CORNERBENCH: IS YOUR DRIVING MODEL SAFE ENOUGH?

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern end-to-end driving planners have improved rapidly on common driving benchmarks, but whether they can safely handle rare, high-consequence corner cases remains unclear. Evaluating such cases is difficult because natural driving logs contain few safety-critical events, while staged data collection is costly and difficult to scale. We present **CornerFactory**, an automated pipeline for constructing corner cases from real-world driving logs according to structured scene specifications. CornerFactory supports multi-view-consistent object insertion, novel-view synthesis from perturbed ego poses, and transfer across vehicle platforms with different camera configurations. Building on this pipeline, we introduce **CornerBench**, a benchmark that organizes generated corner cases into a structured taxonomy and evaluates planners using continuous, severity-aware metrics that capture graded safety differences. Using CornerBench, we evaluate representative planners along several dimensions commonly expected to improve robustness, including training data volume and model size, multimodal sensing, language-based scene understanding and chain-of-thought reasoning in VLA planners, and the use of world models for future prediction and planning.



Figure 1: **Overview of this paper.**

1 INTRODUCTION

End-to-end autonomous driving has advanced rapidly in recent years. Benchmarks such as nuPlan (Caesar et al., 2021), NAVSIM (Dauner et al., 2024), and Bench2Drive (Jia et al., 2024) standardize planner evaluation, but mainly cover regular driving scenes. Their limited corner-case coverage makes it difficult to diagnose and compare robustness, including the benefits of VLA and world-model planners.

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.

Rare events make natural corner-case collection inefficient; physical staging requires costly safety controls, repeated setup, and manual inspection. Simulation platforms such as CARLA (Dosovitskiy et al., 2017; Gerstenecker et al., 2026) simplify scenario creation, but their observations may differ from real sensor data or mismatch real-log-trained planners' input interfaces. Generative models offer an alternative: realistic, targeted cases at lower collection cost. Existing methods generate plausible multi-view observations for routine driving (Gao et al., 2024b; Li et al., 2023; Zhu et al., 2025a;b; Zeng et al., 2026; Li et al., 2025a; Ren et al., 2025; Zheng et al., 2026), but few target safety-critical events (Zhan et al., 2026).

For targeted corner-case evaluation, **CornerFactory** uses structured scene specifications to retrieve compatible real logs and construct the desired cases. It supports multi-view-consistent object insertion, novel-view synthesis from perturbed ego poses, and transfer across vehicle configurations, with corresponding LiDAR updates and VLM-assisted plus human quality filtering before export.

Building on this pipeline, **CornerBench** organizes object-centric, ego-state, and environmental cases into a structured taxonomy. CornerBench further introduces continuous, corner-case-aware metrics that preserve meaningful differences among unsafe behaviors—for example, distinguishing a low-speed collision from a high-speed collision—rather than treating all failed cases identically. Together, these components let us examine whether data and model scaling, multimodal sensing, VLM-based understanding, CoT reasoning, and world modeling translate into safer trajectories under corner cases.

In summary, this work makes three main contributions:

- We introduce **CornerFactory**, an automated, specification-driven pipeline grounded in real driving logs, supporting multi-view-consistent object insertion, perturbed-pose view synthesis, vehicle-configuration transfer, and aligned LiDAR updates for multimodal planners.
- We build **CornerBench**, a benchmark of object-centric, ego-state, and environmental cases, with a structured taxonomy and continuous, severity-aware metrics that distinguish unsafe behaviors rather than treating failures identically.
- We systematically evaluate representative end-to-end driving planners on CornerBench to assess their robustness to generated corner cases. Our analysis examines training-data volume and diversity, model size, multimodal sensing with LiDAR, VLM-based scene understanding, CoT reasoning, world-model predictions, and recovery from off-nominal ego states. These comparisons highlight the importance of data diversity and reveal gaps between scene understanding, reasoning, prediction, and safe trajectory planning.

2 Related Work

Driving Scene Generation and Editing. Recent controllable driving generation methods synthesize realistic driving scenes from structured conditions, while editing methods modify existing driving clips. SceneCrafter (Zhu et al., 2025a), Dream4Drive (Zeng et al., 2026), and HorizonForge (Wang et al., 2026c) further support driving-scene editing, such as inserting or removing vehicles in existing scenes. Risk-oriented methods such as CompoSIA (Zhan et al., 2026) and RiskMV (Lin et al., 2026) move closer to safety-critical scenario generation, but most existing methods do not directly provide multi-camera observations with cross-view consistency for multi-view planners or the corresponding LiDAR sensor records for multimodal-input planners.

Driving Scene Benchmarks. Recent corner-case benchmarks follow two complementary approaches: Fail2Drive (Gerstenecker et al., 2026) constructs failure-oriented scenarios for closed-loop evaluation, whereas WOD-E2E (Xu et al., 2025a), KITScenes (Wagner et al., 2026), and nuReasoning (Huang et al., 2026) collect or annotate long-tail cases in real-world driving data, including scenarios that require explicit reasoning. However, the former depends on predefined simulation assets and the latter on rare events present in existing logs; neither provides an economical and flexible mechanism for creating user-specified corner cases while retaining real-log context. Extended related work and planner descriptions are provided in Appendices D and E.

3 CornerFactory

CornerFactory constructs corner cases from real driving logs, starting with a structured scene specification that defines the intended intervention and guides retrieval of a compatible base scene. From this scene, the pipeline creates object-centered cases through multi-view-consistent object insertion and ego-state cases through view synthesis from perturbed ego poses. To evaluate planners across vehicle platforms on the same cases, it can transfer the generated observations to different camera configurations. For planners that also require LiDAR, it updates the point clouds to match the edited camera views. Finally, generated clips undergo VLM-assisted screening followed by human filtering.

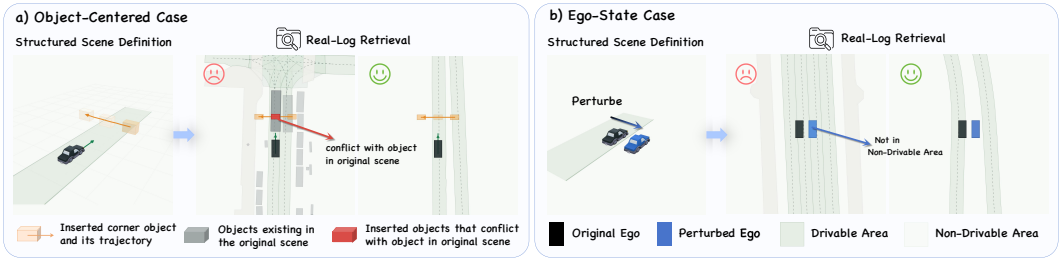

Figure 2: **Scene specification and retrieval.** Object trajectories or ego-pose perturbations guide selection of compatible real logs.

3.1 Structured Scene Specification and Real-Log Retrieval

The specification encodes the intervention's 3D spatial and temporal attributes; retrieval uses them to select a compatible real log (Figure 2). For object-centered cases, it defines a rare semantic type, plausible 3D box size, and position and motion intersecting the original ego trajectory, testing whether the planner slows, stops, or steers away safely. Retrieval excludes conflicts with existing non-ego agents to keep the insertion physically plausible. For ego-state cases, it defines a lane-straddling or non-drivable-area start, testing recovery toward the lane center rather than continued motion from the perturbed state. Retrieval selects road geometry and surrounding traffic compatible with that perturbation.

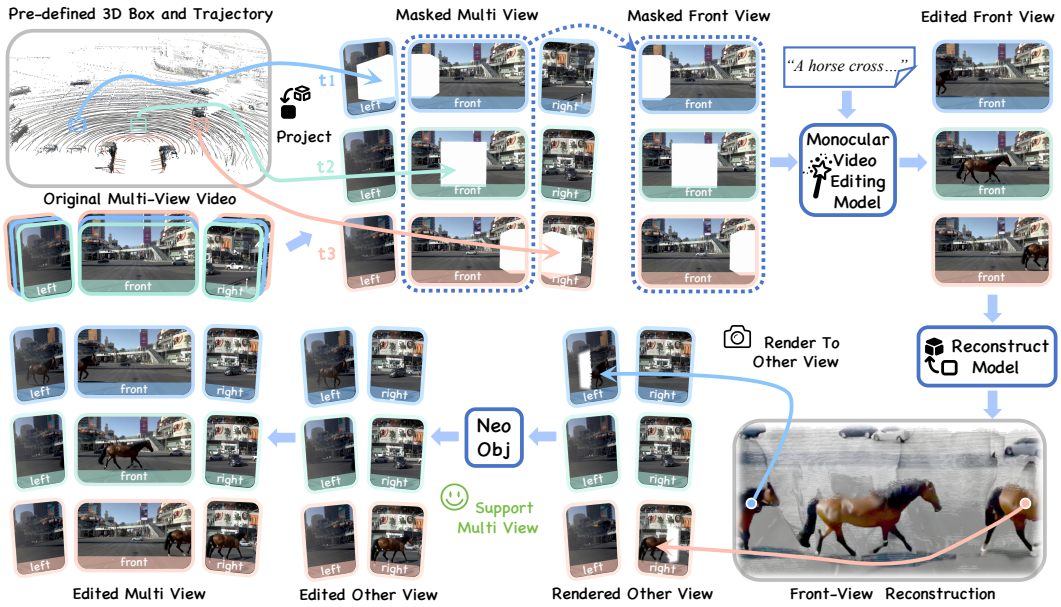

Figure 3: **Multi-view object insertion.** Front-view editing is followed by 3D reconstruction, cross-view rendering, and NeoObj completion.

