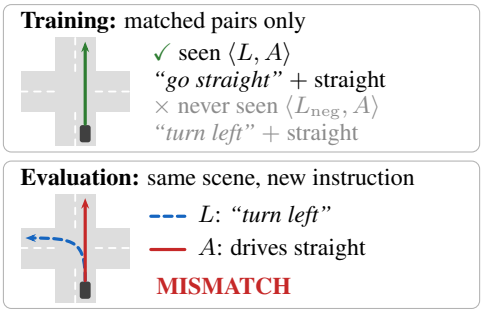
CARTENDER: VERIFIABLE LANGUAGE-CONDITIONED DRIVING BY LEARNING THE DRIVES NOT TAKEN

Anonymous authors
Paper under double-blind review

ABSTRACT

Every driving scene is logged exactly once, yet driving is inherently one-to-many. The scene decides which behaviors are admissible and the driver's intent which one to take: vision alone cannot choose among them, and language alone cannot tell what the scene permits. But the log keeps only the drive that was taken, and its instruction merely describes it. Language-conditioned driving policies are reported to follow instructions, yet on such data this cannot be established: a policy that ignores the instruction and one that obeys it regardless of the scene fit the logs equally well, and neither is distinguishable from one that weighs both. Trained only on such logs, driving vision–language–action models (VLAs) and world–action models (WAMs) receive no direct supervision on actions that violate their instructions or instructions the scene rules out. We present **Cartender**, which learns from the drives not taken. To construct matched and mismatched instruction–action pairs from real recordings without a simulator, we propose *Counterfactual Branching*, which treats each logged trajectory as one branch among several candidate futures: a vision–language model proposes the others and the instruction that selects each, and instructions the scene cannot support become negative labels. Cartender then weighs both explicitly: a learned scene–language compatibility regulates language conditioning during trajectory generation, and a reward-trained language residual re-ranks the candidates over a scene-based score. Branches serve training; instruction following is evaluated by changing instructions under identical recorded inputs, using behavior criteria rather than generated trajectory targets. On WOD-E2E, the best of four proposals reaches an RFS of 8.642 across 3,603 feasible instruction conditions. Across all 8,974 admitted proposals, mean RFS is 8.203 versus 7.906 for their matched logged trajectories, and 67% score higher than their logged counterpart. Relative to matched-only imitation with the same backbone, Cartender more than doubles the rate of fulfilling both instructions of a same-scene pair (57.00% vs. 23.67%) while RFS rises from 6.32 to 7.10; its gate cuts execution of infeasible instructions from 53.36% to 14.18%.

1 INTRODUCTION



The same driving situation admits many competent behaviors — overtake the slow truck now or follow it for three seconds first, take the gap at 40 km/h or at 25, return to the right lane the moment it is clear or stay out until the exit — and the scene settles only which are admissible, never which one is preferred (Liao et al., 2025; Wang et al., 2026b). That is settled by an intention the sensors never see. They record the situation, not the reason one behavior was chosen, and not the behaviors passed over. Nor can more data recover the rest. A vehicle occupies one position in space and time and drives one trajectory; the situation never reassembles itself for a second attempt. Every driving corpus therefore contains single samples from genuinely multi-modal conditional behavior distributions — one sample each, whatever

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.

else is recorded alongside it. The alternatives are not missing from the datasets we happen to have. They are not collectible as outcomes of that same event.

Their absence is felt twice: once in what can be trained and, with far less attention, once in what can be checked. Matched records show the executed behavior, leaving alternative responses unobserved. Establishing instruction control requires testing what the policy does under a *different* instruction in the *same* scene. Matched instruction–trajectory evaluation leaves this response untested. The instruction-following scores reported on such data are taken to show that the policy reads the instruction. They do not. The visual scene may not determine which admissible behavior is preferred, but it can already narrow the space of plausible actions enough for a largely scene-driven policy to fit the recorded trajectory well; an instruction-sensitive policy can fit the same matched pair just as well, while differing only in what happens when the instruction is changed. Both score well. Figure 1 makes the ambiguity concrete. Only an instruction intervention separates them.

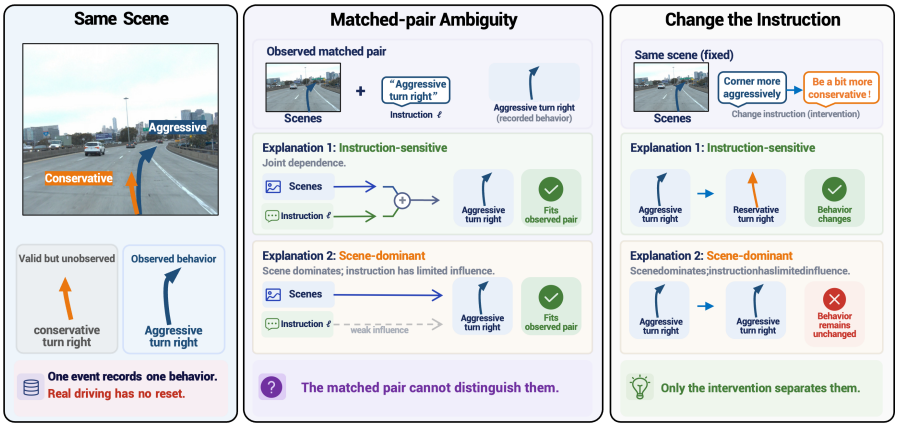


Figure 1: **Matched-pair ambiguity in real driving.** A single event reveals one recorded behavior, so the same matched instruction–trajectory pair can be consistent with policies exhibiting different instruction dependence. Holding the scene fixed and changing the instruction separates them.

Simulation can construct alternative trajectories from a recorded simulator state. SimLingo's Action Dreaming pairs language instructions with alternative trajectories generated from recorded CARLA states through offline, non-reactive simulation (Renz et al., 2025). Our construction instead operates on real recordings without a simulator. Matched-pair evaluation on these recordings retains the ambiguity above. Whether same-scene supervision helps is not the open question; simulation already indicates that it does. Two others are. The first is about data: whether such supervision can be obtained, and trusted, where the situation cannot be replayed. The second does not wait on the first. Instruction dependence can already be measured on released models, and that is where we begin.

Two of the three diagnostics we use to measure instruction dependence need no data beyond what a model was already trained on. Instructions are first shuffled across scenes. If the action loss does not move, the matched objective is insensitive to instruction identity. A rise exposes dependence. At inference, removing the instruction tests whether its presence changes the predicted trajectory. Neither diagnostic requires a behavior that was never driven, so the confound becomes something to measure rather than something to argue about. On LMDrive, shuffling increases waypoint L1 loss by 0.011. Under the evaluated Qwen-Drive interface, instruction removal shifts predictions by only 0.064 m. The third holds the scene fixed and changes the instruction, probing alternative behaviors that a single recorded event cannot jointly reveal.

We construct same-scene, multi-intent supervision on real driving data, treating its two halves differently. Declaring an instruction *inapplicable* to a recorded scene takes a label and no trajectory at all, so scene–instruction applicability is grounded in real observation by construction; supplying the behavior a *feasible* alternative instruction would have produced takes a trajectory that was never driven, and Sec. 3.1 states how it is branched and on what grounds it is admitted. We then keep training and measurement apart. Branched alternatives enter training. Evaluation changes instructions under an identical recorded input, with both requests reviewed for feasibility and predictions assessed by behavior criteria, not by distance to generated teachers.

The feasible half is the hard one: it requires the behavior that a valid alternative instruction would have produced, and we supply it with *Counterfactual Branching*. At selected keyframes, a vision–language model proposes route-relative parameters and corresponding instructions; deterministic construction and rule-based admission produce trajectory branches in the planner's prediction space. Branch quality is an empirical question, not a design assumption. We evaluate admitted branches against matched logged trajectories using Rater Feedback Score (RFS). Across the complete set of 3,603 feasible instruction conditions, the branched set reaches a best-of-four RFS of 8.642. Across all 8,974 admitted proposals, mean RFS is 8.203, compared with 7.906 for their matched logged trajectories, and 67% of branches individually outperform their logged counterparts.

A demonstration identifies a point; a preference among admissible behaviors is an ordering. That distinction follows from multimodality: a demonstration says "drive like this," but does not rank admissible alternatives under an instruction; a reward expresses that ordering. The supervision supplies the candidates, not the ordering. Same-scene alternatives make the comparison expressible; the ordering itself is defined by the frozen key-event reward in Sec. 3.2. Cartender accordingly generates a candidate set, estimates continuous scene–language compatibility to scale language conditioning during generation and ranking, and ranks the candidates with a demonstration-initialized base score plus a reward-trained language residual. Our main contributions are:

1. **The matched-pair confound, and how to measure it.** An instruction-following score on matched instruction–trajectory pairs does not establish that the policy's behavior depends on the instruction. We expose this ambiguity through instruction removal, shuffling, and same-scene swapping. Removal and shuffling, where native action loss is accessible, need no alternative trajectory targets; swapping fixes a recorded input and tests two feasible requests without generated trajectory targets.
2. **Counterfactual Branching.** Same-scene, multi-intent supervision built on real recordings without a simulator, where the logged trajectory anchors route-relative candidate futures, each paired with a proposed instruction. Branches pass rule-based admission before RFS evaluation: admitted branches average 8.203 against 7.906 for matched logged trajectories, with 67% individually scoring higher. Inapplicable instructions require labels but no synthesized trajectories.
3. **Cartender.** A policy that gates language conditioning through scene–language compatibility and ranks candidates with a reward-trained language residual over a demonstration-initialized base score. Cartender more than doubles maneuver paired success over matched-only imitation (57.00% vs. 23.67%), with RFS rising from 6.32 to 7.10. Gating reduces unsupported-request following from 53.36% to 14.18%.

2 Related Work

Multi-Proposal Trajectory Planning. Multi-proposal planners address behavioral multimodality by predicting several trajectories per scene. DiffusionDrive and its successor generate candidates through truncated diffusion and select among them with a learned scorer (Liao et al., 2025; Zou et al., 2025). The candidate set hedges against missing one answer; it does not establish which answer the instruction prefers. We supply what matched demonstrations alone do not—a same-scene criterion for intention-dependent selection.

Driving Vision–Language–Action Models. LMDrive, ORION, AutoVLA, and MindVLA-U1 connect language with trajectory generation through different architectures (Shao et al., 2024a; Fu et al., 2025; Zhou et al., 2025; Huang et al., 2026). NudgeVAD adds language-conditioned residuals to a frozen planner and examines when navigation commands make language redundant (Yang et al., 2026). The central issue here is not architecture but what matched-pair training and evaluation establish. On matched pairs, the instruction and recorded behavior co-occur; agreement with that behavior alone does not establish that the prediction changes when the instruction does. ICR-Drive tests robustness to instruction perturbations under matched simulator routes and seeds (Hamid et al., 2026). Our question is whether different feasible instructions select different appropriate behaviors under the same recorded input. Our contribution is diagnostic before it is architectural: we distinguish instruction dependence from agreement with a matched trajectory, with two tests requiring no new annotation.

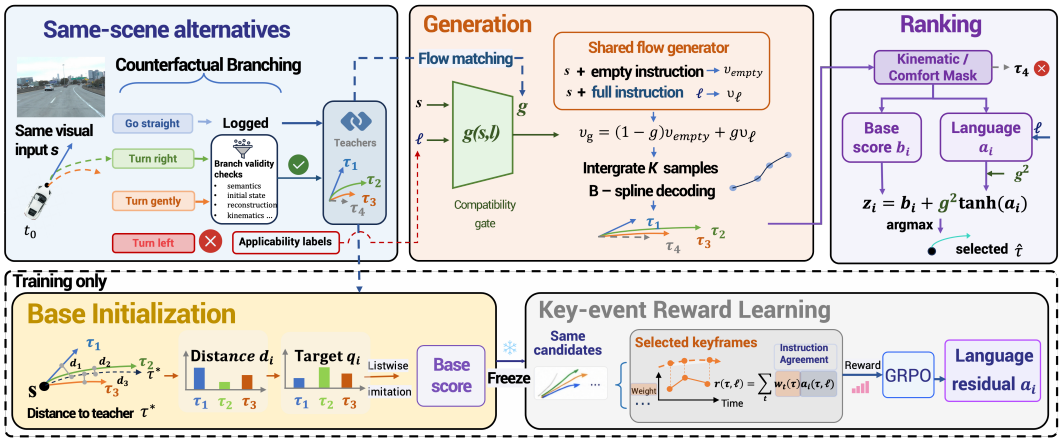


Figure 2: Overview of Cartender. Offline same-scene relations provide applicability labels for compatibility learning and teacher trajectories for flow matching and base-score initialization. Compatibility modulates generation and a language-dependent ranking residual. After base initialization, reward learning updates only the residual. At inference, the policy ranks candidates passing kinematic and comfort checks without teachers or reward evaluation.

