NAVSAFE-∞: BENCHMARKING CLOSED-LOOP DRIVING SAFETY IN PHOTOREALISTIC ENVIRONMENTS

Anonymous authors
Paper under double-blind review

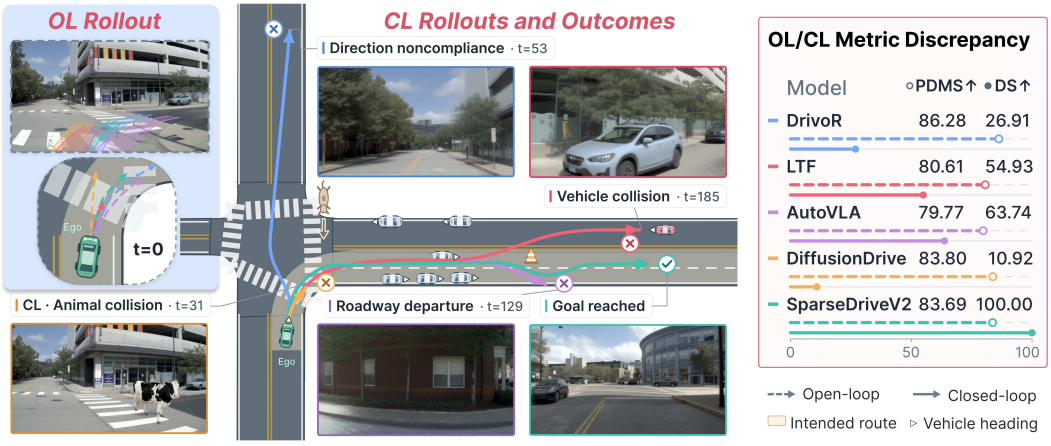

Figure 1: **Motivations.** End-to-end (E2E) driving policies with comparable open-loop (OL) predictive driver model scores (PDMS) diverge under closed-loop (CL) execution, yielding different safety outcomes and Driving Scores (DS), which reveals the OL-to-CL evaluation gap in E2E driving.

ABSTRACT

End-to-end (E2E) driving policies have advanced rapidly on open-loop (OL) benchmarks, yet OL evaluation cannot reveal whether a policy can withstand compounding errors, recover from failures, or interact safely with surrounding actors. We introduce NAVSAFE-∞, a photorealistic closed-loop (CL) benchmark comprising 280 scenarios spanning 28 event types, each with success and failure criteria defined within a structured traffic-safety taxonomy, yielding category-level capability scores for Traffic Crashes, Vulnerable Road User Crashes, Traffic Violations, and Traffic Incidents. After evaluating 20 E2E policies, we find that OL gains do not reliably transfer to CL safety. Analysis of two common remedies reveals that (1) passive demonstration perturbation helps mainly when CL rollouts stay near their perturbed training states, and (2) OL reinforcement-learning fine-tuning exhibits reward hacking by trading safety margin for ego progress, which CL feedback amplifies into compounding safety-critical errors. Together, these results demonstrate the blind spot of OL benchmarks indicating CL safety success. The benchmark and an extensible toolbox for customizable event curation and policy diagnosis will be open-sourced to facilitate future research.

1 INTRODUCTION

Real-world driving safety depends on how a policy resolves safety-critical situations through continuous interaction: in closed-loop (CL) driving, a policy's actions reshape its future observations and the surrounding dynamics, so the same scenario can lead to divergent outcomes. Open-loop (OL) evaluation on static logged states cannot capture these interaction sequences or the compounding errors they cause. Yet recent end-to-end (E2E) driving policies are evaluated primarily on OL bench-

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.

marks such as nuScenes and NAVSIM (Caesar et al., 2020; Dauner et al., 2024; Cao et al., 2025), where safety metrics score predicted trajectories without sequential policy–environment feedback. Figure 1 illustrates this OL-to-CL evaluation gap: policies with similar OL scores yield different CL safety outcomes in the same scenario. Thus, the community still has limited insight into the reliability of current E2E policies confronting safety-critical events.

In this paper, we aim to understand whether existing driving policies can reliably navigate long-horizon CL scenarios across sequences of safety-relevant events. To systematically evaluate these capabilities across diverse, safety-relevant interactions, we introduce NAVSAFE-∞, a photorealistic CL benchmark that evaluates policies against event-specific objectives derived from real-world traffic-safety events (Liu et al., 2026).

Each event specifies success and failure conditions that are evaluated over the executed CL rollout. For example, a red-light event requires the ego to remain behind the stop line for the prescribed interval, recognizing compliant stopping as successful task completion rather than route completion. Through event-level evaluation, NAVSAFE-∞ enables identification of specific safety failures and assessment of whether gains in OL performance translate into improved CL safety.

To assess policy behavior in CL, we conduct a fine-grained analysis across the NAVSAFE-∞ safety taxonomy. Our results show that strong OL performance does not reliably transfer to CL safety: high-ranking policies on OL benchmarks remain vulnerable across multiple safety-critical event types when operating in policy-induced states. We further examine whether common mitigation practices for narrowing the OL-to-CL gap, such as passive demonstration perturbation (Karkus et al., 2025; Tian et al., 2026) and reinforcement learning (RL) fine-tuning (RLFT) methods, alleviate these failures. Notably, we find that 1) perturbation (training on states perturbed independently of the policy) helps mainly when CL rollouts stay close to those states and 2) RLFT exhibits signs of reward hacking in the OL PDM score (PDMS) (Dauner et al., 2024), while CL feedback can amplify the resulting errors and degrade CL performance. These findings shed light on how future methods should be developed to improve CL driving safety.

Our contributions are threefold. **1) NAVSAFE-∞ Benchmark:** we present a photorealistic CL benchmark that evaluates driving safety at the event level, grounded in a real-world traffic-safety taxonomy, and validate its rendering fidelity with perceptual and policy-oriented metrics so that policy behavior is evaluated with limited visual distortion (Section 3.3). **2) Analysis:** we show that OL performance does not reliably translate into CL safety, localize the resulting failure patterns across policies, and diagnose why perturbation and RLFT yield inconsistent CL gains. **3) NAVSAFE-∞ Toolbox:** we design an extensible toolbox for customizable scenario curation and policy diagnosis under controlled changes to ego and environment states, enabling the community to extend NAVSAFE-∞ with new safety-critical events and diagnostic protocols. Together, the benchmark and toolbox provide a reliable and extensible testbed for studying various CL driving behaviors.

2 Related Work

2.1 End-to-End Driving Benchmarks

Open-Loop Benchmarks. OL benchmarks offer a low-cost, scalable proxy on real sensor observations. nuScenes (Caesar et al., 2020) scores trajectories against recorded futures; NAVSIM (Dauner et al., 2024; Cao et al., 2025) replaces displacement error with collision- and compliance-based planning scores; WOMD-Reasoning (Li et al., 2025a) and nuReasoning (Huang et al., 2026b) add reasoning question-answer pairs. Yet the policy never sees the consequences of its own actions, hiding compounding error, failure recovery, and reactive traffic, and OL rankings only loosely track CL performance (Zhao et al., 2026a; Wang et al., 2026b).

Closed-Loop Benchmarks. CL benchmarks execute policies through sequential feedback. CARLA-based (Dosovitskiy et al., 2017) Bench2Drive (Jia et al., 2024), HiDrive (Xia et al., 2026b), and Fail2Drive (Gerstenecker et al., 2026) provide structured behavioral evaluation, with the first two reporting per-ability results, but game-engine rendering leaves a domain gap for real-sensor policies. HUGSIM (Zhou et al., 2026a), WorldEngine (Li et al., 2026a), and AlpaSim (NVIDIA et al., 2025) narrow this gap with 3DGS reconstruction (Kerbl et al., 2023; NVIDIA, 2026b), and DriveArena (Yang et al., 2025) and AlpaSim with generative rendering (OmniDreams (Basant et al.,

2026)), yet they mainly report aggregate scores on their own scenarios. Moreover, existing ability categories follow driving maneuvers rather than real-world safety events, and success is typically generic route completion. NAVSAFE-∞ instead combines photorealistic CL with taxonomy-grounded events, event-specific success criteria, and category-level Ability Scores.

2.2 END-TO-END DRIVING POLICIES

We introduce the following driving policy training paradigms that are covered in this paper:

Imitation learning-based pretraining. Most E2E and vision–language–action (VLA) policies are trained offline through imitation learning (IL) or supervised fine-tuning (SFT): LTF (Dauner et al., 2024), DiffusionDrive (Liao et al., 2025), and DrivoR (Kirby et al., 2026) learn trajectories from fixed demonstrations; RAP (Feng et al., 2026) augments them with rasterized counterfactual views; SparseDriveV2 (Sun et al., 2026) and GTRS (Li et al., 2025b) score a vocabulary factorized into paths and velocity profiles; VLA policies align vision, language, and action (Zhou et al., 2025). All inherit behavior cloning's covariate shift: training never covers the state distribution the policy induces at test time.

Reinforcement learning-based post-training. RL optimizes a driving reward atop an imitation-pretrained policy: ReCogDrive (Li et al., 2026d), MTDrive (Li et al., 2026b), and AutoVLA (Zhou et al., 2025) do so over offline observations. Staying offline confines reward to logged states and leaves reward-exploiting policies uncorrected by the environment.

Passive demonstration perturbation methods. To synthesize states absent from logs, SimScale (Tian et al., 2026) perturbs ego trajectories in reconstructed reactive scenes and co-trains with pseudo-expert supervision. BeyondDrive (Wang et al., 2026a) mines expert-proximate unsafe trajectories as negatives.

World-model-based policies. These policies learn future prediction alongside planning: DriveLaW (Xia et al., 2026a) conditions a flow-matching planner on video-diffusion latents, DriveVLA–W0 (Li et al., 2026c) adds future-image prediction as auxiliary supervision to a VLA policy, and SimWAM (Zhao et al., 2026b) co-trains an action expert with a video expert dropped at inference.

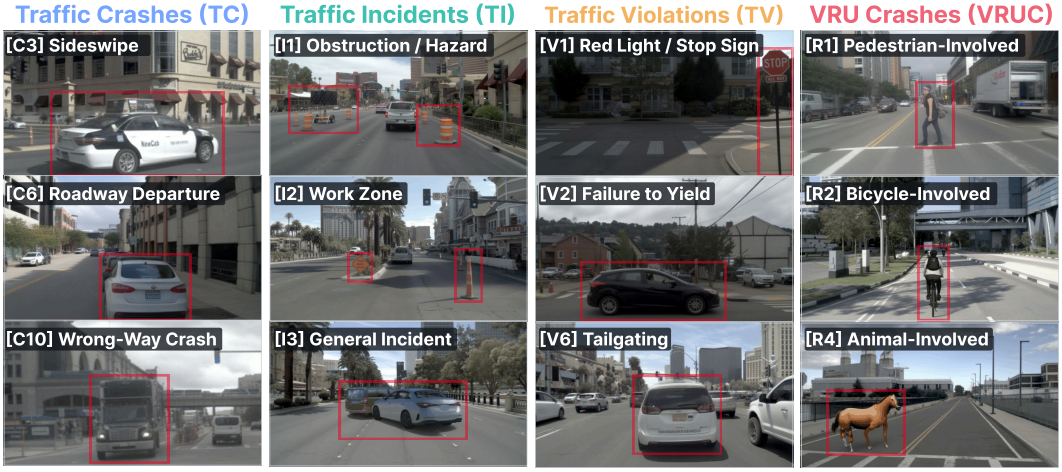

Figure 2: Representative NAVSAFE-∞ events. Appendix A.2 shows the remaining 16 event types.

3 NAVSAFE-∞ BENCHMARK

NAVSAFE-∞ is a photorealistic CL benchmark comprising 280 event-based scenarios organized into 28 event types adapted from RoadSafe365 (Liu et al., 2026). These event types span four categories: *Traffic Crashes (TC)*, *Vulnerable Road User Crashes (VRUC)*, *Traffic Violations (TV)*, and *Traffic Incidents (TI)*. Figure 2 shows representative scenarios from each category. Section 3.1 presents the event-based benchmark design, Section 3.2 defines the evaluation protocol and metrics, and Section 3.3 evaluates rendering fidelity and policy-relevant visual alignment.

3.1 EVENT-BASED BENCHMARK DESIGN

Design Principles. Long-horizon driving requires a policy to resolve a sequence of local interactions and traffic-rule decisions. A single route-level score aggregates behavior across these events, making it difficult to assess specific safety capabilities. Prior work further shows that long-route evaluation can produce high-variance aggregate scores (Jia et al., 2024). NAVSAFE-∞ therefore evaluates bounded, event-centered scenarios, enabling granular capability assessment while preserving CL feedback within each event. Figure 3 illustrates how this design preserves both properties: each bounded event isolates a specific safety capability while retaining policy-environment feedback throughout the rollout.

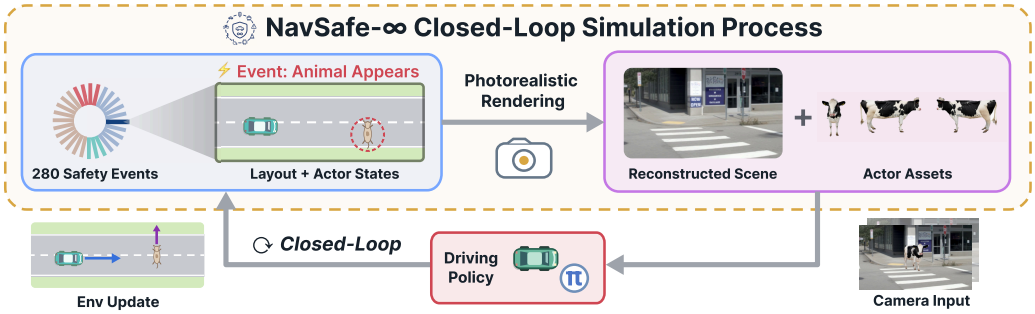

Figure 3: **NAVSAFE-∞ Closed-Loop Simulation Process.** NAVSAFE-∞ loads a bounded, event-specific reconstruction. At each step, the renderer produces camera observations from the current ego and actor states; the policy's plan advances the simulator, yielding the next observation.

Event Definition. We define an *event* as a bounded traffic interaction or rule-governed situation with explicit task and safety requirements. Each scenario instantiates an event type through a specified initial scene, relevant road users or traffic controls, and termination conditions. Figure 2 demonstrates some representative examples including responding to a braking lead vehicle, yielding to a crossing pedestrian, holding behind a red-light stop line, and navigating a work-zone obstruction. The required behavior determines how success is evaluated: a red-light event requires a compliant hold for the prescribed interval, whereas a work-zone event requires completing the prescribed bypass without disallowed infractions. Appendices A.1 and D.2 list the inclusion criteria of all 28 event types and their success and termination rules.