3.2 Multi-View-Consistent Object Insertion

The specified 3D box trajectory is projected into all camera views to obtain temporally and cross-view consistent insertion masks, constraining the object to its intended position (Figure 3). VACE (Jiang et al., 2025) edits the masked front-view video using a semantic prompt describing the object and its behavior while preserving the surrounding scene. However, it does not produce corresponding edits in other cameras, reflecting the limited support for synchronized multi-view editing in current video models.

To overcome this limitation, we reconstruct the edited front view with the feed-forward Depth Anything 3 (DA3) model (Lin et al., 2025) and our fine-tuned Gaussian head, producing per-frame Gaussian representations. We render the reconstructed content into each target view's precomputed masks as geometry and appearance conditions; NeoObj then refines and completes the transferred object. This ties the object to the specified 3D trajectory while preserving consistent, photorealistic appearance across views. Training details are provided in Appendix A.1.

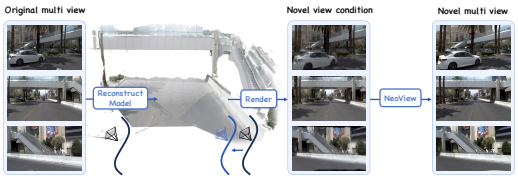

Figure 4: **Perturbed-pose synthesis.** Render the reconstructed scene at perturbed ego poses; NeoView completes newly visible regions.

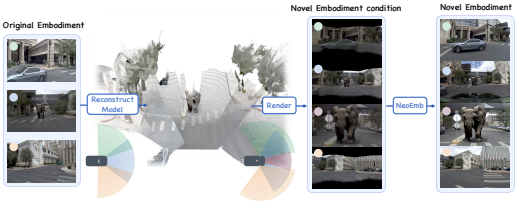

Figure 5: **Vehicle-configuration transfer.** Render the reconstructed scene with target camera parameters; NeoEmb completes exposed regions.

3.3 Free-View Synthesis from Perturbed Ego Poses

Ego-state cases perturb only the ego pose while preserving the map, route, and surrounding traffic. This places the vehicle in an off-nominal state within the same scene, allowing us to assess whether a planner recovers toward a safer driving pose or simply continues from the perturbed state. The DA3-based stack reconstructs the original multi-view scene as per-frame Gaussians and renders them along the perturbed ego trajectory. NeoView completes newly visible regions from these novel-view conditions (Figure 4). Training details are provided in Appendix A.2.

3.4 Vehicle-Configuration Transfer

To evaluate planners from different vehicle platforms on equivalent cases, CornerFactory transfers the camera configuration while retaining the corner-case geometry and traffic context. The same DA3-based stack reconstructs the source scene as per-frame Gaussians and renders them with the target vehicle's camera parameters; NeoEmb completes the newly exposed regions (Figure 5). The resulting observations match the target planner's input format, allowing nuPlan cases to evaluate planners such as NVIDIA's Alpamayo family. Training details are provided in Appendix A.3.

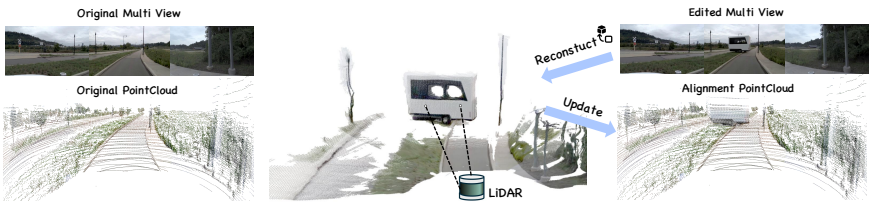

Figure 6: **LiDAR alignment.** Reconstruct edited observations as 3D Gaussians and synthesize returns consistent with the camera views.

3.5 Aligning LiDAR with Edited Camera Views

Camera-based editing alone is insufficient for multimodal planners because an inserted object appears in the edited RGB observations but remains absent from the original LiDAR point cloud. To restore cross-modal consistency, CornerFactory reconstructs the edited multi-view observations as 3D Gaussians using the same feed-forward pipeline and identifies the Gaussians associated with the inserted content. We then use Gaussian-based ray tracing (Zhou et al., 2025a) to synthesize the corresponding LiDAR returns. These returns replace the inconsistent original measurements, yielding geometrically aligned camera and LiDAR observations (Figure 6).

4 CornerBench

Using CornerFactory, we build CornerBench to stress-test end-to-end planners. Each case provides multi-view camera observations and aligned LiDAR. To distinguish planner responses, we evaluate predicted trajectories using continuous, severity-aware metrics.

4.1 Scenario Taxonomy

Figure 7 groups cases into **Ego-state**, **Environmental**, and **Object-centric** challenges; the last separates uncommon objects, behaviors, and textures. **Ego-state.** Unlike nominal lane-centered starts in most real logs, *Lane-Straddling Start* overlaps a lane marking and *Non-Drivable-Area Start* begins outside the drivable area. They test whether planners actively recover toward a safer driving state or simply continue along the initial heading.

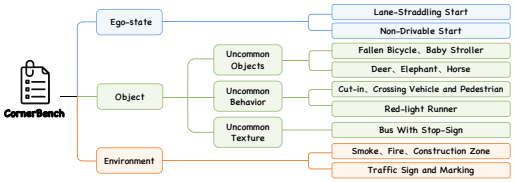

Figure 7: **CornerBench taxonomy:** ego states, uncommon objects, behaviors and textures, and environmental hazards.

Uncommon objects. Rare static or dynamic obstacles near the future ego path include fallen bicycles, tree branches, baby strollers, elephants, horses, and deer. These cases test whether planners can recognize rare objects and agents, avoid collisions, and yield when needed. **Uncommon behavior.** Crossing vehicles, crossing pedestrians, and cut-ins create imminent path-intrusion risks; red-light runners test whether the ego stops instead of following the violating vehicle through the light. **Uncommon texture.** Familiar objects with misleading cues, such as a bus carrying a stop-sign marking, test whether planners distinguish object identity from local texture. **Environment.** Smoke, fire, and construction zones affect traversability; edited lane-arrow markings simulate stale navigation cues, testing responses to scene-level context. Together, these categories distinguish failures in recovery, recognition, interaction, visual-cue interpretation, and environmental response.

Following this taxonomy, we construct 230 cases spanning 19 concrete scenario types from 125 source scenes in the nuPlan test set (Caesar et al., 2021); Appendix B reports the detailed distribution.

4.2 Metrics

CornerBench metrics are designed to expose graded differences among planners. Because CornerBench concentrates on difficult long-tail cases, many planners may receive identical low scores under coarse failure penalties. For example, NAVSIM assigns zero overall scores to at-fault collisions with road users or drivable-area violations (Dauner et al., 2024). This limits discrimination precisely where robustness differences matter most: two planners may both collide, but one slows before impact while the other strikes the hazard at high speed. We therefore score each case on a 0–1 scale using continuous severity terms, preserving differences among failed responses as well as distinguishing success from failure. Let $\mathrm{clip}_{[0,1]}(x) = \min(\max(x, 0), 1)$.

Hazard response. For both static and dynamic object cases, let $c \in \{0, 1\}$ denote collision, v_{col} the ego speed at the collision moment, and $v_{\max}$ a normalization speed. The score is

$$S_{\text{hazard}} = \begin{cases} 1, & c = 0, \\ 0.6\left(1 - \text{clip}_{[0,1]}\left(\frac{v_{\text{col}}}{v_{\max}}\right)\right), & c = 1. \end{cases} \tag{1}$$

Thus collision-free behavior receives full credit, while colliding trajectories remain distinguishable by collision speed. This metric is mainly used for *Uncommon Objects*, *Uncommon Behavior*, and the Environment cases that introduce traversability hazards. **Lane keeping.** Following the NAVSIM-style lane-keeping rule, let t_{out} be the total duration within the 4-second planning horizon for which the ego trajectory deviates from the lane center by more than 0.5 m. Instead of assigning a binary failure once t_{out} reaches 2 s, CornerBench uses

$$S_{\text{lane}} = \begin{cases} 1, & t_{\text{out}} < 2, \\ 0.6\left(1 - \text{clip}_{[0,1]}\left(\frac{t_{\text{out}}-2}{2}\right)\right), & t_{\text{out}} \geq 2. \end{cases} \tag{2}$$

Thus a threshold-level lane-keeping failure receives 0.6, and the score decreases linearly to 0 as the violation duration approaches the full 4-second horizon. This metric is used for ego-state cases, including *Lane-Straddling Start* and *Non-Drivable-Area Start*. **Traffic-rule compliance.** For red-light cases, let ρ_{red} be the maximum overlap ratio between the ego footprint and the red-light forbidden polygon over the trajectory, normalized by the ego footprint area. The score is

$$S_{\text{red}} = \begin{cases} 1, & \rho_{\text{red}} = 0, \\ 0.6\left(1 - \text{clip}_{[0,1]}(\rho_{\text{red}})\right), & \rho_{\text{red}} > 0. \end{cases} \tag{3}$$

This preserves the NAVSIM-style no-overlap requirement while distinguishing minor boundary intrusions from large overlaps. This metric is used for *red-light runners*.