Language–Action Alignment and Driving Supervision. LinkVLA strengthens language–action coupling through shared representations, while Drive My Way learns driving preferences (Wang et al., 2026a;b). SimLingo's Action Dreaming constructs alternatives through offline, non-reactive simulation (Renz et al., 2025); MALT-Drive uses rollout-based counterfactual supervision to align control instructions with planning (Lu et al., 2026). Our setting requires alternative trajectories for real recordings without simulator rollouts. OmniDrive supplies counterfactual descriptions of alternative outcomes (Wang et al., 2025); constructing the trajectories themselves additionally requires checking their validity. Counterfactual Branching does not sidestep this requirement. It constructs alternatives from recorded routes, admits them through rule-based checks, and evaluates their quality afterward using RFS (Sec. 3.1); these checks do not certify full scene feasibility. Generated teachers serve training, not evaluation targets: instruction control is assessed by behavior criteria under fixed recorded inputs. Further architectural and supervision details appear in Appendix A.

3 Method

Given a visual scene s and an instruction ℓ, Cartender predicts an ego trajectory $\tau \in \mathbb{R}^{T\times 2}$. Same-scene relations supply two signals: which requests the scene supports, and which trajectories realize them. Figure 2 shows how these signals govern generation and candidate ranking.

3.1 Same-Scene Relational Supervision

A single visual scene can support multiple distinct driving behaviors while leaving other requested behaviors unsupported. We organize supervision around this shared context: feasible instructions receive teacher trajectories, and infeasible instructions receive negative applicability labels. The two halves are not equally hard to obtain. An infeasible relation needs no counterfactual trajectory; a feasible alternative requires behavior absent from the recording. Counterfactual Branching supplies it. At selected keyframes, a vision–language model proposes alternative futures and their instructions alongside the logged branch. We reconstruct proposals in the planner's continuous trajectory space, enforcing observed initial position and velocity before boundary-preserving smoothing. We check source validity, semantic conflicts, initial-state consistency, reconstruction error, and kinematic and comfort violations, including those after reconstruction. Passing these checks does not establish driving quality. Section 4.4 evaluates the best branches and admitted set using automated Rater Feedback Score (RFS); RFS does not determine admission.

Branched trajectories supervise training. The diagnostic in Sec. 4.3 reuses one recorded input snapshot for two instructions independently reviewed as feasible. It does not pair approximately similar

scenes or assume both responses were recorded. Behavior criteria assess predictions without generated trajectory targets.

Binary applicability labels train a continuous compatibility score. Let $q_\ell(k \mid \ell)$ be the frozen language-intent posterior and $p_s(k \mid s)$ the scene-support probability for intent k. The requested intent $\hat{k}$ and confidence c_ℓ define

$$\hat{k} = \arg\max_k q_\ell(k \mid \ell), \qquad c_\ell = \max_k q_\ell(k \mid \ell), \qquad g(s, \ell) = c_\ell \, p_s(\hat{k} \mid s). \tag{1}$$

We train support with intent-balanced binary cross-entropy on feasible and infeasible relations, then freeze the compatibility model. The language-intent branch remains fixed. The gate estimates maneuver-level support; the generator and ranking residual condition on the full instruction, including timing, speed, and manner. Section 4.3 evaluates maneuver and qualifier control separately.

3.2 Learning from Structured Language–Behavior Relations

Teacher trajectories supervise conditional flow matching in B-spline coefficient space; a fixed decoder D_{BS} maps coefficients c to T-step trajectories (Han et al., 2026). At flow time t and state c_t, the same generator produces empty-instruction and full-instruction fields. Compatibility sets their classifier-free guidance weight:

$$v_g(c_t, t) = v_\emptyset(c_t, t; s) + g(s, \ell) \left[v_\ell(c_t, t; s, \ell) - v_\emptyset(c_t, t; s) \right]. \tag{2}$$

With $g \in [0, 1]$, scene–language compatibility interpolates between the scene-conditioned empty-instruction field and the full-instruction field, controlling the strength of language conditioning. Integrating from K noise samples and decoding yields $\mathcal{T}(s, \ell) = \{\tau_i\}_{i=1}^K$.

We mask candidates violating kinematic or comfort constraints and denote the surviving indices by $\mathcal{A}$. Their logits combine an imitation-initialized base with a bounded language residual:

$$z_i = b_\psi(s, \tau_i) + g(s, \ell)^2 \tanh a_\theta(s, \ell, \tau_i). \tag{3}$$

The base reads the scene and trajectory; the residual also reads the instruction. Scaling by g^2 limits language-driven ranking changes at low compatibility. Inference selects the highest-scoring candidate under a masked softmax. Appendix B.6 covers empty-set handling.

Geometric proximity to a teacher need not imply instruction fulfillment. We therefore initialize ranking by teacher proximity and refine it with the key-event reward. For each feasible relation (s, ℓ), proximity to its admitted A and B teachers $\mathcal{H}(s, \ell)$ defines a target distribution q_i over $i \in \mathcal{A}$. C and D are not teacher targets. We train the base scorer with

$$\mathcal{L}_{\mathrm{IL}} = -\sum_{i \in \mathcal{A}} q_i \log \pi_\psi(i \mid s, \mathcal{T}), \tag{4}$$

averaged over feasible relations, where π_ψ is the masked softmax of base scores. During base initialization, the language residual stays fixed at zero.

We then freeze the generator, compatibility model, and base scorer. Only the language residual learns, through Group Relative Policy Optimization (GRPO) (Shao et al., 2024b). A frozen key-event evaluator supplies the reward

$$r(\tau, \ell) = \sum_{t=1}^{T} w_t(\tau) \, a_t(\tau, \ell), \qquad \sum_{t=1}^{T} w_t(\tau) = 1, \tag{5}$$

where t indexes trajectory steps, w_t emphasizes salient motion changes, and a_t measures trajectory-observable instruction agreement. GRPO normalizes relative advantages within each scene–instruction group before weighting reward-driven updates by g^2. Low-compatibility groups receive a stronger anchor to the frozen imitation policy. Appendix B.5 specifies teacher targets, event scoring, and the complete objective.

Finite candidates and frozen rewards also permit exact summation of the selector's expected reward over unmasked candidates. GRPO is therefore an optimization choice rather than a requirement of the formulation. Section 4.5 compares both optimizers and tests gating, preference rewards, and staged optimization. Teachers and reward evaluation are training-only. Low compatibility attenuates language influence in generation and ranking; it does not produce an explicit refusal.

Table 1: Instruction dependence and same-scene control.

(a) Released-policy diagnostics.

Model	Ref. ADE (m)	Removal shift (m)	Shuffle $\Delta\mathcal{L}_{\text{action}}$	Swap paired success (%)
Qwen-Drive	1.413	0.064	—	21.13
SimLingo	0.842	2.511	+0.232 +1.289	50.00
LMDrive	0.203	0.178	+0.011	3.20

(b) Same-scene instruction control.

Model	Maneuver success (%)	Trajectory shift (m)	Direction consistency (%)
Vision-only	20.37	0.00	0.00
Qwen-Drive	27.78	0.60	22.22
SimLingo	72.80	0.89	62.50
Cartender	**73.83** $\pm$ **2.60**	1.666 $\pm$0.09	**83.09** $\pm$ **7.40**

(a) Frozen checkpoints follow their protocols. Ref. ADE uses original instructions; Removal uses empty inputs. Shuffle reports shuffled-minus-original native action loss, with scenes and targets fixed. SimLingo: route loss above speed-waypoint loss; LMDrive: waypoint L1. Dash: unavailable; shading: within-model contrasts. Paired success requires both requests fulfilled.
(b) Scenes and random draws are fixed. Maneuver success measures individual request fulfillment; trajectory shift measures prediction displacement; direction consistency measures alignment with the requested direction.

4 Experiments

4.1 Experimental Setup

Data and comparisons. Our data pool comprises 10,336 training and 4,580 validation scenes from WOD-E2E (Xu et al., 2025); the quality evaluation uses 1,200 validation scenes (Appendix C.4). Unless otherwise stated, Cartender variants and the language-free control share frozen U1 scene features, a 5 s horizon, and $K = 24$ candidates. Released-policy diagnostics use fixed checkpoints of Qwen-Drive (Zhou et al., 2026), SimLingo (Renz et al., 2025), and LMDrive (Shao et al., 2024a): Qwen-Drive uses our added instruction interface on WOD-E2E; SimLingo and LMDrive use their respective CARLA protocols.

Supervision and training. Counterfactual Branching constructs 3,603 feasible instruction conditions across 2,487 source scenes. Four proposals per condition yield 14,412 trajectories, of which 8,974 pass rule-based admission. The admitted A and B proposals form the teacher sets; all 3,603 A proposals pass admission, while C and D are excluded from teacher supervision. Infeasible instructions supply applicability labels only. Selector comparisons use three seeds with a shared generator checkpoint, candidate sets, and fixed reward (Sec. 3.2); Table 3 reports mean paired success. Construction and training details: Appendices B.2–B.4.

Evaluation. We test instruction dependence through removal, shuffling where native action loss is accessible, and same-scene swapping between feasible instructions (Appendix C). Maneuver-level and qualifier-level results are reported separately. Paired success requires both requests to satisfy action and hard-constraint checks. Behavioral criteria are independent of Cartender's relational training annotations. We also report RFS, ADE, and output coverage. Missing predictions count as failures in success metrics; quality statistics state their coverage.

4.2 Is the Instruction Being Read at All?

Table 1a tests frozen released models under their respective input interfaces and datasets, without our constructed supervision. These are within-model interventions, not a common cross-model benchmark; Qwen-Drive results are specific to the added instruction interface. We evaluate instruction dependence using three controlled interventions: instruction removal, instruction shuffling, and same-scene instruction swapping.

Table 2: Counterfactual branch quality and validation. RFS is evaluated after rule-based admission. Best-of-four averages conditions; mean admitted RFS and "Above logged" pool admitted proposals. "Above logged" is the fraction exceeding matched logged RFS. Two reviewers audit constructed infeasible relations.

Branch quality		Construction set		Negative-relation audit	
Matched logged RFS	7.906	Source scenes	2,487	Audited relations	228
MindVLA-U1 ref. RFS	7.92	Feasible conditions	3,603		
Best-of-4 RFS	8.642	Generated proposals	14,412	Confirmed infeasible	202
Mean admitted RFS	8.203	Admitted proposals	8,974		
Above logged (%)	67.0	Admission rate (%)	62.27	Confirmation rate (%)	88.6

Table 1a exposes the gap between predicting the recorded trajectory and following the requested behavior. Under the evaluated Qwen-Drive interface, instruction removal changes predictions by only 0.064 m, while same-scene paired success is 21.13%. LMDrive attains a reference ADE of 0.203 m but only 3.20% paired success under its evaluated protocol. These results show that low matched trajectory error does not establish reliable instruction control. SimLingo clearly uses the instruction: removal shifts predictions by 2.511 m, and shuffling increases both native route and speed-waypoint losses. Nevertheless, paired success is 50.00%, leaving at least one request unfulfilled in half of the pairs. Using language does not ensure that the requested behavior is realized; instruction dependence and behavioral fulfillment must be evaluated separately.

4.3 Instruction Dependence on Real Same-Scene Pairs

The diagnostic reuses recorded input snapshots; generated teachers are not evaluation targets. We fix observations and sampling noise while comparing two instructions feasible under the same observation. Table 1b shows that the vision-only control fulfills 20.37% of requests without changing its output; fulfillment alone does not establish instruction control. Cartender achieves 83.09% directional consistency. The response extends to execution intensity within the same maneuver. For gradual versus strong acceleration or deceleration, speed-direction accuracy is 64.58% versus 43.75%; requiring the correctly directed speed difference to exceed $0.2\,\mathrm{m/s}$ still yields 53.13% success versus 12.50%. Timing control remains limited, with 35.42% correct early–late ordering across 32 deceleration pairs. Full qualifier results appear in Appendix E.

4.4 Counterfactual Branching: Branch Quality

We evaluate constructed branches using RFS, which does not determine admission. Best-of-four RFS measures the best available candidate among A–D for each feasible instruction condition. Mean admitted RFS and the fraction exceeding matched logged RFS summarize all admitted proposals, not only the A/B teachers used for training.