Task Setting. Following Zhao et al. (2026a), we formulate E2E CL driving as a finite-horizon partially observable Markov decision process traversing through each scenario e of event type ℓ, namely $\mathcal{M}_e = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P_e, \Omega_\kappa, \rho_e, R_\ell, H_e)$, with transition P_e, initial-state distribution ρ_e, sparse terminal success reward R_ℓ, and horizon H_e. Two properties distinguish this formulation. First, the emission model $\Omega_\kappa(o_t \mid s_t)$ provides *configurable camera observations*: because each scene is reconstructed as a 3DGS representation, the same scenario can be rendered as single-view, multi-view, or any other camera rig κ by changing only the camera intrinsics and extrinsics, without re-collecting data. Second, the dynamic state s_t is *fully controllable*, including editing traffic elements and prescribing the behaviors of surrounding vehicles. The objective is to complete the event-specific task while maintaining safe execution throughout the CL interaction.

3.2 EVALUATION PROTOCOL AND METRICS

Appendix D provides the full evaluation protocol, while Appendix H reports runtime and resource requirements.

We report *Driving Score (DS)*, *Success Rate (SR)*, *Driving Efficiency (DE)*, and *Comfort*, following established CL driving evaluation protocols (Jia et al., 2024; Caesar et al., 2021). For each event, DS is computed by multiplying *Event Progress (EP)* by penalties for recorded infractions and scaling the result to $[0, 100]$. It therefore rewards event progress while penalizing unsafe or non-compliant behavior. SR is the percentage of events that satisfy their event-type-specific success criteria within the time budget without a disallowed infraction. DE measures ego speed relative to surrounding traf-

fic, while Comfort measures compliance with predefined bounds on acceleration, jerk, yaw rate, and yaw acceleration. We additionally report an *Ability Score* for each benchmark category (TC, VRUC, TV, and TI), defined as the unweighted mean of the event-type-level SRs within that category.

Table 1: Quantitative visual fidelity comparison across generative renderers. All metrics compare rendered observations with the corresponding recorded clips. DD refers to DiffusionDrive (Liao et al., 2025); SDv2 refers to SparseDriveV2 (Sun et al., 2026).

Generative Renderer	**Venue**	**FID↓**	**FVD↓**	$\mathrm{FD}_\pi \downarrow$	$\mathrm{FD}_\pi^k \downarrow$				
					DrivoR	**DD**	**LTF**	**RAP**	**SDv2**
DriveArena (Yang et al., 2025)	ICCV 2025	27.14	814.81	18.55	12.88	33.96	7.61	27.10	11.18
DreamStream (Leng et al., 2026)	CoRL 2026	11.53	176.83	6.27	5.51	11.06	2.89	6.67	5.24
NavSafe-∞	–	**4.95**	**45.34**	**2.45**	**2.26**	**5.01**	**1.07**	**1.67**	**2.26**

3.3 Rendering Fidelity and Policy-Relevant Visual Alignment

Observations are rendered from 3DGS reconstructions (NVIDIA, 2026b) and refined by the diffusion-based DiffusionHarmonizer (Zhang et al., 2026) (Appendix C). To assess the visual fidelity of the reconstructed observations, we compare rendered frames against the corresponding real camera frames from the source logs. We report FID and FVD (Heusel et al., 2017; Unterthiner et al., 2018) for frame-level and video-level (temporal) distribution gaps, and the policy-oriented FD_π (Leng et al., 2026), which is the Fréchet distance between policy features of real and rendered frames, averaged over five E2E policies (FD_π^k per policy; details in Appendix C.4).

As shown in Table 1 and Figure 4, NavSafe-∞ achieves better rendering quality than existing generative renderers on all three metrics.

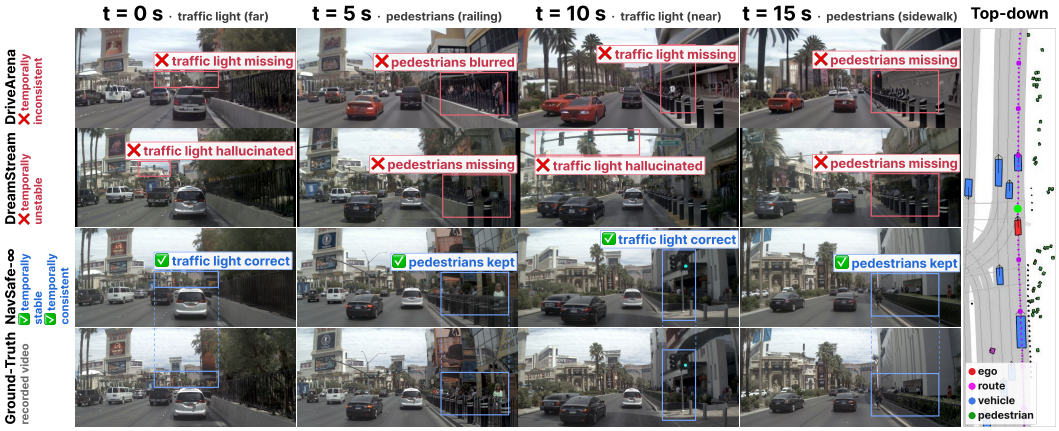

Figure 4: Rendering quality on a long-horizon scenario. **Left:** First-person views at $t \in \{0, 5, 10, 15\}$s. **Right:** Top-down bird's-eye view. DriveArena shows temporal inconsistency and DreamStream exhibits cumulative drift, whereas NavSafe-∞ stays temporally consistent.

4 Experiments

We organize our experiments around three research questions: **(1)** How closely is performance on the OL benchmark NAVSIM (Dauner et al., 2024) associated with success in safety-critical events under CL execution? **(2)** What failure mechanisms underlie representative unsafe CL outcomes across mainstream E2E policy paradigms? **(3)** Why do post-IL methods such as perturbation and RLFT yield inconsistent CL gains across policies?

4.1 NAVSAFE-∞ LEADERBOARD

We evaluate policies using four global CL metrics and four category-specific ability scores (Table 2). Appendix F.1 gives the complete leaderboard.

Table 2: NAVSAFE-∞ leaderboard. Among learned policies, blue, green, and yellow backgrounds denote the first- , second- , and third- highest point estimates in each column, respectively. ± is the sample standard deviation across the three per-seed estimates.

Policy	Venue	Closed-loop metrics				Closed-loop ability scores			
		DS	SR	DE	Comf.	TC	VRUC	TV	TI
Human Driver Reference	–	100.00±0.00	100.00±0.00	316.52±0.42	75.15±1.89	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
Privileged reference									
PDM-Closed (Dauner et al., 2023)	CoRL 2023	90.17±1.00	81.31±1.03	305.27±8.57	82.49±1.39	88.67±1.53	72.50±0.00	84.24±1.39	57.78±1.92
IL-based Methods									
LTF (Dauner et al., 2024)	NeurIPS 2024	53.04±1.92	27.74±1.76	163.71±0.11	**97.81±0.15**	31.67±0.58	25.00±7.50	28.48±1.89	15.56±1.92
DiffusionDrive (Liao et al., 2025)	CVPR 2025	44.77±0.75	20.95±0.55	165.08±1.84	97.68±0.56	21.33±1.15	11.67±1.44	23.64±0.00	22.22±1.92
DrivoR (Kirby et al., 2026)	CVPR 2026	64.74±1.60	50.71±0.71	188.02±1.07	87.92±1.25	58.67±2.52	31.67±5.77	52.42±2.62	43.33±0.00
RAP (Feng et al., 2026)	ICLR 2026	47.70±0.66	27.26±0.41	220.78±3.77	48.38±0.60	32.33±1.15	14.17±6.29	28.79±0.52	22.22±3.85
SparseDriveV2 (Sun et al., 2026)	ECCV 2026	65.26±1.72	48.81±3.04	167.99±1.01	93.41±0.43	54.00±0.00	43.33±8.04	48.18±4.17	41.11±3.85
GTRS-Dense–V2-99 (Li et al., 2025b)	arXiv 2025	64.27±0.76	49.52±1.15	163.88±0.87	95.90±0.63	52.00±1.73	40.00±2.50	54.24±0.52	36.67±0.00
ReCogDrive–2B–IL (Li et al., 2026d)	ICLR 2026	48.98±0.95	28.21±1.43	166.94±3.15	97.58±0.11	30.00±1.00	16.67±3.82	31.21±0.52	26.67±3.33
MTDrive–SFT (Li et al., 2026b)	arXiv 2026	48.78±0.88	31.79±1.43	173.77±0.76	94.51±1.25	30.33±1.15	15.00±6.61	35.76±1.39	44.44±1.92
RLFT-based Methods									
ReCogDrive–2B–RL (Li et al., 2026d)	ICLR 2026	53.64±1.42	30.95±2.06	173.96±1.90	85.10±0.49	35.33±2.31	18.33±6.29	32.42±1.39	27.78±5.09
MTDrive–mtGRPO (Li et al., 2026b)	arXiv 2026	55.86±0.50	40.60±0.90	**277.62±3.17**	93.94±0.41	40.33±2.89	16.67±1.44	46.36±0.91	**52.22±3.85**
AutoVLA (Zhou et al., 2025)	NeurIPS 2025	51.05±0.80	29.52±2.43	186.02±0.58	65.37±0.47	33.00±2.65	15.00±4.33	29.70±2.92	36.67±8.82
Passive Demonstration Perturbation Methods									
DiffusionDrive (SimScale) (Tian et al., 2026)	CVPR 2026	52.19±2.05	35.24±2.38	189.66±4.28	96.92±0.34	38.33±1.15	18.33±7.22	36.67±5.01	42.22±3.85
DrivoR (SimScale, 134k) (Tian et al., 2026)	CVPR 2026	57.66±1.32	42.50±0.36	176.73±0.92	85.88±0.39	44.00±1.00	17.50±2.50	48.18±2.41	50.00±0.00
LTF (SimScale) (Tian et al., 2026)	CVPR 2026	52.65±0.88	34.05±0.21	192.09±0.85	97.41±0.48	35.00±1.73	17.50±2.50	38.48±0.52	36.67±0.00
GTRS-Dense–V2-99 (SimScale) (Tian et al., 2026)	CVPR 2026	**73.54±0.84**	**61.79±1.56**	141.50±1.30	87.69±1.66	**69.33±0.58**	**50.00±2.50**	**64.24±1.89**	43.33±3.33
DiffusionDrive (BeyondDrive) (Wang et al., 2026a)	ECCV 2026	46.61±0.07	21.79±0.94	172.79±1.26	97.24±0.19	22.67±1.53	10.00±0.00	25.15±1.39	22.22±1.92
World-Model-based Methods									
SimWAM (Zhao et al., 2026b)	arXiv 2026	57.53±2.72	37.50±1.56	160.54±1.64	97.19±0.27	41.33±0.58	26.67±7.22	40.61±0.52	27.78±5.09

The leaderboard reveals two important insights. First, strong OL performance does not ensure CL safety. Despite approaching human-level performance on the NAVSIM leaderboard (Figure 13), the evaluated policies exhibit a pronounced CL safety gap: even the strongest learned policy trails the privileged PDM-Closed reference by 19.5 percentage points in SR. Second, safety performance varies substantially across event types, and no learning-based policy leads all four categories. The same training intervention can affect policies differently: SimScale raises DiffusionDrive's SR but lowers DrivoR's (Table 2). A single OL or aggregate CL score therefore cannot reveal how a policy behaves across the states and safety events encountered during interaction.

4.2 FAILURE INVESTIGATIONS

We further conduct case studies to investigate failure modes across representative E2E policy paradigms. We analyze one diagnostic case from each of the four NAVSAFE-∞ categories, designed to probe distinct safety capabilities. This analysis localizes unsafe CL outcomes to specific breakdowns in *planning, scoring, prediction, and reasoning*. Case details are in Appendix F.3.

Planning Failure: Causal confusion in hazard-conditioned planning. The I-3 General Incident event in the Traffic Incidents (TI) category requires a prompt response to an unexpected crossing conflict. Although LTF consistently detects the crossing vehicles, its longitudinal plan resembles nominal car following and delays decisive braking until the collision becomes unavoidable (Figure 5, left). This behavior is consistent with causal confusion in offline behavior cloning: the policy perceives the hazard, yet its plan remains governed by correlations associated with routine car following rather than by the detected crossing conflict.

Scoring Failure: Trajectory scoring gap prevents safe planning. The V-1 Red-Light/Stop-Sign event in the Traffic Violations (TV) category requires the policy to stop at the applicable traffic control. DrivoR's candidate set contains a collision-free stopping trajectory, yet the learned scorer assigns a higher value to the collision-bound proposal (Figure 5, right). The available safe proposal localizes the observed failure to trajectory ranking rather than candidate generation. DrivoR learns

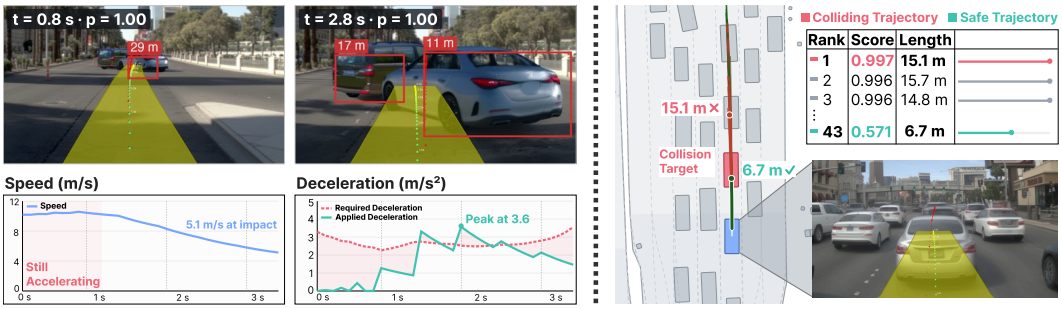

Figure 5: **Left: Planning Failure.** LTF detects crossing traffic at 29 m, but continues accelerating and brakes too late to avoid a collision. **Right: Scoring Failure.** DrivoR executes its top-ranked trajectory and collides with the lead vehicle, while a safe stopping trajectory is ranked 43rd.

its scoring function from trajectory-level oracle labels evaluated at logged states. The observed misranking exposes an OL–CL generalization gap in which the learned safety ordering does not transfer to the policy-induced state encountered during rollout.