5 Experiments

We evaluate a broad range of end-to-end planners on CornerBench to assess their robustness under generated corner cases and address the central question of this paper: *Are modern end-to-end driving models safe enough?* Our quantitative comparisons and qualitative failure analyses examine five factors commonly expected to improve robustness: training-data scale and model size, multimodal sensing, VLM-based scene understanding, CoT reasoning, and world-model prediction. Table 1 summarizes the planners and their attributes; Appendix E details their models and training data.

Table 1: CornerBench planner comparison, sorted in descending order by Hazard-response score from the left block to the right block. Higher is better. VLA denotes vision-language-action; WM, world model; Extra, additional data beyond NAVSIM `navtrain`; and Sens., sensor type (C: camera, C+L: camera plus LiDAR, N: ego status only). Hazard, Lane, and Rules denote Hazard response, Lane keeping, and Traffic-rule compliance. Colored superscripts mark the top three planners for each metric. Model references are given in Appendix E.

Planner	VLA	WM	Extra	Sens.	Hazard	Lane	Rules
Alpamayo R1	✓	✗	✓	C	**0.476**[1]	0.081	**0.740**[3]
SimScale	✗	✗	✓	C	**0.475**[2]	**0.274**[1]	0.312
Drive-JEPA	✗	✓	✓	C	**0.475**[3]	**0.148**[3]	0.084
GTRS-Aug	✗	✗	✗	C	0.468	0.122	0.133
GTRS-Dense	✗	✗	✗	C	0.461	0.146	0.395
Alpamayo 1.5	✓	✗	✓	C	0.457	0.143	**0.890**[2]
Hydra-MDP	✗	✗	✗	C	0.454	**0.177**[2]	0.304
VeteranAD	✗	✗	✗	C	0.452	0.051	0.128
DriveLaW	✗	✓	✓	C	0.444	0.059	0.292
DrivoR	✗	✗	✗	C	0.443	0.139	0.181
SeerDrive	✗	✓	✗	C+L	0.441	0.138	0.277
DriveVLA-W0	✓	✓	✗	C	0.441	0.059	0.166
OneVL	✓	✓	✗	C	0.441	0.085	0.506
DynVLA	✓	✓	✗	C	0.441	0.080	0.246
WoTE	✗	✓	✗	C+L	0.440	0.088	0.228
CuriousVLA	✓	✗	✗	C	0.440	0.101	0.210
WAM-Flow	✓	✗	✓	C	0.439	0.094	**0.923**[1]
TransFuser	✗	✗	✗	C+L	0.438	0.046	0.240

Planner	VLA	WM	Extra	Sens.	Hazard	Lane	Rules
RAP	✗	✗	✓	C	0.438	0.109	0.016
Drive-JEPA-PF	✗	✓	✓	C	0.437	0.060	0.253
MeanFuser	✗	✗	✗	C	0.436	0.099	0.283
AutoVLA	✓	✗	✗	C	0.436	0.000	0.442
Ego-MLP	✗	✗	✗	N	0.436	0.014	0.217
GoalFlow	✗	✗	✗	C+L	0.435	0.062	0.182
DiffusionDriveV2	✗	✗	✗	C+L	0.434	0.120	0.062
Mimir	✗	✗	✗	C+L	0.434	0.129	0.198
LEAD	✗	✗	✗	C	0.434	0.071	0.312
SparseDriveV2	✗	✗	✗	C	0.434	0.104	0.040
DriveWorld-VLA	✓	✓	✗	C	0.433	0.129	0.284
DiffusionDrive	✗	✗	✗	C+L	0.432	0.030	0.214
LTF	✗	✗	✗	C	0.432	0.023	0.337
ReCogDrive-IL-8B	✓	✗	✗	C	0.430	0.064	0.361
ReCogDrive-IL-2B	✓	✗	✗	C	0.430	0.098	0.427
PRIX	✗	✗	✗	C	0.429	0.051	0.279
ReCogDrive-RL-8B	✓	✗	✗	C	0.424	0.099	0.038
ReCogDrive-RL-2B	✓	✗	✗	C	0.424	0.128	0.166

5.1 How do training-data scale and model size affect corner-case robustness?

Alpamayo R1 and 1.5 (NVIDIA et al., 2025; NVIDIA, 2026) perform strongly with large-scale external driving data, without training on NAVSIM `navtrain`, demonstrating strong cross-domain generalization. This suggests that increasing training-data volume can bring gains. However, data volume alone does not explain the quantitative results; Figure 11 (left) relates Hazard-response scores to data volume. Drive-JEPA (Wang et al., 2026b), SimScale (Tian et al., 2026), DriveLaW (Xia et al., 2026), and WAM-Flow (Xu et al., 2025b) use additional data at comparable scales. Nevertheless, Drive-JEPA and SimScale achieve Hazard-response scores of 0.475 and 0.475, respectively, compared with 0.444 for DriveLaW and 0.439 for WAM-Flow. DriveLaW and WAM-Flow rely primarily on standard driving data from nuPlan and nuScenes. By contrast, Drive-JEPA and SimScale expand training-data diversity in different ways: the former uses driving videos covering diverse long-tail events, while the latter perturbs ego trajectories in ordinary driving scenes to synthesize diverse off-nominal situations. Drive-JEPA pretrains its V-JEPA backbone on 208 hours of driving video assembled from CoVLA (Arai et al., 2025), DrivingDojo (Wang et al., 2024b), and other datasets before training the downstream NAVSIM planner. Figure 8 shows representative open-world corner cases from DrivingDojo, including overhanging branches, animals near the ego lane, and nighttime roadside fire. Drive-JEPA ranks among the strongest planners on the CornerBench Hazard-response score in Table 1, and Figure 9 shows cases in which it produces a more conservative response to the corner-case hazard than other planners.



Figure 8: **DrivingDojo examples.** Drive-JEPA pretraining includes road obstructions, animal interactions, and nighttime hazards.



Figure 9: **Drive-JEPA responses.** No green front-view trajectory appears because early slowing makes Drive-JEPA's plan too short to project into those panels.

SimScale (Tian et al., 2026) augments its training data with 384K synthetic off-nominal scenes. To construct these scenes, it first perturbs ego trajectories in ordinary driving scenes to create off-nominal ego states, and then uses rule-based strategies or existing planners to generate trajectories that return the vehicle toward normal lane-centered states, providing explicit supervision for ego-state recovery. SimScale achieves the highest Lane-keeping score of 0.274 in Table 1, and Figure 10 shows cases in which it steers back toward the lane center from *Lane-Straddling Start* and *Non-Drivable-Area Start*, whereas most planners continue along the perturbed initial pose. Together, these cases suggest that diversifying the training data to cover rare objects and off-nominal ego states is more important for corner-case robustness than merely increasing the volume of routine driving data.

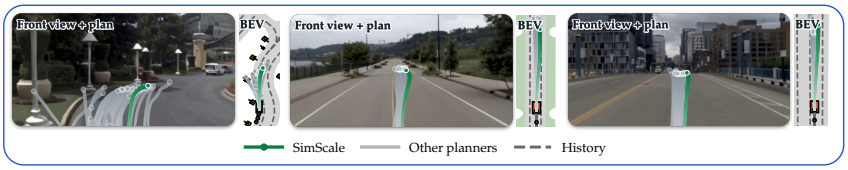

Figure 10: **SimScale recovery.** From lane-straddling and non-drivable-area starts, SimScale returns toward the lane center while most planners follow the perturbed initial state.

Model size. The right panel of Figure 11 shows no clear positive relationship between reported model size and CornerBench Hazard-response score. This is also visible within the same ReCogDrive family (Li et al., 2025d): scaling the backbone from 2B to 8B leaves the Hazard-response score unchanged in both settings, at 0.430 for IL and 0.424 for RL, while models with smaller active planners such as Drive-JEPA (Wang et al., 2026b) and SimScale (Tian et al., 2026) remain among the strongest methods. These comparisons suggest that simply increasing active model size is not sufficient; corner-case robustness also depends on data diversity.

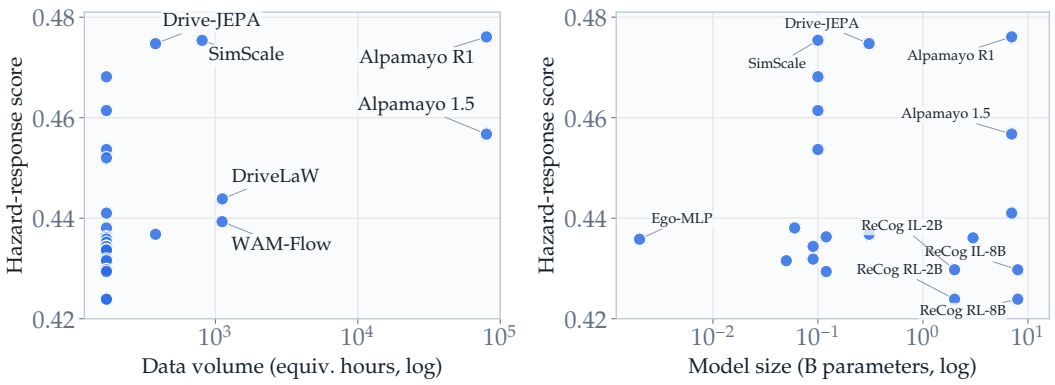

Figure 11: **Scale versus Hazard-response score.** Training-data volume in equivalent driving hours (left) and model size in billions of parameters (right); both x-axes are logarithmic.

