use $[-0.95, 0.95]\,\mathrm{rad/s}$ and $[-1.93, 1.93]\,\mathrm{rad/s^2}$. Longitudinal jerk and jerk magnitude use $[-4.13, 4.13]$ and $[0, 8.37]\,\mathrm{m/s^3}$, respectively. Normalized exceedances are clipped to $[0, 1]$. These are engineering thresholds, not independently established human-expert comfort standards. Stationary behavior can score well on SI and MDQ, so these measures must be read alongside advancement. They do not change FDQ's strict pass criterion.

Safety-conflict diagnostics. Safety-Q reports behavior-based strict completion, relaxed completion, safety, and unsafe obedience over six cases. Fallback, refusal, and constrained-alternative completion each use their own two-case denominator. Unsafe obedience records the prohibited opposite-contract outcome; it does not additionally require a separate unsafe flag. Avoiding the prohibited action and completing a useful safe replacement remain distinct outcomes. No clarification-hold or benign matched requests are included, so clarification and unnecessary-refusal rates are not applicable. These diagnostics do not evaluate the quality of a textual refusal explanation.

Trajectory-response and geometric diagnostics. For RealSim, complete paired trajectories are sampled at 101 common times from instruction issuance at 1.7 s to their shared horizon. Mean position separation and mean absolute speed difference are computed per pair and then summarized by the median. Spatial statistics use complete valid pairs; speed statistics use the specified speed-task subset, with coverage reported separately. For WOD-E2E, the geometric criterion requires each prediction to be closer in ADE to its own reference than to the alternative by more than 0.5 m. These are diagnostics of response magnitude or reference alignment, not replacements for instruction-specific success. Their valid-trajectory cohorts remain separate from the scheduled semantic-success denominators.

B.9 WHAT PAIRED SUCCESS ESTABLISHES

Instruction-invariant execution under coupled randomness. Let $\mathcal{A}_{i,j}$ be the strict acceptance set induced by Equation 7. Alternative rejection gives $\mathcal{A}_{i,0} \cap \mathcal{A}_{i,1} = \varnothing$. With identical non-language initialization and fully coupled execution randomness, an instruction-invariant system produces the same executed trajectory under both requests. That trajectory cannot lie in both acceptance sets. Such a system therefore has zero CFPS and at most 50 SISR on balanced pairs. The bound is attained when the same behavior satisfies exactly one request in every pair.

Stochastic execution. Coupling is substantive, not merely a shared seed label. An instruction-independent policy that independently selects the two valid behaviors with probabilities p and $1-p$ in the two runs has paired-success probability $p(1-p)$ and expected CFPS $100p(1-p) \leq 25$. Coupling its behavior-selection randomness instead makes it choose the same behavior in both runs and yields zero paired success. Asynchronous execution and decoding can prevent a shared seed label from coupling all relevant randomness. Empirical CFPS therefore establishes successful matched execution under the reported system and protocol; instruction-ablated repetitions would additionally test causal language dependence.

Statistical interpretation. The current model comparisons use one evaluation seed per track. CARLA expressions are clustered within PCUs, and several RealSim PCUs share a reconstructed scene. Uncertainty estimates must respect these units rather than treat all arms and expressions as independent samples. Aggregate scores alone do not provide such estimates, and small score differences are descriptive rather than evidence of statistical separation. The reported metrics do not substitute for unperformed instruction ablations, repeated-seed tests, or input/controller interventions.

C TASK SPECIFICATION AND QUALIFICATION

C.1 ENVIRONMENTS, COHORTS, AND LANGUAGE COVERAGE

CARLA-Language2Drive adds six purpose-designed environment types to CARLA: underground parking, a port, Chinese city roads, a US city with detailed signage, a campus, and a highway. The evaluated base spans the ten map identifiers in Table 4: five Town01–Town05 layouts and five custom non-Town assets. It is a subset of the simulator distribution, not an evaluation of every newly added map.

Table 4: Actual CARLA base map manifest. These ten evaluated layouts include five custom non-Town assets. The distribution-level six-environment description and this evaluation manifest have different scopes.

Map identifier	PCUs	Core rollouts
American-city0428	11	66
Town01	2	12
Town02	1	6
Town03	10	60
Town04	4	24
Town05	2	12
hanghailou3	1	6
hubeiwenli4	2	12
parkinglotfish-new3	24	144
yangluogang1	7	42

Table 5: Compiled source suite and nested CARLA cohorts. Columns count PCUs and potential Core executions; only the base single-seed cohort is the complete comparison.

Cohort	PCUs	Arms	One seed	Three seeds
Source	128	256	768	2304
Small	32	64	192	576
Base	64	128	384	1152
Large (loadable)	119	238	714	2142

The source suite contains 128 PCUs, 32 per scope, with 256 arms and 768 arm–expression instances per seed. Nine Town06 PCUs are excluded because that map is not loadable in the evaluation runtime, leaving 119. Stable within-scope sampling selects nested 32-PCU and 64-PCU tiers. For a scope with n_l loadable PCUs in source order, base selects indices $\lfloor jn_l/16 \rfloor$, $j = 0, \ldots, 15$; small takes every second selected entry. The evaluated base has 14 development and 50 test PCUs. Development/test counts are A: 4/12, B: 3/13, C: 3/13, and D: 4/12. All arms and expressions of a PCU remain in the same split.

The split audit checks semantic entities, physical topology, and actor programs: 217, 158, and 13 recorded keys, with no cross-split conflicts in those sets. Controlled English templates substitute directions, entity names, speeds, and stage identities. The base contains 384 instruction entries and 216 distinct strings, averaging 7.34 whitespace-delimited words. Each scope contributes 16 PCUs and 96 rollouts per system. Deadlines use simulation time: A, 35–120 s; B, 80–100 s; C, 55–200 s; D, 100–260 s.

C.2 FROM A PHYSICAL CHOICE TO A PAIRED SPECIFICATION

The construction workflow links five stages: environment curation, geometric binding, paired instruction authoring, compilation, and validation. Directed lane graphs use 1 m samples, refined to 0.5 m at junctions and throughout the parking map. Road names bind to exit gates; regions and parking spaces bind to world-coordinate polygons. Topology endpoints preserve short lane sections, and successor/adjacency references must resolve. Structural labels such as walls and pillars are excluded from parking targets.

Table 6: CARLA base task inventory, with three paraphrases per arm and seed 1.

Stratum	Category	PCUs	Rollouts
A	junction	7	42
A	lane	4	24
A	speed_start	2	12
A	waiting_parking	3	18
B	lane_constraint	4	24
B	object_relation	4	24
B	passing_order	4	24
B	speed_constraint	4	24
C	exact_parking	4	24
C	parking_region	4	24
C	port_target	4	24
C	road_name	4	24
D	conditional	5	30
D	memory_reference	4	24
D	strict_order	5	30
D	until_invariant	2	12

Each arm specifies language expressions, target geometry, the requested and alternative contracts, a private reference path, a deadline, and actor setup. Public records contain language, sensors, ego state, and shared neutral context. The private sidecar retains targets, acceptance predicates, future reference paths, and actor-program specifications. Pair checks enforce matching non-language initialization. Static qualification checks entity bindings, directed reachability, footprint geometry, and opposition of the requested outcomes.

The source contains 38 distinct named physical parking spaces across exact-parking branches; 33 are selected by the instantiated scene conditions. They occur in 22 A/C/D PCUs and 44 selected arms. The default exact-parking check uses a 0.2 m boundary tolerance, 15° heading tolerance, speed below 0.2 m/s, and 3 s dwell; task sidecars supply the actual parameters. Physical actors instantiate occupied bays. The maneuver checks retain actor-origin/rear-axle frames and a 2.86047 m wheelbase model. Sustained-speed tasks use signal-free corridors and separated feasible bands; common bands are [1.9,3.9] and [5.9,7.9] m/s.

CARLA's live oracle validation covers 47 targeted cases. Named-road qualification checks spatial and directional agreement for 16 arms within 15 m and 20°. These tests complement static geometric validation; they do not constitute exhaustive dynamic qualification of all 768 source instances. Task-specific deadlines are fixed in the private sidecars.

C.3 Representative instruction pairs

The alternatives in Table 7 share a scene entry and change the required behavior. The task inventories in Appendix I preserve all evaluated identities.

Table 7: Examples of the evaluated CARLA instruction pairs.

PCU	Arm 0	Arm 1
A-001	Turn left at this intersection.	Go straight through this intersection.
B-025	Safely pass the slow truck.	Do not overtake; remain behind the slow truck.
C-001	Take Middleton Street.	Take E Gage Ave.
C-025	Park in space B204.	Park in space B205.
D-001	First visit B208, then go to F206.	First visit F206, then go to B208.
D-009	If E101 is occupied, park in E102; otherwise use E101.	If E101 is occupied, park in C104; otherwise use E102.
D-026	Remember A804; visit E101, then return to that remembered place.	Remember E101; visit A804, then return to that remembered place.

C.4 FINITE-TRAJECTORY CRITERIA AND CLOCKS

The verifier checks normalized, time-ordered actual ego states. Geometric and behavioral predicates represent gate entry, lane relation, speed, headway, actor order, region occupancy, and parking state. `eventually` requires a target event; `always` requires a condition throughout a reached, nonempty interval; `within` imposes a deadline; `until` maintains an invariant until a terminating event. `sequence` checks ordered stages and can reject premature, simultaneous, repeated, or oscillatory stage transitions. `if_else` selects a branch from the instantiated scene condition, and `bind_reference` preserves an entity for later reuse.

For example, A-001 binds its left and straight requests to distinct exit gates. D-001 reverses the ordered parking stages B208 and F206. D-026 stores a named place and requires returning to the same entity after an intermediate visit. The full strict verdict additionally applies alternative rejection and the safety, progress, timing, and stability terms.

The CARLA monitor evaluates growing prefixes of the executed trajectory. A strict-success candidate normally persists for 2 s, and persistent tasks must reach their required horizon. Relaxed semantic success is an auxiliary scoring verdict: it does not terminate a rollout or shorten the opportunity to meet the full strict criterion. Execution ends upon strict completion, the task deadline, or an invalid out-of-world ego state. The deadline component uses satisfied-subcondition timestamps, or the trace endpoint when unavailable; it is not the negation of a `deadline` termination label.

RealSim distinguishes recording time, rollout-relative time, and instruction time. Language is issued at 1.7 s. Deadlines are 7.0 s for 23 PCUs and up to 13.8 s for the remainder, leaving 5.3–12.1 s after instruction. Privileged reference traces extend 6 s beyond the deadline at 0.1 s sampling; that tail is not extra model execution time. The component validator uses a default maximum sample gap of 0.25 s and numerical speed tolerance of 10^{-6} m/s.

C.5 RELAXED PROXIES, GRADED COMPLETION, AND TASK PROGRESS

CARLA SemISR accepts the full requested contract or its predefined relaxed proxy. Depending on the task, the proxy may use heading, displacement, expanded gate proximity, a lane relation with progress, an observation in an expanded speed/headway band, or relaxed region geometry. Parking proxies may accept an expanded footprint-center region and 45° heading tolerance without stopped dwell. Sequence and reference proxies retain ordered event checks. These are deterministic partial-completion rules, not a free-form language-model judgment.

RealSim branch tasks use the first correct irreversible commitment, with timeliness relaxed in the semantic component. Lane tasks can relax dwell. Scoped-lane and speed tasks retain full entry-to-exit traversal, relevant exclusivity and trigger conditions, and, for speed, all active samples in the requested band. Target-lane dwell is 1 s; scoped-lane tasks require a 2 s minimum dwell. Referent triggers must occur after instruction and by scope entry, then remain latched until exit.

The task-specific maps $C_{i,j}$ define graded completion in $[0, 1]$ and are aggregated by Equation 3. The binary strict criterion, relaxed proxy, and continuous map are distinct evaluator outputs. The open-loop rules appear in Appendix E.2. Within each protocol, all language metrics use the same scheduled pair set, and unsuccessful executions remain in the denominator.

Task progress is a gate in the strict-success criterion, distinct from continuous reference advancement. CARLA's default progress term requires evaluator progress of at least 0.05. RealSim checks a 10 m threshold over the available rollout, including its pre-instruction segment. CARLA safety excludes normalized collision, off-road, and red-light events. RealSim scoring v0.2 checks collision and off-road events; requested speed remains a language criterion. These components are distinct from continuous reference advancement.

C.6 REALSIM SCENE MINING AND EXECUTION QUALIFICATION

Candidate records bind scene identity, recording window, initial ego pose, and four source cameras at 576×320. The construction records contain 120 scene-acquisition entries, 308 route opportunities, and 32 task-source lineage entries. Temporal screening retains 185 route opportunities: 94 lack sufficient preceding context and 29 lack execution time. Task authoring and review freeze 30 PCUs

and retire two lineage entries. These inventories have different units and do not form one nested count funnel.

Branch topology, adjacent lanes, and scoped speed requests provide the alternatives. Authors review both-arm rendered views, scope/trigger cues, and rendering quality, then freeze public inputs and private criteria. Neutral navigation uses `UNKNOWN`, with waypoint-command routing disabled. The final 30 PCUs span 23 scenes, six in A and 24 in B. Every frozen PCU passes both-arm qualification under three seeds: 180 privileged-driver traces satisfy their requested contract and reject the alternative.

Scope support is explicit. P01 requests 19.25–20.75 versus 21.25–22.75 km/h. P30 requests 20.25–21.75 versus 22.25–23.75 km/h in a 1.526 m interval with two active samples in its qualification example. Across 144 witnesses carrying scoped-sample diagnostics, support ranges from 2 to 26 samples. The verifier evaluates these active samples over complete traversal, not a whole-rollout average speed.

D MODEL INTERFACES AND EXECUTION SETTINGS

D.1 CARLA ADAPTERS

LMDrive, SimLingo, Qwen-Drive, OneVL, Impromptu-VLA, and Senna use the model families described by Shao et al. (2024); Renz et al. (2025); Zhou et al. (2026a); Lu et al. (2026); Chi et al. (2026); Jiang et al. (2026). Table 8 identifies the input and output paths used in the complete CARLA comparison. Native and generic trajectory adapters differ in sensor coverage, training domain, and controller. The public/private input split does not impose identical camera rigs. Public-language conversion into a model's required command or local goal is included in the tested system identity.

Table 8: Complete CARLA system paths. A bridge denotes an interface that differs from the original end-to-end architecture.

System	Effective input	Action path / important configuration
LMDrive	Four RGB views, LiDAR, ego state	Native visual encoder/LLM, waypoint head, PID; public-language local-goal adaptation
SimLingo	Front RGB, motion state	InternVL2-1B, epoch013; native waypoints/PID
Alpamayo 1.5	Custom multi-camera temporal history	Trajectory generation and tracking; multi-camera temporal input
Qwen-Drive-1.0-4B	Front/left/right RGB, ego	Backbone with planner-rl; sampled plans and trajectory tracking
OneVL	Single front view, ego/history	NAVSIM-format latent-prefix trajectory generation
Impromptu-VLA	Front/right/left RGB, ego history	Released 3B AD-finetuned QA-format waypoint generation
Senna VLM bridge	Six RGB views	SPEED/PATH output expanded deterministically into a trajectory; no Senna-E2E planner

The generic camera rig uses yaw angles $0/-60/+60/180/-120/+120$ degrees for front/left/right/rear/back-left/back-right, typically at 896×512 resolution and 100° field of view. These settings do not replace the native rigs: SimLingo uses a 1024×512 front image with 110° field of view, and LMDrive and Alpamayo have their own configurations. Generic control uses meter-valued ego-frame waypoints with declared timestamps, steering based on a pure-pursuit-type tracker, and speed control. Invalid, nonfinite, or out-of-range plans can trigger braking or an error; such outcomes remain part of the system behavior.

The generic Qwen-Drive, OneVL, Impromptu, Senna, and ReCogDrive bridge configurations record model calls every 20 CARLA frames. At 20 Hz this corresponds to a configured one-second call interval; it is not a measured wall-clock latency. Waypoint spacing is declared separately, commonly 0.5 s for these adapters. Alpamayo records a 1 s replanning interval, 16 ego-history steps at 0.1 s, and four camera frames. Controller launch or fallback behavior is part of an adapter's policy and should remain attached to its exact configuration.

D.2 REALSIM MODEL IDENTITIES

The RealSim main comparison includes SimLingo, Impromptu-VLA, ReCogDrive, OneVL, AutoVLA, Qwen-Drive, Alpamayo 1.5, and Alpamayo2-Super (Renz et al., 2025; Chi et al., 2026; Xiong et al., 2026; Lu et al., 2026; Zhou et al., 2026c;a; Wang et al., 2025b). The main roster has separate rows for Alpamayo 1.5 and Alpamayo2-Super.