Prediction Failure: Latent prediction errors can undermine planning in world-model-based policies. The C-7 Head-On event in the Traffic Crashes (TC) category requires the policy to maintain sufficient clearance from oncoming traffic in a constrained corridor. For this diagnostic, we re-enable and decode a future frame with SimWAM's retained video expert from the same CL observation. SimWAM omits the oncoming vehicle from its future prediction and plans through the region it occupies (Figure 6, left). This coupled failure exposes a breakdown in action-relevant actor persistence, as the model drops a safety-critical actor from its predicted world state. Latent prediction errors can undermine world-model-based planning by removing safety-critical constraints from the representation used for trajectory generation under CL execution.

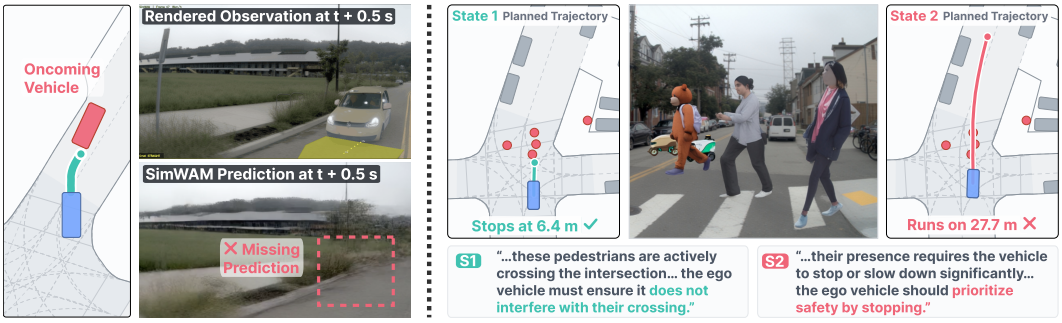

Figure 6: **Left: Prediction Failure.** SimWAM omits the oncoming vehicle visible in the rendered observation; the policy plans through the space it occupies and collides. **Right: Reasoning Failure.** At the same instant, two policy-induced states reached by different rollouts both identify the crossing pedestrians and call for a stop, but only State 1 plans to stop; State 2 accelerates and collides.

Reasoning Failure: Reasoning–action alignment is unstable across policy-induced states for VLA policies. The R-1 Pedestrian-Involved event in the Vulnerable Road User Crashes (VRUC) category requires the policy to stop or yield to pedestrians crossing the ego path. At the same instant, in two rollouts that differ only in the evaluation seed, AutoVLA correctly calls for stopping in two policy-induced states, but only one decoded trajectory realizes that decision (Figure 6, right). The matched comparison isolates a state-conditioned failure of reasoning–action grounding, whereby the reasoning remains invariant while the resulting plans diverge across policy-induced states.

4.3 Understanding the Limits of Passive Demonstration Perturbation and RLFT in Closed-Loop

Passive demonstration perturbation and RLFT are two widely adopted remedies that go beyond imitation learning to improve CL performance (Karkus et al., 2025). Perturbation enhances IL by

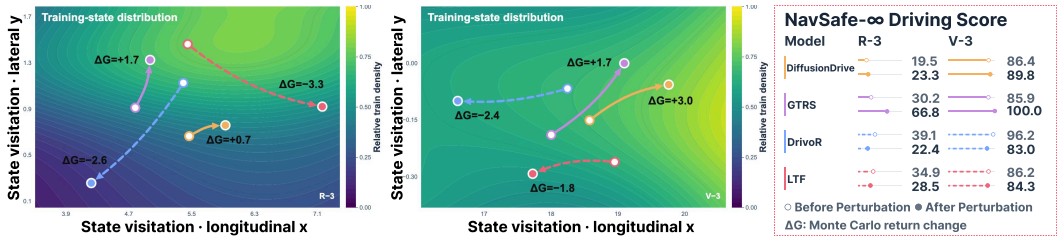

Figure 7: **Effect of perturbation on CL rollouts.** State shifts and Monte Carlo return changes show that perturbation helps when CL rollout states remain covered and hurts otherwise (Appendix F.4).

further training the policy on states constructed independently of its own rollouts. We show that this approach helps mainly when the policy's CL rollouts remain close to these perturbed states. RLFT for E2E driving, on the other hand, typically optimizes an OL proxy reward against a frozen logged future. As a result, OL reward hacking, such as trading safety margin for progress, can carry over to CL execution and compound over repeated replanning steps.

Perturbation effectiveness hinges on CL rollout distribution coverage. Such methods (Tian et al., 2026) extend the logged demonstrations $\mathcal{D}^{\text{log}}$ with $\mathcal{D}^{\text{aug}}$, pairing observations of perturbed ego states s^{aug} with verified expert recovery trajectories a^{aug}:

$$\min_{\theta} \mathbb{E}_{(s,a)\sim\mathcal{D}^{\text{log}}\cup\mathcal{D}^{\text{aug}}}\left[\ell\big(a,\pi_\theta(\cdot\mid s)\big)\right]. \tag{1}$$

This improves CL performance when rollouts remain near the perturbation support, but this depends on the learned trajectory proposal, which the objective does not directly constrain. In practice, the outcome varies across policies (Figure 7). For anchor-based methods such as DiffusionDrive (Liao et al., 2025) and GTRS (Li et al., 2025b), trajectory candidates remain close to expert trajectories and are largely unaffected by training on perturbed states. Training mainly improves the scorer, keeping selected rollouts closer to the perturbed state distribution. For query-based DrivoR and LTF, training on $\mathcal{D}^{\text{aug}}$ improves OL scores on expert states, yet CL visitation still drifts away from the support of $\mathcal{D}^{\text{aug}}$. This limitation arises because $\mathcal{D}^{\text{aug}}$ is constructed independently of the policy and thus does not capture the policy visitation distribution, which only emerges in CL execution.

OL reward hacking transfers and compounds in CL execution. Most existing RLFT methods (Li et al., 2026d;b) optimize an OL objective on states visited by the expert, scoring each sampled trajectory once with a proxy reward R', such as PDMS (Dauner et al., 2024), against a frozen logged future:

$$\max_{\theta} \mathbb{E}_{s^{\text{log}}\sim\mathcal{D}^{\text{log}},\ a\sim\pi_\theta(\cdot|s^{\text{log}})}\left[R'\left(s^{\text{log}},a\right)\right]. \tag{2}$$

Because R' rewards progress directly but constrains safety margin only weakly, the policy can raise R' by trading margin for progress. This exploitation carries over to CL, raising the per-step risk of a safety-critical error; in safety-critical scenarios, such errors compound over the rollout and lower the CL return. The two RLFT policies we evaluate illustrate both regimes (Figure 8; Appendix F.5). RLFT as applied in ReCogDrive leaves the safety-related sub-scores largely intact, so its OL gains transfer to CL. RLFT as applied in MTDrive instead raises PDMS mainly through higher progress at the cost of lower safety scores, causing a sharp drop in CL DS under safety-critical scenarios.

5 NavSafe-∞ Toolbox

To facilitate further research on CL policy behavior, we provide an extensible NavSafe-∞ toolbox that enables researchers to **(1)** build photorealistic CL environments from additional real-world driving logs, **(2)** create controlled tests for user-defined safety events, and **(3)** examine how new policies behave under off-nominal states and evolving traffic interactions.

(1) Photorealistic environment construction. The environment-construction tool converts real-world driving logs (Caesar et al., 2021; Sun et al., 2020) into interactive simulation environments. It maps calibrated sensor streams, ego poses, actor tracks, traffic-light states, and map context into a unified simulation representation, and reconstructs the corresponding visual scene as a 3DGS environment (NVIDIA, 2026b; Huang et al., 2026a).

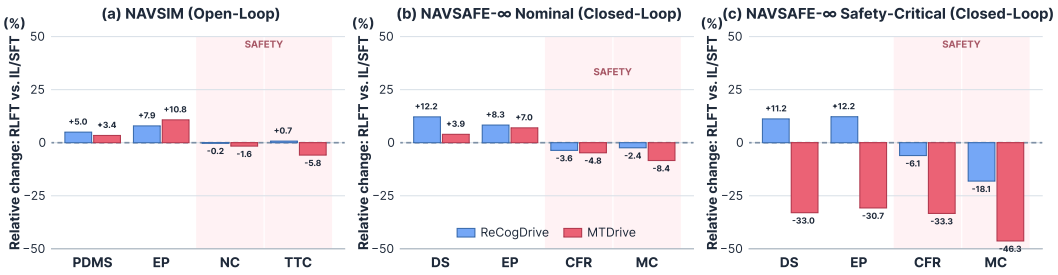

Figure 8: **RLFT reward exploitation from OL to CL.** (a) OL RLFT reward hacking trades safety for progress. (b) The trade-off persists in nominal CL planning. (c) In safety-critical scenarios, the resulting collisions terminate rollouts early and amplify the degradation in CL performance. EP: Ego Progress in (a), Event Progress in (b) and (c); NC: no at-fault collision; TTC: time to collision; DS: driving score; CFR: collision-free rate; MC: minimum clearance.

(2) Customizable scenarios. The event-design tool provides an asset zoo of vehicles, vulnerable road users, traffic-control devices, and roadside hazards, plus 3DGS assets generated from a reference image (Cao et al., 2026). Assets are inserted into a reconstructed scene with user-specified size, pose, and a static, scripted, or reactive behavior, enabling researchers to create events absent from the log and controlled variants that change one factor at a time.

(3) Closed-loop policy diagnosis. The diagnosis tool records policy-induced ego and environment states in a rollout archive, from which controlled counterfactual evaluations can be initialized. Ego-state interventions alter the vehicle's pose or motion state, whereas environment-state interventions modify the behavior of surrounding actors, such as through lead-vehicle braking or cut-in maneuvers. These controlled replays reveal whether policy improvements persist as CL rollouts move beyond the states represented in logged or perturbed demonstrations.

6 Limitations

Sim2Real Interpretation. Although photorealistic rendering narrows the visual-domain gap, the simulator cannot fully capture sensor noise, reconstruction artifacts, contact dynamics, or human driving behavior. Consequently, failures may arise from simulator mismatch, while successful rollouts may benefit from simplified agent responses. The results of NavSafe-∞ should therefore be interpreted as controlled evidence within the evaluated simulation setting rather than as direct estimates of real-world crash risk.

Scenario Evaluation Coverage. Our benchmark primarily focuses on dense urban driving scenarios (Dauner et al., 2024; Caesar et al., 2021), with highway, rural, adverse-weather, and nighttime conditions comparatively underrepresented. Expanding NavSafe-∞ to a broader range of driving environments and conditions is an important direction for future evaluation.

7 Conclusion

E2E driving safety must be assessed through policy-induced interactions. NavSafe-∞ enables it through photorealistic and event-level CL evaluation, revealing specific behavioral failures that non-reactive evaluations can obscure. Our results show that strong OL performance does not reliably translate into CL safety. The analysis attributes these failures to causal confusion, scorer bottlenecks, latent dynamic discrepancy, and reasoning-action misalignment. Common post-IL remedies do not reliably resolve these failures. Passive demonstration perturbation helps mainly when CL rollouts remain near the perturbed support, while reward hacking on expert states with RLFT carries over to CL and compounds through feedback. NavSafe-∞ will be actively maintained to facilitate future research on CL driving.

AI USE STATEMENT

In this work, we used generative AI tools for manuscript editing and assistance with software development and debugging. We did not use generative AI tools to formulate scientific claims, design the evaluation methodology, or interpret experimental results. We manually reviewed and tested AI-assisted code for correctness, and reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work evaluates autonomous-driving policies entirely in simulation and does not deploy vehicles on public roads or collect new data from road users. The benchmark uses publicly released driving logs, third-party model checkpoints, and NVIDIA NRE/NuRec artifacts under their respective licenses and access conditions; photographs used to construct animal and prohibitory-sign assets were captured by the authors. The Human Driver Reference was produced under an IRB-approved protocol by three participants, each of whom completed all 280 scenarios under one assigned seed using a game controller. All participants provided written informed consent, and separate consent was obtained for publication of the evaluation-setup photograph. No sensitive personal data are reported. The human results establish that the evaluated scenario–seed cells are solvable under the specified interface; they are not intended to characterize population-level human driving performance or between-driver variability. Finally, NAVSAFE-∞ is intended for controlled research evaluation: simulation scores should not be interpreted as certification of deployment readiness or as estimates of real-world crash risk.

REPRODUCIBILITY STATEMENT

Appendices A–H document the benchmark taxonomy and scenario provenance, asset construction, scene reconstruction and rendering, evaluation protocol and metrics, evaluated checkpoints and policy adapters, diagnostic experiments, toolbox implementation, and runtime requirements. Every policy is evaluated on the same 280 scenarios under fixed seeds 0, 1, and 1024, with means and sample standard deviations reported across seeds. Each evaluation records its resolved configuration, checkpoint identifier, requested trajectory, state trace, infractions, and terminal reason, allowing metrics to be recomputed from saved traces. Upon publication, we will release the simulator code, benchmark recipes and manifests, policy adapters, evaluation configurations, and scripts needed to reproduce the reported tables. Third-party datasets, model weights, and NVIDIA NRE/NuRec components remain subject to their original licenses and access requirements.

REFERENCES

Aarti Basant, Amlan Kar, Despoina Paschalidou, Fangyin Wei, Francesco Ferroni, Guillermo Garcia Cobo, Haithem Turki, Huan Ling, Jaewoo Seo, James Lucas, Jay Zhangjie Wu, Jialiang Wang, Jonathan Lorraine, Jun Gao, Kai He, Katarina Tothova, Kevin Xie, Michal Tyszkiewicz, Qi Wu, Riccardo de Lutio, Ruilong Li, Sanja Fidler, Seung Wook Kim, Tianchang Shen, Tianshi Cao, Tobias Pfaff, William Lew, Xindi Wu, Xuanchi Ren, Yifan Lu, Yuxuan Zhang, Zan Gojcic, and Zian Wang. NVIDIA OmniDreams: Real-time generative world model for closed-loop autonomous vehicle simulation. *arXiv preprint arXiv:2606.03159*, 2026. doi: 10.48550/arXiv.2606.03159.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11618–11628, 2020. doi: 10.1109/CVPR42600.2020.01164.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. doi: 10.48550/arXiv.2106.11810.