5.2 How much do multimodal sensors help under rare and safety-critical corner cases?

Many evaluated planners use LiDAR together with camera inputs to obtain more reliable geometric perception of object distances and spatial structure (Li et al., 2025c; Chitta et al., 2022; Zhang et al., 2025b; Xing et al., 2025; 2026; Liao et al., 2025; Zou et al., 2025). However, the quantitative ranking in Table 1 shows that several top-performing planners do not use a LiDAR sensor. Figure 12 further shows qualitative results on three CornerBench test scenes: DiffusionDrive receives a 2D LiDAR BEV input, whereas Ego-Status MLP uses no LiDAR and does not even use camera input; nevertheless, the trajectories from both planners fail to avoid the frontal obstacle. This indicates that adding a LiDAR sensor does not by itself ensure stronger corner-case response.

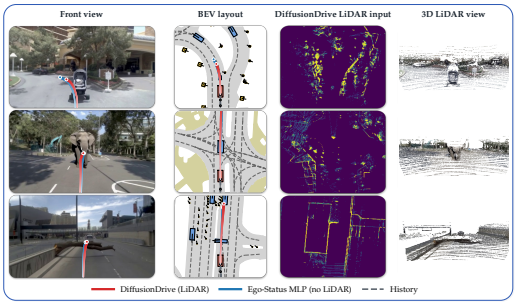

Figure 12: **Failures with and without LiDAR.** DiffusionDrive (LiDAR BEV) and Ego-MLP (no camera or LiDAR) both hit frontal obstacles, yielding similarly low Hazard-response scores.

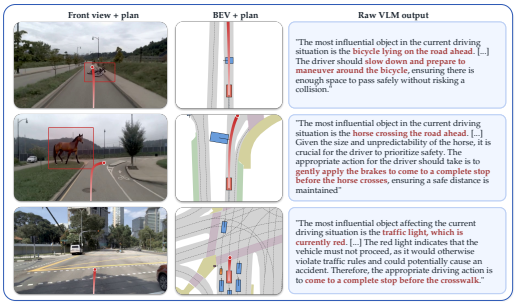

Figure 13: **Understanding–action mismatch.** ReCogDrive identifies hazards and suggests safe responses, but its trajectories collide or violate red-light rules.

5.3 How much does language-based scene understanding help VLA planners under corner-case scenarios?

VLA planners leverage the strong scene-understanding capability of VLMs to incorporate visual observations and language-level representations into action planning (Li et al., 2025d; Zhou et al., 2025c; Chen et al., 2026; Lu et al., 2026; NVIDIA et al., 2025). In Figure 13, ReCogDrive's VLM output identifies the inserted hazard or red traffic light and proposes a reasonable high-level response, but the final trajectory remains inconsistent with that response. For example, in the horse-crossing case, the VLM output first recognizes the horse crossing ahead and then recommends braking to a complete stop, but the planner's trajectory output still collides with the horse. This shows that VLM scene understanding is not reliably converted into correct trajectory-planning output.

5.4 How much does CoT help VLA planners handle corner cases?

CoT is widely regarded as a promising way to enhance the reasoning and planning capabilities of VLA planners in complex driving situations (Lu et al., 2026; Chen et al., 2026; Zhou et al., 2025c; NVIDIA, 2026). We test OneVL, CuriousVLA, AutoVLA, and Alpamayo 1.5 on the same corner-case scenes. In Figure 14, OneVL states that there are no relevant objects, CuriousVLA marks the critical object category as absent, and AutoVLA treats the scene as a straightforward driving case. Alpamayo 1.5 instead recognizes the corner-case hazard and suggests a safer response. The first three planners are trained primarily on regular driving scenes, whereas Alpamayo 1.5 uses 80,000 hours of diverse driving data. This comparison suggests that CoT alone does not ensure generalization to corner cases; its effectiveness also depends strongly on the diversity of the training data. Even when Alpamayo 1.5's CoT correctly identifies the hazard and suggests an appropriate response, its planned trajectory can remain unsafe, reflecting the understanding–action mismatch discussed in Section 5.3.

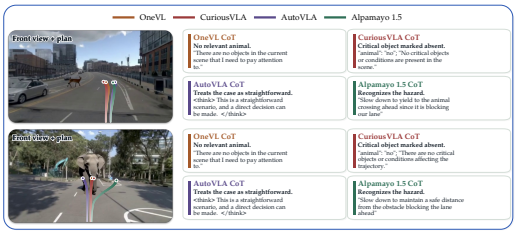

Figure 14: **CoT responses.** OneVL, CuriousVLA, and AutoVLA miss the critical object; Alpamayo 1.5 identifies it and suggests a safer response.



Figure 15: **World-model failures.** OneVL predicts plausible futures at $t{+}0.5$s and $t{+}1.0$s but fails at obstacle response and lane recovery.

5.5 How much do world models help planners respond to corner cases?

World models are widely expected to improve driving planning by predicting how a scene may evolve and using the predicted future to inform planning (Xia et al., 2026; Li et al., 2025c; Zhang et al., 2025b; Jia et al., 2026). To examine whether this benefit extends to corner-case scenarios, we inspect the future images decoded by OneVL (Lu et al., 2026) together with its corresponding planned trajectories on CornerBench. As shown in Figure 15, the decoded futures appear visually plausible, but neither the decoded futures nor the corresponding trajectories show the correct response to a frontal obstacle or a Lane-Straddling Start case. One likely reason is that OneVL is trained mainly on regular driving scenes, so its decoded futures do not generalize reliably to these rare corner-case situations. Thus, visually plausible world-model predictions do not by themselves ensure a correct corner-case response.

6 Conclusion

We presented CornerFactory, an automated pipeline for constructing corner cases from real-world driving logs, and CornerBench, a benchmark with diverse scenarios and severity-aware metrics. Together, they enable systematic evaluation of end-to-end driving planners under rare objects, abnormal behaviors, environmental hazards, and off-nominal ego states. Our evaluation shows that modern end-to-end driving models are not yet reliably safe under the tested corner cases. Data diversity is more important than simply increasing data volume or model size. LiDAR input, chain-of-thought reasoning, and visually plausible world-model predictions do not by themselves ensure safe trajectories when planners are trained mainly on routine driving scenes. Our findings emphasize the importance of training data that cover the relevant corner cases and appropriate driving responses. VLM-based scene understanding exposes a different bottleneck: VLMs can correctly recognize corner-case hazards and recommend safety-aware responses, yet current action planners do not reliably translate this knowledge into safe trajectories. Future work should therefore focus on stronger knowledge-to-action alignment, enabling action planners to make effective use of the rich scene understanding and driving knowledge provided by VLMs.

AI USE STATEMENT

We used generative video models to construct corner-case scenarios through object insertion, novel-view synthesis, and transfer across vehicle camera configurations, as described in the paper. VLMs assisted quality screening by flagging mismatches with scene specifications, temporal inconsistencies, and obvious visual artifacts. Generated clips underwent subsequent human filtering before inclusion in the benchmark. An AI assistant was also used for language polishing to improve grammar, clarity, and readability. The authors take full responsibility for the final manuscript, experimental claims, and benchmark artifacts.

REFERENCES

Hidehisa Arai, Keita Miwa, Kento Sasaki, Yu Yamaguchi, Kohei Watanabe, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *WACV*, 2025. URL https://arxiv.org/abs/2408.10845.

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, 2021. URL https://arxiv.org/abs/2106.11810.

Canyu Chen, Yuguang Yang, Zhewen Tan, Yizhi Wang, Ruiyi Zhan, Haiyan Liu, Xuanyao Mao, Jason Bao, Xinyue Tang, Linlin Yang, Bingchuan Sun, Yan Wang, and Baochang Zhang. Devil is in narrow policy: Unleashing exploration in driving vla models, 2026. URL https://arxiv.org/abs/2603.06049.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving, 2022. URL https://arxiv.org/abs/2205.15997.

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, 2024. URL https://arxiv.org/abs/2406.15349.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, 2017. URL https://arxiv.org/abs/1711.03938.

Lue Fan, Hao Zhang, Qitai Wang, Hongsheng Li, and Zhaoxiang Zhang. Freesim: Toward free-viewpoint camera simulation in driving scenes, 2024. URL https://arxiv.org/abs/2412.03566.

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.

Ruiyuan Gao, Kai Chen, Zhihao Li, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes, 2024a. URL https://arxiv.org/abs/2405.14475.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024b. URL https://arxiv.org/abs/2310.02601.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability, 2024c. URL https://arxiv.org/abs/2405.17398.

Simon Gerstenecker, Andreas Geiger, and Katrin Renz. Fail2drive: Benchmarking closed-loop driving generalization, 2026. URL https://arxiv.org/abs/2604.08535.

Xiangyu Guo, Zhanqian Wu, Kaixin Xiong, Ziyang Xu, Lijun Zhou, Gangwei Xu, Shaoqing Xu, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Genesis: Multimodal driving scene generation with spatio-temporal and cross-modal consistency, 2025. URL https://arxiv.org/abs/2506.07497.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving, 2023a. URL https://arxiv.org/abs/2309.17080.

Yihan Hu et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023b. URL https://arxiv.org/abs/2212.10156.

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 Qi, and Jiaqi Ma. nureasoning: A reasoning-centric dataset and benchmark for long-tail autonomous driving, 2026. URL https://arxiv.org/abs/2605.31572.