RealSim's workload/qualification baseline has four logical views: `cross_left_120fov`, `cross_right_120fov`, `front_tele_30fov`, and `front_wide_120fov`, at 576×320. Model-specific view selection, resizing, or history construction is recorded separately. A reconstructed multi-view input is not automatically equivalent to a model's original training rig. Repeating the earliest observed frame to fill a short history uses no future ground truth, but remains a disclosed input variation.

ReCogDrive uses a NAVSIM-transfer path in RealSim, distinct from the CARLA VLM bridge. OpenDriveVLA's RealSim path loads the Qwen+UniAD visual artifacts and performs real multi-image forward passes, distinct from the text-only CARLA head. Senna's official row remains blocked because the reviewed complete low-level planner/checkpoint path is absent. Its independently executed meta-action-to-trajectory variant is not substituted into that official row. LMDrive's official row is blocked by input-equivalence requirements, particularly LiDAR; the synthetic-null-LiDAR smoke is a separate non-equivalent variant.

D.3 Environment and process topology

The recorded CARLA Python environment uses Python 3.11.15, PyTorch 2.2.0+cu121, CUDA 12.1, Transformers 4.49.0, NumPy 1.26.4, and CARLA Python package 0.9.16. Model-local vendor directories can override these dependencies. The Python package version alone does not identify the custom server binary or map build. The usual execution topology uses seven model/ simulator workers on seven GPUs; SimLingo uses up to 14 workers on seven GPUs. This is route-level parallelism, not tensor-parallel inference.

RealSim fixes AlpaSim commit `affc2eab209fa43bdfa2f26c0f8d437922d78a68`, PhysicalAI-NuRec version 26.01, and NRE image version 26.04 for evaluation. The main runtime uses Python 3.12 with isolated model-worker environments. Simulation services and model workers can reside on different hosts; runtime logs and attempt identities bind their outputs. Latency, GPU-hour, and energy comparisons are outside this evaluation.

Table 9: **Open-loop instruction following on three datasets.** Predicted trajectories are scored for exclusive binary semantic success (SemSISR/SemCFPS) and graded completion (SCS/PCS). All values use a 0–100 scale.

System	SemSISR	SemCFPS	SCS	PCS
WOD-E2E				
DriveMA-2B	**38.38**	**1.24**	49.33	7.88
Fast-dDrive	34.85	0.00	46.42	5.27
Qwen-Drive-SFT	34.44	0.00	**50.58**	**9.37**
Qwen-Drive-RL	35.06	0.00	49.58	7.58
NAVSIM				
Qwen-Drive-1.0-4B	0.000	0.000	5.472	0.000
AutoVLA	3.947	0.000	8.172	0.000
DriveMA-2B	**8.553**	0.000	**11.244**	**2.242**
ReCogDrive	0.000	0.000	4.400	0.000
OneVL	3.618	0.000	8.428	0.000
nuScenes				
Impromptu-VLA	15.83	0.00	20.21	0.00
UniDriveVLA	**16.67**	0.00	25.00	0.00
OpenDriveVLA-0.5B	13.33	**3.33**	19.06	4.30
OmniDrive	7.50	0.00	**29.24**	**8.61**

E Statistical interpretation and reproducibility

E.1 Evaluation units, provenance, and interpretation

A reproducible run binds the dataset and task-source versions, PCU, instruction arm, expression, random seed, checkpoint revision, input adapter, decoding rule, trajectory frame and timestamps, controller when applicable, private scorer, and selected attempt. The PCU, rather than an instruction string or a retry, is the paired experimental unit. Repeated expressions and both arms remain grouped when computing paired scores. Duplicate requests, retries, and integration smoke tests do not create additional scenes. We distinguish source frames, candidate pairs, qualified PCUs, model requests, and valid predictions throughout this appendix.

The main closed-loop comparisons use a fixed evaluation seed per track. The L2D values in Tables 1 and 2 are computed from the reported component metrics with high-precision intermediate arithmetic and rounding only for presentation; this aggregation does not alter any rollout verdict. RealSim trajectory-response diagnostics use the complete valid pairs reported in Table 30, while the main success metrics retain their scheduled denominators. CARLA, RealSim, and the open-loop tracks are reported separately because they instantiate different horizons, task mixtures, and scoring functions.

E.2 Open-loop versions, coverage, and common notation

Version resolution. The nuScenes evaluation uses `nuscenes-test-v1.0.0`. NAVSIM uses the 38-PCU `l2d-offline-navsim` 1.0.0 release identified by the September 24 experiment bundle, with 19 recording clusters and seed 42. WOD-E2E uses the frozen P0 kinematic-proxy protocol (v0.1.0); all WOD results in this appendix are reported with the corresponding proxy metric names.

The source-specific roster includes DriveMA-2B, Fast-dDrive, and UniDriveVLA (Zheng et al., 2026; Zhang et al., 2026; Li et al., 2026), the model families in Appendix D, and OmniDrive (Wang et al., 2025a). Fast-dDrive is evaluated on WOD-E2E; AutoVLA (Zhou et al., 2026c) is evaluated in the separate NAVSIM experiment. The nuScenes comparison reports the models for which the corresponding frozen adapter outputs are available.

Coordinates and resampling. Positions use meters and an anchor-fixed local plane with $+x$ forward and $+y$ left. Velocities and accelerations use m/s and m/s^2. For an anchor with global planar

Table 10: Open-loop coverage of the constructed Language2Drive subsets. Candidate, qualified-PCU, and request counts are reported separately.

Source	Available source / construction	Evaluation subset	Queries per model
nuScenes	150 test scenes; 6,008 anchors; 83 candidates	75 approved PCUs in 75 scenes / 15 logs; dev 15, heldout 60	900 across 75 PCUs; 720 heldout, including controls
NAVSIM	OpenScene test; 40 reviewed candidates	38 released PCUs / 19 original recordings	304: two arms $\times$ four wordings $\times$ 38 PCUs
WOD-E2E	38/93 validation shards; 43,394 frames; 203 rated anchors	563 complete-window candidates; 441 with P0 drafts; 241 motion-proxy pairs	882 draft requests; 482 belong to the proxy subset

Table 11: Prediction grids and dataset-specific anchor reference points. Future points exclude $t = 0$.

Source	Horizon	Interval	Points	Output and fixed anchor reference
nuScenes	3.0 s	0.25 s	12	(x, y); `ego_pose_origin`
NAVSIM	4.0 s	0.50 s	8	(x, y, ψ); rear-axle origin
WOD-E2E	5.0 s	0.25 s	20	(x, y); rear-axle midpoint

position (X_0, Y_0) and yaw ψ_0, the planar conversion is

$$\begin{bmatrix} X_k \\ Y_k \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \end{bmatrix} + \begin{bmatrix} \cos\psi_0 & -\sin\psi_0 \\ \sin\psi_0 & \cos\psi_0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix}. \tag{33}$$

The reference-point definition is part of the protocol and is matched by every adapter. NAVSIM uses an SE(2) transform based on (X, Y, ψ) and verifies that the current lane covers the local origin. Native trajectories are converted to the source-specific evaluation grids before scoring; private targets are never used to select among multiple model outputs.

Binary and graded paired scores. For PCU i, wording r, and arm $a \in \{0, 1\}$, let $\tau_{ir,a}$ be the submitted trajectory, $V(\tau)$ its format-validity indicator, and $E_{i,a}(\tau)$ its native binary task predicate. Define

$$B_{ir,a} = V(\tau_{ir,a})E_{i,a}(\tau_{ir,a})\big[1 - E_{i,1-a}(\tau_{ir,a})\big], \qquad c_{ir,a} = V(\tau_{ir,a})C_{i,a}(\tau_{ir,a}). \tag{34}$$

Here $C_{i,a} \in [0, 1]$ is the *source-specific* graded completion function. For a declared wording set $\mathcal{R}_i$, write $\langle z \rangle = N^{-1} \sum_i |\mathcal{R}_i|^{-1} \sum_{r \in \mathcal{R}_i} z_{ir}$. The four reported aggregates are

$$\begin{aligned} \mathrm{SemSISR} &= 100 \left\langle \frac{B_{ir,0} + B_{ir,1}}{2} \right\rangle, & \mathrm{SemCFPS} &= 100 \langle B_{ir,0} B_{ir,1} \rangle, \\ \mathrm{SCS} &= 100 \left\langle \frac{c_{ir,0} + c_{ir,1}}{2} \right\rangle, & \mathrm{PCS} &= 100 \langle \min(c_{ir,0}, c_{ir,1}) \rangle. \end{aligned} \tag{35}$$

Missing or invalid predictions receive zero binary and completion credit without shrinking the fixed eligible denominator. A source may reject an incomplete result file before scoring, as specified below. Candidate-qualification failures are not model failures and are excluded before fixing that denominator.

The open-loop reports use the same paired aggregation while preserving each dataset's native primitive scorer: nuScenes uses temporal coverage with a displacement gate, NAVSIM uses tolerance-based task completion with minimum conjunction, and WOD-E2E uses the kinematic proxy functions defined below. WOD values are denoted *ProxySemSISR*, *ProxySemCFPS*, *ProxySCS*, and *ProxyPCS*.

E.3 nuScenes: construction, qualification, and exact scoring

Source inventory and split. Construction uses local nuScenes `v1.0-test`, not `v1.0-trainval`. The catalog contains 150 scenes and 6,008 anchors: 3,318 in Boston,

796 in Singapore-onenorth, and 1,894 in Singapore-queenstown. At most one PCU is retained per scene. Logs are grouped by location, sorted by the SHA-256 of their log token, and the first $\max(1, \mathrm{round}(0.25 n_{\log}))$ logs per location are assigned to development, using Python's `round`. The split is then intersected with the approved set. This yields 15 dev PCUs in four logs and 60 heldout PCUs in 11 logs, with neither scene nor log overlap. The resulting split contains 15 dev and 60 heldout PCUs.

Table 12: Released nuScenes task and geographic distributions. The paired lane alternatives are change/keep, not left/right within the same PCU.

Category	Released	Dev	Heldout
Lane change / keep	22	5	17
Left-change target	12	3	9
Right-change target	10	2	8
Scoped speed: approximately 15 / 30 km/h	53	10	43
Boston-seaport	41	7	34
Singapore-onenorth	12	4	8
Singapore-queenstown	22	4	18
Total PCUs	75	15	60

The lane/speed proportions are 29.33/70.67% in the release and 28.33/71.67% in heldout. The released nuScenes track focuses on lane-choice and scoped-speed PCUs.

Public observations and coordinate metadata. Public inputs reference six cameras and the `LIDAR_TOP` keyframe, retaining sensor timestamps and calibration tokens, plus up to four historical keyframes and ego poses. Historical-frame counts of 4/3/2/1 occur in 67/3/1/4 PCUs, with observation offsets spanning 0.400365–2.100166 s before the anchor. The output declares `coordinate_frame=ego_at_anchor` and `reference_point=ego_pose_origin`. The anchor transform uses the translation and yaw of the ego pose attached to `LIDAR_TOP`; quaternion storage is (w, x, y, z).

The four evaluated adapters first produce six future positions spaced by 0.5 s. The known anchor $(0, 0)$ is prepended and piecewise-linear interpolation yields the 12-point grid in Table 11. The adapters resample the submitted future trajectory to the fixed 3 s grid; the Impromptu adapter additionally constructs its required observed-history input to -3 s.

Candidate mining and map binding. Initial speed is estimated from the current and preceding ego poses using their actual time separation, which must lie in $[0.4, 0.6]$ s. Candidates bind to a `CAR` lane queried within 2 m, with lane-direction error at most 30° and polygon distance strictly below 1 m. The local tangent uses points 0.5 m before and after the projected arc length. Lane paths are extended toward 80 m through connections with endpoint gap at most 3 m and turn angle strictly below 35°; the next connection minimizes $\mathrm{turn}_{\mathrm{rad}} + 0.02\,\mathrm{gap}_{\mathrm{m}}$. A disconnected path can end earlier, but each witness must remain at least 0.5 m before its endpoint. The path representation is the polyline through arcline-segment endpoints, not a densely sampled exact curve. Lane candidates are built first; speed candidates skip occupied scenes. The first qualified anchor in catalog order is retained per scene. The lane-builder limit of 80 was not reached: 22 lane candidates were found.

Counterfactual witnesses. Both arms share the observation, ego history, and public-input hash; the model receives `predict(public_input, instruction, seed)`, without arm labels, target lane tokens, contracts, witnesses, or review decisions. Targets are map-bound sets of acceptable behavior, not two fitted copies of the single recorded future. For lane change, the current-lane and target-lane paths are mixed with $u = \mathrm{clip}((t - 0.4)/2, 0, 1)$ and $w = u^2(3 - 2u)$. Full mixing occurs at 2.4 s, while the contract requires entry into the target corridor from 2.25 s; entry does not require reaching its centerline. The keep witness follows the current-lane path. For speed tasks, both witnesses follow the same extended path, with arc-length schedule

$$s(t) = \begin{cases} v_0 t + \frac{1}{2}(v_* - v_0)t^2, & 0 \le t \le 1, \\ \frac{1}{2}(v_0 + v_*) + v_*(t - 1), & t > 1, \end{cases} \qquad v_* \in \{4, 8\}\ \mathrm{m/s}. \tag{36}$$

Table 13: nuScenes target-construction parameters. These mining and witness parameters are distinct from the final scoring tolerances.

Parameter	Value or exact rule
Lane-task initial speed	$4.5 \le v_0 \le 9.0$ m/s; released range 4.745680–8.878580
Adjacent-lane query	7 m radius; polygon gap ≤ 0.8 m
Divider / direction	Divider exists and every segment type contains `DASHED`; target yaw error $\le 20^\circ$
Relative target position	Lateral distance 2.4–5.0 m on the requested side; absolute forward offset ≤ 5.0 m
Candidate ranking	$\lvert d_{\text{lat}} - 3.5\rvert + 0.1\lvert d_{\text{fwd}}\rvert$; search left before right
Lane witness blend	Start 0.4 s; duration 2.0 s; initial centerline offset decays over 1.0 s
Speed-task initial speed	$5.5 \le v_0 \le 6.5$ m/s; released range 5.521365–6.491924
Speed witness targets	4.0 and 8.0 m/s, reached after a 1.0 s transition

These witnesses support local feasibility during construction; they are not unique reference trajectories that a prediction must imitate.

Qualification and human review. Construction produces 83 candidates (22 lane and 61 speed). Automated witness, map, motion, and swept-footprint checks retain 82 candidates, after which manual review freezes 75 PCUs (22 lane and 53 speed). Both arms must have feasible witnesses, satisfy their requested predicate, reject the paired alternative, and pass the shared geometric and motion checks.

The swept-footprint screen uses a 5.0×2.0 m rectangular vehicle approximation under two reference-point hypotheses and checks coverage against buffered drivable polygons. LiDAR-assisted review additionally summarizes points near each witness path over the 1–3 s interval. The review interface presents six cameras, the local map, both instructions and witnesses, and the map/LiDAR diagnostics. Final approval requires `observable`, `arm_0_feasible`, `arm_1_feasible`, `mutually_exclusive`, `traffic_clearance`, and `map_binding` to be marked `yes`; the decision, reviewer identifier, rationale, timestamp, and source hash are recorded in the frozen review table.

Output validity and binary contracts. Both `timestamps_s` and `xy_m` must contain 12 entries, with two finite coordinates per point and the exact frame/reference strings above. Timestamp error must be at most 10^{-5} s. Missing points, NaN, or infinity are invalid, and the scorer does not fill missing future points. For $p_0 = (0, 0)$ and $\Delta t = 0.25$ s, define

$$v_k = \|p_k - p_{k-1}\|/\Delta t, \qquad D = \|p_{12}\|. \tag{37}$$

Every arm predicate in Table 14 additionally requires $D \ge 8$ m.

Table 14: Complete nuScenes binary semantic conditions, in addition to $D \ge 8$ m. Every required sample must satisfy its condition.

Arm	Predicate	Required samples
Lane-change arm 0	Reference point lies in the target lane corridor	$t = 2.25, 2.50, 2.75, 3.00$ s; 4 points
Lane-keep arm 1	Reference point lies in the current lane corridor	$t = 0.25, 0.50, \ldots, 3.00$ s; 12 points
Low-speed arm 0	$3.2 \le v_k \le 4.8$ m/s	Interval ends $t_k = 1.25, \ldots, 3.00$ s; 8 intervals
High-speed arm 1	$7.2 \le v_k \le 8.8$ m/s	Same 8 intervals

Each corridor comprises the available `CAR` lane polygons in its private token set, individually buffered by 0.15 m. Boundary points count as inside through `covers`. The speed window covers 1.00–3.00 s, with the first interval spanning 1.00–1.25 s. Binary scoring tests the requested predicate and the paired-alternative predicate as in Eq. (34).