Tianshi Cao, Jiawei Ren, Yuxuan Zhang, Jaewoo Seo, Jiahui Huang, Shikhar Solanki, Haotian Zhang, Mingfei Guo, Haithem Turki, Muxingzi Li, Yue Zhu, Sipeng Zhang, Zan Gojcic, Sanja Fidler, and Kangxue Yin. Asset Harvester: Extracting 3D assets from autonomous driving logs for simulation. *arXiv preprint arXiv:2604.18468*, 2026. doi: 10.48550/arXiv.2604.18468.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4709–4722. PMLR, 2025. URL https://proceedings.mlr.press/v305/cao25a.html.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the Kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1268–1281, 2023. URL https://proceedings.mlr.press/v229/dauner23a.html.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28706–28719, 2024. doi: 10.52202/079017-0902.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16, 2017.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. RAP: 3D rasterization augmented end-to-end planning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.04333.

Simon Gerstenecker, Andreas Geiger, and Katrin Renz. Fail2Drive: Benchmarking closed-loop driving generalization. *arXiv preprint arXiv:2604.08535*, 2026. doi: 10.48550/arXiv.2604.08535.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jiahui Huang, Jiawei Ren, Michal Tyszkiewicz, Bjoern Haefner, Michael Shelley, Xin Kang, Seung Wook Kim, Ning Xu, Qi Wu, Janick Martinez Esturo, Shengyu Huang, Nick Schneider, Laura Leal-Taixé, Zan Gojcic, and Sanja Fidler. Instant NuRec: Feed-forward 3D Gaussian reconstruction for driving scene simulation. *arXiv preprint arXiv:2607.14203*, 2026a. doi: 10.48550/arXiv.2607.14203.

Zhiyu Huang, Johnson Liu, Rui Song, Zewei Zhou, Ruining Yang, Yun Zhang, Tianhui Cai, Hanyin Zhang, Mingxuan Gao, Valeria Xu, Jiali Chen, Yishan Shen, Yiluan Guo, Tony Xuewei Qi, and Jiaqi Ma. nuReasoning: A reasoning-centric dataset and benchmark for long-tail autonomous driving. *arXiv preprint arXiv:2605.31572*, 2026b. URL https://arxiv.org/abs/2605.31572.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Advances in Neural Information Processing Systems*, volume 37, pp. 819–844, 2024. doi: 10.52202/079017-0025.

Peter Karkus, Maximilian Igl, Yuxiao Chen, Kashyap Chitta, Jef Packer, Bertrand Douillard, Ran Tian, Alexander Naumann, Guillermo Garcia-Cobo, Shuhan Tan, Alperen Degirmenci, Alexander Popov, Nikolai Smolyanskiy, Urs Muller, Boris Ivanovic, and Marco Pavone. Beyond behavior cloning in autonomous driving: A survey of closed-loop training techniques. TechRxiv preprint, 2025. URL https://d1qx31qr3h6wln.cloudfront.net/publications/beyond_bc_survey_preprint.pdf.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023. doi: 10.1145/3592433.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Kirby_Driving_on_Registers_CVPR_2026_paper.html.

Youngwan Lee and Jongyoul Park. CenterMask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13906–13915, 2020.

Ziyang Leng, Sicheng Mo, Seth Z. Zhao, Haoyuan Cai, Yu Zeng, Rowan McAllister, and Bolei Zhou. DreamStream: Towards policy-oriented generative simulation for end-to-end driving. In *10th Annual Conference on Robot Learning*, 2026.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. MetaDrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Tianyu Li, Li Chen, Caojun Wang, Haochen Liu, Kashyap Chitta, Zhenjie Yang, Yuhang Lu, Naisheng Ye, Yihang Qiu, Yufei Wang, Luoxi Zou, Jiaxin Peng, Jin Pan, Zhaoyu Su, Andrei Bursuc, Shengbo Eben Li, Andreas Geiger, Peng Su, and Hongyang Li. World Engine: Towards the era of post-training for autonomous driving. *arXiv preprint arXiv:2606.19836*, 2026a. doi: 10.48550/arXiv.2606.19836.

Xidong Li, Mingyu Guo, Chenchao Xu, Bailin Li, Wenjing Zhu, Yangang Zou, Rui Chen, and Zehuan Wang. MTDrive: Multi-turn interactive reinforcement learning for autonomous driving. *arXiv preprint arXiv:2601.22930*, 2026b. doi: 10.48550/arXiv.2601.22930.

Yiheng Li, Cunxin Fan, Chongjian Ge, Seth Z. Zhao, Chenran Li, Chenfeng Xu, Huaxiu Yao, Masayoshi Tomizuka, Bolei Zhou, Chen Tang, Mingyu Ding, and Wei Zhan. WOMD-reasoning: A large-scale dataset for interaction reasoning in driving. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34288–34311, 2025a.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. DriveVLA-W0: World models amplify data scaling law in autonomous driving. In *International Conference on Learning Representations*, 2026c. doi: 10.48550/arXiv.2510.12796.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. ReCogDrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, 2026d. doi: 10.48550/arXiv.2506.08052.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M. Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025b.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. DiffusionDrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12037–12047, 2025. doi: 10.1109/CVPR52734.2025.01124.

Xinyu Liu, Darryl C. Jacob, Yuxin Liu, Xinsong Du, Muchao Ye, Bolei Zhou, and Pan He. Understanding real-world traffic safety through RoadSafe365 benchmark. *arXiv preprint arXiv:2602.07212*, 2026. doi: 10.48550/arXiv.2602.07212.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

NVIDIA. Isaac Sim. https://developer.nvidia.com/isaac/sim, 2026a.

NVIDIA. Neural reconstruction and 3D Gaussian splatting with NVIDIA Omniverse NuRec. https://developer.nvidia.com/omniverse/nurec, 2026b. Accessed: 2026-07-23.

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

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11410–11420, 2022.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler. Kimodo: Scaling controllable human motion generation. *arXiv preprint arXiv:2603.15546*, 2026.

Jun Saito, Jiefeng Li, Michael de Ruyter, Miguel Guerrero, Edy Lim, Ehsan Hassani, Roger Blanco Ribera, Hyejin Moon, Magdalena Dadela, Marco Di Lucca, Qiao Wang, Xueting Li, Sam Wu, Chaeyeon Chung, Yeongho Seol, Jan Kautz, Simon Yuen, and Umar Iqbal. SOMA: Unifying parametric human body models. *arXiv preprint arXiv:2603.16858*, 2026.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. SparseDriveV2: Scoring is all you need for end-to-end autonomous driving. In *European Conference on Computer Vision*, 2026. doi: 10.48550/arXiv.2603.29163.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Haochen Tian, Tianyu Li, Haochen Liu, Jiazhi Yang, Yihang Qiu, Guang Li, Junli Wang, Yinfeng Gao, Zhang Zhang, Liang Wang, Hangjun Ye, Tieniu Tan, Long Chen, and Hongyang Li. SimScale: Learning to drive via real-world simulation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36365–36374, 2026. doi: 10.48550/arXiv.2511.23369.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000. doi: 10.1103/PhysRevE.62.1805.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric and challenges. *arXiv preprint arXiv:1812.01717*, 2018. URL https://arxiv.org/abs/1812.01717.

Junli Wang, Zhihua Hua, Xueyi Liu, Zebing Xing, Haochen Tian, Kun Ma, Hangjun Ye, Guang Chen, Long Chen, and Qichao Zhang. Beyond imitation: Learning safe end-to-end autonomous driving from hard negatives. In *European Conference on Computer Vision*, 2026a. doi: 10.48550/arXiv.2605.19771. URL https://arxiv.org/abs/2605.19771.

Luobin Wang, Hongzhan Yu, Chenning Yu, Sicun Gao, and Henrik Christensen. Controllable motion generation via diffusion modal coupling. *arXiv preprint arXiv:2503.02353*, 2025.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

Yiru Wang, Anqing Jiang, Shuo Wang, Yuwen Heng, Hai Yang, Yang Chen, and Hao Sun. Do open-loop metrics predict closed-loop driving? a cross-benchmark correlation study of NAVSIM and Bench2Drive. *arXiv preprint arXiv:2605.00066*, 2026b. doi: 10.48550/arXiv.2605.00066.

Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. DriveLaW: Unifying planning and video generation in a latent driving world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. doi: 10.48550/arXiv.2512.23421.

Zhongyu Xia, Guanyu Zhu, Guo Tang, Wenhao Chen, and Yongtao Wang. HiDrive: A closed-loop benchmark for high-level autonomous driving. *arXiv preprint arXiv:2605.09972*, 2026b.

Ruining Yang, Muxing Wang, Yixiao Chen, Tongfei Guo, Yi Xu, Can Cui, Zichong Yang, Yitian Zhang, Ziran Wang, Yun Fu, and Lili Su. Post-training in end-to-end autonomous driving. *arXiv preprint arXiv:2607.08072*, 2026.

Xuemeng Yang, Licheng Wen, Tiantian Wei, Yukai Ma, Jianbiao Mei, Xin Li, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, Liang He, Yong Liu, Botian Shi, and Yu Qiao. DriveArena: A closed-loop generative simulation platform for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26933–26943, 2025. doi: 10.1109/ICCV51701.2025.02500.

Yuxuan Zhang, Katarína Tóthová, Zian Wang, Kangxue Yin, Haithem Turki, Riccardo de Lutio, Yen-Yu Chang, Or Litany, Sanja Fidler, and Zan Gojcic. DiffusionHarmonizer: Bridging neural reconstruction and photorealistic simulation with online diffusion enhancer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2602.24096.

Seth Z. Zhao, Luobin Wang, Hongwei Ruan, Yuxin Bao, Yilan Chen, Ziyang Leng, Abhijit Ravichandran, Honglin He, Zewei Zhou, Xu Han, Abhishek Peri, Zhiyu Huang, Pranav Desai, Henrik Christensen, Jiaqi Ma, and Bolei Zhou. BridgeSim: Unveiling the OL-CL gap in end-to-end autonomous driving. *arXiv preprint arXiv:2604.10856*, 2026a. doi: 10.48550/arXiv.2604.10856.

Zongchuang Zhao, Xin Zhou, Tianyang Xu, Zhengyang Sun, Kaixuan Zhou, Honglin Li, Dingkang Liang, and Xiang Bai. SimWAM: A simple world action model for end-to-end autonomous driving. *arXiv preprint arXiv:2608.07468*, 2026b. URL https://arxiv.org/abs/2608.07468.

Hongyu Zhou, Longzhong Lin, Jiabao Wang, Yichong Lu, Dongfeng Bai, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. HUGSIM: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(4): 4673–4691, 2026a. doi: 10.1109/TPAMI.2025.3647952.

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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 27920–27956. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2843fccca5bedd369a4764848b9bd546-Abstract-Conference.html.

Zewei Zhou, Ruining Yang, Xuewei Qi, Yiluan Guo, Sherry X. Chen, Tao Feng, Kateryna Pistunova, Yishan Shen, Lili Su, and Jiaqi Ma. SpanVLA: Efficient action bridging and learning from negative-recovery samples for vision-language-action model. *arXiv preprint arXiv:2604.19710*, 2026b.

Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

APPENDIX

The appendix is organized as follows.

- Appendix A: Scenario taxonomy, coverage, and scenario gallery (Section 3.1).
- Appendix B: Asset library used for scenario interventions.
- Appendix C: Dataset source, photorealistic reconstruction, and rendering fidelity (Section 3.3).
- Appendix D: Closed-loop protocol, event contracts, metrics, and benchmark quality controls (Section 3.2).
- Appendix E: Evaluated policies and the human driver reference (Table 2).
- Appendix F: OL–CL comparison and full leaderboard, case-selection protocol for the failure investigations, and diagnostic settings (Sections 4.1, 4.2, and 4.3).
- Appendix G: Toolbox implementation (Section 5).
- Appendix H: Runtime and resource requirements.

A NAVSAFE-∞ SCENARIO TAXONOMY AND GALLERY

This appendix supplements Section 3.1 with detailed operational criteria, nuPlan screening labels, scenario-construction routes, and per-event scenario counts. Together with Figure 2, the additional examples provide one rendered scenario for each of the 28 event types.

A.1 TAXONOMY DEFINITIONS AND SCENARIO COVERAGE

Each of the 280 scenarios belongs to exactly one of 28 event types adapted from RoadSafe365 (Liu et al., 2026) (Table 3). An event type is defined by its operational inclusion criterion, which specifies the event that must occur within the evaluated window. nuPlan scenario labels (Caesar et al., 2021) are used only to retrieve candidate windows and do not determine membership.

Scenarios are obtained through one of two routes. For 19 *direct* event types, nuPlan scenario labels expose the defining event, and scenarios are retrieved from the logs. For the remaining nine *recipe* event types (C-7, C-10, V-8, V-10, V-11, R-2, R-3, R-4, and I-3), no label retrieves the event reliably, so every scenario is constructed from a versioned recipe (Appendix G.2). A recipe either retains the host log unchanged or applies an *intervention*, namely a controlled actor, asset, or traffic-control edit to a compatible log.

Each event type contains ten scenarios. The 19 direct event types contribute 190 natural scenarios. The nine recipe event types contribute 90 scenarios, of which 62 are intervened and 28 replay the host log unchanged (all 20 scenarios of C-10 and V-11, and the natural scenarios of C-7, V-10, and R-2). The benchmark therefore contains 218 natural and 62 intervened scenarios. The benchmark manifest records the source log token, event type, provenance, recipe identifier, and event contract of every scenario.

Table 3: Operational taxonomy, nuPlan screening labels, and coverage route. *Direct* event types are taken from unmodified nuPlan logs (screened by scenario type where one applies); *Recipe* event types are constructed from versioned recipes.