Feiyang Jia, Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving, 2026. URL https://arxiv.org/abs/2602.06521.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving, 2024. URL https://arxiv.org/abs/2406.03877.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2303.12077.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing, 2025. URL https://arxiv.org/abs/2503.07598.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Eloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, Anh-Quan Cao, Nermin Samet, Tuan-Hung Vu, and Matthieu Cord. Driving on registers. In *CVPR*, 2026. URL https://arxiv.org/abs/2601.05083.

Bohan Li, Jiazhe Guo, Hongsi Liu, Yingshuang Zou, Yikang Ding, Xiwu Chen, Hu Zhu, Feiyang Tan, Chi Zhang, Tiancai Wang, Shuchang Zhou, Li Zhang, Xiaojuan Qi, Hao Zhao, Mu Yang, Wenjun Zeng, and Xin Jin. Uniscene: Unified occupancy-centric driving scene generation. In *CVPR*, 2025a. URL https://arxiv.org/abs/2412.05435.

Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model, 2023. URL https://arxiv.org/abs/2310.07771.

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, 2025b. URL https://arxiv.org/abs/2510.12796.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model, 2025c. URL https://arxiv.org/abs/2504.01941.

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, 2025d. URL https://arxiv.org/abs/2506.08052.

Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, Yu-Gang Jiang, and Jose M. Alvarez. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation, 2024. URL https://arxiv.org/abs/2406.06978.

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, 2025e. URL https://arxiv.org/abs/2506.06664.

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*, 2025. URL https://arxiv.org/abs/2411.15139.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views, 2025. URL https://arxiv.org/abs/2511.10647.

Hongyi Lin, Wenxiu Shi, Heye Huang, Dingyi Zhuang, Song Zhang, Yang Liu, Xiaobo Qu, and Jinhua Zhao. Risk-controllable multi-view diffusion for driving scenario generation, 2026. URL https://arxiv.org/abs/2603.11534.

Hannan Lu, Xiaohe Wu, Shudong Wang, Xiameng Qin, Xinyu Zhang, Junyu Han, Wangmeng Zuo, and Ji Tao. Seeing beyond views: Multi-view driving scene video generation with holistic attention, 2024. URL https://arxiv.org/abs/2412.03520.

Jinghui Lu, Jiayi Guan, Zhijian Huang, Jinlong Li, Guang Li, Lingdong Kong, Yingyan Li, Han Wang, Shaoqing Xu, Yuechen Luo, Fang Li, Chenxu Dang, Junli Wang, Tao Xu, Jing Wu, Jianhua Wu, Xiaoshuai Hao, Wen Zhang, Tianyi Jiang, Lingfeng Zhang, Lei Zhou, Yingbo Tang, Jie Wang, Yinfeng Gao, Xizhou Bu, Haochen Tian, Yihang Qiu, Feiyang Jia, Lin Liu, Yigu Ge, Hanbing Li, Yuannan Shen, Jianwei Cui, Hongwei Xie, Bing Wang, Haiyang Sun, Jingwei Zhao, Jiahui Huang, Pei Liu, Zeyu Zhu, Yuncheng Jiang, Zibin Guo, Chuhong Gong, Hanchao Leng, Kun Ma, Naiyan Wang, Guang Chen, Kuiyuan Yang, Hangjun Ye, and Long Chen. Xiaomi onevl: One-step latent reasoning and planning with vision-language explanation, 2026. URL https://arxiv.org/abs/2604.18486.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Lead: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2512.20563.

NVIDIA. Alpamayo 1.5: Model card. Hugging Face model card, 2026. URL https://huggingface.co/nvidia/Alpamayo-1.5-10B.

NVIDIA. PhysicalAI-Autonomous-Vehicles. Hugging Face dataset, https://huggingface.co/datasets/nvidia/PhysicalAI-Autonomous-Vehicles, n.d.

NVIDIA, Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, Dongran Liu, Ming-Yu Liu, Langechuan Liu, Zhijian Liu, Jason Lu, Yunxiang Mao, Pavlo Molchanov, Lindsey Pavao, Zhenghao Peng, Mike Ranzinger, Ed Schmerling, Shida Shen, Yunfei Shi, Sarah Tariq, Ran Tian, Tilman Wekel, Xinshuo Weng, Tianjun Xiao, Eric Yang, Xiaodong Yang, Yurong You, Xiaohui Zeng, Wenyuan Zhang, Boris Ivanovic, and Marco Pavone. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail, 2025. URL https://arxiv.org/abs/2511.00088.

Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, Seung Wook Kim, Jun Gao, Laura Leal-Taixe, Mike Chen, Sanja Fidler, and Huan Ling. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models, 2025. URL https://arxiv.org/abs/2506.09042.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, Zhaoxiang Zhang, and Tieniu Tan. Dynvla: Learning world dynamics for action reasoning in autonomous driving, 2026. URL https://arxiv.org/abs/2603.11041.

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, 2026. URL https://arxiv.org/abs/2603.29163.

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*, 2026. URL https://arxiv.org/abs/2511.23369.

Royden Wagner, Omer Sahin Tas, Jaime Villa, Felix Hauser, Yinzhe Shen, Marlon Steiner, Dominik Strutz, Carlos Fernandez, Quentin Delfosse, Christoph Weinhuber, Christian Kinzig, Guillermo S. Gutierrez-Cabello, Hendrik Königshof, Fabian Immel, Richard Schwarzkopf, Nils Alexander Rack, Kevin Rösch, Kaiwen Wang, Jan-Hendrik Pauls, Martin Lauer, Igor Gilitschenski, Holger Caesar, and Christoph Stiller. Reasoning models do not yet follow their reasoning in autonomous driving: The kitscenes longtail dataset, 2026. URL https://arxiv.org/abs/2603.23607.

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Guang Li, Kun Ma, Guang Chen, Hangjun Ye, Zhongpu Xia, Long Chen, and Qichao Zhang. Meanfuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via meanflow for end-to-end autonomous driving, 2026a. URL https://arxiv.org/abs/2602.20060.

Linhan Wang, Zichong Yang, Chen Bai, Guoxiang Zhang, Xiaotong Liu, Xiaoyin Zheng, Xiao-Xiao Long, Chang-Tien Lu, and Cheng Lu. Drive-jepa: Video jepa meets multimodal trajectory distillation for end-to-end driving, 2026b. URL https://arxiv.org/abs/2601.22032.

Qitai Wang, Lue Fan, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Freevs: Generative view synthesis on free driving trajectory, 2024a. URL https://arxiv.org/abs/2410.18079.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving, 2023a. URL https://arxiv.org/abs/2309.09777.

Yifan Wang, Francesco Pittaluga, Zaid Tasneem, Chenyu You, Manmohan Chandraker, and Ziyu Jiang. Horizonforge: Driving scene editing with any trajectories and any vehicles. In *CVPR*, 2026c. URL https://arxiv.org/abs/2602.21333.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving, 2023b. URL https://arxiv.org/abs/2311.17918.

Yuqi Wang, Ke Cheng, Jiawei He, Qitai Wang, Hengchen Dai, Yuntao Chen, Fei Xia, and Zhaoxiang Zhang. Drivingdojo dataset: Advancing interactive and knowledge-enriched driving world model. In *NeurIPS*, 2024b. URL https://arxiv.org/abs/2410.10738.

Waymo. The waymo world model: A new frontier for autonomous driving simulation, 2026. URL https://waymo.com/blog/2026/02/the-waymo-world-model-a-new-frontier-for-autonomous-driving-simulation/.

Wayve. Gaia-3: Scaling world models to power safety and evaluation. https://wayve.ai/thinking/gaia-3/, 2025.

Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving, 2023. URL https://arxiv.org/abs/2311.16813.

Maciej Wozniak, Lianhang Liu, Yixi Cai, and Patric Jensfelt. Prix: Learning to plan from raw pixels for end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 11(5):6400–6407, 2026. doi: 10.1109/LRA.2026.3678836. URL https://arxiv.org/abs/2507.17596.

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*, 2026. URL https://arxiv.org/abs/2512.23421.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving, 2025. URL https://arxiv.org/abs/2503.05689.

Zebin Xing, Yupeng Zheng, Qichao Zhang, Zhixing Ding, Pengxuan Yang, Songen Gu, Zhongpu Xia, and Dongbin Zhao. Mimir: Hierarchical goal-driven diffusion with uncertainty propagation for end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 11(2):2178–2185, 2026. doi: 10.1109/LRA.2025.3641129. URL https://arxiv.org/abs/2512.07130.

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

Yifang Xu, Jiahao Cui, Feipeng Cai, Zhihao Zhu, Hanlin Shang, Shan Luan, Mingwang Xu, Neng Zhang, Yaoyi Li, Jia Cai, and Siyu Zhu. Wam-flow: Parallel coarse-to-fine motion planning via discrete flow matching for autonomous driving, 2025b. URL https://arxiv.org/abs/2512.06112.

Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, and Sida Peng. Streetcrafter: Street view synthesis with controllable video diffusion models, 2024. URL https://arxiv.org/abs/2412.13188.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Genad: Generalized predictive model for autonomous driving. In *CVPR*, 2024. URL https://arxiv.org/abs/2403.09630.

Yuxue Yang, Lue Fan, Ziqi Shi, Junran Peng, Feng Wang, and Zhaoxiang Zhang. Neoverse: Enhancing 4d world model with in-the-wild monocular videos, 2026. URL https://arxiv.org/abs/2601.00393.