Table 2 reports two aggregation units. Best-of-four takes the maximum among A–D for each of 3,603 feasible conditions before averaging. Mean admitted RFS and "Above logged" pool all 8,974 rule-admitted proposals, with each proposal compared against its condition's logged trajectory. Under this proposal-level aggregation, admitted branches average 8.203 RFS, exceeding the matched logged score of 7.906 by 0.297. Moreover, 67% score above their corresponding logged trajectories. The RFS advantage over matched logged trajectories therefore extends to a majority of admitted branches. These statistics use the complete construction across 2,487 source scenes rather than a sampled subset. A separate audit has two reviewers assess each of 228 constructed infeasible relations independently. Cases with different judgments receive an additional review. The audit confirms 202 relations (88.6%); 26 remain unconfirmed. Scoring, aggregation, and audit protocols are detailed in Appendix C.3.

4.5 Main Results and Elimination of Alternative Explanations

We first evaluate whether instruction control can be improved while preserving trajectory quality. Beyond this evaluation, we analyze whether the gains can be attributed to additional same-scene examples, training schedules, reward optimization choices, or specific supervision designs.

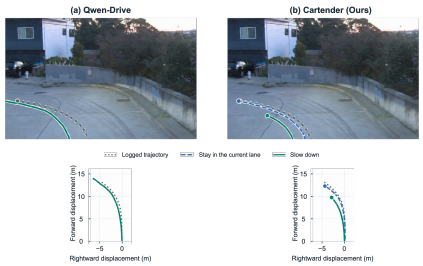
Instruction control and trajectory quality. Table 3 shows that Cartender more than doubles maneuver paired success over matched-only imitation with the same backbone, from 23.67% to 57.00%, while RFS rises from 6.32 to 7.10. Against the language-free control, maneuver and qualifier paired success rise from zero to 57.00% and 35.07%, respectively. RFS rises from 6.24 to 7.10, while output coverage falls from 98.98% to 97.70%. Paired success requires both requests to be fulfilled; missing outputs count as failures. Appendix C.4 details the evaluation sets.

Contribution beyond additional trajectory examples. Table 3 compares matched-only and same-scene imitation under the same imitation procedure and optimization budget. Adding admitted A/B teachers raises maneuver paired success from 23.67% to 39.50%, a gain of 15.83 percentage points. On the same expanded training set, Cartender further improves maneuver and qualifier paired success by 17.50 and 26.04 points, reaching 57.00% and 35.07%. Its RFS is 7.10 compared with 7.04 for same-scene imitation, while coverage increases from 94.78% to 97.70%. The full training procedure therefore improves instruction fulfillment beyond imitation on the additional examples.

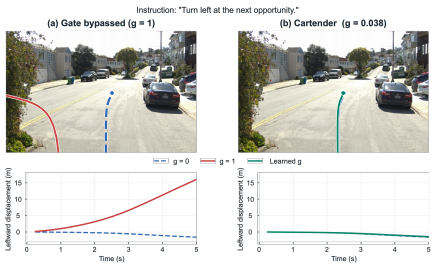
Compatibility and preference reward. Table 3 shows that adding compatibility to same-scene imitation raises maneuver and qualifier paired success from 39.50% and 9.03% to 45.83% and 12.85%, respectively, while RFS decreases from 7.04 to 6.95. The full method reaches 57.00% and 35.07%, with RFS increasing to 7.10. Removing compatibility from the full method reduces maneuver paired success by 15.33 percentage points to 41.67%, eliminates qualifier paired success, and lowers RFS from 7.10 to 7.00 and coverage from 97.70% to 94.40%. Replacing reward refinement with language-conditioned residual imitation, using the same initialization and update count, reduces maneuver and qualifier paired success by 7.00 and 30.56 points, respectively, and lowers RFS from 7.10 to 6.37. Both components therefore contribute to instruction fulfillment, with compatibility also supporting output availability.

Staged versus joint optimization. Both variants start from the same IL endpoint; the joint variant additionally retains the imitation loss during preference refinement. Table 3 shows that joint optimization and staged optimization both achieve 57.00% maneuver paired success. Staged optimization achieves 35.07% qualifier paired success versus 5.21% for joint optimization, and RFS of 7.10 versus 6.94. Staged optimization therefore favors qualifier fulfillment and trajectory quality without changing maneuver paired success in the reported runs.

GRPO versus exact expected-reward optimization. We compare GRPO with exact expected-reward optimization over the same finite candidate set, using the same reward and imitation anchor. Table 3 shows similar maneuver paired success: 57.00% for GRPO and 58.17% for exact expectation. GRPO achieves 35.07% qualifier paired success, compared with 4.86% for exact expectation, and RFS of 7.10 versus 7.00. This fixed-budget result favors GRPO on qualifier fulfillment, but does not isolate a general optimizer advantage.



(a) Same-scene instruction intervention.



(b) Compatibility gating under an unsupported instruction.

Figure 3: Cartender responds to a supported slow-down request under fixed scene inputs (left) and suppresses an unsupported turn through compatibility gating (right).

Table 3: Instruction control, trajectory quality, and output coverage. All methods use logged data and admitted A/B teachers except matched-only imitation. Imitation baselines share procedure and budget; exact expectation shares GRPO's candidates, reward, and imitation anchor. Paired success averages three seeds and requires both requests to succeed; missing outputs fail. Coverage measures output availability.

Method	Maneuver paired (%) ↑	Qualifier paired (%) ↑	RFS ↑	Coverage (%) ↑
Language-free control	0.00	0.00	6.24	98.98
Matched-only imitation	23.67	3.13	6.32	96.40
Same-scene imitation	39.50	9.03	7.04	94.78
Same-scene imitation + compatibility	45.83	12.85	6.95	97.70
Cartender	57.00	35.07	7.10	97.70
Cartender, no compatibility	41.67	0.00	7.00	94.40
Cartender, no preference reward	50.00	4.51	6.37	97.70
Joint IL + preference	57.00	5.21	6.94	97.70
Exact expected reward	58.17	4.86	7.00	97.70

Qualitative instruction interventions. In Fig. 3a, Qwen-Drive predictions nearly overlap for lane keeping and slowing down, whereas Cartender predicts a shorter trajectory for slowing down. In Fig. 3b, bypassing the gate produces pronounced leftward motion, while the learned gate keeps the prediction close to the $g = 0$ reference when the turn is unsupported within the evaluated horizon.

4.6 Behavior Under Infeasible Instructions

We test whether scene–language compatibility distinguishes unsupported instructions and reduces their influence on predicted behavior.

Compatibility separation. Mean compatibility scores are 0.771 for feasible instructions and 0.310 for infeasible instructions, with overlapping distributions. The score achieves an AUROC of 0.863 for identifying infeasible instructions. At $\theta = 0.5$, classifying low-compatibility instructions as infeasible yields 76.74% precision, 74.16% recall, and 75.43% F1. See Appendix D.1.

Behavioral suppression of unsupported instructions. With checkpoint, inputs, and random draws fixed, learned compatibility reduces infeasible execution from 53.36% with $g = 1$ to 14.18%, and drift from 3.612 to 0.337 m (Appendix D.3). The separate diagnostic in Table 9 also reports a cost: Feasible Follow falls from 61.6% to 43.8%, while Relation Handling rises from 54.1% to 64.8% (61.9% with a constant gate). These request-level metrics differ from paired success. Gating therefore trades feasible-request fulfillment for reduced unsupported execution in that diagnostic; it provides no explicit refusal.

5 Conclusion

Logged driving records one behavior even when several are admissible, so matched pairs provide neither a direct comparison among feasible behaviors nor evidence that language caused the selected one. Counterfactual Branching constructs same-scene alternatives with rule-based admission and post-hoc RFS evaluation. These relations supervise training, while fixed-scene interventions on recorded inputs use behavior criteria to test whether predictions follow the changed instruction.

Cartender generates multiple candidates, gates language conditioning by continuous scene–language compatibility, and ranks them with a GRPO-trained language residual. It achieves 57.00% maneuver paired success versus 23.67% for matched-only imitation; gating reduces unsupported-request following from 53.36% to 14.18%. Limitations are discussed in Appendix F.1.

AI USE STATEMENT

We used Qwen to propose alternative instructions, assess their scene applicability, and generate parameters for trajectory construction. Generative AI tools also assisted with data annotation, labeling, and manuscript language editing. All AI-assisted annotations were reviewed by the authors, and the final labels used in our experiments were checked for consistency and correctness. AI-assisted writing was limited to language editing and presentation; the research questions, methodology, experimental design, analysis, and conclusions were developed and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, annotations, and artifacts produced with the aid of generative AI.

REFERENCES

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation, 2025. URL https://arxiv.org/abs/2503.19755.

Kaiser Hamid, Can Cui, and Nade Liang. Icr-drive: Instruction counterfactual robustness for end-to-end language-driven autonomous driving, 2026. URL https://arxiv.org/abs/2604.05378.

Xiaoshen Han, Haoyu Xiong, Haonan Chen, Chaoqi Liu, Antonio Torralba, Yuke Zhu, and Yilun Du. B-spline policy: Accelerating manipulation policies via b-spline action representations, 2026. URL https://arxiv.org/abs/2607.09648.

Yuzhou Huang, Benjin Zhu, Hengtong Lu, Victor Shea-Jay Huang, Haiming Zhang, Wei Chen, Jifeng Dai, Yan Xie, and Hongsheng Li. Mindvla-u1: Vla beats va with unified streaming architecture for autonomous driving. *arXiv preprint arXiv:2605.12624*, 2026.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/2411.15139.

Ziheng Lu, Yingfeng Cai, Wei Dong, Hai Wang, and Long Chen. Malt-drive: Moment-aligned language-to-trajectory planning for autonomous driving. *CMES - Computer Modeling in Engineering and Sciences*, 148(2), 2026. ISSN 1526-1492. doi: https://doi.org/10.32604/cmes.2026.086003. URL https://www.sciencedirect.com/science/article/pii/S1526149226003760.

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)*, pp. 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)*, pp. 15120–15130. IEEE, 2024a.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024b. URL https://arxiv.org/abs/2402.03300.

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, 2025. URL https://arxiv.org/abs/2405.01533.

Xinyang Wang, Qian Liu, Wenjie Ding, Zhao Yang, Wei Li, Chang Liu, Bailin Li, Kun Zhan, Xianpeng Lang, and Wei Chen. Unifying language-action understanding and generation for autonomous driving. *arXiv preprint arXiv:2603.01441*, 2026a.

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, 2026b. URL https://arxiv.org/abs/2603.25740.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. Wod-e2e: Waymo open dataset for end-to-end driving in challenging long-tail scenarios, 2025. URL https://arxiv.org/abs/2510.26125.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Chieh-Chi Yang, Yu-Hsiang Chen, and Yi-Ting Chen. Nudgevad: Language-nudged end-to-end driving via film residuals, 2026. URL https://arxiv.org/abs/2605.24531.

Xin Zhou, Zongchuang Zhao, Zhibo Yang, Mingsheng Li, Humen Zhong, Shuai Bai, Du Chu, Ruizhe Chen, Zhaohai Li, Jun Tang, Qiuyue Wang, Mingkun Yang, Jiazhao Zhang, Dayiheng Liu, Dingkang Liang, and Xiang Bai. Qwen-drive-1.0: An initial step towards a vision-language foundation model for autonomous driving, 2026. URL https://arxiv.org/abs/2609.00111.

Zewei Zhou, Tianhui Cai, Seth Z. Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2506.13757.

Jialv Zou, Shaoyu Chen, Bencheng Liao, Zhiyu Zheng, Yuehao Song, Lefei Zhang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Diffusiondrivev2: Reinforcement learning-constrained truncated diffusion modeling in end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/2512.07745.

A Additional Related Work

Multi-Proposal Generation Mechanisms. Multi-proposal planners differ in how they represent and generate candidate trajectories. Anchor-based methods use a fixed vocabulary of motion primitives. DiffusionDrive and its successor instead draw multiple candidates from a truncated diffusion process and select among them with a learned scorer (Liao et al., 2025; Zou et al., 2025). These mechanisms provide candidate behaviors for downstream selection. Our use of multi-proposal generation concerns the supervision of that selection: same-scene alternatives allow candidates to be compared under a stated instruction.

Architectural Coupling in Driving VLAs. Driving VLA models realize the connection between language and action in different ways. LMDrive demonstrates direct language conditioning for closed-loop driving (Shao et al., 2024a). ORION connects vision–language reasoning with generative planning (Fu et al., 2025). AutoVLA combines reasoning and action generation with reinforcement fine-tuning (Zhou et al., 2025). MindVLA-U1 exposes a measurable language-to-action route for continuous trajectory generation (Huang et al., 2026). These architectural differences motivate examining instruction dependence through observable changes in predicted behavior, rather than inferring it solely from the presence of a language-conditioning pathway.