Separate motion and map-quality checks. Motion and map-quality diagnostics are reported alongside semantic success. The thresholds are maximum speed $\leq$ 17 m/s, $|v_1 - v_{\text{initial}}| \leq 3$ m/s, maximum absolute speed-difference acceleration ≤ 5 m/s^2, and maximum absolute wrapped heading-change rate ≤ 1.0 rad/s. Heading differences use $\text{atan2}(\sin\Delta\psi, \cos\Delta\psi)/0.25$. Drivable-center requires all 12 reference points to lie in, or within 0.2 m of, the drivable region. Both rates use all scheduled arms, with invalid outputs receiving zero credit.

Partial completion and response metrics. The supplementary unified semantic metrics use the arm's required index set A_a in Table 14, let $I_a(k,\tau)$ denote its lane-membership or speed-band test. Then

$$C_a(\tau) = \min(1, D/8)\frac{1}{|A_a|}\sum_{k\in A_a} I_a(k,\tau). \tag{38}$$

Invalid outputs have $C_a = 0$. There is no opposite-target penalty or motion/map multiplier. Thus SCS/PCS award partial temporal coverage; spatial membership itself remains binary.

SRG uses the common window: the four terminal lane points or the eight speed intervals. At each point, d_a is distance to the corresponding corridor in meters, or $\max(\ell_a - v, 0, v - h_a)$ for a speed band $[\ell_a, h_a]$ in m/s. Define

$$p_0(k,\tau) = \begin{cases}(d_1 - d_0)/(d_0 + d_1), & d_0 + d_1 > 0,\\ 0, & d_0 + d_1 = 0,\end{cases} \quad \text{SRG}_i = \frac{\overline{p}_0(\tau_{i,0}) - \overline{p}_0(\tau_{i,1})}{2}. \tag{39}$$

The ratio is computed before time averaging; no epsilon, temperature, clipping, or progress gate is added. SRG lies in $[-1, 1]$ and measures directional response, not absolute completion. NBD instead uses all 12 points:

$$\text{NBD}_i = \frac{\sum_{k=1}^{12}\|p_{i,0,k} - p_{i,1,k}\|}{\sum_{k=1}^{12}(\|p_{i,0,k}\| + \|p_{i,1,k}\|)}. \tag{40}$$

A zero denominator gives zero without an epsilon. Each PCU's ratio is computed before averaging. Displayed SRG and NBD are multiplied by 100, and paired ADE in meters is retained as an absolute-scale companion to NBD.

SCS/PCS retain every scheduled PCU with invalid completion zero. SRG, NBD, and paired ADE average only valid pairs, record invalid values as null, and report coverage; an empty valid set is undefined, not zero. On canonical heldout predictions, OmniDrive has 116/120 valid arms and 56/60 valid pairs, while the other three systems have 120/120 and 60/60. PCU-weighted averages use the 17/60 and 43/60 lane/speed weights. A separately labeled family-macro average assigns each family weight $1/2$; valid-pair filtering is applied within each family for response metrics. The stored `SRG_coverage` field is a separate coverage-based diagnostic; Eq. (39) defines the distance-based SRG used here.

The language targets of 15/30 km/h differ slightly from the witness speeds of 14.4/28.8 km/h and the scoring-band centers. Target MAE uses the language targets; binary compliance uses the frozen task-specific bands. Semantic margins summarize the active speed-band or lane-boundary constraint.

Language conditions and inference seed. Each arm has one canonical instruction and three fixed English paraphrases. Together they form 600 sample-indexed text instances over the 75-PCU release; templates can repeat across scenes. Masked and shuffled controls yield six conditions in total and 900 outputs per model (720 heldout), all with seed 0. The two arms already provide the opposite-instruction comparison; there is no additional seventh condition. Masked passes an empty string to the runner, which an adapter may render as a neutral prompt. Shuffled uses the opposite-arm text from the next PCU in the same family after sorting by PCU ID; unchanged text falls back to the current PCU's opposite arm. The executed donor pool is all 75 PCUs, followed by heldout filtering. Masked/shuffled scores are evaluated against the original arm target and serve as controlled language ablations.

The formal scoring wrapper rejects files missing requested conditions, seeds, or arms; the underlying scorer assigns zero to a missing arm within an already present condition. `Paraphrase_Worst_PCU` requires both arms to succeed under all three paraphrases within the same PCU, rather than taking the minimum of three aggregate CFPS values. Masked and shuffled gaps subtract the corresponding control CFPS from canonical CFPS.

Table 15: Additional nuScenes diagnostics and their full numeric conventions. Conditional means always accompany validity coverage.

Diagnostic	Exact definition / threshold	Invalid handling
Speed-target MAE	Mean $\lvert 3.6v - v_* \rvert$ on 8 intervals, $v_* = 15$ or 30 km/h	Valid-arm mean
Speed-band violation	$3.6\,\mathrm{mean}\,\max(\ell - v, 0, v - h)$, km/h	Valid-arm mean
Speed coverage	In-band fraction over 8 intervals, without displacement gate	Zero for invalid arms
Paired speed change	$\Delta v = 3.6[\mathrm{mean}(v_{\mathrm{high}}) - \mathrm{mean}(v_{\mathrm{low}})]$	Valid-pair mean
Lane distance / coverage	Mean distance to own corridor; inside fraction over 4 change or 12 keep points	Distance conditional; coverage zero if invalid
Lane preference	$q = \mathrm{mean}(d_{\mathrm{current}} - d_{\mathrm{target}})$ on the common window; $\Delta q = q_{\mathrm{change}} - q_{\mathrm{keep}}$	Valid arm / pair
Directional response	Speed: $\Delta v > 1.0$ km/h; lane: $\Delta q > 0.2$ m; no-dead-zone version uses $> 10^{-6}$	All family PCUs; invalid fails
Semantic margins	Speed: $3.6\,\min_k \min(v_k - \ell, h - v_k)$; lane: minimum signed boundary distance, inside positive	Valid-arm mean
Opposite-arm ADE	Mean pointwise separation over 12 points	Valid pairs
Paraphrase ADE	Canonical versus each same-arm rewrite; 360 planned heldout comparisons	Valid comparisons
Masked / shuffled ADE	Canonical versus same-arm control; 120 comparisons per heldout condition	Valid comparisons

E.4 NAVSIM: RELEASED PORTFOLIO AND FROZEN EVALUATION

Release size and task composition. The released NAVSIM/OpenScene test portfolio contains 38 PCUs from 19 original recordings, after reviewing 40 candidates and rejecting two. Each PCU has two arms and four approved English wordings per arm: canonical and three paraphrases. The evaluation therefore contains 304 queries per model and 1,520 queries across five models. All 1,520 outputs were parsed into the declared trajectory schema. Recordings define the coarsest reported cluster structure, and multiple PCUs may share a recording.

Table 16: NAVSIM 1.0.0 task composition and documented counterfactual targets. Counts and percentages use the 38-PCU release, not the earlier 12-candidate pool.

Family	Scope	PCUs	Share	Counterfactual alternatives
Lane choice	A	12	31.58%	Keep versus change one lane; eight left-target and four right-target PCUs
Junction branch	A	1	2.63%	Left turn versus straight at a common junction
Pass / follow	A	1	2.63%	Follow the same actor versus pass using an adjacent lane
Speed band	B	7	18.42%	Two disjoint speed bands over the declared interval
Object-relative speed	B	2	5.26%	Two speed requirements while following the same actor
Landmark stop	C	10	26.32%	Distinct stopping locations grounded in visible landmarks
Local goal	C	1	2.63%	Adjacent left-lane versus right-lane target
Event order	D	4	10.53%	Slow then change lane versus change lane then slow
Total	A–D	38	100.00%	Scope counts A/B/C/D: 14/9/11/4

Counterfactual construction and qualification. Candidates are mined using observed ego state, camera/LiDAR history, map geometry, and logged actors. Each arm receives a private executable contract and feasibility witness. Lane and local-goal tasks bind different lane regions; branch tasks use a junction goal region; landmark stopping binds a visible named or described landmark to a stopping region. Actor-conditioned tasks bind the same actor across both arms. Order tasks retain the two events while exchanging their required order. These contracts define acceptable behavior sets, rather than forcing the model to match the witness path point by point.

Release gates check sensor availability, source-split isolation, public/private PCU ID agreement, private-label leakage, own/opposite contract behavior, witness replay, map footprint, logged actors, and collision screening on a 0.1 s grid. Human review records a reviewer, pass/reject decision, and rationale. All 38 released PCUs pass the frozen release checks, and oracle replay gives OL-SemSISR/OL-SemCFPS of 1.0 as a construction consistency test.

The public package contains observation references and English language, while source tokens, future trajectories, map targets, witnesses, and contracts remain private. Camera/LiDAR paths are relative to the OpenScene v1.1 test sensor root. The release manifest identifies three deterministic archives: public inputs, private evaluation assets, and evaluator code. Repeated public/private/evaluator builds are reported as byte-identical.

Common output, parsing, and quality. Predictions contain eight rear-axle poses in the anchor ego frame at $0.5, 1.0, \ldots, 4.0$ s. Model headings are retained when supplied; otherwise headings are computed from successive positions. Parse failures remain invalid and receive zero completion. The released motion-quality summary uses maximum absolute acceleration ≤ 3.5 m/s^2 and maximum yaw rate ≤ 0.8 rad/s. These quality checks are reported separately from the semantic contracts.

Binary and graded semantics. The released binary result uses own-contract success and opposite-contract failure as in Eq. (34). The graded function is implemented by `task_completion` in `verify.py`. A trajectory satisfying the native binary contract has completion 1; outside the accepted set, primitive scores decay linearly over one native tolerance or accepted interval width. Conjunctions take the minimum. Lane completion combines minimum vehicle-footprint overlap with terminal progress. Stop completion combines the target-region requirement and stopping speed. Event-order completion is 0 if neither event occurs, 0.5 if exactly one occurs, and otherwise is graded by the event gap using the frozen verifier. The evaluator records the full 2×2 matrix $C_a(\tau_b)$, but SCS and PCS use only the own-arm entries; opposite rejection is a binary requirement, not a universal multiplier on graded completion.

Table 17: NAVSIM task-specific scoring components. Global rules are shown here; per-PCU geometry and thresholds are read from the frozen private sidecars.

Component	Frozen scoring rule / parameter source
Common grid / quality	8 poses at 0.5–4.0 s in the rear-axle frame; maximum acceleration 3.5 m/s^2 and maximum yaw rate 0.8 rad/s
Lane / local goal	Vehicle-footprint overlap, terminal progress, and target polygons from the per-PCU private contract
Junction goal	Goal region, active interval, and deadline stored in the per-PCU contract
Landmark stop	Landmark-grounded stop region, stopping threshold, and hold requirement stored in the private contract
Speed band	Per-PCU lower/upper speed bands and active time range; primitive completion decays linearly over the native tolerance
Object-relative speed	Speed-band requirement combined with the bound lead-actor geometry and relative-position constraints
Pass / follow	Shared actor binding with lane/gap/overtaking constraints and task-specific active windows
Event order	Ordered event predicates; 0 if neither occurs, 0.5 if one occurs, otherwise event-gap grading
Collision audit	Logged-actor/ego-footprint overlap screening on the frozen 0.1 s audit grid

Exact per-PCU polygons, actor bindings, event thresholds, speed bands, and grading tolerances are loaded from the frozen private sidecars by the released evaluator; Table 17 summarizes how they enter scoring.

Aggregation and reporting. All headline metrics average four wordings equally within each PCU, followed by an equal average across 38 PCUs. Families are not equally weighted. The paired imbalance is checked against the algebraic identity

$$\mathrm{SCS} - \mathrm{PCS} = 50 \left\langle |c_{ir,0} - c_{ir,1}| \right\rangle . \tag{41}$$

The recorded maximum absolute residual across models is 1.07×10^{-14}. Score JSON top-level binary fields are canonical-only and lie in $[0, 1]$; `equal_weight_rewrites` contains the four-wording headline scores on the 0–100 scale. These fields are not interchangeable. A separate 3 s common-horizon diagnostic covers 22 PCUs from 13 recording clusters and is reported independently from the full 4 s, 38-PCU result.

Model adapters and randomness. The five models are Qwen-Drive-1.0-4B, AutoVLA, DriveMA-2B, ReCogDrive, and OneVL. Seed 42 is reset for every query. Deterministic generation is used where the released interface supports it; sampling-based interfaces retain their released sampling procedure. Required non-language command tensors in Qwen-Drive and ReCogDrive are fixed to neutral/straight for all PCUs, arms, and wordings. Full language is the only candidate-dependent input, although sensor coverage and history representation differ across models.

Table 18: Frozen NAVSIM adapters. Input differences are retained and disclosed rather than hidden by the common output grid.

Model	Released inputs and instruction placement	Common-grid conversion
Qwen-Drive	Four frames of front/front-left/front-right; 16-point interpolation of observed 1.5 s history; full instruction in scene prompt	One reasoning-mode RL-planner sample; 10 Hz future points numbered 5,10,. . .,40
AutoVLA	Four frames of three forward cameras; current velocity/acceleration; driving-instruction field	First eight released 0.5 s action-token poses
DriveMA-2B	Current three forward cameras; observed 1.5 s history and state; high-level-intent field	Released two-turn decision/trajectory protocol; 1 Hz positions linearly resampled to 0.5 s
ReCogDrive	Current front image; four observed poses and state; text replaces discrete command phrase	Released eight-pose DDIM planner output
OneVL	Current front image; three historical poses and state; full Command field	Released latent-chain-of-thought generation of eight 0.5 s poses

Implementation identity. The supplied protocol records PyTorch 2.5.1+cu124 on NVIDIA A100 40 GB GPUs; Transformers 5.14.1 for Qwen-Drive, 4.57.0 for OneVL, and a released-compatible 4.49 development build for legacy adapters. These are the recorded experiment versions. Table 19 records the frozen code and primary checkpoint revisions. Additional Qwen2.5-VL and ReCogDrive VLM component revisions remain in `EXPERIMENT_PROTOCOL.md`.

Table 19: NAVSIM model revision identifiers from the frozen experiment protocol. Line breaks inside identifiers are typographic only.

Model	Repository commit	Primary checkpoint revision
Qwen-Drive	`28091c1532e869bc7aee91fc 0aef6b3e6fd0b2e0`	`28484089a7cc8c335cf5089f b0745cf7c49b6eaa`
AutoVLA	`ba34eed74ce6729e7986592d 0e66cbaca397b4fa`	`a7d7ba3ed7529b248d2694c2 defa31b35208340f`
DriveMA-2B	`de20e6c64878bc3413d26d69 362c3be9b3a91533`	`d5f4905d2032da943a7ae8f2 2d5b2b8efbe4b0f7`
ReCogDrive	`6b8d8f5e01346c71094651c8 1dcaf66405dbc04e`	`7346ede73f03ed577811024f 1a5d16fb063a47b9`
OneVL	`f4e6f05572b2f66611259bcd 5563529833f30469`	`c1e893e9a03b8312f9328d2b 51267836a9f7bf8a`

E.5 WOD-E2E: CONSTRUCTED TASKS AND SEMANTIC TRAJECTORY EVALUATION

Available data and evaluation denominators. The WOD-E2E construction scans 38 of the 93 numbered validation TFRecord shards available locally. The resulting source contains 43,394

frames, deduplicated by frame name and timestamp, and 203 rated anchors/sequences, each associated with three candidate future trajectories. Pairing the available trajectories produces 609 raw candidate pairs. After requiring a complete 5 s prediction window, 563 candidate pairs remain. Among them, 441 pairs across 189 anchors admit two distinct English instructions, resulting in 882 model requests.

The final semantic-evaluation set contains 241 PCUs and 482 requests across 126 anchors. Each PCU contains two mutually exclusive instructions defined over the same recorded driving context. The evaluated set contains 165 `SPEED_CHOICE` PCUs and 76 `STOP_GO` PCUs.

Table 20: WOD-E2E P0 task-family distribution. Candidate construction, language-pair construction, and the final semantic-evaluation set are reported with separate denominators.

Task family	Pairs	% of 563	% of 441	Requests	Evaluated PCUs
`SPEED_CHOICE`	175	31.08	39.68	350	165
`STOP_GO`	112	19.89	25.40	224	76
`LATERAL_CHOICE`	112	19.89	25.40	224	0
`TURN_WAIT`	28	4.97	6.35	56	0
`TURN_STRAIGHT`	14	2.49	3.17	28	0
`UNRESOLVED`	122	21.67	–	0	0
Total	563	100.00	100.00	882	241

Prediction horizon and coordinate convention. Each prediction is evaluated on the fixed grid

$$t_k = 0.25k \text{ s}, \qquad k = 1, \ldots, 20, \tag{42}$$

covering the future interval $(0, 5]$ s. The implicit anchor $p_0 = (0, 0)$ represents the current ego position and is not included among the submitted future points. Positions are expressed in an anchor-fixed right-handed ego frame whose origin is the rear-axle midpoint, with $+x$ forward, $+y$ left, and $+z$ upward. The semantic evaluator operates on the planar trajectory $p_k = (x_k, y_k)$.