Kai Zeng, Zhanqian Wu, Kaixin Xiong, Xiaobao Wei, Xiangyu Guo, Zhenxin Zhu, Kalok Ho, Lijun Zhou, Bohan Zeng, Ming Lu, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, and Wentao Zhang. Rethinking driving world model as synthetic data generator for perception tasks. In *ICLR*, 2026. URL https://arxiv.org/abs/2510.19195.

Yifan Zhan, Zhengqing Chen, Qingjie Wang, Zhuo He, Muyao Niu, Xiaoyang Guo, Wei Yin, Weiqiang Ren, Qian Zhang, and Yinqiang Zheng. Composing driving worlds through disentangled control for adversarial scenario generation, 2026. URL https://arxiv.org/abs/2603.12864.

Bozhou Zhang, Jingyu Li, Nan Song, and Li Zhang. Perception in plan: Coupled perception and planning for end-to-end autonomous driving, 2025a. URL https://arxiv.org/abs/2508.11488.

Bozhou Zhang, Nan Song, Jingyu Li, Xiatian Zhu, Jiankang Deng, and Li Zhang. Future-aware end-to-end driving: Bidirectional modeling of trajectory planning and scene evolution. In *NeurIPS*, 2025b. URL https://arxiv.org/abs/2510.11092.

Guosheng Zhao, Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Xueyang Zhang, Yida Wang, Guan Huang, Xinze Chen, Boyuan Wang, Youyi Zhang, Wenjun Mei, and Xingang Wang. Drivedreamer4d: World models are effective data machines for 4d driving scene representation, 2024. URL https://arxiv.org/abs/2410.13571.

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. In *AAAI*, 2025. URL https://arxiv.org/abs/2403.06845.

Chaoda Zheng, Sean Li, Jinhao Deng, Zhennan Wang, Shijia Chen, Liqiang Xiao, Ziheng Chi, Hongbin Lin, Kangjie Chen, Boyang Wang, Yu Zhang, and Xianming Liu. X-world: Controllable ego-centric multi-camera world models for scalable end-to-end driving, 2026. URL https://arxiv.org/abs/2603.19979.

Chenxu Zhou, Lvchang Fu, Sida Peng, Yunzhi Yan, Zhanhua Zhang, Yong Chen, Jiazhi Xia, and Xiaowei Zhou. LiDAR-RT: Gaussian-based ray tracing for dynamic LiDAR re-simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a. URL https://arxiv.org/abs/2412.15199.

Jingqiu Zhou, Lue Fan, Linjiang Huang, Xiaoyu Shi, Si Liu, Zhaoxiang Zhang, and Hongsheng Li. Flexdrive: Toward trajectory flexibility in driving scene reconstruction and rendering, 2025b. URL https://arxiv.org/abs/2502.21093.

Lijun Zhou, Hongcheng Luo, Zhenxin Zhu, Cheng Chi, Mingfei Tu, Kaixin Xiong, Lei Gong, Zhanqian Wu, Zehan Zhang, Fangzhen Li, Hao Li, Yingying Shen, et al. Xiaomi auto world model: A joint world model integrating reconstruction and generation for autonomous driving, 2026. URL https://arxiv.org/abs/2605.18137.

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, 2025c. URL https://arxiv.org/abs/2506.13757.

Zehao Zhu, Yuliang Zou, Chiyu Max Jiang, Bo Sun, Vincent Casser, Xiukun Huang, Jiahao Wang, Zhenpei Yang, Ruiqi Gao, Leonidas Guibas, Mingxing Tan, and Dragomir Anguelov. Scenecrafter: Controllable multi-view driving scene editing. In *CVPR*, 2025a. URL https://arxiv.org/abs/2506.19488.

Ziyue Zhu, Zhanqian Wu, Zhenxin Zhu, Lijun Zhou, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Jin Xie, and Jian Yang. Worldsplat: Gaussian-centric feed-forward 4d scene generation for autonomous driving, 2025b. URL https://arxiv.org/abs/2509.23402.

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

A TRAINING NEOOBJ, NEOVIEW, AND NEOEMB

NeoObj, NeoView, and NeoEmb are specialized for the three completion tasks in CornerFactory through distinct conditioning interfaces and task-specific training data. Their shared generation backbone is initialized from the pretrained NeoVerse model (Yang et al., 2026) and kept frozen, while the task-specific control branches are optimized with the rectified-flow denoising objective. All three modules are trained on 81-frame clips resized to 336×560, with a batch size of one, bfloat16 precision, and a learning rate of 10^{-5}. This design supports three distinct completion settings without retraining the video-generation backbone. They differ primarily in how the conditioning–target pairs are constructed, as described below.

A.1 NEOOBJ: MASKED CROSS-VIEW OBJECT COMPLETION

Training-Pair Construction. NeoObj is designed to complete the target-view appearance of an object transferred from the front camera to a neighboring camera while preserving the observed target-view background, thereby enabling the multi-view-consistent object insertion described in Section 3.2. We construct training pairs from nuPlan front, front-left, and front-right videos, using tracked real objects as proxies for cross-view object transfer. For each clip, we select either the front-left or front-right view together with a tracked vehicle. We project the tracked 3D boxes into the selected view to obtain a temporally varying 2D mask, and first mask out this object region from the neighboring-view video. We then reconstruct the front-view video with a DA3-based Gaussian reconstruction stack and render the resulting Gaussians into the masked region using the calibrated poses and intrinsics of the neighboring camera. The RGB condition combines this cross-view Gaussian rendering inside the mask with the original neighboring-view pixels outside it. The corresponding unmodified neighboring-view video serves as the supervision target. The second and third columns of Figure 16 illustrate the resulting training pairs.

Difference from the Original NeoVerse. Although NeoVerse provides a strong general-purpose completion prior, its original weights are not specialized for the localized cross-view artifacts introduced when transferring an object between cameras. As shown in Figure 16, both NeoVerse and NeoObj can complete the transferred vehicle under identical inference settings. The main difference lies in the surrounding background: NeoVerse often leaves visible black residuals from the input mask, whereas NeoObj fills these regions naturally and blends the completed object with the observed target-view context. This comparison shows that NeoObj resolves the localized cross-view artifacts encountered in multi-view-consistent object insertion, with additional examples shown in Figure 17.

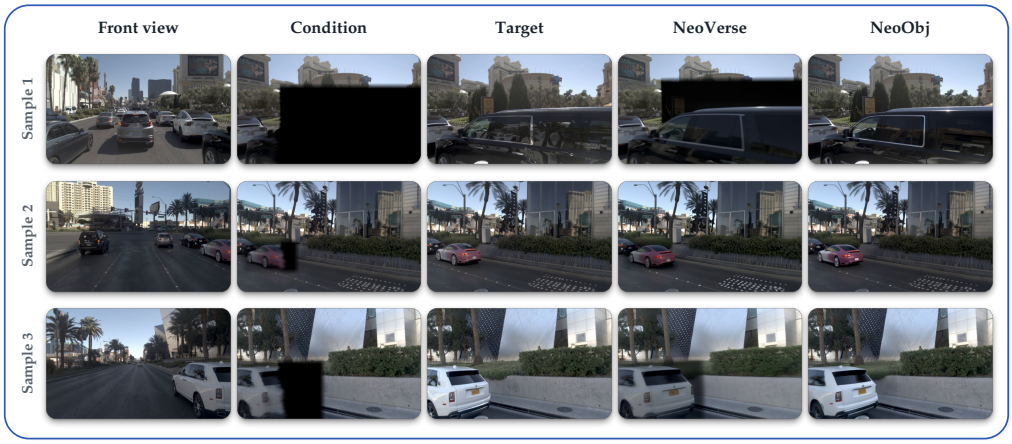

Figure 16: **NeoObj training-pair construction and comparison with NeoVerse.** In each row, the first three columns show the front view, composite RGB condition, and target RGB supervision that form a training example; the last two columns compare the outputs of NeoVerse and NeoObj under identical inference settings.



Figure 17: **Examples of NeoObj.** Each row shows the original multi-view observations and the results after multi-view-consistent object insertion.

A.2 NeoView: Perturbed-Pose Novel-View Completion

Training-Pair Construction. NeoView is designed to complete the novel views produced by perturbing the ego-camera pose, as described in Section 3.3. We construct training pairs from nuPlan multi-view videos. For each clip, we reconstruct the original scene into Gaussian representations using a DA3-based feed-forward reconstruction pipeline (Lin et al., 2025). Following the visibility-based degradation simulation of NeoVerse (Yang et al., 2026), we apply a random lateral displacement to the camera trajectory, remove Gaussian primitives that are invisible from the displaced viewpoints, and render the remaining Gaussians back into the original views. The resulting degraded renderings serve as conditions, while the corresponding original videos serve as supervision targets. The first two columns of Figure 18 illustrate the resulting training pairs.

Difference from the Original NeoVerse. NeoView adopts the same overall condition–target construction and visibility-based degradation simulation as NeoVerse, but differs in its reconstruction setting. To scale 4D world-model training with in-the-wild data, NeoVerse reconstructs monocular videos, whereas NeoView reconstructs multi-camera driving videos with DA3, including all eight camera views in nuPlan. This multi-view setting introduces reconstruction artifacts that are uncommon in the monocular training data of NeoVerse. As shown by Samples 1 and 2 in Figure 18, exposure differences across cameras can produce abrupt local brightness changes. Moreover, inconsistent geometry estimates across views can place the same object at slightly different locations, leading to fragmented or ghosted structures, as illustrated by Sample 3 in Figure 18. NeoVerse tends to preserve these multi-view-specific artifacts, whereas NeoView learns to reconcile the inconsistent appearance and geometry, producing more uniform illumination and coherent object structures. Additional examples are shown in Figure 19.