Representations and Supervision Sources. LinkVLA couples language and action through a shared discrete representation and an auxiliary action-understanding objective (Wang et al., 2026a). Drive My Way models long-term driving preferences together with real-time language guidance (Wang et al., 2026b). These approaches complement efforts to enrich the instruction–behavior supervision itself. SimLingo's Action Dreaming associates instructions with alternative driving behaviors generated through simulation (Renz et al., 2025), while OmniDrive connects language with counterfactual descriptions of alternative action outcomes (Wang et al., 2025). The relevant distinction for our setting is the supervision supplied: shared representations, preference information, counterfactual descriptions, simulator-generated trajectories, and trajectories constructed for real recordings provide different forms of evidence. Our construction and admission procedure for the last category is described in Sec. 3.1.

B Reproducibility and Implementation Details

B.1 Data, Backbone, and Prediction Setup

Our data pool contains 10,336 training scenes and 4,580 validation scenes from the Waymo Open Dataset End-to-End (WOD-E2E). Here, a scene denotes a keyframe identified by its episode and anchor timestamp, rather than an entire episode. RFS evaluation uses the corresponding Waymo-provided rated reference trajectories at each evaluated keyframe. The main trajectory-quality evaluation uses a frozen subset of 1,200 validation scenes (Appendix C.4). Unless otherwise stated, Cartender and its language-free control use the same frozen U1 scene features, a five-second prediction horizon, and $K = 24$ trajectory candidates. Future trajectories are represented by B-spline coefficients and decoded to planar ego positions at 0.25 s intervals over the five-second prediction horizon ($T = 20$).

The released-policy diagnostics use frozen checkpoints of Qwen-Drive, SimLingo, and LMDrive. Qwen-Drive is evaluated on WOD-E2E through the instruction interface added for this study. SimLingo and LMDrive are evaluated under their respective CARLA protocols. Qwen-Drive-1.0 uses the released Qwen-Drive-1.0-4B weight repository, which provides both `planner-sft` and `planner-rl`, with official code at commit `28091c1`. We evaluate the `planner-rl` checkpoint. SimLingo uses `simlingo/checkpoints/epoch=013.ckpt/pytorch_model.pt` from RenzKa/simlingo with official code at commit `743b243`. LMDrive uses `llava-v1.5-checkpoint.pth` from LMDrive-llava-v1.5-7b-v1.0, together with the original LLaVA-v1.5-7B weights and the released R50 vision encoder; its official code is fixed at commit `308d8da`.

The language-free, matched-only, and expanded-generator controls are trained separately while sharing the same encoder, trajectory codec, and initialization source. The full method reuses the expanded generator. Compatibility imitation, the full method, and the exact expected-reward control

share the corresponding generator, compatibility model, candidate sets, and post-imitation selector. Reward ablations retrain only the language residual affected by the ablation. Released baselines retain their frozen public weights and native input interfaces. Each run records the weight hash, code revision, realized inference configuration, and data manifest.

B.2 COUNTERFACTUAL BRANCH CONSTRUCTION

The construction pipeline is adapted from the data-generation setup of MindVLA-U1 (Huang et al., 2026) and repurposes its scene and trajectory representations for same-scene instruction supervision.

Keyframes are selected around turning, lane-change, and acceleration events. When no such event occurs for an extended interval, fixed-rate sampling supplies additional keyframes. At each selected keyframe, a locally deployed Qwen3.8-27B model (Yang et al., 2025) receives up to nine front-camera frames sampled at 2 Hz over the four seconds preceding and including the anchor. Missing frame indices are omitted rather than replaced with images from another view.

Construction has two Qwen stages. The first evaluates the 15-intent vocabulary and returns a score and rationale for each intent. The pipeline retains at most three feasible instructions with confidence greater than 0.5 and selects three infeasible instructions. These generation-time scores are used only to construct the supervision and are distinct from the language-intent confidence c_ℓ in Eq. 1. For each selected feasible instruction, the second stage produces four route-relative parameterizations with predefined roles: A (BETTER) is the best-supported safe execution, B (ALTERNATIVE) is a distinct but still reasonable execution, C (CONSERVATIVE) travels less far or preserves a larger safety margin, and D (WORSE) is deliberately poorer and may violate the intent or a safety constraint while remaining continuous. Qwen does not emit planar waypoints or B-spline control points. Each output instead specifies `progress_ratio`, `lateral_offset_m`, `speed_profile`, and `return_to_reference`, together with a rationale. Before trajectory construction, deterministic rules correct the lateral-offset direction and magnitude and decide whether the branch should return to the reference route.

Route-relative trajectory construction. The recorded future defines the reference route. We prepend the current position, remove duplicate points, parameterize the route by cumulative arc length, and interpolate its x and y coordinates independently with a cubic spline. When the recorded future moves less than 0.5 m, the recent motion direction is used to construct the reference line instead.

For normalized time $u \in [0, 1]$, the reference progress profile along a recorded route of length L is

$$s_{\mathrm{ref}}(u) = L\,\rho\,h(u), \qquad h(u) = \begin{cases} u, & \text{maintain,} \\ u^{1.35}, & \text{accelerate,} \\ 1-(1-u)^{1.65}, & \text{decelerate,} \\ 1-(1-u)^{2.4}, & \text{yield/stop,} \end{cases} \tag{6}$$

where ρ is `progress_ratio`. A smoothstep profile applies the predicted lateral offset along the route normal. If `return_to_reference` is enabled, a $\sin^2(\pi v)$ profile instead returns the branch to the reference route, where v is normalized route progress.

These profiles specify the reference shape rather than an exact constraint on the final trajectory. Using 20 construction nodes including $t = 0$ and $t = 5$ s, the constructor fits a spline subject to

$$p(0) = (0, 0), \qquad \dot{p}(0) = u_{\mathrm{ego}}, \tag{7}$$

where u_{ego} is the observed velocity in the anchor ego frame and the derivative is with respect to physical time. Subsequent smoothing uses a second-difference penalty of 4.0, preserves these initial boundaries and the terminal position, and limits each node's displacement from the boundary-constrained construction to 0.15 m. This limit applies to smoothing, not to the total adjustment from the reference profile. Proposals that cannot satisfy the construction constraints are rejected. The boundary-preserving cubic B-spline representation is densified by five samples per segment (96 points when no nodes are removed). Geometric sampling uses chord length, while time evaluation retains the node-time correspondence and initial-velocity boundary; the densified geometric samples are not treated as uniformly spaced in time. We evaluate the resulting curve at $0.25, 0.5, \ldots, 5.0$ s

to obtain the 20-step teacher trajectory used downstream. This is a deterministic geometric construction from Qwen parameters and the recorded future; no learned planner is called. Encoding the resulting teacher into the B-spline coefficient space used for flow matching is a separate training-representation step. Encoding and decoding preserve the initial position and velocity boundaries in Eq. 7; decoded trajectories still undergo the full admission checks.

Both Qwen stages request JSON-schema output with `strict=false`, followed by application-level validation. Invalid outputs are regenerated with the validation error appended to the prompt. Intent assessment uses temperatures $0.2, 0.4, 0.6$ across successive attempts and a 4,096-token output budget. A/B/C/D parameter generation uses temperatures $0.25, 0.45, 0.65$ and a 2,048-token output budget. Both stages set `enable_thinking=false` and `stream=false`; they do not explicitly set top-p, top-k, frequency penalty, or a sampling seed.

The intent validator requires all 15 classes without duplicates, scores in $[0, 1]$, and complete rationales. The parameter validator checks the expected number and roles of outputs, finite values, `progress_ratio`$\in [0.15, 1]$, `lateral_offset_m`$\in [-4, 4]$, and a valid speed profile. These checks enforce the generation contract; they are separate from branch admission in Sec. B.3. Keyframes are selected first at salient changes in acceleration, turning, lateral motion, and stop–start state. Stable intervals not covered by these events are then supplemented at a maximum anchor spacing of 2.0 s; this fixed-rate supplementation also applies locally between events rather than only to recordings with no detected event.

Across 2,487 source scenes, this procedure yields 3,603 feasible instruction conditions and 14,412 generated proposals. Rule-based admission retains 8,974 proposals (62.27%). Only admitted A and B proposals enter the downstream teacher set; C and D are not teacher targets. All 3,603 A proposals pass admission.

This construction varies progress and lateral offset relative to the recorded route; it does not enumerate alternative road-topology branches. In particular, it does not establish that a straight reference route can support a genuine turn onto another road.

B.3 Admission Checks and Relation Labels

Candidate branches follow a fixed admission order: source, format, and coordinate checks; semantic-conflict, initial-state, and motion checks on the raw trajectory; codec reconstruction error; and then the semantic, initial-state, and motion checks again on the decoded trajectory. Every A and B proposal follows the same procedure. The motion limits must hold at every evaluated step, and the reconstruction limit is

$$\begin{aligned} -4.05 \le a_{\mathrm{lon}} \le 2.40\ \mathrm{m/s^2}, \qquad & |a_{\mathrm{lat}}| \le 4.89\ \mathrm{m/s^2}, \\ |\omega| \le 0.95\ \mathrm{rad/s}, \qquad & |\dot{\omega}| \le 1.93\ \mathrm{rad/s^2}, \\ |j_{\mathrm{lon}}| \le 8.37\ \mathrm{m/s^3}, \qquad & \frac{1}{T}\sum_{t=1}^{T} \|\tau_t - \hat{\tau}_t\|_2 < 0.5\ \mathrm{m}, \end{aligned} \tag{8}$$

where $\hat{\tau}$ is the decoded reconstruction and ω is the course-rate proxy used by the motion checks.

Admission here denotes passage through the stated checks, not a certificate of collision avoidance or drivable-area containment. No collision or drivable-area test is specified in this protocol; direction-conflict rejection also does not establish maneuver completion.

Initial-state reachability. Coordinates are expressed in the anchor ego frame, with $p(0) = (0, 0)$. When an upstream proposal explicitly includes its $t = 0$ point, it must satisfy

$$\|p(0)\|_2 \le 10^{-9}\ \mathrm{m}. \tag{9}$$

The formal 20-point trajectory begins at 0.25 s, so this constraint is not applied to the first future point p_1. The representation preserves both the initial position and observed velocity, but these boundary conditions do not replace the first-interval reachability and subsequent motion checks.

Let the observed planar ego velocity be transformed into the anchor frame as

$$u_0 = R(-\psi_{\mathrm{ego}}) v_{\mathrm{observed}}, \qquad v_0 = \|u_0\|_2. \tag{10}$$

With $\Delta = 0.25\,\mathrm{s}$, the initial constant-acceleration reachability estimate is

$$A_0 = \frac{2(p_1 - u_0\Delta)}{\Delta^2}. \tag{11}$$

When $\|u_0\| > 10^{-6}\,\mathrm{m/s}$, A_0 is projected onto the direction of u_0 and its left normal; otherwise the anchor ego forward and left axes are used. Admission requires

$$-4.05 - \epsilon \le A_{0,\parallel} \le 2.40 + \epsilon, \qquad |A_{0,\perp}| \le 4.89 + \epsilon, \qquad \epsilon = 10^{-6}\ \mathrm{m/s^2}. \tag{12}$$

Equivalently, for the first-interval average velocity $\bar{u}_1 = p_1/\Delta$,

$$(\bar{u}_1 - u_0)_{\parallel} \in [-0.50625, 0.30000]\ \mathrm{m/s}, \qquad |(\bar{u}_1 - u_0)_{\perp}| \le 0.61125\ \mathrm{m/s}. \tag{13}$$

Missing observed velocity is a data error. We do not impose a separate initial heading-angle tolerance: the direction derived from p_1 is a first-interval course rather than the body heading at $t = 0$. Initial lateral reachability provides the connection check, and subsequent course-rate and course-acceleration limits apply only where the supporting speeds exceed $0.10\,\mathrm{m/s}$. The first-step jump is therefore checked by Eq. 11, without an additional fixed distance threshold.

Direction-conflict rejection. The semantic checks reject only clear motion opposite to the request:

Table 4: Terminal direction-conflict checks used for admission. A value exactly on a boundary passes.

Requested intent	Reject when
Left turn or left lane change	$y_{20} < -1.0\,\mathrm{m}$
Right turn or right lane change	$y_{20} > +1.0\,\mathrm{m}$
Acceleration or deceleration	$x_{20} < -0.5\,\mathrm{m}$
Merge, obstacle avoidance, or overtake	No additional direction rule

These conditions are not completion tests. In particular, admission does not require a signed speed change for acceleration or deceleration, terminal lateral displacement does not establish entry into a target lane, and avoidance or overtaking may return to the original lateral position. Those semantics are evaluated by the reward and independent action evaluator.