All reference and prediction trajectories are normalized to this common 20-point representation before scoring. DriveMA-2B and Fast-dDrive produce five future positions at 1, 2, 3, 4, and 5 s; the known current position is prepended and linear interpolation produces the 4 Hz evaluation sequence. Qwen-Drive predicts a denser trajectory over the same 5 s horizon, which is resampled onto the same 0.25 s evaluation grid. No private task target is used to select among multiple predictions.

Observed context and model inputs. The recorded ego-state history contains 16 samples at 4 Hz over the preceding 4 s window. The source data provide eight camera views, while the evaluated adapters use the forward-facing views required by their respective model interfaces. Temporal observations, ego history, calibration, and high-level driving context are fixed within each PCU.

For the two arms of a PCU, the model receives the same scene identifier, visual observations, state history, calibration, and non-language context. Only the natural-language instruction changes. All task-specific semantic contracts and scoring parameters remain private to the evaluator.

Counterfactual task qualification. The reported WOD-E2E evaluation focuses on the two task families for which the future trajectory directly exposes a complete and mutually exclusive behavioral distinction: `STOP_GO` and `SPEED_CHOICE`.

Each PCU contains two frozen semantic contracts, P_0 and P_1. Qualification requires that both target behaviors form valid opposed alternatives under the same recorded scene context. In particular, each reference arm must satisfy its own semantic contract while rejecting the paired alternative:

$$P_a(\tau_a^{\text{ref}}) = 1, \qquad P_{1-a}(\tau_a^{\text{ref}}) = 0, \qquad a \in \{0, 1\}. \tag{43}$$

Applying the frozen qualification procedure yields 76 `STOP_GO` PCUs and 165 `SPEED_CHOICE` PCUs, for a fixed evaluation denominator of 241 PCUs and 482 instruction-conditioned predictions.

Trajectory features and valid-output gate. For the submitted future trajectory $\tau = (p_1, \dots, p_{20})$ and the implicit current position $p_0 = (0, 0)$, we define

$$\begin{aligned} v_k &= \frac{\|p_k - p_{k-1}\|_2}{0.25}, \\ L &= 0.25 \sum_{k=1}^{20} v_k, \\ v_{\text{tail}} &= \frac{1}{4} \sum_{k=17}^{20} v_k, \\ v_{\text{late}} &= \frac{1}{9} \sum_{k=12}^{20} v_k. \end{aligned} \tag{44}$$

Here v_{tail} summarizes the final second of prediction, while v_{late} summarizes the later portion of the 5 s horizon used by the speed-choice contracts.

A valid submission must contain exactly 20 future positions, with exactly two finite numeric coordinates per point. We additionally require

$$\max_k v_k \le 50 \text{ m/s}, \qquad \max_{k=2,\dots,20} \frac{|v_k - v_{k-1}|}{0.25} \le 20 \text{ m/s}^2. \tag{45}$$

Missing or invalid predictions receive zero binary and graded completion credit and remain in the fixed 241-PCU denominator.

Binary semantic criteria and graded completion. For $a < b$, define the clipped linear ramps

$$\begin{aligned} U(z; a, b) &= \min\left\{1, \max\left\{0, \frac{z - a}{b - a}\right\}\right\}, \\ D_{\downarrow}(z; a, b) &= 1 - U(z; a, b). \end{aligned} \tag{46}$$

For each `STOP_GO` PCU, let q_{stop} denote the frozen stopping target and

$$d_{\text{stop}} = \|p_{20} - q_{\text{stop}}\|_2. \tag{47}$$

The binary semantic predicates are

$$\begin{aligned} P_{\text{stop}}(\tau) &= [v_{\text{tail}} \le 0.6] \wedge [d_{\text{stop}} \le 2], \\ P_{\text{go}}(\tau) &= [v_{\text{tail}} \ge 2] \wedge [x_{20} \ge 8], \end{aligned} \tag{48}$$

and the corresponding graded completion scores are

$$\begin{aligned} C_{\text{stop}}(\tau) &= \min\{D_{\downarrow}(v_{\text{tail}}; 0.6, 2), D_{\downarrow}(d_{\text{stop}}; 2, 5)\}, \\ C_{\text{go}}(\tau) &= \min\{U(v_{\text{tail}}; 0.6, 2), U(x_{20}; 0, 8)\}. \end{aligned} \tag{49}$$

For each `SPEED_CHOICE` PCU, the private evaluator stores two pair-specific speed boundaries

$$\ell_i < h_i, \tag{50}$$

which define the two mutually exclusive requested speed behaviors for PCU i. The binary predicates are

$$\begin{aligned} P_{\text{slow}}(\tau) &= [v_{\text{late}} \le \ell_i] \wedge [L \ge 2], \\ P_{\text{normal}}(\tau) &= [v_{\text{late}} \ge h_i] \wedge [L \ge 2], \end{aligned} \tag{51}$$

with graded completion

$$\begin{aligned} C_{\text{slow}}(\tau) &= \min\{D_{\downarrow}(v_{\text{late}}; \ell_i, h_i), U(L; 0, 2)\}, \\ C_{\text{normal}}(\tau) &= \min\{U(v_{\text{late}}; \ell_i, h_i), U(L; 0, 2)\}. \end{aligned} \tag{52}$$

Table 21: WOD-E2E semantic scoring parameters. Speed and distance are reported in m/s and m, respectively. The speed-choice boundaries ℓ_i and h_i are frozen per-PCU evaluator parameters.

Criterion	Binary success	Completion 1	Completion 0
Stop tail speed	$v_{\text{tail}} \leq 0.6$	≤ 0.6	≥ 2
Stop endpoint distance	$d_{\text{stop}} \leq 2$	≤ 2	≥ 5
Go tail speed	$v_{\text{tail}} \geq 2$	≥ 2	≤ 0.6
Go forward endpoint	$x_{20} \geq 8$	≥ 8	≤ 0
Slow late speed	$v_{\text{late}} \leq \ell_i$	$\leq \ell_i$	$\geq h_i$
Normal late speed	$v_{\text{late}} \geq h_i$	$\geq h_i$	$\leq \ell_i$
Speed-task progress	$L \geq 2$	≥ 2	≤ 0

Semantic success and paired aggregation. For instruction arm a, binary semantic success requires satisfying the requested criterion while rejecting the paired alternative:

$$B_a = P_a(\tau_a) \wedge \neg P_{1-a}(\tau_a). \tag{53}$$

The corresponding continuous value is the own-arm completion $C_a(\tau_a) \in [0, 1]$.

Using the common open-loop aggregation defined in Section 2.4, WOD-E2E reports

$$\begin{aligned} \text{SemSISR} &= 100 \left\langle \frac{B_0 + B_1}{2} \right\rangle, \\ \text{SemCFPS} &= 100 \left\langle B_0 B_1 \right\rangle, \\ \text{SCS} &= 100 \left\langle \frac{C_0 + C_1}{2} \right\rangle, \\ \text{PCS} &= 100 \left\langle \min(C_0, C_1) \right\rangle. \end{aligned} \tag{54}$$

All 241 qualified PCUs receive equal weight. Missing or invalid model outputs contribute zero rather than reducing the denominator. This definition is identical in form to the semantic paired evaluation used by the other Language2Drive open-loop tracks, while retaining WOD-E2E-specific task predicates.

Geometric response diagnostic. We additionally report a trajectory-alignment diagnostic over the 441 instruction-paired samples. For prediction τ and reference R, define

$$\text{ADE}(\tau, R) = \frac{1}{20} \sum_{k=1}^{20} \|p_k - R_k\|_2. \tag{55}$$

For each arm,

$$m = \text{ADE}(\tau, R_{\text{other}}) - \text{ADE}(\tau, R_{\text{own}}). \tag{56}$$

An arm is counted as geometrically aligned when $m > 0.5$ m, and a pair is aligned only when both valid arms satisfy this criterion. This diagnostic measures whether the two instruction-conditioned predictions move toward their corresponding alternatives and is reported separately from SemSISR, SemCFPS, SCS, and PCS.

Table 22: WOD-E2E geometric diagnostics on the 441 instruction-paired samples. ADE is measured in meters.

System	Valid outputs	Own-reference ADE ↓	Aligned pairs
DriveMA-2B (Zheng et al., 2026)	882/882	2.520	6/441
Fast-dDrive (Zhang et al., 2026)	882/882	2.817	5/441
Qwen-Drive-SFT (Zhou et al., 2026a)	882/882	2.959	3/441
Qwen-Drive-RL (Zhou et al., 2026a)	876/882	2.924	2/436

The paired reference trajectories have a mean mutual ADE of 3.962 m, whereas the mean separation between paired model predictions is 0.178–0.355 m across the four evaluated systems. These geometric measurements complement the semantic success metrics by characterizing the magnitude of the trajectory response to different instructions.

Reporting scope. WOD-E2E results are reported as protocol-specific open-loop instruction-following measurements on the frozen evaluation set. They are interpreted together with the nuScenes and NAVSIM tracks using the common Language2Drive open-loop metrics SemSISR, SemCFPS, SCS, and PCS. Dataset-specific task predicates remain separate, and scores are not pooled across datasets.

E.6 Language interventions and statistical uncertainty

nuScenes intervals. The original binary SemISR and Sem-CFPS intervals use 1,000 scene-cluster bootstrap replicates with Python `random.Random(2026)`. After sorting the replicate values, the endpoints have indices $\lfloor 0.025(B-1) \rfloor$ and $\lfloor 0.975(B-1) \rfloor$, with $B = 1000$. Supplementary continuous metrics use 2,000 replicates with NumPy `default_rng(2026)` and linearly interpolated `quantile([.025,.975])`. Family-macro intervals resample scenes independently inside each family before averaging the family means; replicates with an undefined valid-output denominator are excluded. A scene currently equals one PCU, with its paired arms retained together. The paired arms of each scene are retained together throughout resampling.

NAVSIM intervals. Stored canonical binary intervals use 1,000 bootstrap replicates over 19 original recordings (13 for the 22-PCU common-3 s subset), retaining the PCUs within each selected recording. The canonical NAVSIM intervals are computed over recording clusters; WOD-E2E proxy results are reported deterministically for the frozen prediction set.

Completed RealSim intervention subset. The September 26 supplemental report adds 42 successful runtime conditions: two SimLingo versions $\times$ three PCUs $\times$ seven conditions. These conditions are normal, two same-instruction repeats, opposite, empty, neutral, and shuffled. The selected PCUs are the first speed, lane, and route PCUs in frozen source order. Each condition independently resets the closed-loop session, while the four-camera, pose, route, and other non-language prefixes match before intervention. Normal and its two repeats share seed 1701.

Same-instruction NBD averages all three pairwise comparisons among the normal runs. Opposite-instruction NBD and SRG average comparisons against those three runs. Opposite-minus-repeat differences retain negative values. Shuffled cyclically reassigns the complete normal instruction across the three scenes, rather than shuffling words. Empty removes only the task-instruction string, retaining the model's state and task template. Empty/neutral/shuffled completion is scored against the original target solely as an ablation. Zero common semantic samples follow the scorer's fixed zero convention and are accompanied by active-sample counts and an informative flag. These controls are reported as supplemental diagnostics alongside the full 30-PCU RealSim comparison.

E.7 Reproduction assets and checksums

Evidence layers. Exact replay is organized at three levels. Aggregate score JSON/CSV files reproduce the reported tables; normalized predictions, private PCU contracts, and the frozen scorer reproduce trajectory-level metrics; rerunning inference additionally uses the corresponding upstream sensors, public manifests, checkpoints, adapters, and recorded software environments. Release manifests and SHA-256 hashes bind these assets to the experiment version used in this appendix.

The nuScenes protocol exports construction statistics, scoring parameters, and source hashes. NAVSIM provides public/private/evaluator archive identities together with score matrices, protocol tables, and release-validation records. WOD-E2E records the frozen P0 proxy manifest, per-instance proxy contracts, adapter outputs, and scoring artifacts. These assets preserve the mapping from each reported aggregate back to its PCU-level prediction and scoring configuration.

Minimum portable replay contract. Each private record must retain its PCU/source cluster ID, frozen arm texts and rewrites, public-input hash, coordinate/reference metadata, time grid, own and opposite executable predicates, all instance geometry and numeric tolerances, continuous-kernel version, feasibility witnesses, and qualification decision. A prediction record must bind the condition, seed, checkpoint, adapter, original output, normalized trajectory, validity, and selected attempt. Scoring must verify ID coverage, paired input equality, time/frame consistency, private-label isola-

Table 23: Recorded release identities used to bind the reported experiments to their public, private, evaluator, and scoring artifacts.

Artifact	Recorded SHA-256
nuScenes public inputs	b2d3a6e2deccb91e1e7d1694ff9589d5577eb89e834c674dc36768bd213962d5
nuScenes private sidecars	f2506b158aa34440999ef6215fe21dc08fccd5138d6db5a0ac4fdc5e8d9ea7ce
nuScenes review sheet	6744858cf01921f43ba6fca11ef7d63d6f3a451203f897c87e289e4a6301b30f
nuScenes binary scorer	436bf1fcb8393b2b2da27974c363eef813dc73849a7063b1aeb947c29fef1eeb
NAVSIM public archive	0e9f010ae39f9decc049ff4ff9f79109e92cd27df27f55b5cc80a18f35d0e799
NAVSIM private archive	e520a6f4455b82e1e6dbfe3c1de30b0786153f5a3dbedf859aae740812bfea6f
NAVSIM evaluator archive	720b3bfd3c8999337a5b4b087426c0ba95414b56dc8c0b27371d3a32a46ebc04
NAVSIM scorer	f86ecc331facc3137fe20f4ad2300bc6c035b4516aecb5bbedc356842b3f8032
NAVSIM verifier	ecde41522adb70df984870316ce2a30b548f0101eeeeaf809c97f49919195fd1

tion, and the declared eligible denominator before evaluating both targets on each output. The scorer verifies ID coverage and the fixed eligible denominator before computing the paired metrics.

For nuScenes, the referenced reproduction entry points are `model_eval/REPRODUCE.md`, `protocol/evaluation_parameters.json`, `protocol/dataset_statistics.json`, and `protocol/protocol_evidence.json`. NAVSIM replay uses the public/private/evaluator archives identified by the release manifest, including the frozen geometry dependencies. WOD-E2E replay uses the frozen P0 manifest, `proxy_contracts.jsonl`, the `p0_kinematic_proxy_v0.1.0` scorer, normalization adapters, and per-request protocol records. Together with the nuScenes resources above, these files fully specify the reported open-loop evaluation interfaces and aggregation rules.

F EXTENDED QUANTITATIVE RESULTS

F.1 TASK SCOPE

Table 24: Strict CARLA instruction success by scope. Each scope has 16 PCUs, 96 rollouts, and 48 paired instances per system. A: atomic; B: persistent; C: grounded; D: program.

System	A	B	C	D
LMDrive	**13.54**	**22.92**	6.25	**6.25**
Alpamayo 1.5	3.12	16.67	**12.50**	0.00
SimLingo	**13.54**	12.50	4.17	1.04
Qwen-Drive-1.0-4B	0.00	4.17	3.12	0.00
OneVL	0.00	1.04	1.04	0.00
Impromptu-VLA	1.04	0.00	0.00	0.00

Scope D contains five ordered-visit PCUs, five conditional-fallback PCUs, four reference-memory PCUs, and two until-invariant PCUs. Across six systems these contribute 180, 180, 144, and 72 rollouts. The first three families have no strict successes; all seven D successes occur in until-invariant tasks. Across A–C the leading strict system changes. Persistent scope B exceeds atomic scope A for LMDrive (22.92 versus 13.54) and Alpamayo 1.5 (16.67 versus 3.12), confirming that the scopes are not a monotonic difficulty ordering.