Event type	Operational inclusion criterion	nuPlan screening labels	Coverage route	N (nat./int.)
Traffic Crashes				
C-1 Rear-End	Same-lane longitudinal conflict in which inadequate headway or braking can produce front-to-rear contact.	`stopping_with_lead`; `behind_long_vehicle`; `near_long_vehicle`; `stopping_at_traffic_light_with_lead`; `stopping_at_stop_sign_with_lead`; `stationary_at_traffic_light_with_lead`; `accelerating_at_traffic_light_with_lead`.	Direct.	10/0

Continued on next page

Event type	Operational inclusion criterion	nuPlan screening labels	Coverage route	N (nat./int.)
C-2 Angle/T-Bone	Crossing paths create a lateral-impact conflict at an intersection or access point.	`crossed_by_vehicle`; `starting_protected_cross_turn`.	Direct.	10/0
C-3 Sideswipe	Overlapping lateral motion between adjacent vehicles creates same-direction side-contact risk.	`changing_lane_with_lead`; `changing_lane_with_trail`; `near_high_speed_vehicle`.	Direct.	10/0
C-4 Multi-Vehicle	At least four interacting vehicles permit a primary conflict to propagate into secondary braking or collision risk.	`near_multiple_vehicles`.	Direct.	10/0
C-5 Intersection	The safety-critical conflict and required maneuver occur within an intersection footprint.	`traversing_intersection`; `on_intersection`; `starting_left_turn`; `starting_right_turn`; `starting_low_speed_turn`; `starting_high_speed_turn`; `starting_protected_noncross_turn`.	Direct.	10/0
C-6 Roadway Departure	Ego must re-enter the through lane from a curbside or pickup/drop-off position.	`traversing_pickup_dropoff`; `on_pickup_dropoff`.	Direct.	10/0
C-7 Head-On	Opposing traffic competes for insufficient longitudinal or lateral clearance in a shared corridor.	`traversing_narrow_lane`.	Recipe: natural log plus targeted oncoming actor.	2/8
C-8 Pull-Over	Ego must leave the through lane into a curbside or parking position, where the low-speed lateral transition creates contact risk with objects or road users near the boundary.	`on_carpark`.	Direct.	10/0
C-9 Single-Vehicle	No other traffic participant is present in the ego's conflict region, so any safety event arises from the ego's own control, the surface, or a fixed object.	None.	Direct.	10/0
C-10 Wrong-Way Crash	Ego enters a lane assigned to the opposite direction of travel, so oncoming traffic produces a head-on conflict.	None.	Recipe: natural road geometry.	10/0
Traffic Violations				
V-1 Red-Light/Stop-Sign	A traffic light or stop sign governs ego entry at a control line.	`traversing_traffic_light_intersection`; `on_traffic_light_intersection`; `on_stopline_traffic_light`; `starting_straight_traffic_light_intersection_traversal`; `stopping_at_traffic_light_without_lead`; `stationary_at_traffic_light_without_lead`; `accelerating_at_traffic_light`; `accelerating_at_traffic_light_without_lead`; `on_stopline_stop_sign`; `starting_straight_stop_sign_intersection_traversal`; `stopping_at_stop_sign_without_lead`; `stopping_at_stop_sign_no_crosswalk`; `accelerating_at_stop_sign`; `accelerating_at_stop_sign_no_crosswalk`.	Direct.	10/0
V-2 Failure to Yield	Ego must reject or accept a gap under an explicit right-of-way obligation.	`starting_unprotected_cross_turn`; `starting_unprotected_noncross_turn`; `on_all_way_stop_intersection`.	Direct.	10/0
V-3 Unsafe Lane Change	A lane transfer or pass is unsafe because of clearance, boundary, signaling, or maneuver-completion constraints.	`changing_lane`; `changing_lane_to_left`; `changing_lane_to_right`.	Direct.	10/0
V-4 Reckless/Aggressive	Ego action exceeds a preregistered kinematic or interaction-risk envelope without necessity.	`high_lateral_acceleration`; `high_magnitude_jerk`.	Direct.	10/0
V-5 Speeding	Ego speed violates the applicable posted or scenario-specific limit.	`high_magnitude_speed`; `medium_magnitude_speed`; `low_magnitude_speed`.	Direct.	10/0

Continued on next page

Event type	Operational inclusion criterion	nuPlan screening labels	Coverage route	N (nat./int.)
V-6 Tailgating	Ego follows a lead below the declared time-headway or stopping-distance threshold.	`following_lane_with_lead`; `following_lane_with_slow_lead`.	Direct.	10/0
V-7 Lane Maintenance	Uncommanded lateral drift crosses or persistently approaches the lane or drivable-corridor boundary.	`following_lane_without_lead`.	Direct.	10/0
V-8 Illegal Turn	A turn is geometrically feasible but prohibited by the applicable control or lane rule.	`starting_u_turn`.	Recipe: prohibitory-sign intervention.	0/10
V-9 Blocking Intersection	Ego enters without downstream clearance or remains stopped in a conflict area without necessity.	`stationary`; `stationary_in_traffic`.	Direct.	10/0
V-10 Unsafe Merge	Ego enters a traffic stream without the declared longitudinal and lateral safety margins.	None.	Recipe: natural log plus targeted merge actor.	3/7
V-11 Wrong-Way Driving	Ego traverses a lane against its lawful direction even without a collision.	None.	Recipe: natural road geometry.	10/0
Vulnerable Road User Crashes				
R-1 Pedestrian-Involved	A pedestrian occupies, approaches, or is occluded near the ego's potential path and requires yielding or clearance.	`waiting_for_pedestrian_to_cross`; `near_pedestrian_on_crosswalk`; `near_pedestrian_on_crosswalk_with_ego`; `near_multiple_pedestrians`; `behind_pedestrian_on_driveable`; `behind_pedestrian_on_pickup_dropoff`; `near_pedestrian_at_pickup_dropoff`; `traversing_crosswalk`; `on_stopline_crosswalk`; `stopping_at_crosswalk`; `stationary_at_crosswalk`; `accelerating_at_crosswalk`.	Direct.	10/0
R-2 Bicycle-Involved	A cyclist is followed, crossed, passed, or encountered in a shared conflict region.	`behind_bike`; `crossed_by_bike`; `near_multiple_bikes`.	Recipe: natural log plus targeted cyclist actor.	3/7
R-3 Micromobility-Involved	A micromobility device or other atypical road user, such as a delivery robot, enters shared space.	None.	Recipe: targeted micromobility actor.	0/10
R-4 Animal-Involved	An animal enters or approaches the drivable corridor without traffic-rule-compliant behavior.	None.	Recipe: targeted animal actor.	0/10
Traffic Incidents				
I-1 Obstruction/Hazard	An unmapped static object obstructs the nominal path and requires stopping or a bounded avoidance maneuver.	`near_trafficcone_on_driveable`; `near_barrier_on_driveable`.	Direct.	10/0
I-2 Work Zone	Temporary signs, barriers, or workers alter the valid corridor relative to the base map.	`near_construction_zone_sign`.	Direct.	10/0
I-3 General Incident	A preregistered safety event not represented by another event type requires a specific fallback response.	None.	Recipe: targeted incident.	0/10

A.2 Canonical Scenario for Each Event Type

Figure 9 complements the 12 event types visualized in Figure 2 by showing one scenario for each of the remaining 16 event types. Together, the two figures cover all 28 taxonomy leaves. The gallery highlights the diversity of road layouts, traffic participants, and traffic controls, while Table 3 specifies the operational inclusion criterion, nuPlan screening labels, coverage route, and natural/intervened scenario counts for each event type.

B NavSafe-∞ Asset Library

Intervened scenarios insert actors and objects that are absent from the host log. This appendix describes how these assets are constructed and lists the asset library. Dynamic actors follow a frozen pose track or a declared controller, and static objects are placed at a fixed pose (Appendix G.2).



Figure 9: **Additional NAVSAFE-∞ scenario examples.** Complementing the 12 event types shown in Figure 2, this figure presents one scenario for each of the remaining 16 event types; together, the two figures cover all 28 taxonomy leaves. Each panel shows a rendered front-camera observation at event onset under seed 0 using a common crop. The displayed scenarios were fixed before rendering and were not selected based on rendering quality or policy outcome.



Figure 9: **Additional NAVSAFE-∞ scenario examples (continued).**

B.1 ASSET CONSTRUCTION

All assets are 3DGS models generated with Asset Harvester (Cao et al., 2026). Given segmented images of an object, Asset Harvester synthesizes 16 consistent views with a multi-view diffusion model (SparseViewDiT) and lifts them to 3D Gaussians (TokenGS). Because unobserved views are generated, an asset approximates rather than reproduces the geometry and texture of its source object. The assets differ only in the source of the input images.

Vehicles from the source logs. We build a pipeline on Asset Harvester that converts vehicles in the nuPlan logs. Each scenario is processed as its four 5 s clips. In each clip, vehicle tracks are projected into the calibrated cameras using their 3D bounding boxes; observations that are invalid, too small, or heavily occluded are discarded, and the remaining ones are cropped and segmented. We retain cars, trucks, buses, and trailers with at least two valid observations and a net displacement of at least 2 m over the 20 s scenario, and keep at most ten per scenario, ranked by their closest observation distance. Parked vehicles are excluded because the scene reconstruction already observes them from

multiple viewpoints. A track that appears in several clips is converted once, from the clip in which it is observed at the closest range. Each generated asset is rotated to the NuRec coordinate convention, and assets that are empty or whose dimension ratios deviate from the source bounding box by more than 35% are discarded. A retained asset replaces the reconstructed appearance of its source track so that the vehicle can be rendered from viewpoints not observed in the log. The asset is scaled to the bounding box of the track, and the identity and motion of the track are unchanged.

Standalone assets. Pedestrians, cyclists, delivery robots, and additional vehicles are converted from single nuPlan camera frames, and animals and prohibitory signs from photographs taken by the authors. Each image is converted with the single-image mode of Asset Harvester, which segments the object with its built-in segmentation model and estimates the camera viewpoint before multi-view generation and lifting. Unlike in-log vehicles, these assets are not bound to a track and can be inserted into any scenario.

Gait animation. Inserted pedestrians and animals are static 3DGS assets, and the NuRec render interface accepts only one rigid pose per actor per frame. To keep them from sliding across the road in a frozen posture, we bake limb motion into the geometry as a small bank of copies, each frozen at a different phase of one gait cycle. For pedestrians, we fit the SOMA body model (Saito et al., 2026) to the asset's Gaussians, starting from walking poses and refining with Chamfer distance. We then bind each Gaussian to the skeleton with skinning weights taken from the nearest body surface and pose the body at K evenly spaced frames of a Kimodo (Rempe et al., 2026) walking motion. For animals, we use the SMAL model (Zuffi et al., 2017) with a procedurally generated trot, solving the knee angle at each frame to keep the stance foot planted. Forward travel and heading drift are removed from every phase, so each copy carries only limb motion and the trajectory still comes from the simulator. At render time, every phase is registered as a separate track. Each frame, phase $\lfloor dK/s \rfloor \bmod K$ is placed at the actor pose and the others are moved out of view, where d is the distance the actor has traveled and s is the stride measured from the baked cycle. Selecting phases by distance rather than by time keeps the feet planted at whatever speed the actor policy chooses. Physics, collision checking, and metrics still see a single track per actor, so the gait affects rendering only. Because each phase occupies render-server GPU memory, the benchmark uses three phases for pedestrians and five for animals.

B.2 Asset Inventory

Table 4 lists the asset library by category, source, and the event types that use it. Figures 10 and 11 show the assets rendered in isolation.

Table 4: NavSafe-∞ asset inventory. nuPlan (track): converted from the vehicle tracks of the source logs; nuPlan (frame): converted from single nuPlan camera frames; Photo: converted from photographs taken by the authors.

Category	# Assets	Source	Used by event types
Pedestrians	5	nuPlan (frame)	R-3
Cyclists	4	nuPlan (frame)	R-2
Delivery robots	2	nuPlan (frame)	R-3
Animals	4	Photo	R-4
Vehicles	6	nuPlan (frame)	C-7, V-10, I-3
Vehicles	up to 10 per scenario	nuPlan (track)	All
Prohibitory signs	2	Photo	V-8

C Dataset Source and Photorealistic Reconstruction

This appendix supplements Section 3.3. It describes how each scenario is reconstructed and rendered, and how the fidelity comparison in Table 1 is conducted.

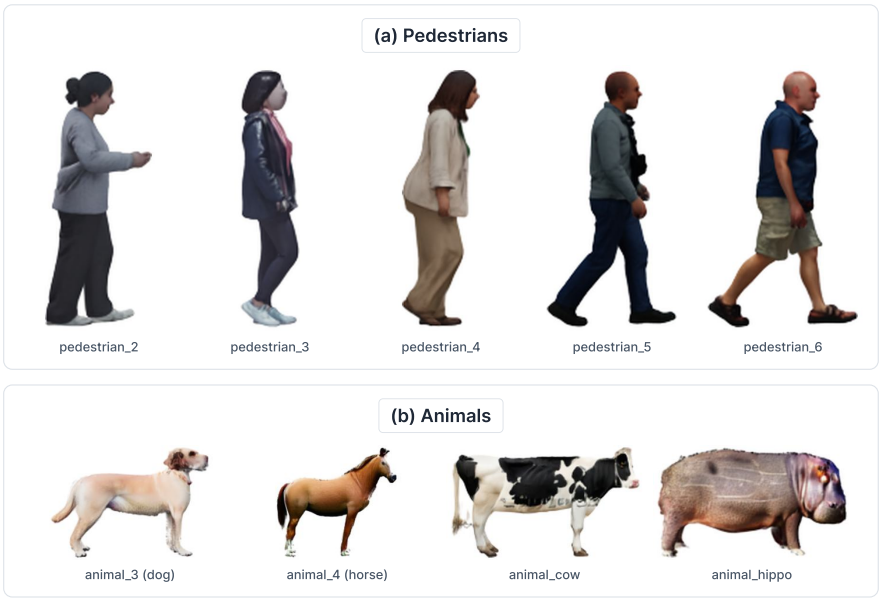

Figure 10: Deformable assets animated with walking gaits: (a) pedestrians and (b) animals.

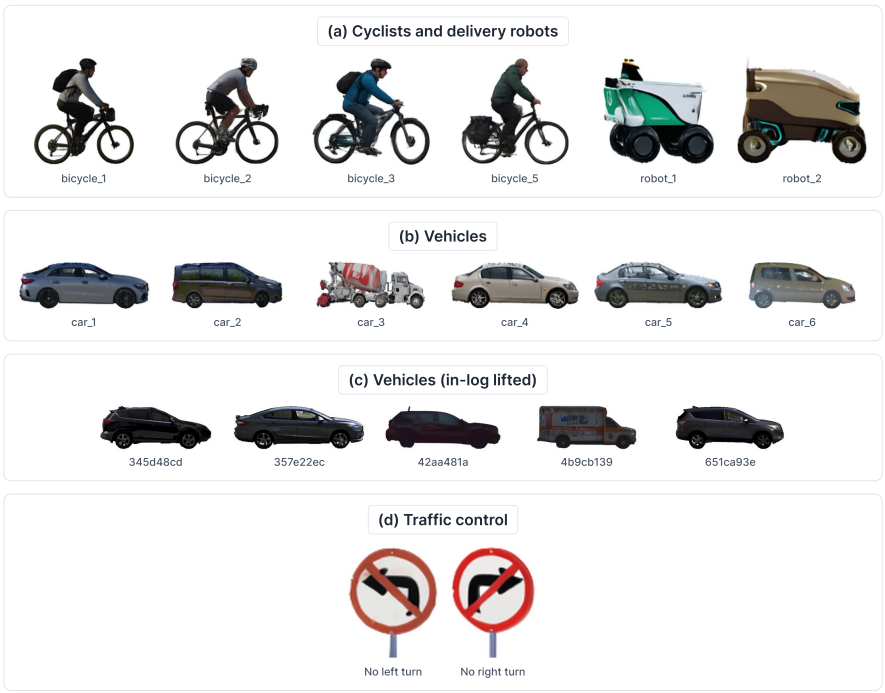

Figure 11: Rigid assets: (a) cyclists and delivery robots, (b) vehicles converted from single nuPlan frames, (c) vehicles converted from the vehicle tracks of the source logs (five examples), and (d) prohibitory signs.