Waiting additionally requires a maximum displacement of $0.10\,\mathrm{m}$ over the horizon and a maximum speed of $0.20\,\mathrm{m/s}$ over the final four intervals. Raw and decoded trajectories use the same observed velocity, coordinate frame, time grid, functions, thresholds, low-speed mask, and tolerances; failure after decoding rejects the proposal even when the raw trajectory passes. The continuous raw proposal constructor separately enforces $\min_t s'(t) \ge -10^{-8}\,\mathrm{m/s}$, but decoded trajectories do not have a complete, independent no-reversing check. We therefore do not claim that terminal x checks establish forward motion throughout the horizon.

Feasible relations retain their admitted A and B proposals as a teacher set. All 3,603 A proposals pass these checks. Infeasible instructions contribute only a binary scene–instruction applicability label and never contribute a counterfactual trajectory target. Rater Feedback Score (RFS) is computed only after rule-based admission and is not used to select branches. Admission pass rates are diagnostic statistics and are not targets for tuning these thresholds.

B.4 Training Configuration

Training proceeds in three policy stages. Their configurations are summarized in Table 5. The 600 GRPO optimizer steps comprise 150 rollout rounds with four PPO epochs per round. The compatibility model, generator, and base scorer are frozen during this final preference stage; only the language-dependent residual is updated. Selector comparisons use three seeds with a shared generator checkpoint and candidate sets.

Table 5: Training configuration. Batch sizes are global.

Stage	Optimizer	Learning rate	Global batch	Optimizer steps
Generator adaptation	AdamW	5×10^{-5}	32 relations, one teacher each	2,000
Selector imitation	AdamW	3×10^{-4}	48 groups, 24 candidates each	1,200
GRPO language residual	AdamW	1×10^{-4}	48 groups, 8 sampled indices each	600

All three stages use AdamW with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, weight decay 0.01, gradient-norm clipping at 1.0, a constant learning rate, and FP32 precision. We use the checkpoint at the fixed final update rather than selecting the best checkpoint on the evaluation set. A pool of 32 NVIDIA A800-SXM4-80GB GPUs is used for data generation and processing. The scene-support, generator-adaptation, selector-imitation, and GRPO training runs each use eight GPUs. The global batch sizes in Table 5 are per run and are not multiplied by the size of the GPU pool.

The scene-support model is trained first and then frozen. It uses AdamW for 200 full-batch updates over all labeled training scenes, with learning rate 2×10^{-4}, weight decay 0.01, gradient clipping at 1.0, a constant schedule, and FP32 precision. Intents are weighted equally. Within each intent, the mean positive and mean negative binary-cross-entropy terms receive weight 0.5 each; if only one labeled class is present, its term receives weight one, and unlabeled entries are ignored.

The reward defined in Sec. 3.2 uses trajectory-derived motion events to locate informative times and instruction-dependent agreement to score them. Teacher distance, RFS, and the A/B/C/D proposal roles do not enter this reward. The reward definition is held fixed across the main comparisons.

B.5 Teacher Targets and Reward Optimization

For each feasible relation (s, ℓ), the base selector is initialized from a listwise target distribution q_i over the unmasked candidates. For the decoded admitted A/B teacher set $\mathcal{H}(s, \ell)$, candidate i has minimum teacher ADE

$$d_i = \min_{h \in \mathcal{H}(s,\ell)} \frac{1}{20} \sum_{t=1}^{20} \|\tau_{i,t} - h_t\|_2. \tag{14}$$

Over the candidates $\mathcal{A}$ that pass the hard mask, the target is

$$q_i = \frac{\exp\left(-\frac{d_i}{0.5\,\mathrm{m}}\right)}{\sum_{j \in \mathcal{A}} \exp\left(-\frac{d_j}{0.5\,\mathrm{m}}\right)}, \qquad i \in \mathcal{A}, \tag{15}$$

and $q_i = 0$ for masked candidates. Relations with no admitted teacher or no surviving candidate do not enter imitation training. This stage minimizes Eq. 4 while the language residual remains fixed at zero.

The no-preference control retains the full language-conditioned selector. It shares the generator, compatibility model, candidate sets, teacher sets, and base initialization with the full method. After base initialization, the base remains frozen in both variants, and the same language residual is trained from the same starting point for the same number of optimizer updates. The no-preference control trains this residual by imitation of the admitted A/B teachers for the current instruction, reusing the target distribution q_i in Eq. 15 but optimizing the residual through the full selector with the base fixed. The full method instead trains the residual with GRPO using the instruction-agreement reward. Thus, this control replaces reward-based refinement with language-conditioned imitation; it does not remove the residual or its instruction input.

The preference stage evaluates each candidate with the frozen key-event reward in Eq. 5. The event weights are computed from the candidate trajectory alone and therefore do not depend on the instruction.

Motion signals. Let $p_i = (x_i, y_i)$ denote the ith future point on the $\Delta t = 0.25\,\mathrm{s}$ grid, and prepend the current position $p_0 = (0, 0)$. Interval velocity is right-aligned to the future point:

$$\mathbf{v}_i = \frac{p_i - p_{i-1}}{\Delta t}, \qquad v_i = \|\mathbf{v}_i\|_2. \tag{16}$$

For a sampled scalar signal f, Df uses centered differences at interior points and first-order one-sided differences at the two ends, equivalent to `np.gradient(f, 0.25, edge_order=1)`. Let S denote one pass of the kernel $[0.25, 0.50, 0.25]$, with endpoint values replicated at the boundary. The evaluator uses

$$\alpha_i = S(Dv)_i, \qquad \omega_i = S(D\psi)_i, \qquad v_i^{\mathrm{lat}} = S(v_y)_i, \tag{17}$$

where $\psi = \mathrm{unwrap}(\mathrm{atan2}(v_y, v_x))$. Thus α_i is the derivative of speed magnitude, rather than $D(v_x)$, so constant-speed turning is not labeled as deceleration.

All quantities are expressed in the fixed ego frame at the anchor time, with x forward and y left. The frame does not rotate with the predicted path and does not use the recorded route. Heading is invalid below $0.10\,\mathrm{m/s}$; ω_i is evaluated only when all samples supporting its differencing and smoothing stencils are valid, and no angular difference crosses a stopped interval. Accordingly, ω_i is a course-rate proxy rather than measured yaw rate. Fixed-frame lateral motion on a curved road is not, by itself, evidence of true lane membership or a lane change.

The event detector considers acceleration and deceleration, turning, lateral motion, and stop–start transitions. Acceleration or deceleration is significant when $|\alpha_i| \geq 0.5\,\mathrm{m/s^2}$; turning uses $|\omega_i| \geq 0.10\,\mathrm{rad/s}$; and lateral motion uses $|v_i^{\mathrm{lat}}| \geq 0.30\,\mathrm{m/s}$. Stop–start detection uses hysteresis, with low- and high-speed thresholds of $0.20\,\mathrm{m/s}$ and $0.50\,\mathrm{m/s}$, respectively.

For each contiguous motion event, the detector retains its onset, peak, and end. Whenever two retained times are more than $0.5\,\mathrm{s}$ apart, temporal filler points are inserted so that the remaining gap is at most $0.5\,\mathrm{s}$. If no motion event is detected, the entire horizon is sampled at intervals of at most $0.5\,\mathrm{s}$. The first and last future points are always retained, and the number of retained points is not capped below the $T = 20$ trajectory steps.

Let $\tilde{w}_t$ denote the unnormalized weight of a retained point. Motion-event points receive weight 2, temporal fillers and forced endpoints receive weight 1, and unselected points receive weight 0. A point with multiple roles is counted once, with the motion-event weight taking precedence. We then set

$$w_t(\tau) = \frac{\tilde{w}_t(\tau)}{\sum_{u=1}^{T} \tilde{w}_u(\tau)}. \tag{18}$$

Instruction agreement. All inputs to the following scoring functions are converted to their numerical values in SI units. Let

$$\rho(z) = \frac{1}{1 + e^{-z}}, \qquad G(x; s) = \exp\left(-\frac{x^2}{2s^2}\right), \qquad I(x; L, U, s) = \exp\left(-\frac{\mathrm{dist}(x, [L, U])^2}{2s^2}\right). \tag{19}$$

An infinite endpoint denotes the corresponding half-line. The agreement score decomposes as

$$a_t(\tau, \ell) = m_t\, c_{\mathrm{mag}}\, c_{\mathrm{time}}\, c_{\mathrm{comp}}, \tag{20}$$

where the factors evaluate action and phase, magnitude, timing, and completion. A factor not requested by the instruction is one.

Let $v_{\mathrm{ref}} = v_1$ be the candidate's first-interval speed, not an observed instantaneous speed, and let $\delta v_i = v_i - v_{\mathrm{ref}}$. For directional lateral actions, $q = +1$ denotes left and $q = -1$ right. We use $Y_i = q y_i$ and $V_i = q v_i^{\mathrm{lat}}$ when a direction is requested, and their absolute values otherwise; $E_i = \max(0, Y_1, \ldots, Y_i)$. Let h_i be the most recent reliable course heading and χ_i indicate whether such a heading has been observed. We set $H_i(b) = \exp(-|h_i|/b)$ when $\chi_i = 1$ and $H_i(b) = 1$ otherwise. The base action scores are listed in Table 6.

For an action with a requested onset window $[L, U]$, the pre-action maintenance score is

$$P_i^{\mathrm{long}} = G(\delta v_i; 0.5) I(|\alpha_i|; 0, 0.25, 0.25), \tag{21}$$

$$P_i^{\mathrm{lat}} = G(\delta v_i; 0.5) G(y_i; 0.3) G(v_i^{\mathrm{lat}}; 0.15). \tag{22}$$

Table 6: Trajectory-observable base action scores. Left/right categories are expanded using $q \in \{+1, -1\}$.

Intent	Base score m_i^0
Lane keeping	$\exp(-\lvert y_i\rvert/1.5)H_i(0.35)$
Following	$0.65\exp(-\lvert y_i\rvert/1.8)H_i(0.4) + 0.35\exp(-\lvert\alpha_i\rvert/2)$
Accelerating	$0.65\rho((\delta v_i - 0.5)/1.5) + 0.35\rho(\alpha_i)$
Decelerating	$0.65\rho((-\delta v_i - 0.5)/1.5) + 0.35\rho(-\alpha_i)$
Stopping or yielding	$0.75\rho(2(1.2 - v_i)) + 0.25\rho(-1.5\alpha_i)$
Waiting	$I(v_i; 0, 0.2, 0.3)I(\lVert p_i\rVert_2; 0, 0.1, 0.1)$
Starting	$0.65\rho(1.5(v_i - 1)) + 0.35\rho(\alpha_i)$
Turning left/right	$0.65\chi_i\rho(8(qh_i - 0.08)) + 0.35\rho(1.5(qy_i - 0.5))$
Lane change left/right	$0.65\rho(2(qy_i - 0.7)) + 0.35\rho(2qv_i^{\text{lat}})$
Merging	$0.70\rho(2(Y_i - 0.7)) + 0.30\rho(2(V_i - 0.25))$
Avoiding an obstacle	$0.60\rho(2(E_i - 0.8)) + 0.40\rho(2(V_i - 0.30))$
Overtaking	$0.50\rho(2(Y_i - 0.8)) + 0.50\rho((\delta v_i - 0.4)/1.2)$

We use P_i^{long} for acceleration, deceleration, stopping, yielding, and starting, and P_i^{lat} for turning, lane change, merging, obstacle avoidance, and overtaking. The phase-aware action factor is

$$m_i = \begin{cases} P_i, & t_i < L, \\ \max(P_i, m_i^0), & L \le t_i \le U, \\ m_i^0, & t_i > U. \end{cases} \tag{23}$$

Without an onset request, $m_i = m_i^0$. Lane keeping, following, and waiting are treated as sustained states and do not accept an onset qualifier. The maintenance term credits correct behavior before a delayed action rather than treating that phase as disagreement.

Let $\bar{v}_{\text{end}}$ be the mean of the final four interval speeds. We define $\Delta v = \bar{v}_{\text{end}} - v_{\text{ref}}$ for acceleration and $\Delta v = v_{\text{ref}} - \bar{v}_{\text{end}}$ for deceleration. Magnitude agreement is one when no magnitude is requested. Otherwise,

$$c_{\text{mag}}^{\text{gentle}} = I(\Delta v; 0.5, 2.0, 0.35), \tag{24}$$

$$c_{\text{mag}}^{\text{strong}} = I(\Delta v; 2.0, \infty, 0.35), \tag{25}$$

$$c_{\text{mag}}^{d} = I(\Delta v; \max(0, d - 0.25), d + 0.25, 0.35), \tag{26}$$

with all four arguments measured in $\mathrm{m/s}$. These ranges apply to acceleration and deceleration; they are not reused for, for example, a gentle turn. For these two longitudinal actions only, *gentle*, *gently*, *gradual*, and *gradually* are normalized to the same *gradual* category. We do not map *smoothly* to this category.