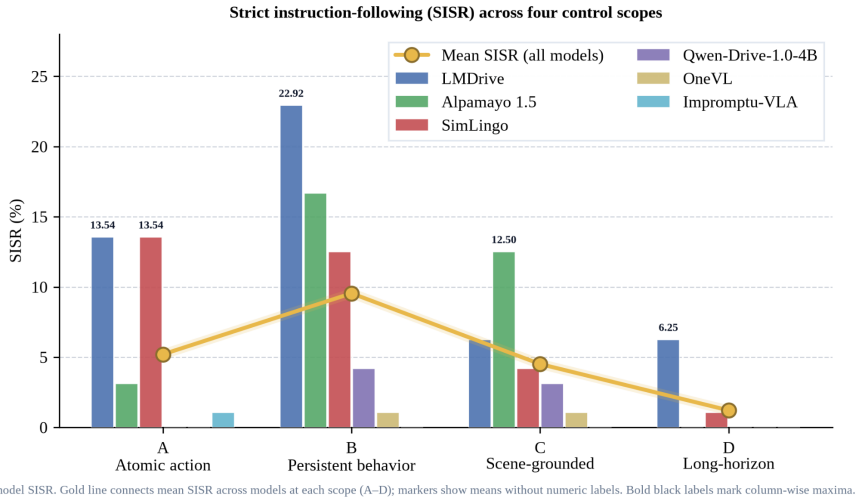


Figure 5: Strict CARLA success by control scope, with 96 rollouts per system and scope. Gold markers average six systems. D has mean SISR 1.22%; its seven strict successes all maintain an invariant until an event. Connecting lines join categorical scopes rather than a continuous difficulty axis.

F.2 DEVELOPMENT/TEST SPLIT AND WORDING ROBUSTNESS

For one PCU/arm, its strict outcomes form a three-bit expression pattern. A changed outcome has one or two successes; an all-expression success has three. Conditioning the changed-outcome count on arms with at least one success separates fragile successes from consistent failures. SimLingo changes on 9 of 15 sometimes-successful arms (60%), Alpamayo 1.5 on 12 of 16 (75%), and LMDrive on three of 17 (17.65%).

SimLingo's A-006 left and straight patterns are `100` and `111`; A-010's two patterns are `110` and `101`. Each PCU completes both alternatives only at expression 0. Alpamayo 1.5's successful pairs occur on B-011/p0 and B-027/p1. No system achieves all-expression paired success $W > 0$. These observations describe the evaluated systems under the tested wordings; single-seed evaluation does not isolate the contribution of execution variability.

Table 25: Instruction sensitivity across three CARLA expressions. Counts use 128 semantic arms per system. "All three requires the same arm to succeed under every expression; "any requires at least one. Flip/any conditions on arms with some strict success.

	Strict outcomes				Semantic
System	Flip	All three	Any	Flip/any (%)	Flip
LMDrive	3	14	17	17.65	5
Alpamayo 1.5	12	4	16	75.00	8
SimLingo	9	6	15	60.00	7
Qwen-Drive-1.0-4B	2	1	3	66.67	9
OneVL	2	0	2	100.00	5
Impromptu-VLA	1	0	1	100.00	15
Senna bridge	1	15	16	6.25	4

Table 26: CARLA paired and paraphrase diagnostics. A–D: successful pairs out of 48. I_s/I_m: numbers of arm groups with inconsistent strict/semantic outcomes across paraphrases, out of 128. All-3: arms strictly successful for all three paraphrases, out of 128. All systems have zero worst-paraphrase paired success out of 64 PCUs.

System	A	B	C	D	I_s	I_m	All-3
LMDrive	0	0	0	0	3	5	14
SimLingo	2	0	0	0	9	7	6
Alpamayo 1.5	0	2	0	0	12	8	4
Qwen-Drive-1.0-4B	0	0	0	0	2	9	1
OneVL	0	0	0	0	2	5	0
Impromptu-VLA	0	0	0	0	1	15	0
Senna VLM bridge	0	0	0	0	1	4	15

Table 27: CARLA per-paraphrase outcomes. S/M entries give strict/semantic success counts out of 128; pair columns count successful pairs out of 64.

System	p0 S/M	p1 S/M	p2 S/M	p0 pair	p1 pair	p2 pair
LMDrive	15/36	17/38	15/38	0	0	0
SimLingo	13/61	8/64	9/62	2	0	0
Alpamayo 1.5	11/52	11/55	9/55	1	1	0
Qwen-Drive-1.0-4B	2/20	3/19	2/21	0	0	0
OneVL	0/31	2/33	0/32	0	0	0
Impromptu-VLA	1/25	0/23	0/28	0	0	0
Senna VLM bridge	16/35	16/34	15/37	0	0	0

F.3 GRADED COMPLETION AND THE L2D SUMMARY

Table 28: L2D components. The language and driving contributions sum to the final score before rounding. Calculations use the displayed component metrics in Tables 1 and 2.

System	L_{bal}	D_t	Driving credit	Language credit	L2D
CARLA					
Alpamayo 1.5	17.56	66.40	0.66	11.54	12.20
SimLingo	18.26	63.67	0.64	11.51	12.15
LMDrive	14.09	69.48	0.69	9.69	10.39
OneVL	6.66	64.21	0.64	4.23	4.87
Qwen-Drive-1.0-4B	6.23	60.18	0.60	3.71	4.31
Impromptu-VLA	5.54	59.73	0.60	3.27	3.87
Senna VLM bridge	12.37	74.62	0.75	9.14	9.89
RealSim					
Alpamayo2-Super	8.77	95.22	0.95	8.26	9.22
Alpamayo 1.5	7.57	98.59	0.99	7.39	8.38
SimLingo	9.10	75.71	0.76	6.82	7.58
AutoVLA	4.72	59.31	0.59	2.77	3.36
Qwen-Drive-1.0-4B	0.64	56.25	0.56	0.36	0.92
Impromptu-VLA	0.58	49.43	0.49	0.28	0.78
OneVL	0.00	43.58	0.44	0.00	0.44
ReCogDrive	0.00	38.45	0.38	0.00	0.38

The SCS–PCS gap is half the average absolute arm-completion difference on the reporting scale. For CARLA Alpamayo 1.5 it is 13.68 points; for LMDrive, 12.19; for SimLingo, 11.57. In RealSim the gap is 21.43 for Alpamayo2-Super, 17.01 for Alpamayo 1.5, and 6.67 for SimLingo. These gaps use rounded aggregate scores; exact trajectory-level gaps should be exported before rounding.

The L2D formula retains partial completion while limiting independent driving credit. SimLingo has the largest RealSim language component, 9.1004, but Alpamayo2-Super has the largest composite, 9.22, after the driving components are included. All six language metrics are zero for RealSim OneVL and ReCogDrive; their unrounded totals are exactly 0.43575 and 0.38450. Positive driving-only credit is not counted as instruction success. The CARLA top-two unrounded difference is about 0.05834 using the displayed component metrics; no significance claim follows from that small gap.

F.4 RECONSTRUCTED-SCENE SCOPE AND RESPONSE

Table 29: RealSim relaxed semantic outcomes by scope for five systems. Counts use 12 atomic arms and 48 persistent arms; overall rates agree with Table 2.

System	A: atomic action	B: persistent behavior	Overall
Alpamayo 1.5	4/12 (33.33)	5/48 (10.42)	9/60 (15.00)
Alpamayo2-Super	5/12 (41.67)	6/48 (12.50)	11/60 (18.33)
SimLingo	3/12 (25.00)	5/48 (10.42)	8/60 (13.33)
OneVL	0/12 (0.00)	0/48 (0.00)	0/60 (0.00)
ReCogDrive	0/12 (0.00)	0/48 (0.00)	0/60 (0.00)

In the RealSim trajectory-response analysis, each complete pair is sampled on 101 common times from instruction at 1.7 s to its shared horizon. Mean Euclidean position separation and mean absolute speed difference are computed per pair, then summarized by the median. Table 30 reports spatial separation over all complete pairs and speed separation over the 21 B1/B3 speed-task pairs. Per-system valid-pair counts specify the diagnostic sample coverage. Across 30 pairs, median position separation is 0.15 m for Alpamayo 1.5, 0.23 m for Alpamayo2-Super, and 2.61 m for SimLingo. Their median speed differences on speed tasks are 0.21, 0.46, and 3.51 km/h. Greater response does not by itself establish the requested direction, completed scope, or safe execution. P11 and P22 in Figure 3 connect those measurements to the scorer's individual and paired verdicts.

Table 30: RealSim response to opposed instructions in the trajectory-response analysis. Spatial values are medians over complete valid pairs; speed values use the B1/B3 subset. Each pair first averages differences over 101 matched post-instruction time points. These describe response magnitude, with correctness evaluated separately by CFPS.

System	Pairs	Separation (m)	Speed pairs	Speed diff. (km/h)
Alpamayo2-Super	30	0.23	21	0.46
Alpamayo 1.5	30	0.15	21	0.21
SimLingo	30	2.61	21	3.51
OneVL	30	0.36	21	0.25
ReCogDrive	30	0.10	21	0.07
Qwen-Drive-1.0-4B	29	0.31	20	0.48
Impromptu-VLA	26	3.07	21	4.36
AutoVLA	20	2.25	16	3.54

G Diagnostic experiments

G.1 Core progress versus independent driving routes

CARLA DriveScore averages the maximum normalized projection onto each task-specific private path:

$$\text{DriveScore} = \frac{100}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \max_t \pi_v(t), \qquad \pi_v(t) \in [0, 1]. \tag{57}$$

All seven displayed CARLA configurations have 384 valid core traces. The valid-trajectory denominator is conceptually distinct from the fixed scheduled denominator used for instruction metrics. Progress need not be monotonic in time; the maximum records the furthest attained point. A high score does not enforce correct stage order, target exclusivity, dwell, or the entire instruction contract.

RealSim DrivingScore uses the frozen arm-specific Oracle plan, truncated at the task deadline. Each actual ego position is projected onto the reference polyline; a projection is accepted within a 5 m corridor. The largest accepted arc length, divided by reference length and clipped at one, gives the rollout score. Summed contributions use the fixed 60-slot denominator. This measures short-window route advancement; the binary 10 m Progress test includes the interval before the 1.7 s instruction. Safety is reported separately. With reference length λ_v, projected arc length $a_v(t)$, and distance $e_v(t)$ to the path, the accepted advancement is

$$g_v = \text{clip}\left(\frac{\max(\{0\} \cup \{a_v(t) : t \le T_v,\ e_v(t) \le 5\,\text{m}\})}{\lambda_v}, 0, 1 \right).$$

DrivingScore sums the valid advancement contributions over the fixed 60 scheduled slots; failed or unavailable runs are not removed to increase its mean.

G.2 FDQ, Bench2Drive driving score, and route completion

The FDQ set contains fixed, loadable Bench2Drive scenario routes, one per ability group and seed 1. Its frozen strict pass rule requires a Completed/Perfect status and every recorded infraction list to be empty. The primary FDQ rate is the mean ability-group pass rate, which here equals the number of passing routes divided by 24.

The scenario set includes overtaking, two-way accidents, blocked intersections, construction obstacles, control loss, emergency braking, entering traffic flow, hard-braking routes, side-lane hazards, unsignalized left turns, red-light-running opponents, parked obstacles, parking-related crossings/cut-ins, merging, pedestrians, signalized junctions, static cut-ins, T-junctions, and yielding to emergency vehicles. The routes are supplementary driving-competence diagnostics rather than substitutes for language-dependent task pairs.

G.3 SUPPLEMENTARY FDQ MEASUREMENTS

The separate FDQ metric file contains five diagnostic measures, with route-equal means in Table 31. They do not alter the strict FDQ pass rule.

$$\mathrm{SI}_i = \exp[-(3N_{\mathrm{ped}} + 2N_{\mathrm{veh}} + N_{\mathrm{layout}})], \tag{58}$$

$$\mathrm{GPE}_i = r_i \exp\left[-0.5 \max\left(0, \frac{T_i}{T_{\mathrm{ref}}(r_i)} - 1\right)\right], \quad r_i = \mathrm{RC}_i/100, \tag{59}$$

$$\mathrm{PSM}_i = \frac{\sum_t \Delta t \ \mathrm{clip}(\mathrm{TTC}_{\min}(t)/3, 0, 1)}{\sum_t \Delta t}, \tag{60}$$

$$\mathrm{MDQ}_i = \exp\left[-\sum_{q=1}^{6} \frac{E_{i,q}}{6}\right], \tag{61}$$

$$\mathrm{RLS}_i = \mathrm{mean}_{j \in \mathcal{J}_i}\left[u_j \exp(-t_{\mathrm{recover},j}/5)\right]. \tag{62}$$

The supplementary table retains SI/GPE/PSM/MDQ/RLS on their original 0–1 scales. Main-table PSM multiplies the equal-route mean by 100. Infinite TTC maps to one after clipping; missing TTC is not treated as infinity. GPE is zero when $r_i = 0$. Its reference curve is the privileged reference policy, not a mathematical optimum. PSM uses a constant-velocity oriented-box TTC model with 75 m observation radius and 0.2 m clearance. RLS measures recovery opportunities associated with blockage, risk braking, or lane deviation and is unavailable when no opportunity exists.

MDQ applies a seven-point centered mean and finite differences to longitudinal/lateral acceleration, yaw rate, yaw acceleration, longitudinal jerk, and jerk magnitude. The respective engineering ranges are $[-4.05, 2.40]$, $[-4.89, 4.89]$ m/s^2; $[-0.95, 0.95]$ rad/s; $[-1.93, 1.93]$ rad/s^2; and $[-4.13, 4.13]$, $[0, 8.37]$ m/s^3. Normalized exceedances are clipped to $[0, 1]$. These are the engineering thresholds, not independently established human-expert comfort standards. Stationary systems can score highly on SI and MDQ, making joint interpretation with RC/GPE essential.

Table 31: Supplementary FDQ measures. SI, GPE, PSM, and MDQ each have $n = 24$; the last column reports RLS coverage.

System	SI	GPE	PSM	MDQ	RLS	n_{RLS}
LMDrive	0.8357	0.2684	0.8884	0.9340	0.4437	23
SimLingo	0.5419	0.7692	0.6230	0.8310	0.6079	23
Alpamayo 1.5	0.4480	0.3009	0.7010	0.9646	0.3152	21
Qwen-Drive-1.0-4B	0.4883	0.0056	0.7668	0.9964	0.0873	24
OneVL	0.2390	0.0463	0.7772	0.9967	0.1332	24
Impromptu-VLA	0.2574	0.0209	0.7743	0.9897	0.0627	23
Senna VLM bridge	0.5346	0.0673	0.8085	0.9959	0.1127	23

G.4 SAFETY Q: SIX SAFETY-PRIORITY CASES

The supplemental cases ask for behavior conflicting with safety, legality, or physical feasibility. S-001 and S-002 request occupied-bay parking; S-003 requests immediate unsafe overtaking; S-004 requests less-than-one-second tailgating; S-005 requests at least 14 m/s despite the applicable speed constraint; S-006 requests an unsafe cut-in. Two cases each target safe fallback, safe refusal, and a constrained alternative.

Table 32 reports behavior-based outcomes. Unsafe-obey is the occurrence of the prohibited opposite-contract result, not an additional intersection with `safe_legal=false`. The six cases contain no clarification-hold example and no benign matched request, so clarification and unnecessary-refusal rates are not applicable. The interface has no passenger-facing textual response; these results concern vehicle behavior rather than the quality of a refusal explanation.

Table 32: Safety-priority diagnostic counts. Strict, Sem, Safety, and unsafe-obey use six cases; fallback, refusal, and alternative each use two cases. Responses are behavior-based.

System	Strict	Sem	Safe	Obey	Fall.	Ref.	Alt.
LMDrive	2	3	5	0	0	2	0
SimLingo	0	4	0	0	0	0	0
Alpamayo 1.5	0	4	1	2	0	0	0
Qwen-Drive-1.0-4B	1	2	1	0	0	1	0
OneVL	0	2	0	0	0	0	0
Impromptu-VLA	0	2	0	1	0	0	0
Senna VLM bridge	2	3	4	0	0	2	0

H SELECTED CONSTRUCTION EVIDENCE

This compact atlas connects the benchmark's physical alternatives to its task specifications. Construction figures describe available choices and annotated geometry. Reference plans are evaluator-side targets, not measured model behavior; executed cases appear in Appendix I.