A.3 NeoEmb: Full-View Vehicle-Configuration Transfer

Training-Pair Construction. NeoEmb is designed to complete the views rendered using the camera intrinsics and extrinsics of a different vehicle platform, as described in Section 3.4. We construct training pairs from front-wide videos in the NVIDIA PhysicalAI Autonomous Vehicles dataset (NVIDIA, n.d.). For each clip, we reconstruct the original video into Gaussian representations using the same DA3-based feed-forward reconstruction pipeline (Lin et al., 2025). Following the visibility-based degradation simulation of NeoVerse (Yang et al., 2026), we construct degraded RGB renderings from the Gaussian representations. We additionally mask the lower region of each frame containing the ego-vehicle hood, requiring the model to recover the platform-specific foreground appearance. The resulting RGB renderings form the conditions, while the corresponding original front-wide videos serve as supervision targets. The first two columns of Figure 20 illustrate the resulting training pairs.

Difference from the Original NeoVerse. NeoVerse is trained on generic scenes and therefore lacks the vehicle-specific prior needed to recover the ego hood of a target platform. As shown in

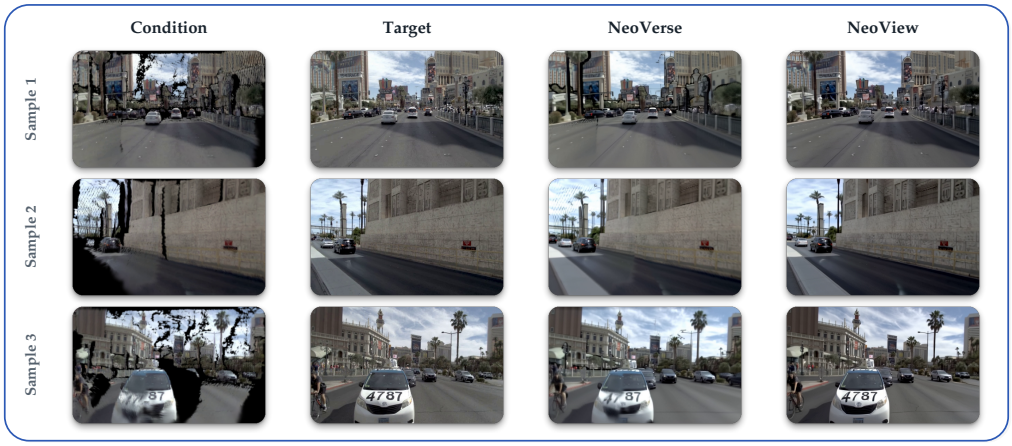

Figure 18: **NeoView training-pair construction and comparison with NeoVerse.** In each row, the first two columns show the perturbed-pose RGB condition and its target RGB supervision; the last two columns compare the outputs of NeoVerse and NeoView under identical inference settings.

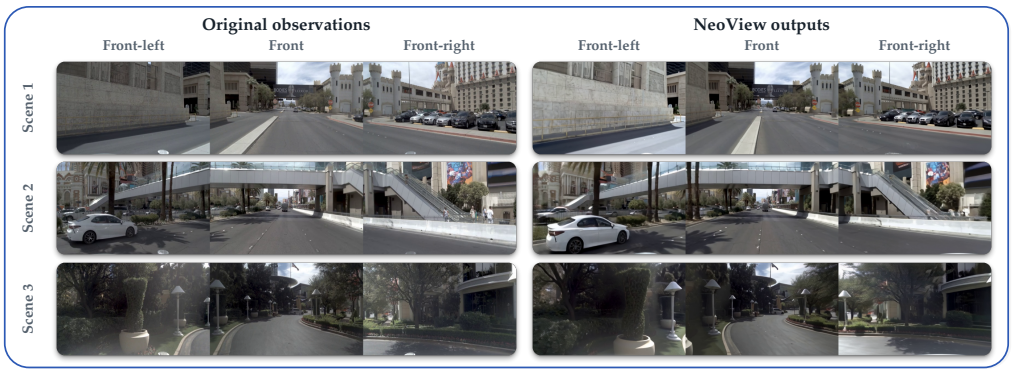

Figure 19: **Examples of NeoView.** Each row shows the original multi-view observations and the results after free-view synthesis from a perturbed ego pose; the views are ordered as front-left, front, and front-right.

Figure 20, NeoVerse produces hoods with incorrect appearance and geometry in Samples 1 and 2, and omits the hood entirely in Sample 3. Through training on data collected with the target vehicle configuration, NeoEmb learns the characteristic shape, appearance, and image-space location of its ego hood, enabling consistent recovery of the target-platform foreground together with the surrounding driving scene. Additional examples are shown in Figure 21.

B CornerBench Statistics

The current CornerBench release comprises 230 generated cases across 19 concrete scenario types, constructed from 125 distinct source scenes in the nuPlan test set (Caesar et al., 2021). Object-centric scenarios account for 132 cases (57.4%): 99 involve uncommon objects, 28 involve uncommon behavior, and 5 involve uncommon texture. The remaining cases comprise 81 environmental scenarios (35.2%) and 17 ego-state scenarios (7.4%). Because different corner-case edits can be applied to the same source scene, the number of benchmark cases exceeds the number of distinct source scenes. Table 2 reports the complete scenario-level distribution.



Figure 20: **NeoEmb training-pair construction and comparison with NeoVerse.** In each row, the first two columns show the degraded front-wide condition and its target RGB supervision; the last two columns compare the outputs of NeoVerse and NeoEmb under identical inference settings.



Figure 21: **Examples of NeoEmb.** Each row shows the original multi-view observations and the results after vehicle-configuration transfer.

Table 2: Scenario distribution in the current CornerBench release. Scenario types are grouped according to the taxonomy in Section 4.1.

Taxonomy family	Concrete scenario type	Cases
Ego state	Lane-Straddling Start	9
	Non-Drivable-Area Start	8
Uncommon object	Baby Stroller	21
	Large Tire Ahead	19
	Fallen Tree Branch	16
	Static Elephant Ahead	16
	Fallen Bicycle	14
	Crossing Horse	5
	Crossing Deer	8
Uncommon texture	Bus with Stop-Sign Marking	5
Uncommon behavior	Same-Direction Vehicle Running a Red Light	16
	Crossing Pedestrian	7
	Crossing Vehicle Collision	5
Environment	Construction-Sign Stand	26
	Fire Cloud	22
	Burning Car	16
	Smoke Cloud	8
	Traffic-Cone Cluster	6
	Straight-Only Arrow Before Turn	3
Total		**230**

C Capability Comparisons

Tables 3 and 4 provide the full comparisons summarized in the main text.

Table 3: Comparison with representative controllable driving generation and editing methods. Rows denote methods and columns denote capabilities.

Method	Real-log grounding	Corner cases	Multi-camera	Cross-vehicle transfer	LiDAR	Open source
MagicDrive (Gao et al., 2024b)	✓	✗	✓	✗	✗	✓
DrivingDiffusion (Li et al., 2023)	✓	✗	✓	✗	✗	✓
SceneCrafter (Zhu et al., 2025a)	✓	✗	✓	✗	✗	✓
WorldSplat (Zhu et al., 2025b)	✓	✗	✓	✗	✗	✓
Dream4Drive (Zeng et al., 2026)	✓	✗	✓	✗	✗	✓
HorizonForge (Wang et al., 2026c)	✓	✗	✗	✗	✗	✓
CompoSIA (Zhan et al., 2026)	✓	✓	✗	✗	✗	✓
Fail2Drive (Gerstenecker et al., 2026)	✗	✓	✓	✓	✓	✓
UniScene (Li et al., 2025a)	✓	✗	✓	✗	✓	✓
Cosmos Drive Dreams (Ren et al., 2025)	✓	✗	✓	✗	✓	✓
XWorld (Zheng et al., 2026)	✓	✗	✓	✗	✗	✗
Xiaomi Auto WM (Zhou et al., 2026)	✓	✓	✓	✗	✗	✗
GAIA-3 (Wayve, 2025)	✓	✓	✓	✓	✗	✗
Waymo WM (Waymo, 2026)	✓	✓	✓	✗	✓	✗
CornerFactory	✓	✓	✓	✓	✓	✓

Table 4: Comparison with representative autonomous-driving benchmarks. Real-log grounding indicates whether scenarios are grounded in real-world driving records; corner-case focus indicates whether long-tail safety-critical cases are the benchmark's primary target; severity-aware metrics indicate whether failed trajectories remain distinguishable through graded scores based on violation severity rather than receiving a shared binary outcome; economical indicates whether the benchmark avoids expensive simulation or new data collection.