Timing agreement is

$$c_{\text{time}} = I(t_{\text{onset}}; L, U, 0.35\,\text{s}). \tag{27}$$

The early window is $[0, 1.5]$ s and the late window is $[2.5, 3.5]$ s. No timing request gives $c_{\text{time}} = 1$; a requested but undetected action gives zero. Onset is detected on the complete 20-step signal before key-event subsampling, using sign-specific thresholds of $0.10\,\mathrm{m/s^2}$ for acceleration or deceleration, $0.03\,\mathrm{rad/s}$ for turning, and $0.10\,\mathrm{m/s}$ for lateral motion. These lower thresholds are used only for instruction agreement; they do not alter the significant-event thresholds that define w_t.

For the relevant signed signal, let b_i indicate that the signal is valid and meets its onset threshold. The onset is the first sample of the earliest pair of adjacent active samples:

$$k = \min\{i \in \{1, \ldots, T-1\} : b_i \wedge b_{i+1}\}, \qquad t_{\text{onset}} = k\Delta t. \tag{28}$$

Samples equal to the threshold count as active. A below-threshold, opposite-direction, or invalid sample interrupts the pair; invalid samples are not removed and reconnected. We do not interpolate the threshold crossing. If several active segments occur, the earliest one is used regardless of whether the instruction requests an early or late action. The second sample confirms the onset but does not move it, and a single active final sample does not define an onset. Because adjacent samples are 0.25 s apart, this rule is not described as requiring 0.5 s of activity.

Acceleration uses $\alpha_i \geq 0.10\,\mathrm{m/s^2}$; deceleration, stopping, and yielding use $\alpha_i \leq -0.10\,\mathrm{m/s^2}$; turning uses $q\omega_i \geq 0.03\,\mathrm{rad/s}$; lane change uses $qv_i^{\mathrm{lat}} \geq 0.10\,\mathrm{m/s}$; and merging and obstacle avoidance use $V_i \geq 0.10\,\mathrm{m/s}$. Starting uses the stop–start hysteresis state and requires two adjacent samples at or above $0.50\,\mathrm{m/s}$ after a stopped state has been observed, with the onset assigned to the first sample. Overtaking uses the later of the first acceleration and lateral-motion onsets and is undetected if either is absent.

This reward onset is distinct from the diagnostic onset in Sec. E. The latter is the first single interval satisfying $v_i \leq v_{\mathrm{obs}} - 0.5\,\mathrm{m/s}$; it uses the observed initial speed and does not require two adjacent samples.

The main completion factors are

$$c_{\mathrm{stop}} = I\left(\max_{i=T-3}^{T} v_i; 0, 0.2\,\mathrm{m/s}, 0.3\,\mathrm{m/s}\right), \tag{29}$$

$$c_{\mathrm{wait}} = c_{\mathrm{stop}} I\left(\max_i \|p_i - p_0\|_2; 0, 0.1\,\mathrm{m}, 0.1\,\mathrm{m}\right), \tag{30}$$

$$c_{\mathrm{start}} = I\left(\frac{1}{4}\sum_{i=T-3}^{T} v_i; 0.5\,\mathrm{m/s}, \infty, 0.35\,\mathrm{m/s}\right), \tag{31}$$

$$c_{\mathrm{stable}} = I\left(\frac{1}{4}\sum_{i=T-3}^{T} |v_i^{\mathrm{lat}}|; 0, 0.15\,\mathrm{m/s}, 0.30\,\mathrm{m/s}\right), \tag{32}$$

$$c_{\mathrm{return}} = \exp\left[-\frac{1}{2}\left(\frac{y_T}{0.5\,\mathrm{m}}\right)^2\right] c_{\mathrm{stable}}. \tag{33}$$

The completion factor is c_{stop} for stopping, c_{wait} for waiting, c_{start} for starting, and c_{stable} for lane-change and merge requests. Obstacle avoidance and overtaking use c_{return} only when the instruction explicitly requests a return to the initial lateral position; otherwise their completion factor is one. Yielding does not require a full stop. The terminal-stability factor does not claim to verify true lane membership, and returning to the anchor-frame lateral coordinate does not establish a return to a curved route or lane.

Before training, the instruction manifest is scanned for unsupported intents, qualifiers, and combinations. Any unsupported request is reported with its relation identifier and original text and prevents training updates until it is resolved. It is neither assigned zero reward nor silently dropped, since the former confuses an undefined evaluator with failed execution and the latter changes the training set. Thus an unrequested factor equals one, a supported but unexecuted request is penalized by its defined factor, and an unsupported request is an error.

These scores measure trajectory motion rather than full interaction semantics. In particular, a trajectory alone cannot identify which actor is followed or yielded to, nor establish that a merge, avoidance maneuver, or overtake succeeds in the scene. We therefore interpret these terms as motion-shape agreement rather than complete interaction verification.

Neither teacher proximity, RFS, nor the A/B/C/D generation role contributes to a_t. The normalized weighted sum in Eq. 5 is used directly; the earlier fixed event–uniform–terminal mixture is not used.

Within each scene–instruction group, GRPO computes relative advantages over the retained candidate set. The update is weighted by $g(s,\ell)^2$, while low-compatibility groups are anchored more strongly to the frozen imitation policy. Selector logits use temperature 1.5. For each group, eight candidate indices are sampled with replacement, rewards are standardized with a standard-deviation floor of 10^{-3}, and the PPO ratio is clipped to $[0.8, 1.2]$. With S_g denoting the resulting clipped surrogate, the group compatibility $g_g = g(s_g, \ell_g)$, and the frozen post-imitation policy π_{ref}, the loss is

$$\mathcal{L}_{\mathrm{GRPO}} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}}\left[-g_g^2 S_g + \left(0.03 + 0.20(1-g_g^2)\right) D_{\mathrm{KL}}(\pi_{\mathrm{ref}}^g \| \pi_\theta^g) - 0.01\,\mathcal{H}(\pi_\theta^g)\right]. \tag{34}$$

Each rollout batch is reused for four PPO epochs. The reference policy is fixed at the end of imitation training, and GRPO updates only the language residual. For the exact expected-reward control, the same frozen candidate set, reward, post-imitation starting point, reference policy, and compatibility-dependent anchor are used. It evaluates the reward expectation exactly over all unmasked candidates

instead of estimating a clipped group-relative surrogate. The KL, anchor, and entropy terms are otherwise unchanged, and no PPO ratio clipping is applied. This control uses AdamW with learning rate 10^{-4}, global batch 48, and 600 optimizer updates, and reports the checkpoint at the fixed final update.

B.6 Inference and Empty-Candidate Handling

At inference, Cartender draws $K = 24$ candidate trajectories, masks candidates that violate the available kinematic or comfort constraints, and returns the highest-scoring remaining trajectory under Eq. 3. Teacher trajectories and reward evaluation are not used at inference. If every candidate is masked, the system sets the output flag to false and the selected index to -1, returns no trajectory, and records `all_candidates_masked` as the reason. This missing output is counted as a failure in success metrics and is reflected separately in output coverage.

C Evaluation Protocols

C.1 Instruction Removal and Shuffling

Instruction removal replaces the original instruction with an empty input while holding the scene and sampling setup fixed. We report the mean pointwise displacement between the original and empty-instruction predictions over the prediction horizon. The original, empty-instruction, and cross-scene-shuffled requests use identical non-language inputs and sampling noise. A missing output counts as a failure for action success; prediction-to-prediction displacement is computed only when both outputs are valid, and its coverage is reported separately. The same fixed manifest of 500 scene–instruction queries is used for all three conditions within each model. Thus original and empty inputs require 1,000 inference calls per seed, and including the shuffled condition gives 1,500 calls per seed.

Instruction shuffling permutes instructions across scenes while holding the scene inputs and action targets fixed. Where a released implementation exposes its native action objective, we report the shuffled-minus-original change in that objective. SimLingo contributes its route and speed-waypoint losses, LMDrive contributes waypoint L1 loss, and Qwen-Drive is omitted because its public release does not expose the corresponding native-loss evaluation.

C.2 Instruction Swap Evaluation Details

Swapping compares two feasible instructions under the same scene input, measuring trajectory displacement and instruction success.

We construct the diagnostic set from real driving recordings. Each pair reuses one input snapshot s exactly and changes only the instruction from ℓ_1 to ℓ_2. The episode, anchor time, camera frames, ego state and history, navigation input, map and object information exposed to the model, and initial memory or cache are identical. Snapshot identity is verified by content hash and the timestamp difference is zero; no approximate scene retrieval is used. Unlike the training relational supervision, this evaluation contains no synthesized trajectory reference. Request fulfillment is decided by road- and action-based criteria defined separately from the training reward.

Maneuver response accuracy. For each request $q = (s, \ell)$, an independent evaluator $A(s, \hat{\tau}_q, \ell)$ determines whether the predicted trajectory executes the requested maneuver.

Road- and action-based criteria. We evaluate a five-second trajectory at $\Delta t = 0.25\,\mathrm{s}$ intervals. The initial speed v_0 is observed, not inferred from the first predicted interval. With $\mathbf{p}_0$ the current position, define

$$v_i = \frac{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|_2}{\Delta t}, \qquad v_{\mathrm{end}} = \frac{1}{4}\sum_{i=17}^{20} v_i. \tag{35}$$

Thus v_{end} averages the final second, $(4, 5]\,\mathrm{s}$. Table 7 defines the ten intent criteria.

The final 0.5 s means $[4.5, 5.0]\,\mathrm{s}$, with target-lane occupancy maintained through the end of prediction. If the trajectory stops near the end, direction is assessed using the last two consecutive

Table 7: Independent maneuver-completion criteria. Road occupancy uses the vehicle footprint with a 0.2 m boundary tolerance. Strict paired success additionally requires valid outputs and the initial-state, kinematic, and comfort checks for both requests.

Intent	Completion criterion
Lane keeping	Remain in the current lane and designated straight corridor throughout, with net forward progress $\geq$ 1 m. For keep/turn pairs, also pass the straight-exit cross-section by at least 2 m. When paired with acceleration, deceleration, or stopping, the instruction also requires $\lvert v_{\text{end}} - v_0 \rvert \leq 0.25$ m/s.
Left turn	Follow the left-turn connector into the designated left exit and progress at least 2 m beyond its cross-section; remain in the target lane over the final 0.5 s, with motion direction within 20° of the target-lane tangent.
Right turn	Apply the left-turn criterion to the designated right connector and exit, with the same 2 m, 0.5 s, and 20° requirements.
Left lane change	Cross the shared boundary in a permitted lane-change region into the left adjacent same-direction lane; the vehicle footprint must remain in the target lane over the final 0.5 s, with motion direction within 20° of its tangent.
Right lane change	Apply the left-lane-change criterion to the right adjacent same-direction lane, with the same final-hold and direction requirements.
Acceleration	$v_{\text{end}} - v_0 \geq 0.5$ m/s while remaining in the designated driving corridor.
Deceleration	$v_0 - v_{\text{end}} \geq 0.5$ m/s while remaining in the designated driving corridor; deceleration to a stop is allowed.
Stopping	Eligible inputs have $v_0 \geq 2$ m/s. Each of the final four interval speeds is ≤ 0.2 m/s, and the vehicle remains in the annotated permitted stopping region throughout the final second.
Waiting	$v_0 \leq 0.2$ m/s; maximum distance from the start is ≤ 0.1 m throughout; each of the final four interval speeds is ≤ 0.2 m/s, and remaining stationary is permitted in that region.
Starting	$v_0 \leq 0.2$ m/s, $v_{\text{end}} \geq 1.0$ m/s, and net forward progress along a permitted corridor is ≥ 1 m, with motion direction within 20° of the corridor tangent.

valid motion intervals after entry into the target region, each with speed ≥ 0.1 m/s. This is motion direction, not vehicle-body heading. Lane changes cannot be established from ego-frame lateral displacement alone, nor turns from heading change alone.

Request-level accuracy. The maneuver response accuracy is defined as

$$\mathrm{Acc}_{man} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}[O_q \wedge A(s_q, \hat{\tau}_q, \ell_q) = 1], \tag{36}$$

where O_q denotes a valid output; missing outputs count as failures. The explicit scene argument allows road and object evidence to enter the evaluator. Separate definitions do not imply statistical independence from the reward: both can use trajectory motion features.

Trajectory shift. To quantify whether the model changes its motion after instruction swapping, we measure the displacement between predictions before and after the instruction intervention:

$$\mathrm{Shift} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T} \sum_{t=1}^{T} \|\hat{\tau}_{i,t}^{swap} - \hat{\tau}_{i,t}^{orig}\|_2. \tag{37}$$

A larger shift indicates stronger sensitivity to instruction changes.