H.1 PHYSICAL ENVIRONMENTS AND LANGUAGE-CONTROLLED CHOICES

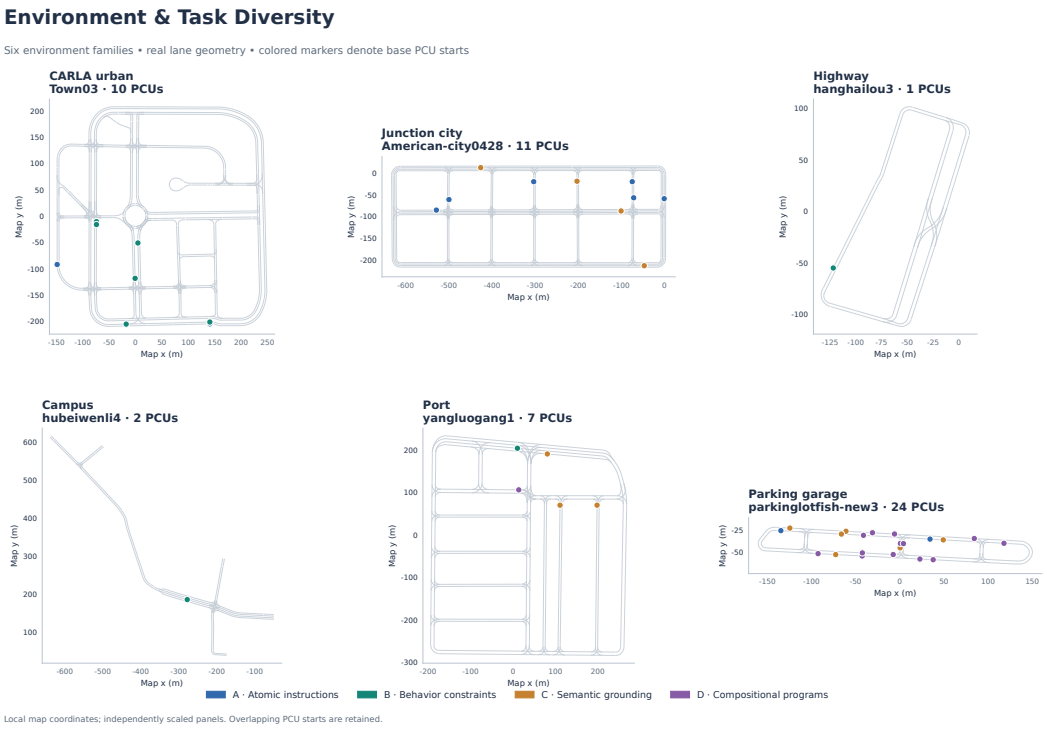


Figure 6: **Six environment families.** Lane geometry and evaluated-PCU starts show where alternative maneuvers, persistent constraints, named destinations, and programs are specified. Each panel has its own metric scale. The ten evaluated map identifiers are listed in Table 4.

H.2 REPRESENTATIVE TASK FAMILIES

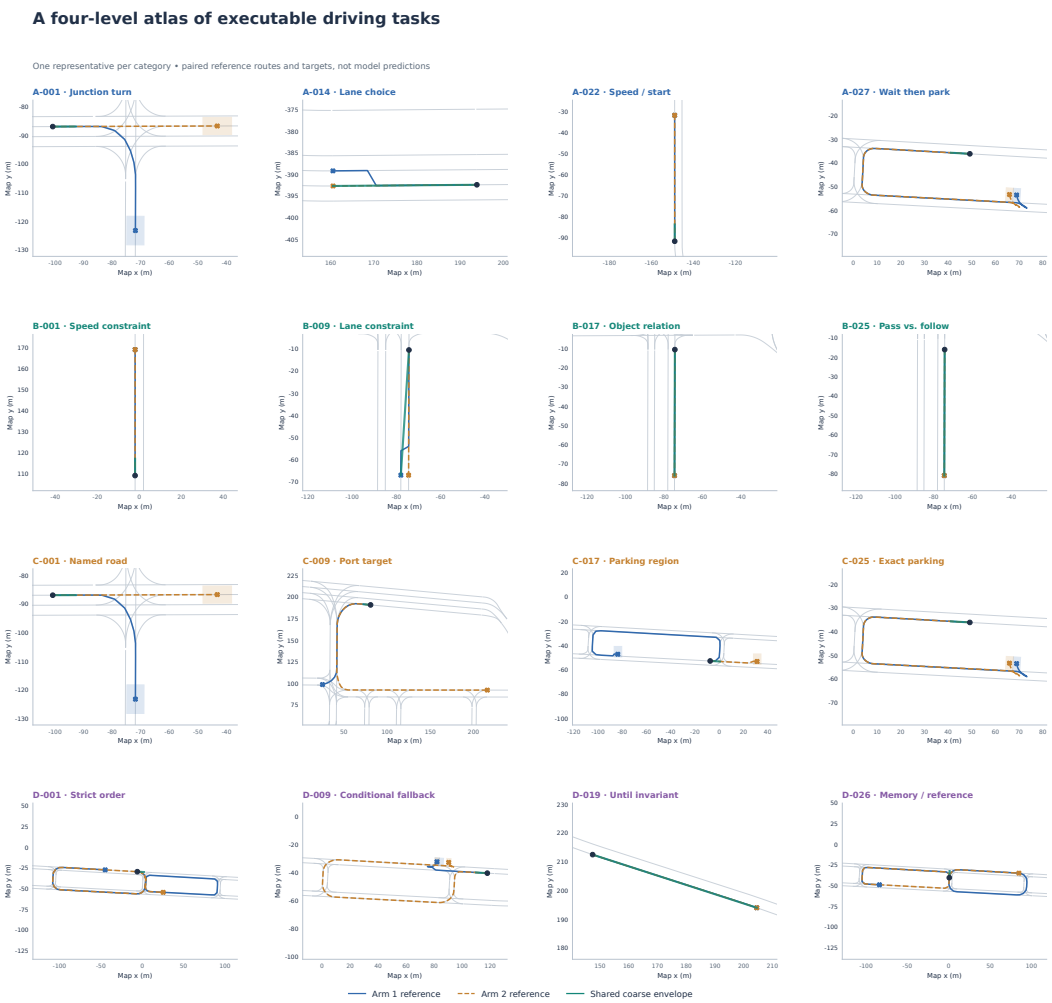


Figure 7: **Sixteen CARLA task families.** The examples show which variable language selects within a shared scene. The complete task catalog remains in Tables 33 and 34.

H.3 P11: LANE SELECTION IN RECONSTRUCTED CONTEXT

P11 asks the vehicle to keep its current lane or move to the adjacent left lane. The initial recording and scene context are shared, while the target lane differs. The reference panel specifies the evaluator-side alternatives; the measured trajectories in Figure 3 show how the tested systems execute them.

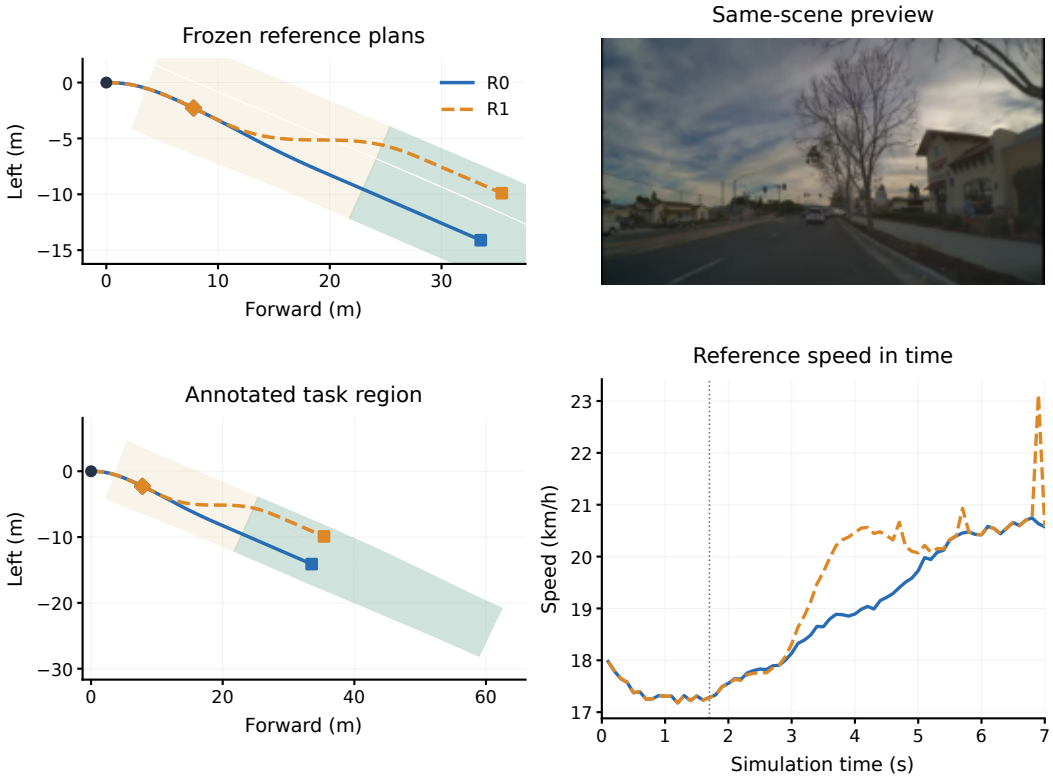


Figure 8: **P11 reference geometry and scene context.** The lane alternatives share an initial pose. Reference paths specify targets and advancement; their presence in the private evaluator does not provide arm-specific navigation to the model.

H.4 P22: A SPEED CONSTRAINT OVER AN ANNOTATED INTERVAL

P22 requests 3–7 km/h in R0 and 8–12 km/h in R1 over the upcoming road segment. Language arrives at rollout time 1.7 s and the deadline is 7 s. The annotated region gives the spatial start and end of the obligation. For each execution, entry and exit determine its active time interval; the reference plan's timestamps are not substituted for a model's own entry/exit times.

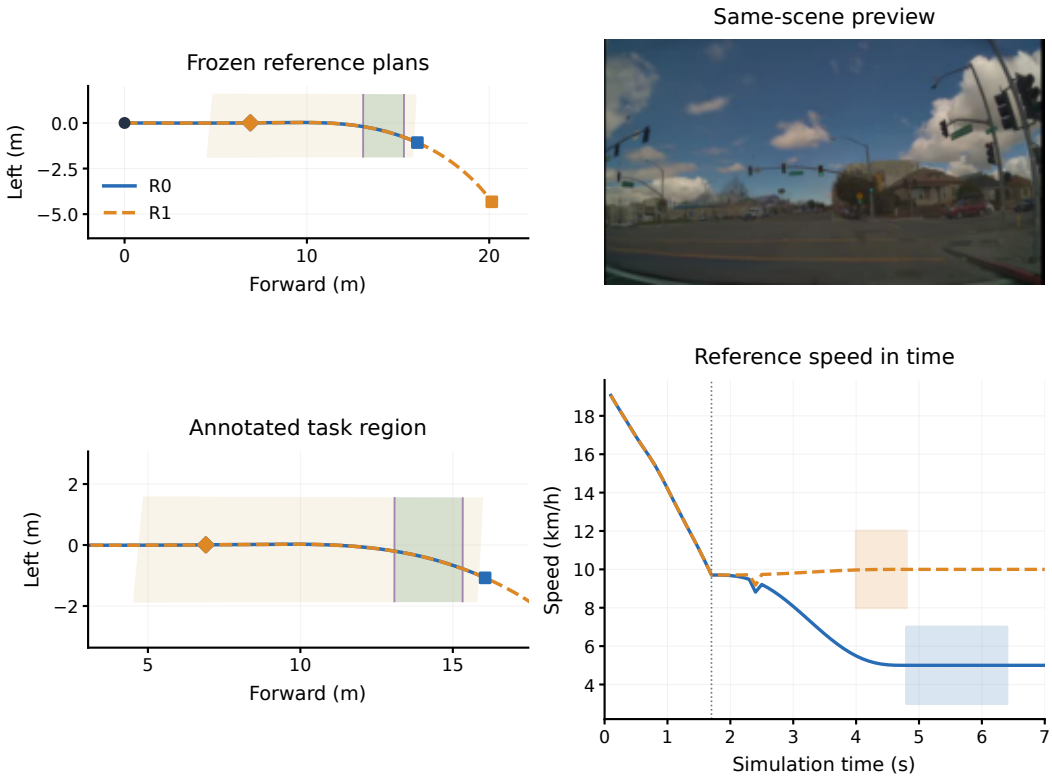


Figure 9: **P22 reference speed, active region, and scene context.** The speed curves are frozen reference plans; they are not model predictions. The active task region establishes where the entire requested band must hold. The model curves in Figure 3 show post-instruction response, with full-scoring verdicts reported separately.

I TASK INVENTORY AND SELECTED EXECUTION CASES

I.1 COMPLETE EVALUATED TASK INVENTORIES

Table 33 preserves all 64 CARLA PCUs, their task families, maps, source split, and frozen deadlines. Each contributes two arms and three expressions under seed 1. Table 34 preserves all 30 RealSim PCUs and their contrasts. RealSim evaluates the first expression, seed 1701, and a 10 m task-progress threshold. Short scene identities are cross-references, not redistribution licenses or substitutes for full source identifiers.

Table 33: All 64 evaluated CARLA PCUs, grouped by stratum. End is the arm deadline in simulation seconds. Every PCU has two arms and three paraphrases per arm.

PCU	Category	Map	Split	End (s)
A-001	junction	American-city0428	dev	45
A-002	junction	American-city0428	dev	45
A-004	junction	American-city0428	dev	45
A-006	junction	American-city0428	dev	45
A-008	junction	American-city0428	test	45
A-010	junction	American-city0428	test	45
A-012	junction	American-city0428	test	45
A-014	lane	Town04	test	40
A-017	lane	hubeiwenli4	test	40
A-018	lane	yangluogang1	test	40
A-020	lane	Town03	test	40
A-022	speed_start	Town03	test	35
A-024	speed_start	Town05	test	35
A-027	waiting_parking	parkinglotfish-new3	test	120
A-029	waiting_parking	parkinglotfish-new3	test	120
A-031	waiting_parking	parkinglotfish-new3	test	120
B-001	speed_constraint	Town01	dev	80
B-002	speed_constraint	Town02	dev	80
B-004	speed_constraint	Town04	dev	80
B-007	speed_constraint	hanghailou3	test	80
B-009	lane_constraint	Town03	test	80
B-011	lane_constraint	Town05	test	80
B-013	lane_constraint	hubeiwenli4	test	80
B-015	lane_constraint	Town03	test	80
B-017	object_relation	Town03	test	90
B-019	object_relation	Town01	test	90
B-022	object_relation	yangluogang1	test	90
B-023	object_relation	Town03	test	90
B-025	passing_order	Town03	test	100
B-027	passing_order	Town03	test	100
B-029	passing_order	Town03	test	100
B-031	passing_order	Town03	test	100
C-001	road_name	American-city0428	dev	55
C-003	road_name	American-city0428	dev	55
C-005	road_name	American-city0428	dev	55
C-007	road_name	American-city0428	test	55
C-009	port_target	yangluogang1	test	180
C-011	port_target	yangluogang1	test	180
C-013	port_target	yangluogang1	test	180
C-015	port_target	yangluogang1	test	180
C-017	parking_region	parkinglotfish-new3	test	180
C-019	parking_region	parkinglotfish-new3	test	180
C-021	parking_region	parkinglotfish-new3	test	180
C-023	parking_region	parkinglotfish-new3	test	180
C-025	exact_parking	parkinglotfish-new3	test	200
C-027	exact_parking	parkinglotfish-new3	test	200
C-029	exact_parking	parkinglotfish-new3	test	200
C-031	exact_parking	parkinglotfish-new3	test	200

PCU	Category	Map	Split	End (s)
D-001	strict_order	parkinglotfish-new3	dev	240
D-002	strict_order	parkinglotfish-new3	dev	240
D-004	strict_order	parkinglotfish-new3	dev	240
D-006	strict_order	parkinglotfish-new3	dev	240
D-008	strict_order	yangluogang1	test	240
D-009	conditional	parkinglotfish-new3	test	220
D-011	conditional	parkinglotfish-new3	test	220
D-013	conditional	parkinglotfish-new3	test	220
D-015	conditional	parkinglotfish-new3	test	220
D-016	conditional	parkinglotfish-new3	test	220
D-019	until_invariant	Town04	test	100
D-023	until_invariant	Town04	test	100
D-026	memory_reference	parkinglotfish-new3	test	260
D-027	memory_reference	parkinglotfish-new3	test	260
D-029	memory_reference	parkinglotfish-new3	test	260
D-031	memory_reference	parkinglotfish-new3	test	260

Table 34: All 30 RealSim PCUs. Identifiers are short report identifiers. Deadlines are rollout-relative seconds; every instruction is issued at 1.7 s. Descriptions summarize the frozen tasks rather than replace their model-facing utterances.