Benchmark	Real-log grounding	Corner-case focus	Severity-aware metrics	Economical
NAVSIM (Dauner et al., 2024)	✓	✗	✗	✗
Bench2Drive (Jia et al., 2024)	✗	✗	✗	✗
Fail2Drive (Gerstenecker et al., 2026)	✗	✓	✗	✗
WOD-E2E (Xu et al., 2025a)	✓	✓	✗	✗
KITScenes (Wagner et al., 2026)	✓	✓	✗	✗
nuReasoning (Huang et al., 2026)	✓	✓	✗	✗
CornerBench	✓	✓	✓	✓

D Extended Related Work

Driving Scene Generation and Editing. Recent controllable driving generation methods synthesize realistic driving scenes from structured conditions, while editing methods modify existing driving clips. MagicDrive (Gao et al., 2024b), DrivingDiffusion (Li et al., 2023), DriveDreamer (Wang et al., 2023a), GAIA-1 (Hu et al., 2023a), Drive-WM (Wang et al., 2023b), Panacea (Wen et al., 2023), GenAD (Yang et al., 2024), Vista (Gao et al., 2024c), DriveDreamer-2 (Zhao et al., 2025), CogDriving (Lu et al., 2024), UniScene (Li et al., 2025a), and Genesis (Guo et al., 2025) use layout, 3D, agent, or action conditions for controllable driving-scene generation. SceneCrafter (Zhu et al., 2025a), Dream4Drive (Zeng et al., 2026), and HorizonForge (Wang et al., 2026c) further support driving-scene editing, such as inserting or removing vehicles in existing scenes. WorldSplat (Zhu et al.,

2025b), MagicDrive3D (Gao et al., 2024a), FreeVS (Wang et al., 2024a), DriveDreamer4D (Zhao et al., 2024), FreeSim (Fan et al., 2024), StreetCrafter (Yan et al., 2024), and FlexDrive (Zhou et al., 2025b) improve 3D or free-view driving-scene generation and rendering. These methods mainly focus on generating plausible ordinary driving scenes or controllable traffic variations. Risk-oriented methods such as CompoSIA (Zhan et al., 2026) and RiskMV (Lin et al., 2026) move closer to safety-critical scenario generation, but most existing methods do not directly provide multi-camera observations with cross-view consistency for multi-view planners or the corresponding LiDAR sensor records for multimodal-input planners.

Driving Scene Benchmarks. Driving scene benchmarks establish standardized data, protocols, and metrics for evaluating autonomous-driving planners. nuPlan (Caesar et al., 2021) provides large-scale real-world driving logs, while NAVSIM (Dauner et al., 2024) builds on these logs with standardized routes and planning metrics for end-to-end evaluation. Bench2Drive (Jia et al., 2024) complements real-log evaluation with closed-loop testing in simulation. Although these general-purpose benchmarks are effective for measuring performance under the nominal driving distribution, they offer limited coverage of rare, safety-critical events. Recent efforts address this gap from two complementary perspectives: Fail2Drive (Gerstenecker et al., 2026) constructs failure-oriented scenarios for closed-loop evaluation, whereas WOD-E2E (Xu et al., 2025a), KITScenes (Wagner et al., 2026), and nuReasoning (Huang et al., 2026) collect or annotate long-tail cases in real-world driving data, including scenarios that require explicit reasoning. However, the former depends on predefined simulation assets and the latter on rare events present in existing logs; neither provides an economical and flexible mechanism for creating user-specified corner cases while retaining real-log context.

End-to-End Autonomous Driving Planners. Early planning-oriented methods, such as UniAD (Hu et al., 2023b) and VAD (Jiang et al., 2023), established the modern end-to-end paradigm by jointly learning scene representations and future trajectories. Subsequent planners explore several complementary directions that are commonly expected to improve robustness in corner cases. First, VLA planners incorporate VLM-based scene understanding, language-conditioned actions, or intermediate reasoning, as exemplified by ReCogDrive (Li et al., 2025d), DriveVLA-W0 (Li et al., 2025b), OneVL (Lu et al., 2026), DynVLA (Shang et al., 2026), AutoVLA (Zhou et al., 2025c), CuriousVLA (Chen et al., 2026), and Alpamayo (NVIDIA et al., 2025). Second, world-model planners introduce future-representation learning, predictive video objectives, or video-pretrained features to support planning, including DriveLaW (Xia et al., 2026), WoTE (Li et al., 2025c), SeerDrive (Zhang et al., 2025b), DriveWorld-VLA (Jia et al., 2026), and Drive-JEPA (Wang et al., 2026b). Third, sensor-rich planners, such as TransFuser (Chitta et al., 2022), GoalFlow (Xing et al., 2025), Mimir (Xing et al., 2026), DiffusionDrive (Liao et al., 2025), and DiffusionDriveV2 (Zou et al., 2025), incorporate LiDAR to provide explicit geometric information. Finally, methods such as SimScale (Tian et al., 2026), Drive-JEPA (Wang et al., 2026b), and RAP (Feng et al., 2026) exploit additional training data or simulated experience to improve generalization. These directions are not mutually exclusive, but together represent the principal design choices currently used to strengthen end-to-end driving planners.

E Planner Families and Training Data

Our evaluation covers a broad set of end-to-end planners to address five questions about corner-case robustness: training-data scale and model size, multimodal sensing, VLM-based scene understanding, CoT reasoning, and world-model prediction. We also track whether each planner uses extra data beyond the default training setting to examine whether additional training sources translate into measurable gains.

VLA planners and CoT reasoning. Vision-language-action (VLA) planners, including CoT-enabled variants, use the scene understanding and language reasoning capabilities of VLMs to support action planning. Rather than relying only on visual feature extraction, these methods introduce textual descriptions, intermediate reasoning traces, or language-conditioned action representations so that object relations and traffic context can inform the planner. The evaluated VLA planners, including CoT-enabled variants, are Alpamayo R1 and Alpamayo 1.5 (NVIDIA et al., 2025; NVIDIA, 2026), OneVL (Lu et al., 2026), DynVLA (Shang et al., 2026), WAM-Flow (Xu et al., 2025b),

DriveVLA-W0 (Li et al., 2025b), AutoVLA (Zhou et al., 2025c), CuriousVLA (Chen et al., 2026), and ReCogDrive-IL/RL with 2B and 8B VLM backbones (Li et al., 2025d).

World-model planners. World-model planners explicitly learn scene evolution, future representations, future-conditioned scoring signals, or video-pretrained representations that capture driving dynamics for downstream planning. The evaluated planners in this group are Drive-JEPA and Drive-JEPA-PF (Wang et al., 2026b), DriveLaW (Xia et al., 2026), OneVL (Lu et al., 2026), DynVLA (Shang et al., 2026), WoTE (Li et al., 2025c), SeerDrive (Zhang et al., 2025b), DriveVLA-W0 (Li et al., 2025b), and DriveWorld-VLA (Jia et al., 2026).

Planners with LiDAR. Camera-plus-LiDAR planners let us test whether additional geometric sensing is sufficient for rare object response. The evaluated LiDAR-enabled planners are WoTE (Li et al., 2025c), TransFuser (Chitta et al., 2022), SeerDrive (Zhang et al., 2025b), GoalFlow (Xing et al., 2025), Mimir (Xing et al., 2026), DiffusionDrive (Liao et al., 2025), and DiffusionDriveV2 (Zou et al., 2025).

Extra-data planners. We mark a planner as using Extra data when it uses additional driving scenes, video pretraining, or synthetic augmentation beyond the default NAVSIM `navtrain` setting. Alpamayo R1 and Alpamayo 1.5 (NVIDIA et al., 2025; NVIDIA, 2026) use large-scale multi-camera driving data, reported as 80,000 driving hours. SimScale (Tian et al., 2026) uses synthetic-real co-training with 100K NAVSIM real-world scenarios plus 384K diverse off-nominal scenes synthesized by perturbing ego trajectories in ordinary driving scenarios, comprising 237K planner-based and 147K recovery-based examples. DriveLaW (Xia et al., 2026) uses nuPlan and nuScenes for latent driving-world/video generation. WAM-Flow (Xu et al., 2025b) reports training on the full 668K-sample nuPlan training set rather than only the 103K NAVSIM `navtrain` subset. Drive-JEPA and Drive-JEPA-PF (Wang et al., 2026b) use 208 hours of driving-video pretraining. RAP (Feng et al., 2026) uses scalable 3D rasterization to generate synthetic counterfactual viewpoints, recovery maneuvers, and cross-agent views for planner training. Additional QA or reasoning annotations built on the same NAVSIM scenes, such as those used by ReCogDrive (Li et al., 2025d) and AutoVLA (Zhou et al., 2025c), are treated as annotation enrichment rather than extra driving-scene data in Figure 11.

Additional baselines. The comparison also includes GTRS-Aug (Li et al., 2025e), Hydra-MDP (Li et al., 2024), VeteranAD (Zhang et al., 2025a), DrivoR (Kirby et al., 2026), MeanFuser (Wang et al., 2026a), Ego-MLP (Dauner et al., 2024), LEAD (Nguyen et al., 2026), SparseDriveV2 (Sun et al., 2026), PRIX (Wozniak et al., 2026).

F LIMITATIONS

Although our multi-view-consistent object insertion achieves strong visual quality, its explicit geometric consistency mechanism introduces a requirement on the camera configuration. Specifically, we reconstruct the edited object from one camera and render the resulting geometry into a neighboring camera before completing the target view, which requires sufficient overlap between the fields of view of adjacent cameras. When this overlap is too small, the geometry observed and reconstructed in one view cannot be rendered into the neighboring view at the required location, and the proposed insertion pipeline is therefore not directly applicable. The nuPlan camera rig provides substantial overlap between the adjacent views used in our experiments, whereas datasets with narrow-field-of-view or sparsely arranged cameras may not satisfy this requirement. One possible extension is to introduce intermediate virtual cameras and propagate the reconstructed object through a sequence of overlapping viewpoints, using each virtual view as a bridge between the original cameras. This strategy could extend the method to camera rigs with limited overlap, but would require additional rendering and completion stages and therefore increase inference cost.