Directional consistency. For behavior pairs involving directional changes, such as left and right maneuvers, we further evaluate whether the motion change follows the desired direction.

$$\mathrm{Dir} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\mathrm{sign}(\Delta y_i^{pred}) = \mathrm{sign}(d_i^{req})]. \tag{38}$$

where Δy_i^{pred} is the lateral motion change induced by instruction swapping and $d_i^{req} \in \{-1, +1\}$ is the direction specified by the instruction pair.

The final maneuver-level diagnostic, after feasibility and evaluability screening, contains 200 same-scene pairs from 200 independent scenes, or 400 individual requests. Its eight pair types cover ten intents: lane keeping paired separately with left turn, right turn, left lane change, right lane change, acceleration, deceleration, and stopping, plus waiting paired with starting. For acceleration, deceleration, and stopping contrasts, the keeping instruction explicitly requires maintaining both the current lane and current speed, with $|v_{\mathrm{end}} - v_0| \leq 0.25\,\mathrm{m/s}$; lane keeping alone is not a valid contrast because it can be satisfied while accelerating.

Table 8: Composition of the maneuver-level diagnostic. Each pair contains two requests under an identical input snapshot.

Instruction pair	Number of pairs
Keep / left turn	22
Keep / right turn	8
Keep / left lane change	23
Keep / right lane change	26
Keep speed / accelerate	44
Keep speed / decelerate	20
Keep speed / stop	17
Wait / start	40
Total	200

Train, development, and evaluation data are partitioned by episode. Derived teachers and language paraphrases inherit the source-scene split, and both source-frame identifiers and input hashes are

checked. Physical locations are additionally isolated by location group. When no global location identifier is available, pairs of records within 5 m and 10° trigger manual leakage review; these tolerances are not used to construct an evaluation pair.

Each pair fixes one observation and changes only the instruction; both requests are reviewed as feasible for that observation before evaluation. Two reviewers independently confirm that the input is complete, both instructions are feasible and distinguishable within the five-second horizon, navigation does not conflict with either request, and the road and traffic evidence is sufficient. Where road information is incomplete, lanes, exits, and adjacency relations may be manually annotated from calibrated data and checked by two reviewers. If the target road or feasibility remains undetermined, the entire pair is marked unevaluable and excluded before inference, with exclusion reasons and counts recorded. Insufficient evidence is not an infeasible label. Once the manifest is frozen, missing outputs, wrong-lane predictions, departures from the annotated region, and incomplete actions count as failures; pairs are not removed based on model results. Evaluations using only speed or displacement are reported separately as motion proxies, not as full maneuver success. Success is decided by the independent behavior criteria rather than by the training reward. Paraphrases and repeated random seeds do not create additional independent pairs.

For Cartender, the observation, candidate noise, and generator checkpoint are held fixed within each intervention pair. Maneuver response accuracy, trajectory shift, and directional consistency are aggregated over three selector-training seeds; the 400 requests therefore yield 1,200 seed-specific predictions, without increasing the number of independent scenes or pairs. The reported $\pm$ values denote the sample standard deviation of the three seed-level results.

C.3 Branch Quality and Human Audit

Branch quality is evaluated after admission on the complete constructed set rather than on a sampled subset. Best-of-four is condition-level: for each of 3,603 feasible instruction conditions, we take the maximum RFS among A–D and then average, obtaining 8.642. Mean admitted RFS and "Above logged" are proposal-level: they pool all 8,974 admitted proposals, with each one compared against the logged trajectory matched to its condition. Under this aggregation, admitted proposals average 8.203 RFS compared with 7.906 for the matched logged trajectories, and 67% score above their logged counterpart.

RFS is not used by the admission procedure. We compute it with the public WOD-E2E metric (Xu et al., 2025). For each of up to three human-rated reference trajectories j, with rating $s_j \in [0, 10]$, the metric compares the predicted and reference positions at $t \in \{3, 5\}$ s. Let $\Delta_{j,t}^{\mathrm{lng}}$ and $\Delta_{j,t}^{\mathrm{lat}}$ denote the longitudinal and lateral errors. The base trust-region thresholds $(\bar{\tau}_t^{\mathrm{lng}}, \bar{\tau}_t^{\mathrm{lat}})$ are $(4.0, 1.0)$ m at 3 s and $(7.2, 1.8)$ m at 5 s. They are scaled by the reference speed v using

$$\alpha(v) = \begin{cases} 0.5, & v < 1.4, \\ 0.5 + 0.5(v - 1.4)/(11 - 1.4), & 1.4 \le v < 11, \\ 1, & v \ge 11, \end{cases} \tag{39}$$

so that $\tau_t^d = \alpha(v)\bar{\tau}_t^d$ for $d \in \{\mathrm{lng}, \mathrm{lat}\}$. The score for reference j at time t is

$$r_{j,t} = s_j \, 0.1^{\max\left(\max\left\{\frac{\Delta_{j,t}^{\mathrm{lng}}}{\tau_t^{\mathrm{lng}}}, \frac{\Delta_{j,t}^{\mathrm{lat}}}{\tau_t^{\mathrm{lat}}}\right\} - 1, 0\right)}. \tag{40}$$

The trajectory score is

$$\mathrm{RFS} = \max\left\{4, \frac{1}{2} \sum_{t \in \{3,5\}} \max_j r_{j,t}\right\}. \tag{41}$$

Thus, RFS is a deterministic metric over released human-rated references rather than a prompted evaluator. Best-of-four is aggregated over feasible instruction conditions, whereas mean admitted RFS and "Above logged" pool the admitted proposals; these reported construction statistics do not use scenario-cluster-balanced aggregation.

These scores quantify agreement with available rated references; they are not instruction-conditioned success or a collision-safety certificate. A different instruction can induce a trajectory

within the same reference trust region, so a higher RFS alone does not establish either low diversity or successful alternative behavior. Best-of-four is an oracle statistic over A–D, including proposals that are not teachers. The admitted-proposal mean also differs from an A/B-only teacher statistic. Neither is a deployed-policy score. The MindVLA-U1 reference value in Table 2 does not establish a matched comparison with the 1,200-scene Cartender evaluation without a common checkpoint, manifest, and aggregation protocol.

The negative-relation audit samples 228 constructed infeasible relations. Two reviewers assess every relation independently and blind to one another's decisions. Disagreements receive adjudication by a third reviewer; cases lacking sufficient evidence retain an *uncertain* label. The audit confirms 202 relations, corresponding to an 88.6% confirmation rate; 26 remain unconfirmed. Before adjudication, the two reviewers agree on 217 of 228 relations and disagree on 11, giving 95.18% raw agreement. Reasons, initial decisions, and adjudications are retained. The two instructions in every evaluation pair undergo a separate complete feasibility review.

C.4 Main Evaluation Sets and Seed Aggregation

Maneuver-level paired success requires valid outputs, action completion, and Hard checks for both requests in a same-scene pair. Writing $A_q = A(s_q, \hat{\tau}_q, \ell_q)$ and letting H_q denote the initial-state, kinematic, and comfort checks, we use

$$\mathrm{PS}_{\mathrm{man}} = \frac{1}{|\mathcal{P}|} \sum_{(q_1,q_2)\in\mathcal{P}} \prod_{j=1}^{2} \mathbf{1}[O_{q_j} \wedge A_{q_j} \wedge H_{q_j}]. \tag{42}$$

Here $\mathcal{P}$ is the frozen set of 200 maneuver pairs. Unlike request-level action accuracy, this strict metric requires both requests to pass Hard. Qualifier-level paired success applies the same conjunction rule to requests that share a parent maneuver but differ in speed or timing. Missing predictions count as failures. Output coverage is the fraction of individual requests for which the planner returns a trajectory. For a valid prediction, ADE is the mean Euclidean distance over the 20 points of the five-second horizon,

$$\mathrm{ADE}(\hat{\tau}, \tau) = \frac{1}{20} \sum_{t=1}^{20} \|\hat{\tau}_t - \tau_t\|_2. \tag{43}$$

Logged ADE uses the matched instruction and its recorded future. RFS uses the official evaluator and available human-rated references described in Appendix C.3. Missing predictions are not imputed as zero in RFS or ADE; both quality metrics state their valid-output coverage.

The main selector comparisons use three training seeds with a shared generator checkpoint and candidate sets. Table 3 reports the arithmetic mean across those seeds. The reported main-table quality metrics are first computed per valid request, averaged within scene, and then averaged within each seed; the table reports the arithmetic mean of the seed-level values. Paired-success rates give every instruction pair equal weight within a seed before the same seed-level aggregation. The trajectory-quality evaluation uses a frozen manifest of 1,200 distinct keyframes, each evaluated under all three selector-training seeds. The maneuver diagnostic contains the 200 pairs above, and the qualifier diagnostic contains 64 speed pairs and 32 timing pairs. Qualifier-specific measurement and aggregation are detailed in Appendix E.

D Additional Ablations

D.1 Compatibility Score Distributions

Feasible instructions receive higher mean compatibility scores than infeasible instructions, 0.771 versus 0.310, although the distributions overlap. The dashed line at $\theta = 0.5$ marks the diagnostic threshold used to classify low-compatibility instructions as infeasible. The score achieves an AUROC of 0.863; at this threshold, precision, recall, and F1 are 76.74%, 74.16%, and 75.43%, respectively.

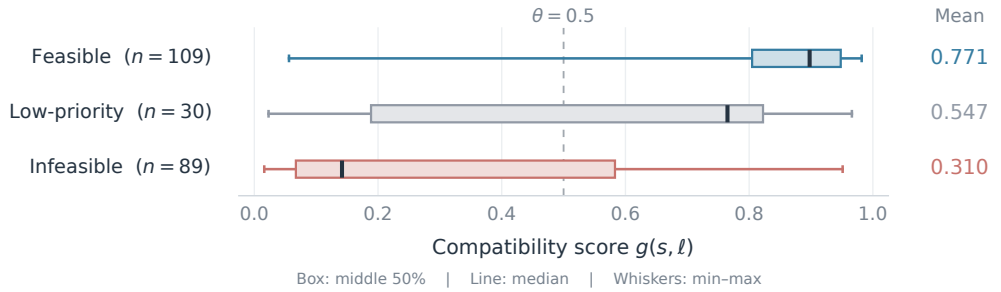


Figure 4: Compatibility scores by instruction applicability. Boxes show the interquartile range, internal lines mark the median, and whiskers span the minimum and maximum. Sample counts and mean scores are shown for each group. The dashed line marks the diagnostic threshold $\theta = 0.5$.

D.2 Detailed Analysis of Scene–Language Compatibility

We further analyze the role of the compatibility score $g(s, \ell)$.

Unlike the main table, this experiment focuses on whether the compatibility model can distinguish supported and unsupported instructions. For a request, let O denote a valid model output, H the conjunction of the initial-state and hard-constraint checks, and A a positive decision from the independent action evaluator. On the manually confirmed feasible subset, Feasible Follow is

$$\Pr(O \wedge H \wedge A \mid \text{feasible}). \tag{44}$$

On the frozen infeasible subset, Infeasible Follow is $\Pr(O \wedge H \wedge A \mid \text{infeasible})$, while Nonexecution is $\Pr(O \wedge H \wedge \neg A \mid \text{infeasible})$. Relation Handling gives the feasible and infeasible subsets equal weight:

$$\text{RelationHandling} = \tfrac{1}{2}\,\text{FeasibleFollow} + \tfrac{1}{2}\,\text{Nonexecution}. \tag{45}$$

Teacher minADE is evaluated on the fixed subset of the 400 maneuver requests that has a complete admitted decoded A/B teacher set. For request q, it is

$$d(q) = \min_{h \in \mathcal{H}_{\text{teacher}}(q)} \frac{1}{20} \sum_{t=1}^{20} \|\tau^{q}_{\text{selected}}(t) - h(t)\|_2. \tag{46}$$

ADE is computed against each complete teacher trajectory before the minimum is taken: teachers cannot be switched independently at each time point, and the reported trajectory is the actually selected output rather than the best prediction chosen post hoc. These metrics are not assumed to be interchangeable with the paired-success and RFS metrics in Table 3.

Table 9: Detailed ablation of scene–language compatibility modeling.

Method	Feasible Follow ↑	Infeasible Follow ↓	Relation Handling ↑	Teacher minADE ↓
Learned $g(s, \ell)$	43.8	14.2	64.8	4.02
No Gate	61.6	53.4	54.1	4.25
Shuffled Gate	44.8	27.2	58.8	4.14
Constant Gate	51.0	27.3	61.9	4.77

The compatibility model mainly improves the rejection of unsupported instructions rather than universally increasing instruction following. The reported values are arithmetic means over three selector-training seeds.