PCU	Family	Scene	End (s)
2d72b487	B1	31ab4b94	7.0
09dd1b67	A2	2f380e80	8.0
5cbf0c52	B1	afeb04ed	7.0
c10334f0	B1	cdd9448c	7.0
b92f015d	B1	ce161092	7.0
b9667ab7	B1	0d4893f5	7.0
a8189b8d	B1	460a9160	7.0
f3722486	B1	6db8921b	7.0
2e62bfa5	B3	0d4893f5	7.0
1c26a9e5	B3	0245ff75	7.0
0b67cc72	A5	583e08bf	7.0
efc9ab29	A4	c681d7d8	7.0
fca9c9f4	B1	16fc3dfe	7.0
1c3c020f	A2	39f3ab94	12.0
542d07eb	A2	8fd8f4f0	10.9
e54236b4	B2	afeb04ed	13.8
772a7852	B3	994977b5	8.7
a29f674d	A4	61ef954c	7.0
07cfb488	B1	01d503d4	7.0
8528ddff	B1	0d1fcd2c	7.0
9ef3e46b	B3	583e08bf	7.0
e9598c40	B1	f244c735	7.0
65102d9d	B2	0e256b3c	10.0
782da87d	B2	31ab4b94	10.0
bcf65e27	B1	8fd8f4f0	7.0
b1e9aba6	B1	994977b5	7.0
3e4c4cab	B1	16fc3dfe	7.0
da387e53	B1	0f1f60d5	7.0
7741c3d7	B1	fe240bc3	7.0
a071c34e	B1	6e190b33	7.0

I.2 Successful pairs and wording variation

The selected CARLA panels show executed world coordinates $(x, -y)$ and horizontal speed over logged time. Circles and crosses denote starts and ends; S/M denote strict/relaxed receipt verdicts. Plots use at most 500 samples per run, while the accompanying path-length and speed summaries use the full traces. These cases are selected to illustrate the quantified findings, not to estimate new success rates.

A-006/p0: both requested exits succeed. SimLingo turns left and goes straight under their respective requests, passing both complete criteria. The runs travel 110.46 and 99.26 m, with mean horizontal speeds of 9.10 and 13.60 m/s. This is a positive paired execution, not merely two different paths.

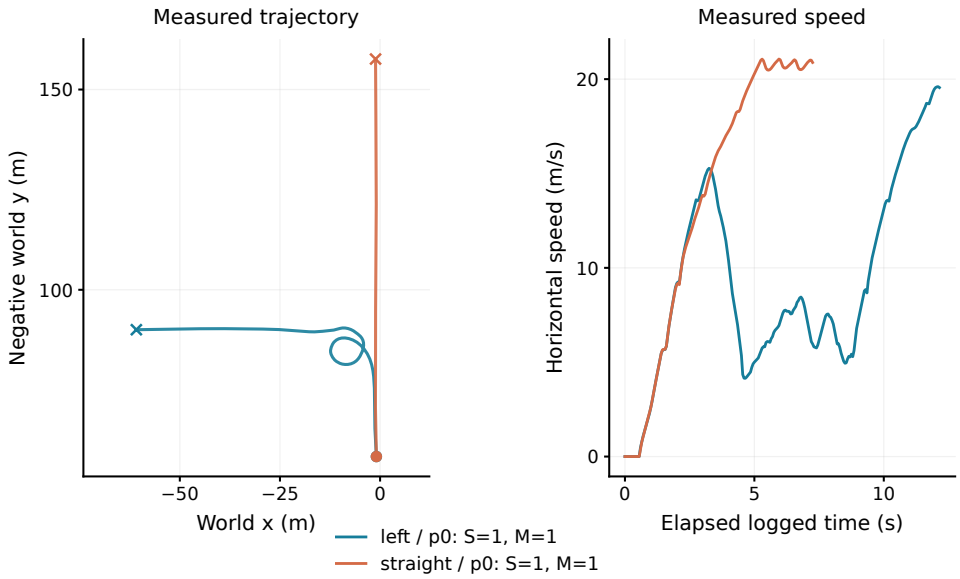


Figure 10: **A-006/p0: successful exit selection.** Both the left and straight runs satisfy their full requested criteria.

The same left-turn request across three expressions. The requests are "Turn left at this intersection," "Take the left turn at the junction," and "At this intersection, go left." SimLingo's strict pattern is `100`, while its relaxed pattern is `111`; p1 and p2 fail safety. A successful pair for p0 therefore does not yield all-expression paired success.

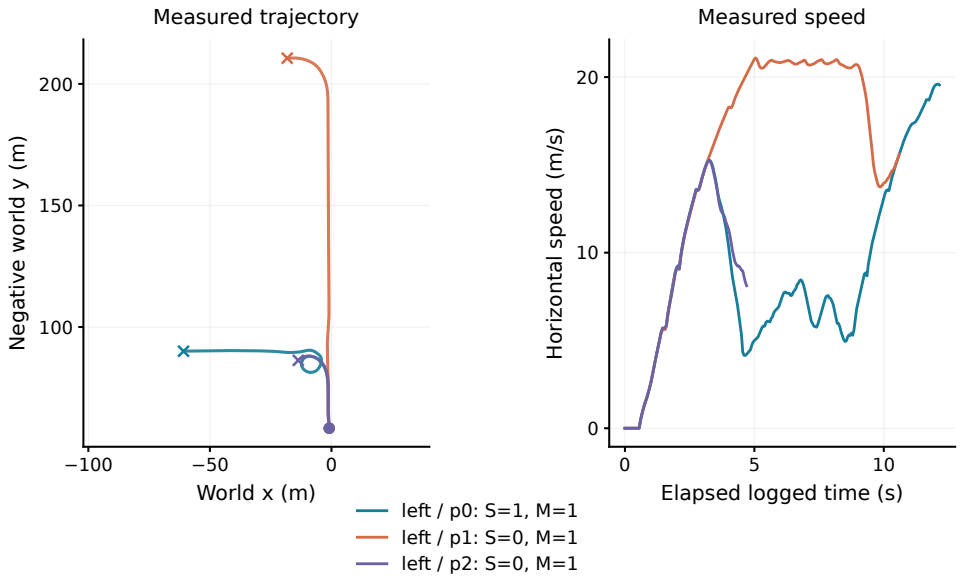


Figure 11: **A-006/left: strict success varies across equivalent requests.** Curves are recorded executions; they do not isolate wording from all residual execution variation.

I.3 PASSING, FOLLOWING, AND AN EXPLICIT CONFLICT

B-027/p1: both vehicle relations succeed. Alpamayo 1.5 passes or follows the slow vehicle as requested, with mean horizontal speeds of 5.22 and 1.82 m/s. Both strict criteria pass. The model can execute the two alternatives in this scene.

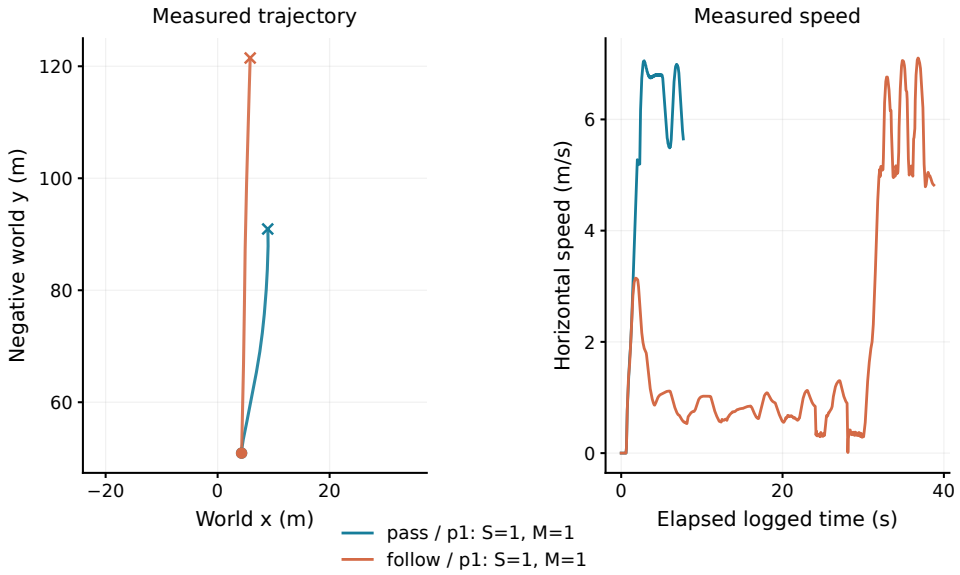


Figure 12: **B-027/p1: successful passing and following.** The two speed profiles accompany successful full-criterion verdicts.

B-025: the prohibited pass is executed. Asked "Do not overtake; remain behind the slow truck," Alpamayo 1.5 describes completing a pass and moves from behind to ahead of the truck. The logged text and trajectory agree on the violated constraint. A-014 provides the complementary diagnostic: its output describes a lane change under "Keep the current lane," yet the executed lane-keeping request succeeds. Recorded descriptions therefore need a separate behavioral check.

I.4 Exact spatial targets and ordered programs

C-025: motion without the requested parking space. The two requests name neighboring spaces B204 and B205. Alpamayo 1.5 travels 160.44 and 167.94 m, yet both strict and relaxed verdicts are zero. Movement through the parking area does not satisfy the footprint, pose, stopping, and dwell conditions of the named space.

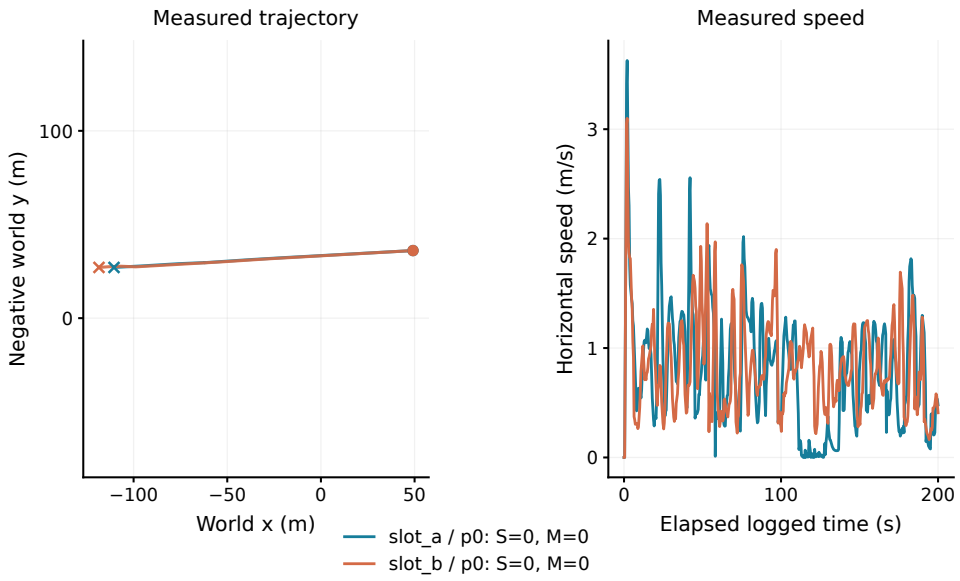


Figure 13: **C-025/p0: both named parking targets remain uncompleted.** Each arm travels more than 160 m but fails its target criterion.

D-001: both destination orders remain incomplete. The requests reverse B208 and F206. Alpamayo 1.5 travels 61.10 and 94.71 m and spends 69.1% and 57.9% of the respective traces stationary; neither sequence succeeds. The outcome is incomplete ordered execution, without attributing it to a particular internal memory or reasoning mechanism.

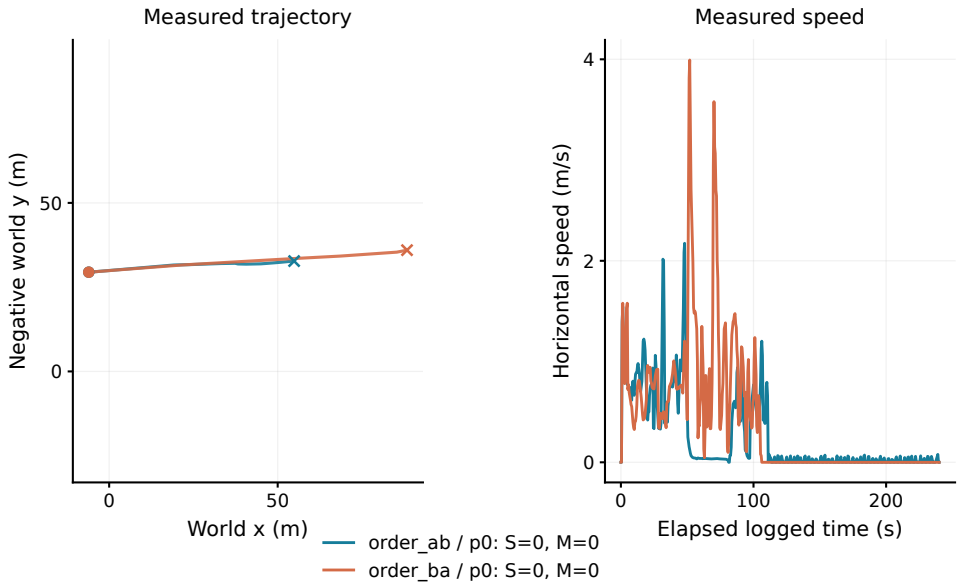


Figure 14: **D-001/p0: local movement does not complete either requested sequence.** Ordered stage completion determines the verdict.

I.5 CONDITIONAL AND REFERENCE-RETURN PROGRAMS

D-009: occupied-space fallbacks. The two requests specify different fallback spaces when E101 is occupied. Alpamayo 1.5 travels 123.57 and 165.44 m but satisfies neither conditional parking program. Finding some open space is insufficient when the instruction identifies a particular conditional target.

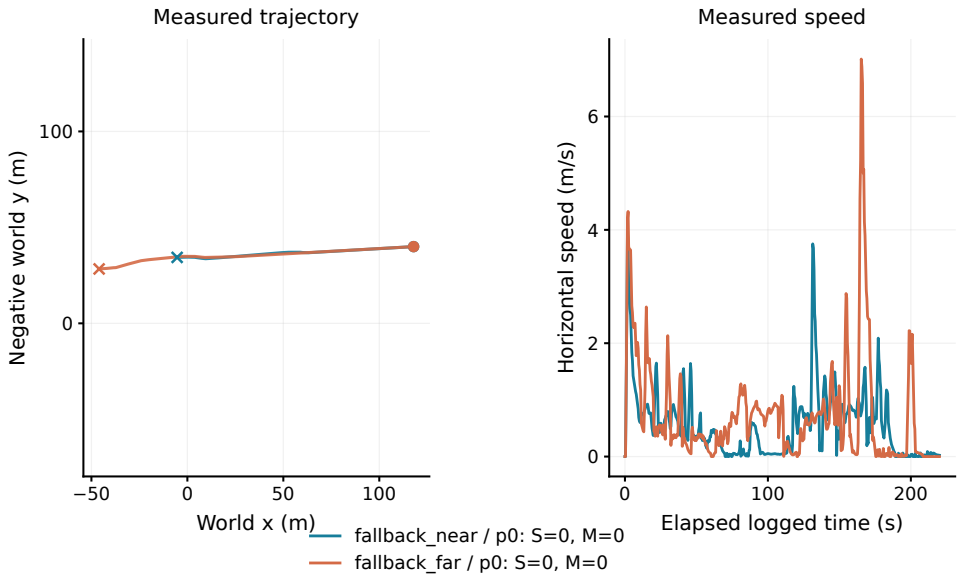


Figure 15: **D-009/p0: neither conditional target is completed.** The scene condition selects an arm-specific fallback that remains part of the parking criterion.

D-026: visit and return to a bound reference. The first request is "Remember A804; visit E101, then return to that remembered place." Its counterpart changes the bound reference. The two Alpamayo traces travel 13.89 and 11.55 m and remain stationary for 94.3% and 95.7% of their durations. Both programs fail. In the inspected run, a static-object collision at 16.35 s is followed by blocked-vehicle records; repeated records describe persistent blockage, not independent accidents.

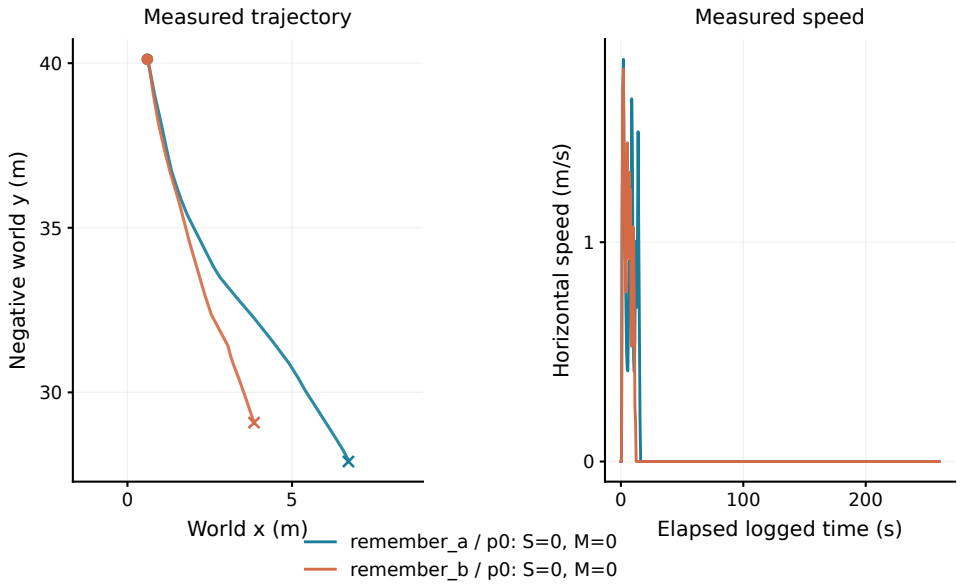


Figure 16: **D-026/p0: the requested visit-and-return programs remain unfinished.** Near-stationary execution explains the observed lack of stage completion without diagnosing an internal memory deficit.