C.1 Source Logs and Evaluation Bundles

Each scenario is a 20 s window from the nuPlan test split (Caesar et al., 2021). Windows are drawn only from the 147 test logs that provide full-rate camera images, since the remaining test logs contain 2 Hz images that are too sparse for reconstruction. For every window, the manifest stores the nuPlan

scenario token, source log and map, start and end timestamps, event type, recipe hash, and whether an edit is active. Each scenario is packaged as a self-contained bundle containing synchronized ego state, tracked boxes, traffic-light states, route and map context, reconstruction metadata, and referenced assets. Simulation and scoring read this symbolic state, whereas the policy receives only rendered sensor data and navigation inputs.

C.2 Four-Segment NuRec Reconstruction

Each scenario is reconstructed as four independently trained 5 s NuRec models (NVIDIA, 2026b; Huang et al., 2026a) rather than as one model over the full window: a single 20 s model spreads the same Gaussian budget over four times as many frames and a correspondingly larger scene extent, which in our preliminary experiments visibly blurred static structure. Reconstruction uses the NRE 26.04 container (`nvcr.io/nvidia/nre/nre-ga@sha256:6e0caa70a9148490552520c3dde9ee665c8d094ca10b814601cb8bb306567c90`) with the `car2sim_6cam` configuration. Each segment is trained on five calibrated nuPlan cameras (F0, B0, L0, L1, and R0), using every fourth frame. The top LiDAR initializes the road Gaussians but provides no training rays, since LiDAR supervision yielded only a marginal improvement in visual fidelity at a substantially longer training time. In addition to the background, road, and dynamic-object layers of `car2sim_6cam`, we add a layer for static boxed objects (parked vehicles, cones, barriers, and signs), so that every annotated object, moving or static, is a separate node that can be moved, removed, or replaced at render time. To support novel-view rendering of moving vehicles, up to ten of the nearest valid in-log vehicles per scenario are further replaced with assets from the NavSafe-∞ asset library (Appendix B). Each segment is optimized for 160k iterations with at most 5M Gaussians per segment; Gaussian addition and relocation stop at 140k iterations, and the learning rate follows a step-wise cosine annealing schedule over all 160k iterations.

At render time, each segment is placed in the shared log frame by a stored translation from its local reconstruction frame. Because a segment covers only the road that the logged ego traverses within its 5 s window, segments are selected by position rather than by simulation time: at each frame, the ego position is matched to the nearest pose on the logged ego trajectory, and the timestamp of that pose determines both the segment to render (the one whose window contains it) and the time at which it is queried. The ego thus enters the next segment when it reaches the corresponding location on the road, regardless of its speed relative to the logged ego, and the switch is made without blending. Because boundaries follow the log rather than event timing, a switch may fall within an event window; such switches are not excluded from evaluation.

C.3 Rendering and Harmonizer

The renderer serves all four segments and renders each camera at its native resolution (1920×1080). Because each scene is a 3DGS reconstruction, any camera rig κ can be rendered by changing its intrinsics and extrinsics (Section 3.1), including cameras that were not used for training. Each policy adapter selects, crops, resizes, and normalizes the cameras used by its policy: ReCogDrive, MTDrive, SimWAM, DriveLaW, and DriveVLA–W0 use F0; LTF, DiffusionDrive, SparseDriveV2, and GTRS use F0, L0, and R0; DrivoR and RAP use F0, L0, R0, and B0; and AutoVLA uses F0, L1, and R1. DiffusionHarmonizer (Zhang et al., 2026) post-processes every rendered frame independently, without temporal conditioning, to reduce reconstruction artifacts and the appearance mismatch of inserted assets.

C.4 Rendering-Fidelity Evaluation

Evaluation set. All renderers are evaluated on the front camera (`CAM_F0`) over 392 reconstructed 20 s scenarios, which include the 280 benchmark scenarios, and each rendered frame is compared with the recorded frame of the same timestamp. FID and FVD use the 15,680 frames sampled at 2 Hz, and FD_π uses the 4,163 NAVSIM `navtest` tokens (Dauner et al., 2024) among them, since each policy requires a complete NAVSIM scene as input. NavSafe-∞ renders along the logged ego trajectory at 1920×1080 with the Harmonizer enabled, as in closed-loop evaluation. DriveArena (Yang et al., 2025) generates each frame at 400×224 with its WorldDreamer checkpoint, conditioned on the BEV layout, a text caption, and the previously generated frame; the reference frame is reset at the start of each scenario, and the three logs in DriveArena's training split are

excluded. DreamStream (Leng et al., 2026) generates frames at 1280×704 with its Wan 2.2 (Team Wan et al., 2025) (5B) autoregressive video model, conditioned on the logged first frame, per-frame traffic layouts (the projected HD map and 3D agent boxes), and a text prompt. FVD uses the 1,568 clips formed by these frames.

Metrics. FID (Heusel et al., 2017) uses Inception-v3 (Szegedy et al., 2016) features (clean-fid (Parmar et al., 2022)) on individual frames, with the recorded frames resized to the resolution of each renderer: 1920×1080 for NAVSAFE-∞ and DreamStream, whose frames are upsampled from 1280×704, and 400×224 for DriveArena. FVD (Unterthiner et al., 2018) uses I3D (Carreira & Zisserman, 2017) features on clips of 16 consecutive 2 Hz frames with a stride of 8, resized to 224×224. For policy k with scene-context encoder ϕ_k, let (μ_r, Σ_r) and (μ_g, Σ_g) be the mean and covariance of ϕ_k features on recorded and rendered frames. Following Leng et al. (2026),

$$\mathrm{FD}_\pi^k = \frac{\|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\big(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\big)}{\mathrm{Tr}(\Sigma_r)}, \qquad \mathrm{FD}_\pi = \frac{1}{K}\sum_{k=1}^{K} \mathrm{FD}_\pi^k,$$

with $K = 5$ policies (DrivoR, DiffusionDrive, LTF, RAP, and SparseDriveV2). ϕ_k is the scene-context feature that each policy passes to its planning head, and Table 1 reports FD_π multiplied by 100. Because FID and FVD depend on the sample size, all renderers are scored on the same frames and clips.

D Evaluation Protocol and Metrics

This appendix supplements Sections 3.1 and 3.2. It specifies how an episode is executed, when it terminates, and how its trace is scored.

D.1 Closed-Loop Protocol

Timing and warm-up. The simulator advances at 10 Hz. Each episode begins with a log-replay warm-up of 20 frames (2 s) for direct scenarios and 8 frames (0.8 s) for recipe scenarios, after which every frame is closed loop. The shorter warm-up of recipe scenarios is fixed by the recipe, because the placement of each inserted actor is solved relative to the hand-off frame. Every policy receives the same warm-up on a given scenario.

Ego control. The policy replans every five frames (2 Hz). A shared tuned-LQR lateral controller with a kinematic bicycle model executes each predicted trajectory, so all policies are evaluated through the same plant.

Semi-reactive traffic. Surrounding traffic runs in a *semi-reactive* mode ported from MetaDrive's reactive scenario traffic (Li et al., 2022). Every vehicle starts in log replay, and a moving vehicle is handed, one-way, to the Intelligent Driver Model (IDM) (Treiber et al., 2000) once it is oriented within 90° of the ego, lies more than 1 m behind and within 15 m laterally of the ego, and has at least 5 m of logged track remaining. Unlike MetaDrive, which tests these conditions only when a vehicle spawns, we re-test them every frame, so a vehicle that the ego overtakes becomes reactive once it falls behind. Once a vehicle's logged track is exhausted, these conditions no longer apply and it is handed to IDM unconditionally. Only following vehicles are therefore made reactive: vehicles ahead of the ego, pedestrians, cyclists, and static objects replay their logs, and recipe-controlled actors retain their authored behaviors. IDM uses MetaDrive's constants (10 m minimum gap, 1.5 s time headway, and 20 m lead-vehicle search range), evaluated in MetaDrive's km/h speed convention.

Seeds. Each policy is evaluated on all 280 scenarios under seeds 0, 1, and 1024. The seed initializes the Python, NumPy, and PyTorch random number generators before the environment and the policy are constructed. Because the simulator retains minor nondeterminism, results can differ across seeds even for deterministic policies.

D.2 Event Contracts and Termination Rules

For route-reaching tasks, the goal is reached when the ego is simultaneously within a 10 m radius of the target and has at most $\max(2\,\mathrm{m}, 0.01 L_i)$ of route arc remaining, where L_i is the route length of scenario i; arc progress is tracked by a monotone cursor, so leaving and rejoining the route does

not retro-credit un-driven stretches. The same predicate is evaluated live and post hoc, so the two cannot disagree. An episode ends at its first terminal event: any contact, 0.5 s continuously outside the drivable surface, more than 6 m of wrong-way travel, running a red light, an illegal turn, or 5 s below 0.1 m/s (deadlock). The deadlock counter is suspended while the ego holds lawfully at a signal or stop line. Following NAVSIM (Dauner et al., 2024), a collision is charged to the ego only if the ego is moving and the other agent lies ahead of it; rear-end contacts by following agents and contacts while the ego is stationary terminate the episode without being charged. Every episode has a time budget of 60 s; an episode that reaches neither the goal nor another terminal event ends when the budget expires. Because the episode ends at the goal, and an ego that strays far from the logged path typically leaves the drivable surface and is terminated, rollouts remain close to the reconstructed region. Episodes interrupted by simulator or renderer failures are rerun rather than scored as policy failures.

For most event types, the contract uses the termination rule above, measures Event Progress (EP) as monotone progress along the logged route (100 on goal completion), and counts an episode as successful when it reaches the goal with no evaluated infraction. Four of the 28 event types modify this contract, listed in Table 5; V-1 appears as two rows because its traffic-light and stop-sign precursors take different contracts.

Table 5: Event types whose contract differs from the default.

Event type	Modification
C-3 Sideswipe	When the ego is assigned a lane change, success additionally requires a lane transition with at least 0.5 m lateral offset held for five frames.
V-1 Red-Light	The goal and deadlock terminals are suspended. Crossing the stop line fails the episode; holding for the full window succeeds. EP is the fraction of the window held before crossing.
V-1 Stop-Sign	Success additionally requires 3 s below 0.3 m/s before the stop line when the approach is observed.
V-8 Illegal Turn	The goal terminal is suspended and the outside-route-lanes infraction is waived, since the logged route *is* the prohibited turn and a compliant policy must leave it. Entering the prohibited exit lane fails the episode; clearing the intersection without entering the prohibited lane succeeds.
I-2 Work Zone	Wrong-way travel is neither a terminal nor an infraction, since the prescribed bypass uses the opposing lane.

D.3 Metrics and Aggregation

All metrics are computed from the saved closed-loop trace. For scenario i and seed s,

$$\mathrm{DS}_{i,s} = \mathrm{EP}_{i,s}\,(0.50)^{n_{\mathrm{ped}}}(0.60)^{n_{\mathrm{veh}}}(0.65)^{n_{\mathrm{layout}}}(0.70)^{n_{\mathrm{red}}+n_{\mathrm{timeout}}+n_{\mathrm{yield}}}(0.80)^{n_{\mathrm{stop}}}\left(1-\frac{p_{\mathrm{outside}}}{100}\right),$$

where n_{ped}, n_{veh}, and n_{layout} count at-fault collisions with pedestrians, vehicles, and static objects; n_{red}, n_{timeout}, and n_{yield} count red-light violations, scenario timeouts, and failures to yield to an emergency vehicle; n_{stop} counts stop-sign violations; and p_{outside} is the percentage of driven distance outside the drivable area, excluding the first 0.5 m of each excursion. The coefficients follow CARLA (Dosovitskiy et al., 2017) Leaderboard 2.0 and Bench2Drive (Jia et al., 2024). An episode succeeds ($S_{i,s} = 1$) if it satisfies its event contract: it completes the event and incurs no infraction other than a minimum-speed infraction. A timeout therefore counts as a failure, as does any nonzero p_{outside}. SR is the percentage of successful episodes, $\mathrm{SR}_s = \frac{100}{N}\sum_i S_{i,s}$, over the $N = 280$ scenarios.

DE is the mean ratio, in percent, between the ego speed and the mean speed of background vehicles at 20 route checkpoints spaced at 5% of the route. It is undefined, and excluded from the average, when the ego reaches no checkpoint or the median background speed is below 1.5 m/s, since the ratio is then dominated by queued traffic. Individual checkpoints whose ratio exceeds 1000% are additionally discarded before averaging. Comfort is the percentage of non-overlapping 2 s windows in which the smoothed longitudinal and lateral acceleration, jerk, yaw rate, and yaw acceleration all stay within the nuPlan comfort bounds (Caesar et al., 2021).

DS, SR, and Comfort are averaged over the 280 scenarios, and DE over the scenarios where it is defined. The Ability Score of a category is the unweighted mean of the success rates of its event types. We report the mean and the sample standard deviation of each metric over the three seeds.

D.4 Benchmark Quality Controls and Scope

Every benchmark cell is solvable under the specified evaluation interface: three participants, each assigned one evaluation seed, collectively complete all 840 scenario–seed cells in the Human Driver Reference (Appendix E.2). All scenarios are drawn from the nuPlan test logs underlying the NAVSIM `navtest` and `navhard` splits, which share no logs or scenes with the NAVSIM training split `navtrain`. Each evaluation records the resolved configuration, seed, state trace, infractions, and terminal reason, together with the checkpoint checksum for policy runs, allowing every reported score to be recomputed from its saved trace. These controls make the evaluation auditable while limiting its conclusions to bounded, taxonomy-defined events within the simulator.

E Evaluated Policies and Human Reference

This appendix supplements Table 2 with the identity and inputs of each evaluated policy and the construction of the Human Driver Reference.