D.3 Behavioral Suppression of Unsupported Instructions

We evaluate compatibility gating under infeasible instructions by comparing the learned gate $g(s, \ell)$ with a bypassed gate, $g = 1$. The checkpoint, scene inputs, and random draws are held fixed.

Infeasible execution measures how often the predicted trajectory follows the unsupported request; unsupported-instruction drift measures its deviation from the reference prediction. Learned compatibility reduces both measures, although 14.18% infeasible execution remains. The mechanism provides silent fallback without an explicit refusal output.

Table 10: Behavioral effect of compatibility gating under infeasible instructions. Lower values are better for both metrics.

Method	Infeasible execution (%) ↓	Unsupported-instruction drift (m) ↓
No gate ($g = 1$)	53.36	3.612
Learned $g(s, \ell)$	**14.18**	**0.337**

The reported values are arithmetic means over three selector-training seeds.

D.4 Effect of Two-stage Selector Learning

This diagnostic tests whether demonstration initialization provides a useful base ranking before reward optimization. It uses the same evaluation set and metric definitions as Table 9. The reported values are arithmetic means over three selector-training seeds.

Table 11: Effect of demonstration initialization before reward optimization.

Setting	Feasible Follow	Infeasible Follow	Relation Handling	Teacher minADE
IL → GRPO	43.8	14.2	64.8	4.02
Random Base → GRPO	51.0	50.0	50.5	5.66

D.5 Reward Optimization Analysis

We compare GRPO with exact expected-reward optimization over the same finite candidate set. Both variants use the same frozen candidate rewards and imitation anchor; they use different optimization objectives: GRPO uses sampled, standardized group-relative advantages and a clipped surrogate, whereas the exact control sums the raw expected reward over all unmasked candidates. Thus this is not merely a comparison of an exact gradient with an unbiased estimate of that same gradient.

Table 12: Comparison of reward optimization strategies under the main evaluation protocol. Values reproduce the corresponding rows of Table 3 and are arithmetic means over three selector-training seeds.

Optimizer	Maneuver paired ↑	Qualifier paired ↑	RFS ↑	Coverage ↑
GRPO	57.00	35.07	7.10	97.70
Exact expected reward	58.17	4.86	7.00	97.70

The two optimizers obtain similar maneuver paired success, with exact expectation 1.17 points higher. In the reported runs, GRPO yields substantially higher qualifier paired success and slightly higher RFS. The GRPO objective and hyperparameters are given in Appendix B.5; the learning rate, update count, and checkpoint rule for the exact expected-reward optimizer are matched as specified there. Exact summation removes candidate-sampling error from its reward expectation, but does not guarantee a better final policy under a fixed update budget or better independent evaluation metrics. The aggregate results alone do not identify the cause of the qualifier gap; per-seed learning curves and outcomes are needed to separate optimization effects from evaluation or data-version differences.

D.6 Effect of Hard Constraints

At inference, the hard constraints mask candidates before the selector chooses among the remaining trajectories. Table 13 reports a supplementary diagnostic of this filtering step on a fixed model and fixed candidate set. Comfort requires the kinematic and comfort limits in the first five inequalities of Eq. 8 to hold; the codec reconstruction term is not an inference-time constraint. Action measures $O_q \wedge A_q$ using the independent criteria in Table 7; it does not include Hard. Joint Validity measures $O_q \wedge A_q \wedge H_q$, requiring the initial-state, Comfort, and Action checks jointly. Strict Proxy adds $\min$-reference-ADE $\leq 3\,\mathrm{m}$ to Joint Validity. If all candidates are masked, the request counts as a failure for these success rates, and coverage is reported separately. The diagnostic uses the fixed set of 400 maneuver requests.

Table 13: Effect of kinematic and comfort constraints.

Setting	Comfort	Action	Joint Validity	Strict Proxy
Without Hard	71.92	95.00	53.33	31.67
With Hard	99.00	92.50	90.00	49.17

Applying the hard constraints increases Comfort, Joint Validity, and Strict Proxy under this diagnostic, while the Action score decreases from 95.00 to 92.50. The table does not by itself establish the effect on the main paired-success metrics. The reported values are arithmetic means over three selector-training seeds.

D.7 Candidate-Budget Sensitivity

An earlier candidate-budget diagnostic evaluates a selector trained with $K = 24$ under three seeds by taking fixed nested candidate subsets at $K \in \{8, 16, 24\}$ within each seed. Because these are nested subsets from the same trained selector in each seed rather than separately retrained models, this diagnostic measures sensitivity to candidate availability at inference and does not isolate a training-time effect.

Table 14: Candidate-budget diagnostic using fixed nested subsets from a selector trained with $K = 24$. Values are arithmetic means over three selector-training seeds. The instruction proxy is the pass rate of the independent Action criterion; missing outputs count as failures. The diagnostic uses the same fixed set of 400 maneuver requests as Table 13.

K	Instruction proxy (%) ↑	Coverage (%) ↑
8	44.83	83.50
16	48.67	88.67
24	51.42	90.33

The proxy score and output coverage both increase with the available candidate count in this diagnostic. Since the selector is not retrained for each K, these results should not be read as a complete scaling study.

E Qualifier Evaluation Protocol and Results

Diagnostic results. Table 15 reports instruction responses with the maneuver held fixed. Speed tests cover 64 scenes and compare gradual versus strong acceleration or deceleration; timing tests compare early versus late deceleration on the 32 deceleration scenes. Each pair shares the same scene input and sampling noise. Cartender achieves 64.58% speed-direction accuracy and 53.13% direction-and-magnitude success, compared with 43.75% and 12.50% for Qwen-Drive. The latter criterion requires $D_v > 0.2\,\mathrm{m/s}$, where D_v is defined below. Timing-order accuracy is 35.42% for Cartender and 0.00% for Qwen-Drive. The vision-only control scores zero on all three response metrics. These metrics assess directional or magnitude responses; they are distinct from strict qualifier paired success, which also requires both individual targets and the kinematic and comfort checks to be satisfied.

Table 15: Qualifier-level instruction responses with the maneuver held fixed. All values are percentages. Direction-and-magnitude success additionally requires $D_v > 0.2\,\mathrm{m/s}$. Timing results use 32 early–late pairs.

Model	Speed direction	Direction + magnitude	Timing order
Vision-only	0.00	0.00	0.00
Qwen-Drive	43.75	12.50	0.00
Cartender	**64.58**	**53.13**	**35.42**

Speed measurement. Each instruction pair shares the same scene and parent maneuver; only the speed or timing qualifier changes. The checkpoint, non-language inputs, and sampling noise are fixed within each pair. We evaluate planar trajectories over a five-second horizon at $\Delta t = 0.25\,\mathrm{s}$ intervals. Let $\mathbf{p}_i$ denote the predicted position at $t_i = i\Delta t$, for $i = 1, \ldots, 20$, and let $\mathbf{p}_0$ be the observed current position. We compute interval speeds and the terminal speed change as

$$v_i = \frac{\|\mathbf{p}_i - \mathbf{p}_{i-1}\|_2}{\Delta t}, \qquad \Delta v = \frac{1}{4}\sum_{i=17}^{20} v_i - v_{\mathrm{obs}}, \tag{47}$$

where v_{obs} is the magnitude of the observed current velocity. Thus, the terminal statistic averages speed over the final second, $(4, 5]\,\mathrm{s}$, rather than using a single endpoint speed or a predicted initial-speed reference.

Deceleration onset. We define onset as the first sampled time at which speed is at least $0.5\,\mathrm{m/s}$ below the observed initial speed:

$$t_{\mathrm{on}} = \min\{t_i : v_i \le v_{\mathrm{obs}} - 0.5\,\mathrm{m/s}\}. \tag{48}$$

The detector uses individual interval speeds without additional smoothing or a minimum-duration requirement; one qualifying interval is sufficient. If the threshold is never crossed within five seconds, onset is undefined. A timing pair with an undefined onset on either side is counted as a failure in both directional response and strict paired success, and remains in the denominator.

Success criteria. For speed pairs, let $\sigma = +1$ for accelerating and $\sigma = -1$ for decelerating. Directional response requires

$$D_v = \sigma\big(\Delta v_{\mathrm{strong}} - \Delta v_{\mathrm{gradual}}\big) > 0. \tag{49}$$

Material speed response additionally requires $D_v > 0.2\,\mathrm{m/s}$. For timing pairs, directional response requires both onsets to be defined and $t_{\mathrm{on}}^{\mathrm{late}} > t_{\mathrm{on}}^{\mathrm{early}}$. These directional metrics measure the requested ordering; they do not require the individual targets or the kinematic and comfort checks to be satisfied.

Strict paired success requires both outputs to pass the kinematic and comfort checks used in evaluation and to satisfy their respective qualifier targets. For speed, these targets are

$$\begin{aligned} 0.5\,\mathrm{m/s} \le \sigma\Delta v_{\mathrm{gradual}} < 2.0\,\mathrm{m/s}, \\ \sigma\Delta v_{\mathrm{strong}} \ge 2.0\,\mathrm{m/s}. \end{aligned} \tag{50}$$

For timing, evaluated on decelerating maneuvers, the early target requires $t_{\mathrm{on}} \le 1.5\,\mathrm{s}$ and the late target requires $2.5\,\mathrm{s} \le t_{\mathrm{on}} \le 3.5\,\mathrm{s}$; both also require $\Delta v \le -0.5\,\mathrm{m/s}$. Missing outputs count as failures for all applicable pair metrics. Output coverage is reported separately.

Aggregation. The speed diagnostic contains 64 pairs from 64 scenes: 32 compare gradual versus strong acceleration and 32 compare gradual versus strong deceleration. Each of the 32 deceleration scenes additionally contributes one early–late timing pair. Thus the timing diagnostic contains 32 pairs, and no scene contributes more than one pair of a given qualifier type. The combined diagnostic therefore contains 96 instruction pairs from 64 distinct scenes. Cartender is evaluated with three evaluation replicates with a fixed generator checkpoint. Within each replicate, we first average

binary pair outcomes over all scenes in the relevant category, including failures and missing outputs. We then report the arithmetic mean across replicates; $\pm$ denotes the sample standard deviation of the replicate-level rates, with denominator $R - 1$ for R replicates. It is not a confidence interval or a standard error.

For the combined qualifier score, pairs are weighted equally:

$$\text{Combined} = \frac{\#\text{ successful speed pairs} + \#\text{ successful timing pairs}}{96}. \tag{51}$$

Equivalently, the 64 speed pairs and 32 timing pairs contribute weights $2/3$ and $1/3$, respectively, before aggregation across replicates. Repeated seeds do not increase the number of independent scenes. Vision-only uses three sampling-noise replicates, and Qwen-Drive uses three evaluation replicates; no across-seed uncertainty is currently reported for Qwen-Drive. All rates are reported as percentages.

F Additional Discussion

F.1 Further Discussion of Limitations and Future Work

Control remains incomplete: strict qualifier paired success is 35.07%, and the learned gate reduces but does not eliminate infeasible execution.

The diagnostic uses identical recorded input snapshots with two reviewed feasible instructions. It does not observe both actions being executed in the same physical event. Its conclusions depend on the coverage of the selected scenes and the behavior criteria; broader coverage requires reviewing feasibility in additional settings.

Admission is decided by rule-based checks, and branch quality is evaluated afterward with aggregate RFS; driver-specific feedback could further distinguish individual preferences among the admitted behaviors.

Route-relative construction limits the range of alternative paths. The admission checks do not establish collision freedom or containment within drivable areas, and RFS does not certify instruction fulfillment. Branch-quality averages pool admitted roles rather than isolating the A/B teachers used for training. These boundaries limit claims about the safety and diversity of the constructed supervision.

Compatibility can suppress feasible requests as well as infeasible ones, as Table 9 shows. Separately defined reward and evaluation rules may share motion features; their agreement does not establish complete interaction-level success. The current experiments also lack direct inference-time reward ranking, $\arg\max_{i \in \mathcal{A}} r(\tau_i, \ell)$, on the same candidates. Reward-free inference describes Cartender's implementation; it does not establish an accuracy or latency advantage over that baseline. Three selector seeds sharing a generator do not capture variation from retraining the generator, and seed variation is not a confidence interval over independent scenes.

The setting here decides once per scene, whereas an instruction issued mid-maneuver and revised before it completes makes the ranking sequential and the compatibility estimate time-varying. Continuous instruction updates require updating both ranking and compatibility as the scene evolves while maintaining consistency with actions already under execution. Future work will extend this formulation to such continuous updates during closed-loop execution, including highly interactive traffic in which other agents respond to the vehicle's actions.