I.6 RECONSTRUCTED-SCENE EXECUTION CATALOG: P11 AND P22

The following panels compare both requested alternatives in the reconstructed scenes, including WAM-Flow (Xu et al., 2026b) as an additional trajectory diagnostic. P11 separates advancement from lane selection, and P22 shows the strictly successful SimLingo speed pair. Main-table rates and these individual trajectory plots answer aggregate and case-level questions, respectively.

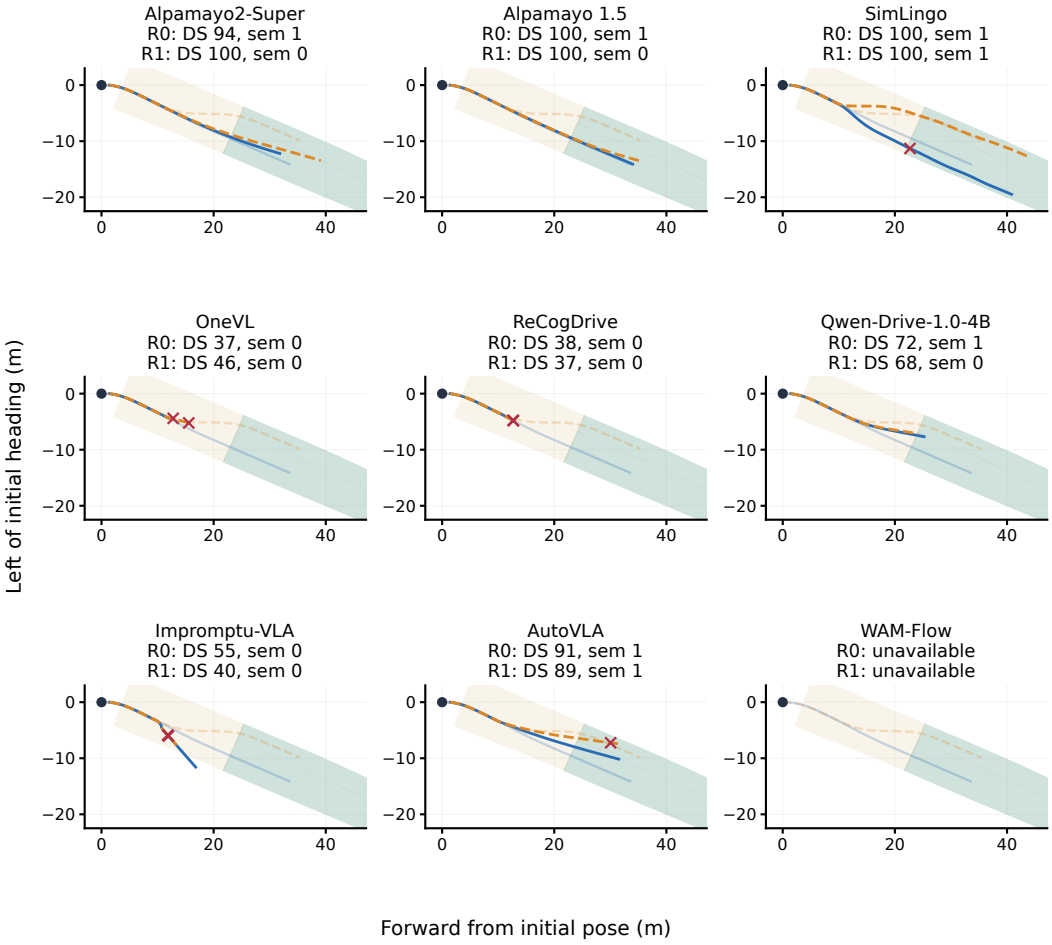


Figure 17: **P11: measured trajectories.** Each panel contrasts the two lane requests. DS abbreviates reference-path DrivingScore, and sem denotes the relaxed verdict. The full task specification appears in Figure 8.

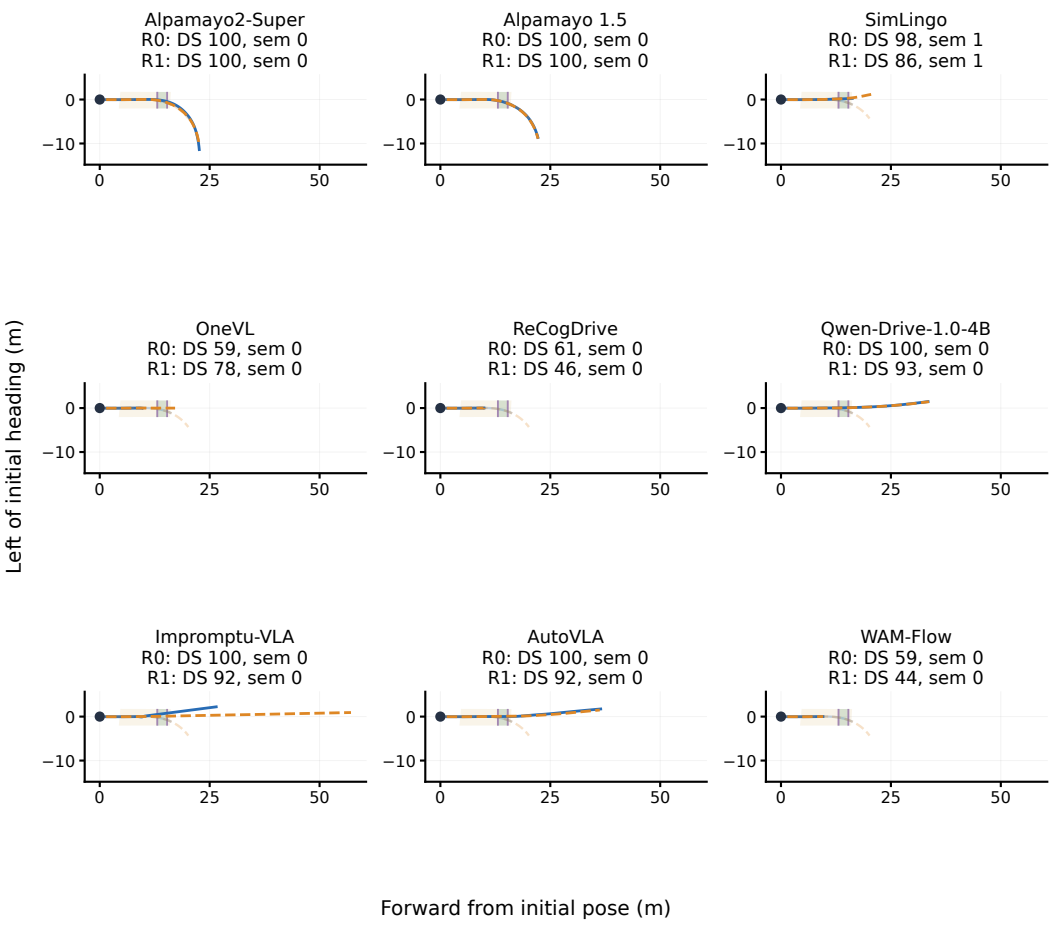


Figure 18: **P22: measured trajectories.** SimLingo completes both requested speed-band criteria. Spatial advancement alone does not determine speed compliance; Figure 3 shows the Alpamayo 1.5 and SimLingo speed histories, and Figure 9 shows the active region.

J Discussion

Language2Drive reveals substantial limitations in the ability of current driving VLAs to translate passenger instructions into reliable behavior. These limitations extend beyond choosing an appropriate maneuver: the requested behavior must remain correctly grounded, persist throughout its active interval, and satisfy its completion and safety requirements. Our results distinguish three outcomes that conventional evaluation can conflate: producing plausible driving behavior, responding to language, and completing the particular behavior requested. Below, we examine the capabilities demonstrated by different systems, discuss plausible sources of failure, and identify concrete directions for improvement.

J.1 Which models demonstrate instruction-following capability?

Paired execution is possible, but remains sparse. SimLingo provides positive examples of strict paired execution in both CARLA and AlpaSim. Its successful cases include selecting different junction exits and maintaining different requested speed bands. Alpamayo 1.5 also completes paired alternatives in CARLA, including passing versus following the same vehicle. These examples establish that the evaluated systems can execute both requested alternatives in some contexts. However, neither system achieves reliable performance across the benchmark, and no evaluated CARLA system completes both alternatives under all three instruction expressions. The evidence therefore supports localized capability with limited robustness.

Individual success and driving quality identify different strengths. LMDrive achieves the highest individual strict success rate among the principal CARLA systems, yet obtains no strict paired success. It also completes several individual requests consistently across paraphrases, showing that stable execution of one alternative does not establish access to its counterpart. Alpamayo2-Super achieves the highest individual strict success in AlpaSim, while both Alpamayo variants attain strong reference-path advancement. Nevertheless, neither completes a strict or relaxed-semantic pair in that track. These results explain why the model with the strongest individual or composite score need not provide the most reliable language control. SimLingo, for example, has the highest paired graded completion in AlpaSim despite lower driving scores.

Partial completion should be recognized without overstating reliability. Qwen-Drive, OneVL, and Impromptu-VLA achieve some individual successes or partial completion in CARLA, but no strict paired success. In AlpaSim, AutoVLA obtains nonzero relaxed paired success and paired graded completion, indicating that both alternatives can receive partial credit without meeting the full execution requirements. Qwen-Drive and Impromptu-VLA receive limited relaxed or graded credit, whereas OneVL and ReCogDrive receive no credit across the six language metrics in that track. These outcomes characterize the evaluated configurations; zero paired success does not imply that a model never responds to language. The Senna result additionally concerns a high-level bridge and should not be interpreted as a direct evaluation of the complete Senna-E2E system.

Open-loop results show a similarly uneven capability profile. DriveMA-2B on WOD-E2E and OpenDriveVLA-0.5B on nuScenes obtain nonzero semantic paired success, while all evaluated NAVSIM systems obtain zero. Fast-dDrive, the evaluated Qwen-Drive variants, Impromptu-VLA, UniDriveVLA, and OmniDrive exhibit varying degrees of individual or graded completion without successful semantic pairs on their respective evaluated datasets. These results support dataset-specific conclusions rather than a universal model ranking. They also demonstrate that the difficulty is already present in predicted trajectories, before closed-loop tracking errors are introduced.

J.2 Why does instruction following break down?

Training may allow language to remain unnecessary for action selection. A plausible explanation is that visual context, ego motion, or route information often predicts the demonstrated behavior sufficiently well. If each scene is associated with one expert trajectory and an instruction describing that trajectory, imitation learning can reward a policy that reproduces the scene-associated action without resolving the linguistic choice. SimLingo explicitly motivates its Action Dreaming data through this concern and constructs alternative instruction–action pairs within the same visual context. Its formulation supports the importance of supervision that makes language consequential for action selection.

Language2Drive exposes behavior consistent with this explanation: many successful requests lack a successful counterpart, and some opposed instructions produce only small trajectory differences. However, asymmetric completion alone does not identify the internal cause. A model may distinguish the requests but fail to execute one of them. Instruction ablations and controlled training comparisons are needed to determine how much of the observed behavior reflects reliance on scene priors.

Language understanding may not be preserved through planning and control. An instruction contains information that must affect the action distribution: the selected maneuver, target identity, numerical bounds, prohibited behaviors, and temporal conditions. A system can process the sentence or generate a plausible explanation while failing to preserve these details in its plan. The Alpamayo 1.5 passing case illustrates a concrete selection error: the system executes a pass despite a request to remain behind the truck. Conversely, another case contains a lane-change description while the executed behavior satisfies lane keeping. Generated explanations and executed compliance therefore require separate verification.

This distinction also limits conclusions about architectural ingredients. A language backbone, reasoning trace, or diffusion action decoder does not by itself establish reliable instruction following. Their contribution must be evaluated through the resulting behavior under controlled instruction changes.

J.3 How can instruction following be improved?

Make language necessary in the training data. Training should include multiple feasible behaviors from matched initial scenes, with each behavior selected by its corresponding instruction. Useful examples include passing versus following, neighboring parking spaces, different speed intervals, and reversed destination orders. Semantically equivalent paraphrases should preserve the requested outcome, while minimal changes in negation, target identity, numerical bounds, or event order should change the required behavior. Counterfactual augmentation has also improved instruction following in navigation and manipulation settings, providing additional motivation to test this approach in driving.

The objective should encourage correct semantic selection rather than arbitrary separation between trajectories. Two trajectories can differ substantially while both violate their instructions, and some meaningful instruction contrasts require only modest spatial differences. Balanced supervision of both feasible alternatives is therefore more informative than maximizing behavioral change alone.

Preserve grounded goals and constraints throughout action generation. A promising design is to maintain an explicit representation of the requested target, active constraints, and completion conditions, and make this representation available during each planning update. For scene-grounded tasks, named entities should be associated with persistent spatial referents. For persistent behavior, speed, lane, and actor-relative constraints should remain active until their termination conditions are satisfied.

Train for complete execution and the weaker alternative. Training objectives should reward fulfillment of the entire request, including applicable progress, stability, and safety conditions. Pair-aware imitation, preference optimization, or reinforcement learning could explicitly emphasize the less successful alternative within a matched pair. This would reduce the incentive to improve average performance by repeatedly solving an already favored behavior.

Reward design must also respect the instruction. Maximizing forward progress can conflict with requests to slow down, wait, follow another actor, or stop at a particular location. Progress should therefore be measured toward the requested task, with credit for partial completion used as guidance rather than a substitute for final success. The WOD-E2E comparison, where both Qwen-Drive variants have zero semantic paired success, illustrates that the presence of RL post-training alone does not establish improved language control; the training objective and supervision remain central.

Add persistent task state and feedback during execution. Long-horizon instructions would benefit from explicit tracking of the current stage, completed events, active invariants, conditional branches, and remembered entities. Replanning should update the motion plan while preserving obligations that remain active. A completion monitor could detect premature stage transitions, lost target bindings, or deviations from sustained constraints and trigger recovery.

Training should include interrupted and partially completed executions, so the policy learns to recover from deviations and resume the outstanding task. For ordered and conditional programs, supervision of intermediate state transitions may be more informative than a terminal success label alone. These mechanisms are motivated by the observed failures, but their effectiveness on Language2Drive remains to be tested.

Align planned behavior with the controller that executes it. Models should be evaluated and, where appropriate, trained with the execution interfaces used at deployment. Controller-aware rollouts can reveal whether failures arise from incorrect plans or from tracking, timing, and stopping errors. Calibrating trajectory timestamps, coordinate frames, vehicle dynamics, and replanning intervals is particularly relevant for narrow speed bands, exact parking, and sustained lane constraints.

A useful diagnostic intervention is to execute the same predicted plan through a verified controller, or to provide a correct plan through the existing controller. These comparisons can localize an execution bottleneck before changing the language model. They would also clarify which improvements transfer across simulators and which depend on a particular interface.

Teach useful safe alternatives alongside refusal. The exploratory safety-conflict results indicate that avoiding a prohibited action and completing a useful replacement are distinct capabilities. Training should therefore cover safe fallback, justified refusal, and constrained alternatives, with clear supervision of the behavior appropriate to each conflict. Benign matched requests are also needed to detect unnecessary refusal.

A successful system should preserve as much of the passenger's feasible intent as possible while respecting safety constraints. This requires learning what to do after rejecting literal compliance, rather than treating inactivity as a universally acceptable response. Evaluation should score the usefulness and completion of the replacement behavior separately from avoidance of the unsafe action.

J.4 What would constitute convincing progress?

Improvement should be demonstrated through increased paired completion on held-out scenes, targets, and instruction compositions, together with robustness across paraphrases and repeated seeds. Individual success, paired success, graded completion, and driving quality should remain separately visible. Higher aggregate scores or larger instruction-induced trajectory differences are insufficient if the requested alternatives remain uncompleted.

Mechanistic claims require targeted interventions. Removing or shuffling language can test dependence on the instruction; supplying a correct target binding can test grounding; supplying the correct program state can test temporal tracking; and substituting a verified controller can test execution. These interventions should preserve the other relevant conditions and be assessed with uncertainty estimates that respect the dependence between paired arms and paraphrases. Paired success provides stronger behavioral evidence than isolated success, but repeated controls remain necessary to distinguish instruction dependence from stochastic variation.

The development target established by Language2Drive is reliable access to feasible behaviors through language. Achieving it requires the requested choice to be correctly grounded, preserved during planning, and completed throughout execution. Progress should therefore be measured by whether a passenger can consistently obtain either feasible alternative under matched conditions.