E.1 Evaluated Policies and Checkpoints

Table 6 lists the 20 learned policies, which are described in Section 2.2, and the privileged reference PDM-Closed (Dauner et al., 2023), a rule-based planner that reads ground-truth simulator state and selects among rolled-out proposals. By how they generate the ego trajectory, the 20 learned policies fall into four families:

- *Regression*, which predicts one or several trajectories from learned queries (Dauner et al., 2024; Kirby et al., 2026): LTF, LTF (SimScale), DrivoR, DrivoR (SimScale, 134k), and RAP.
- *Vocabulary selection*, which scores a fixed set of candidate trajectories (Li et al., 2025b; Sun et al., 2026): SparseDriveV2, GTRS-Dense–V2-99, and GTRS-Dense–V2-99 (SimScale).
- *Diffusion or flow matching*, which generates trajectories by iteratively denoising sampled motions (Wang et al., 2025; Liao et al., 2025), including VLA backbones with diffusion or flow-matching action heads (Zhou et al., 2026b): DiffusionDrive, DiffusionDrive (SimScale), DiffusionDrive (BeyondDrive), ReCogDrive–2B–IL, ReCogDrive–2B–RL, DriveLaW, DriveVLA–W0, SimWAM, and SimWAM–RL. DiffusionDrive denoises from 20 anchors obtained by k-means clustering of training trajectories, using only two denoising steps, so its outputs remain close to these anchors; Section 4.3 therefore treats it as anchor-based.
- *Autoregressive token decoding* (Zhou et al., 2025): AutoVLA, MTDrive–SFT, and MTDrive–mtGRPO.

Released weights are evaluated without retraining, and each adapter preserves the policy's declared sensor inputs, normalization, and trajectory parameterization (camera preprocessing in Appendix C.3; adapter interface in Appendix G.3). The evaluation manifest pins the model revision, checkpoint checksum, adapter revision, sensor configuration, inference parameters, and controller configuration of every row. Conclusions about a post-training intervention (Yang et al., 2026), such as RL fine-tuning or demonstration perturbation, are drawn only from checkpoint pairs that share the same architecture and differ in that intervention (e.g., ReCogDrive–2B–IL and ReCogDrive–2B–RL, or DiffusionDrive and DiffusionDrive (SimScale)); other comparisons are descriptive. All checkpoints are also mirrored in the NavSafe-∞ model zoo.

Table 6: Evaluated policies, grouped as in Table 7. Inputs list the rendered cameras (F0: front; B0: rear; L0/R0 and L1/R1: left/right). The last column gives the NAVSIM version each released checkpoint was trained for. Every post-training pair is version-matched, so an intervention is never confounded with a change of training split. All policies are executed through the same controller at the same replan rate, so the parameterization affects only how far ahead each model plans, not how its output is tracked. †RAP predicts 10 poses over 5 s at 2 Hz and AutoVLA predicts 10 poses; both retain the v1 metric heads.

Policy	Group	Inputs	Description	NAVSIM
PDM-Closed	Privileged	Simulator state	Rule-based planner that scores rolled-out proposals using ground-truth scene state.	Rule-based
LTF	IL	F0/L0/R0	NAVSIM Latent TransFuser baseline that regresses an ego trajectory from logged demonstrations.	v1
DiffusionDrive	IL	F0/L0/R0	Truncated diffusion planner that generates multimodal ego trajectories.	v1
DrivoR	IL	F0/L0/R0/B0	Register-based planner trained on logged NAVSIM demonstrations.	v1
RAP	IL	F0/L0/R0/B0	Planner trained with rasterized counterfactual recovery and cross-agent views and raster-to-real alignment.	v1†
SparseDriveV2	IL	F0/L0/R0	Vocabulary-based planner that scores factorized paths and velocity profiles in a coarse-to-fine manner.	v1
GTRS-Dense–V2-99	IL	F0/L0/R0	Dense-vocabulary trajectory scorer; V2-99 denotes its VoVNet-99 (Lee & Park, 2020) image backbone. Scores 16,384 candidate trajectories with per-metric EPDMS heads.	v2
ReCogDrive–2B–IL	IL	F0	2B-parameter VLA with a diffusion trajectory planner, before reward-based post-training.	v1
MTDrive–SFT	IL	F0	Supervised VLA stage used as the initialization for MTDrive's multi-turn reward post-training.	v1
ReCogDrive–2B–RL	RLFT	F0	ReCogDrive–2B–IL after DiffGRPO reward-based post-training.	v1
MTDrive–mtGRPO	RLFT	F0	MTDrive–SFT after multi-turn group-relative reward optimization on offline scenes.	v1
SimWAM–RL	RLFT	F0	SimWAM action expert after flow-based group-relative post-training.	v1
AutoVLA	RLFT	F0/L1/R1	VLA with adaptive reasoning and reinforcement fine-tuning; only the final released stage is evaluated.	v1†
DiffusionDrive (SimScale)	Perturbation	F0/L0/R0	DiffusionDrive co-trained on real logs and SimScale perturbed scenes with pseudo-expert supervision.	v1
DrivoR (SimScale, 134k)	Perturbation	F0/L0/R0/B0	DrivoR trained on 103k real and 134k SimScale samples for 30 epochs.	v1
LTF (SimScale)	Perturbation	F0/L0/R0	LTF co-trained on real logs and SimScale perturbed scenes.	v1
GTRS-Dense–V2-99 (SimScale)	Perturbation	F0/L0/R0	GTRS-Dense co-trained on real logs and SimScale perturbed scenes.	v2
DiffusionDrive (BeyondDrive)	Perturbation	F0/L0/R0	DiffusionDrive trained with expert-proximate unsafe trajectories as negatives under BeyondDrive's repulsive objective.	v1
DriveLaW	World model	F0	Diffusion planner conditioned on latent features of a jointly trained video world model.	v1
DriveVLA–W0	World model	F0	Emu3 (Wang et al., 2024) visual-token backbone with the released flow-matching action expert.	v1
SimWAM	World model	F0	Action-diffusion policy co-trained with a video expert whose isolated attention mask makes the action head independent of future frames, so the video branch is dropped at inference.	v1

E.2 Human Driver Reference

Human participants. We recruited three volunteer university students aged 20–30. All participants held valid driver's licenses and reported prior video-gaming experience. The study was conducted under an IRB-approved protocol, and each participant provided written informed consent before participation. Before the recorded evaluation, participants were briefed on the experimental procedure, the rendered front-camera display with overlaid ground-truth route waypoints, and the game-controller interface (Figure 12). Each participant then completed a familiarization session in the simulator.

Evaluation procedure. Each participant was assigned one of the three evaluation seeds (0, 1, or 1024) and drove all 280 scenarios for that seed, yielding 840 fully human-controlled scenario–seed cells. Participants were instructed to prioritize collision avoidance and compliance with traffic rules while following the displayed route. They observed only the rendered front-camera view and had

no access to privileged simulator state, bird's-eye-view observations, future actor trajectories, or the logged ego trajectory. Human commands were issued through a gamepad and executed by the same controller and bicycle model used for policy outputs; vehicle dynamics, traffic evolution, event contracts, termination rules, and scoring were identical to those used in the policy evaluations. The Human Driver Reference establishes scenario solvability under the specified interface rather than population-level human performance or between-driver variability.

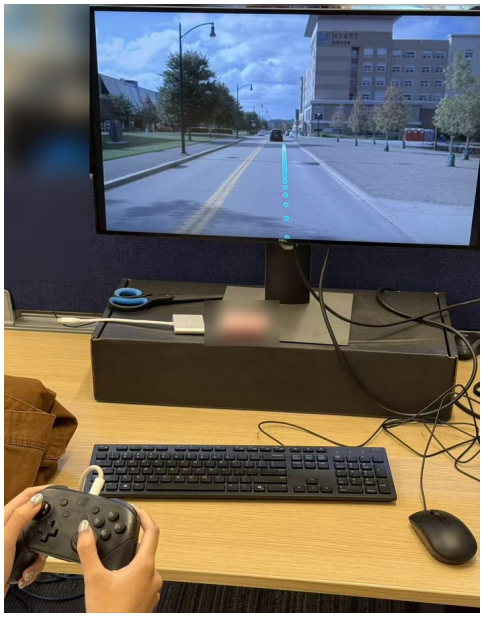

Figure 12: **Human-reference evaluation setup.** The rendered front-camera view with overlaid ground-truth route waypoints and the gamepad used for human-controlled rollouts.

F Additional Experimental Results

This appendix supplements Section 4 with the OL–CL comparison and complete leaderboard for Section 4.1, the case-selection protocol for the failure investigations in Section 4.2, and the diagnostic setting for Section 4.3.

F.1 Full Leaderboard

Table 7 extends Table 2 with three policies omitted for space: SimWAM–RL, DriveLaW, and DriveVLA–W0. All other rows are identical to Table 2. Rankings are recomputed over the 20 learned policies.

F.2 Open-Loop and Closed-Loop Performance

Figure 13 compares the 13 evaluated checkpoints with reported NAVSIM v1 PDMS against their NavSafe-∞ DS. Because the two scores are defined by different evaluation protocols, their numerical difference is not a calibrated measure of transfer. Instead, the aligned panels show each policy's performance relative to the corresponding human reference.

Table 7: Full NAVSAFE-∞ leaderboard, including the world-model-based policies omitted from Table 2. Among the 20 learned policies, blue, green, and yellow backgrounds denote the first-, second-, and third- highest point estimates in each column, respectively. Values are means over the same 280 scenarios and seeds 0, 1, and 1024; $\pm$ is the sample standard deviation across the per-seed estimates. Because every event type contributes exactly ten scenarios, per-seed category scores fall on a discrete grid with steps of $10/10$, $10/11$, $10/4$, and $10/3$ points for TC, TV, VRUC, and TI, respectively, and SR equals the event-type-weighted mean of the four ability scores. § indicates a human controller using NUREC front-camera observations; ⋆ denotes a privileged, non-learned reference. Rankings are recomputed over all learned policies; DriveVLA–W0 enters the top three in DE, displacing LTF (SimScale), and DE is the only column whose highlighting differs from Table 2.

Policy	Venue	Closed-loop metrics				Closed-loop ability scores			
		DS	SR	DE	Comf.	TC	VRUC	TV	TI
Human Driver Reference§	–	100.00±0.00	100.00±0.00	316.52±0.42	75.15±1.89	100.00±0.00	100.00±0.00	100.00±0.00	100.00±0.00
Privileged reference									
PDM-Closed⋆ (Dauner et al., 2023)	CoRL 2023	90.17±1.00	81.31±1.03	305.27±8.57	82.49±1.39	88.67±1.53	72.50±0.00	84.24±1.39	57.78±1.92
IL-based Methods									
LTF (Dauner et al., 2024)	NeurIPS 2024	53.04±1.92	27.74±1.76	163.71±0.11	**97.81±0.15**	31.67±0.58	25.00±7.50	28.48±1.89	15.56±1.92
DiffusionDrive (Liao et al., 2025)	CVPR 2025	44.77±0.75	20.95±0.55	165.08±1.84	97.68±0.56	21.33±1.15	11.67±1.44	23.64±0.00	22.22±1.92
DrivoR (Kirby et al., 2026)	CVPR 2026	64.74±1.60	50.71±0.71	188.02±1.07	87.92±1.25	58.67±2.52	31.67±5.77	52.42±2.62	43.33±0.00
RAP (Feng et al., 2026)	ICLR 2026	47.70±0.66	27.26±0.41	220.78±3.77	48.38±0.60	32.33±1.15	14.17±6.29	28.79±0.52	22.22±3.85
SparseDriveV2 (Sun et al., 2026)	ECCV 2026	65.26±1.72	48.81±3.04	167.99±1.01	93.41±0.43	54.00±0.00	43.33±8.04	48.18±4.17	41.11±3.85
GTRS-Dense–V2-99 (Li et al., 2025b)	arXiv 2025	64.27±0.76	49.52±1.15	163.88±0.87	95.90±0.63	52.00±1.73	40.00±2.50	54.24±0.52	36.67±0.00
ReCogDrive–2B–IL (Li et al., 2026d)	ICLR 2026	48.98±0.95	28.21±1.43	166.94±3.15	97.58±0.11	30.00±1.00	16.67±3.82	31.21±0.52	26.67±3.33
MTDrive–SFT (Li et al., 2026b)	arXiv 2026	48.78±0.88	31.79±1.43	173.77±0.76	94.51±1.25	30.33±1.15	15.00±6.61	35.76±1.39	44.44±1.92
RLFT-based Methods									
ReCogDrive–2B–RL (Li et al., 2026d)	ICLR 2026	53.64±1.42	30.95±2.06	173.96±1.90	85.10±0.49	35.33±2.31	18.33±6.29	32.42±1.39	27.78±5.09
MTDrive–mtGRPO (Li et al., 2026b)	arXiv 2026	55.86±0.50	40.60±0.90	**277.62±3.17**	93.94±0.41	40.33±2.89	16.67±1.44	46.36±0.91	**52.22±3.85**
SimWAM–RL (Zhao et al., 2026b)	arXiv 2026	55.41±1.72	29.76±2.86	173.08±2.41	78.62±0.68	35.67±3.06	22.50±4.33	28.48±1.89	24.44±5.09
AutoVLA (Zhou et al., 2025)	NeurIPS 2025	51.05±0.80	29.52±2.43	186.02±0.58	65.37±0.47	33.00±2.65	15.00±4.33	29.70±2.92	36.67±8.82
Passive Demonstration Perturbation Methods									
DiffusionDrive (SimScale) (Tian et al., 2026)	CVPR 2026	52.19±2.05	35.24±2.38	189.66±4.28	96.92±0.34	38.33±1.15	18.33±7.22	36.67±5.01	42.22±3.85
DrivoR (SimScale, 134k) (Tian et al., 2026)	CVPR 2026	57.66±1.32	42.50±0.36	176.73±0.92	85.88±0.39	44.00±1.00	17.50±2.50	48.18±2.41	50.00±0.00
LTF (SimScale) (Tian et al., 2026)	CVPR 2026	52.65±0.88	34.05±0.21	192.09±0.85	97.41±0.48	35.00±1.73	17.50±2.50	38.48±0.52	36.67±0.00
GTRS-Dense–V2-99 (SimScale) (Tian et al., 2026)	CVPR 2026	**73.54±0.84**	**61.79±1.56**	141.50±1.30	87.69±1.66	**69.33±0.58**	**50.00±2.50**	**64.24±1.89**	43.33±3.33
DiffusionDrive (BeyondDrive) (Wang et al., 2026a)	ECCV 2026	46.61±0.07	21.79±0.94	172.79±1.26	97.24±0.19	22.67±1.53	10.00±0.00	25.15±1.39	22.22±1.92
World-Model-based Methods									
DriveLaW (Xia et al., 2026a)	CVPR 2026	34.68±0.25	1.90±0.21	175.50±1.32	95.27±0.01	1.00±0.00	2.50±0.00	2.12±0.52	3.33±0.00
DriveVLA–W0 (Li et al., 2026c)	ICLR 2026	45.06±0.64	27.98±1.80	256.31±1.01	62.74±1.14	26.67±2.31	7.50±0.00	33.94±2.29	37.78±1.92
SimWAM (Zhao et al., 2026b)	arXiv 2026	57.53±2.72	37.50±1.56	160.54±1.64	97.19±0.27	41.33±0.58	26.67±7.22	40.61±0.52	27.78±5.09

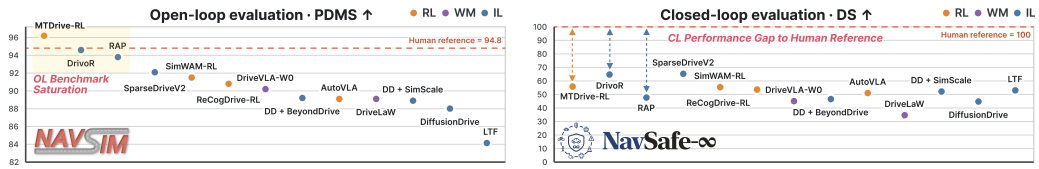

Figure 13: **Open-loop performance does not determine closed-loop safety. Left:** Published NAVSIM v1 PDMS for the 13 evaluated checkpoints with reported v1 scores. **Right:** NAVSAFE-∞ DS for the same checkpoints under CL execution. Dashed horizontal lines mark the human reference for each benchmark. Several policies approach or exceed the NAVSIM reference but remain substantially below the NAVSAFE-∞ reference, and their ordering changes across the two protocols. Colors distinguish IL-only checkpoints (blue), RLFT checkpoints (orange), and policies using world-model or model-generated supervision (purple).

F.3 CASE SELECTION FOR SECTION 4.2

Case selection. The four diagnostic cases are qualitative illustrations, one per benchmark category, of a failure mode that recurs across that category rather than of a single anomalous episode. Each case was chosen as a representative instance of its category's dominant failure and was fixed before the analysis was carried out; no case was reselected after inspecting its trace. The cases are not a random sample and are not used to estimate how often each failure mode occurs. Every rollout is produced and scored under the benchmark's own protocol—the same termination rules, contracts,

and metrics as any other cell (Appendix D.2)—and every panel is regenerated from that cell's saved closed-loop trace. The first three cases use seed 1; the fourth compares two rollouts of one scenario under seeds 1 and 1024. Two cases additionally instrument the policy for inspection without altering what it executes: SimWAM's video expert is re-enabled to decode a future frame, and AutoVLA is run with its chain-of-thought prompt so that a reasoning trace exists at each replan. Both are described below.

Planning Failure (LTF, I-3). The episode ends as an at-fault frontal collision with the incident actor 3.3 s into the rollout, for a Driving Score of 7.3. The failure is not one of perception: the conflicting actor is present in the policy's own detections well before contact, while the predicted longitudinal profile continues to resemble nominal car following.

Scoring Failure (DrivoR, V-1). The episode ends as an at-fault frontal collision with the lead vehicle 4.9 s into the rollout, for a Driving Score of 2.6. DrivoR scores its candidate proposals at each replan and executes the highest-ranked one. The rank reported in Section 4.2 is the position, in that same ordering, of the highest-scoring proposal that is collision-free under the benchmark's own collision check, evaluated at the replan preceding contact; the presence of such a proposal localizes the failure to ranking rather than to candidate generation.

Prediction Failure (SimWAM, C-7). SimWAM discards its video branch at inference by design: an isolated attention mask makes the action expert independent of future frames, so the released policy consumes one frame and emits one trajectory. For this diagnostic only, we re-enable the retained video expert and decode one future frame from the same closed-loop observation. The decoded frame is used solely to inspect the model's predicted world state; it does not enter the action head and does not change the executed trajectory. The episode ends as an at-fault frontal collision with the oncoming vehicle 5.3 s into the rollout, for a Driving Score of 6.9; the decoded future frame omits that vehicle, and the executed plan passes through the space it occupies.

Reasoning Failure (AutoVLA, R-1). AutoVLA is trained with adaptive reasoning and, under its released prompt, may emit action tokens directly without producing a chain of thought; left to that default it frequently skips reasoning altogether, and there is then no reasoning to compare against the action. For this diagnostic we therefore run the policy with its chain-of-thought system prompt at every replan, so that a reasoning trace is available at each decision point. The prompt is the one released with the checkpoint and is applied identically to both rollouts; the decoded action tokens, and hence the executed trajectory, are read from the same forward pass as the reasoning text. The two policy-induced states are taken from two rollouts of the same scenario under different evaluation seeds, which place the ego in different states by the time the pedestrians are encountered, and are compared at the same simulation timestamp. Both states decode reasoning that calls for a stop, yet only one plan does so: the seed-1 rollout accelerates and ends as an at-fault collision with the crossing pedestrian 7.4 s into the episode, whereas the seed-1024 rollout stops for the same crossing.

F.4 Diagnostic Setting for Passive Demonstration Perturbation

The perturbation analysis in Section 4.3 (Figure 7) compares the CL state visitation of the four policies evaluated with SimScale checkpoints in Table 2 (DiffusionDrive, GTRS-Dense–V2-99, DrivoR, and LTF) before and after SimScale co-training (Tian et al., 2026).

Scenarios and pairing. We use two scenarios, R-3 (token `132307b3c1a55f97`) and V-3 (token `3a6525cd871a5fff`), drawn at random from the scenarios in which the policy-dependent effect described below occurs. Each policy is evaluated before and after co-training in a paired setting: both rollouts start from the same logged ego pose at $t = 0$, route, surrounding actors, and environment configuration, use the same 30 seeds (0–29), and hand control to the policy from $t = 0$ without ego replay or additional perturbation. Differences between paired rollouts therefore arise from the checkpoints and their CL behavior. The simulator runs at 10 Hz, and rollouts are truncated at $t = 4$ s.

Coverage. Coverage is measured in ego-frame position, which spans the longitudinal and lateral perturbations of SimScale (up to 20 m and 2 m). The ego position at $t = 2$ s is expressed relative to the initial pose, $(x_t, y_t)^\top = R(\psi_0)^\top(\mathbf{p}_t - \mathbf{p}_0)$, where $\mathbf{p}_0$ and ψ_0 are the ego position and heading at $t = 0$, $\mathbf{p}_t$ is the ego position at time t, and $R(\psi_0)$ is the 2D rotation matrix by angle ψ_0, so that x_t is the forward and y_t the lateral (left-positive) displacement in meters. The background shows the training-state density of the co-trained policies, a weighted mixture of the logged and SimScale state

densities: a kernel centered at the logged state at $t = 2\,\mathrm{s}$ and an adaptive kernel density estimate over the SimScale states of the same scene at $t = 2\,\mathrm{s}$. Colors indicate normalized density, and a rollout state is regarded as covered when it lies in a high-density region of the background.

Metrics. To measure consequences in the benchmark's own units, we reconstruct the cumulative Driving Score on every prefix, $\mathrm{DS}_t = \mathrm{EP}_t\, P_t$, where EP_t and P_t are the event progress and the penalty product accumulated up to t (Appendix D.3), define $r_t = \mathrm{DS}_{t+1} - \mathrm{DS}_t$, and compute the discounted return $G_t = \sum_{k=t}^{T-1} \gamma^{k-t} r_k$ with $\gamma = 0.99$ from $t = 2\,\mathrm{s}$ until termination or $t = 4\,\mathrm{s}$. The reconstructed terminal score matches the evaluated DS for every rollout. Arrows in Figure 7 report $\Delta G = \bar{G}_{\text{after}} - \bar{G}_{\text{before}}$. The two-second window isolates the immediate effect of co-training on state visitation; longer horizons add interaction changes and compounding errors that would confound attribution to the augmentation. All results are averaged over seeds.

Relation to benchmark results. The policy-dependent effect is visible across all 280 scenarios (Table 2). SimScale co-training raises the DS of the anchor-based DiffusionDrive (44.77→52.19) and GTRS-Dense–V2-99 (64.27→73.54), whereas it lowers the DS of the query-based DrivoR (64.74→57.66) and leaves that of LTF essentially unchanged (53.04→52.65) despite a higher SR (27.74→34.05), indicating that the additional successes are offset by lower scores elsewhere. The two scenarios in Figure 7 illustrate the mechanism behind this split.

F.5 Diagnostic Setting for RLFT

The RLFT analysis in Section 4.3 uses two scenario sets derived from the benchmark: a proxy set for the nominal setting (Figure 8(b)) and a safety-critical set built on it (Figure 8(c)).

Proxy set. To reduce evaluation cost, we select one scenario per event type (28 in total) from the benchmark, following the performance-based clustering of tinyBenchmarks (Maia Polo et al., 2024). Each scenario is represented by the nominal closed-loop DS and success of 15 development policies. Within each event type, we cluster the scenarios with k-means and retain the cluster representative closest to the event-type mean. Five policies excluded from the selection serve as a held-out check: their aggregate DS and SR on the proxy set deviate from those on the full benchmark by 1.58 and 2.43 points on average (mean absolute error). The proxy set supports controlled diagnosis and does not replace the full benchmark. The proxy scenarios are used unmodified, without inserted hazards, as the nominal setting in Figure 8(b).

Safety-critical set. Each proxy scenario is augmented with two safety-critical events chosen from three types: a vehicle cut-in, lead-vehicle braking, and a vulnerable road user (VRU) crossing the ego path. Event types are assigned according to the road layout and traffic context of each scenario, so that every inserted event is plausible in its host scene; for example, VRU crossings are not inserted on multi-lane arterials with dense, fast-moving traffic and no sidewalks, where pedestrian crossings are unrealistic. The events are authored with the scene-customization workbench of the toolbox (Section 5; Appendix G.2), which places and times each event actor while preserving the route, background traffic, and ego initialization of the host scenario. This set yields Figure 8(c).

Metrics. CFR is the percentage of episodes without an at-fault collision. MC is the minimum distance between the oriented bounding boxes of the ego and the hazard actor within 4 s after hazard onset, which is negative when the boxes overlap. In nominal scenarios, which contain no designated hazard actor, MC is the minimum distance to any other actor within 4 s after scoring begins. Each bar in Figure 8 reports the relative change of the mean metric over matched rollouts, $100\,(\bar{M}_{\mathrm{RLFT}}/\bar{M}_{\mathrm{IL/SFT}} - 1)$. All results are averaged over three seeds.

G Toolbox Implementation

This appendix supplements Section 5 with the implementation of the three toolbox components.

G.1 Environment Construction

The environment-construction tool converts a driving log into a scenario bundle: four reconstructed segments (`<token>s1.usdz`–`<token>s4.usdz`), Arrow state, frame offsets, scenario metadata, assets, and a manifest. Reconstruction runs offline in the pinned NRE container (Ap-

pendix C.2); during a scored run, only rendering, traffic simulation, policy inference, control, and trace recording are performed. The simulator runs on Isaac Sim (NVIDIA, 2026a) 6.0.0, which hosts the scene and advances the simulation clock; in the evaluated configuration, the ego follows the kinematic bicycle model and images are rendered by the NuRec server. The remaining stack comprises Linux x86-64, Python 3.12, PyTorch 2.10.0 (CUDA 12.8), NumPy 2.3.1, and py123d 0.5.1, resolved by a lock file. Besides nuPlan, the tool supports Waymo Open Dataset logs (Sun et al., 2020).

G.2 Declarative Scenario Authoring

Every scenario of the nine recipe event types, and every edit, is encoded as a versioned YAML recipe rather than as a simulator-side special case. A recipe specifies the host scenario, frame conventions, ego initialization, actor edits, paired reference, and review metadata. Actor edits support removal, replacement, and insertion. Each external asset carries a content hash and physical dimensions, and each dynamic actor carries a frozen pose track or an explicit controller. The renderer, collision checker, traffic runtime, and scorer resolve the same actor identity and state, so an edit cannot alter the observation without also altering physical interaction. The paired reference disables the edit while retaining the route, background traffic, initialization, and seed. Because a recipe event type is defined entirely through recipes, including scenarios that replay the host log unchanged, all of its scenarios share the same construction and warm-up conventions.

G.3 Policy Integration, Tracing, and Scoring

A policy adapter translates the common sensor packet and route context into the native inputs of a checkpoint and converts its output into a common ego-trajectory command. The runtime executes this command through the configured controller and records observations, the requested trajectory, ego and actor states, infractions, traffic transfers, and the terminal reason. Model loading and adapter parameters are resolved from the evaluation manifest rather than stored in the scenario. Because scoring consumes only the saved trace, metric code can be tested independently and results can be re-scored without rerunning inference.

H Runtime and Resource Requirements

Simulation advances at 10 Hz, and camera observations are rendered only at the timesteps where the policy consumes them, so the rendering rate matches the replanning rate (2 Hz here) plus any additional frames needed to populate an observation history. When visualization is enabled, rendering runs at the full 10 Hz simulation rate.

We evaluate DrivoR with semi-reactive traffic and the NuRec renderer on two NVIDIA GeForce RTX 3090 GPUs per worker, one for rendering and one for policy evaluation. Our evaluation replans at 2 Hz, so cameras are rendered at 2 Hz. Visualization is disabled. Across 280 scenarios and three random seeds, DrivoR averages 223.68 simulation frames and 206.20 scored frames per episode. The evaluation loop includes rendering, simulation, policy inference, control, and scoring. Table 8 summarizes the resulting runtime and resource requirements.

Table 8: Runtime and resource requirements for the DrivoR policy-in-loop evaluation path.

Quantity	**Median**	**p90**	**Maximum**	N	**Conditions**
Episode evaluation	114.0 s	203.3 s	271.7 s	280 scenarios	$\leq$ 60 s simulated
Steady-state frame	0.510 s	0.742 s	1.032 s	280 scenarios	End-to-end
Renderer startup	50 s	50 s	50 s	10 launches	Once per worker
GPU memory/worker	24.1 GiB	24.7 GiB	25.1 GiB	10 measurements	Summed over 2 GPUs
Scenario-bundle storage		8.43 GB		one bundle	16-file snapshot

The renderer starts once per worker and is reused across scenarios.