ODYSSEY: A CLOSED-LOOP BENCHMARK FOR LONG-HORIZON REAL-WORLD DRIVING WITH EXPLICIT NAVIGATION ROUTES

Anonymous authors
Paper under double-blind review

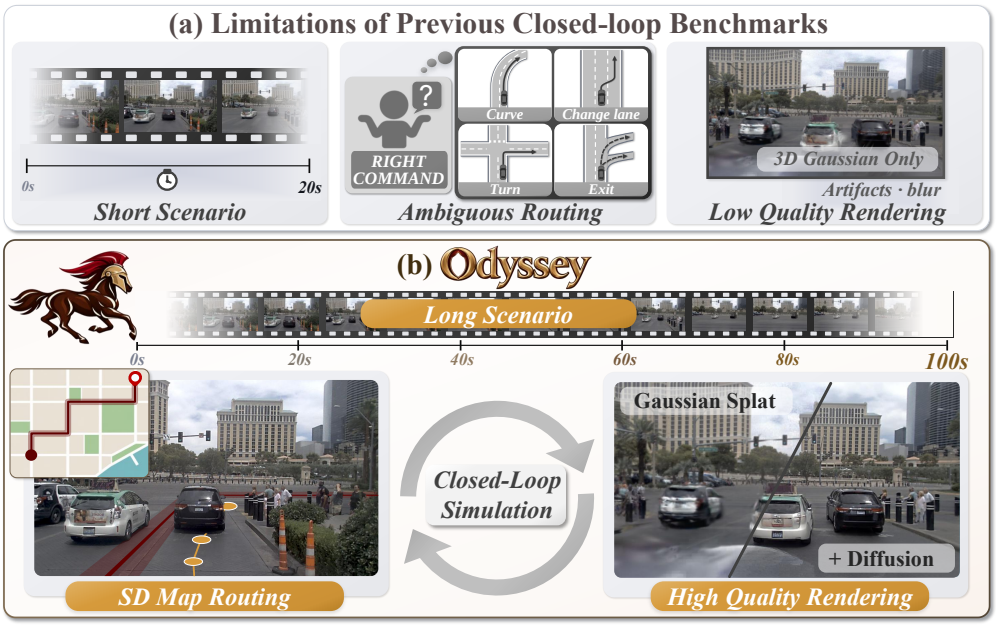


Figure 1: **Odyssey framework.** (a) Previous closed-loop benchmarks suffer from limited scenario horizons, ambiguous command-based routing, and artifacts from 3DGS-only rendering. (b) Odyssey evaluates E2E planners over continuous closed-loop rollouts based on 100-second driving sequences with explicit SD-route guidance and diffusion-refined 3DGS observations, assessing sustained safe navigation toward the intended destination.

ABSTRACT

Closed-loop evaluation of end-to-end driving requires continuous rollouts that reveal how earlier decisions affect subsequent driving. However, existing benchmarks evaluate only short segments and fail to capture later consequences. Ambiguous directional commands also obscure the intended navigation objective. We introduce Odyssey, a closed-loop benchmark for long-horizon driving comprising 100 scenarios, each reconstructed from a 100-second nuPlan driving log to preserve the context of navigation maneuvers and traffic interactions. To provide a consistent navigation objective, Odyssey replaces directional commands with explicit standard-definition (SD) map routes that specify which roads to follow, while sensor-based planning determines local driving actions. Throughout these rollouts, diffusion-based refinement of 3DGS-rendered images reduces rendering artifacts along the ego trajectory. To assess how effectively planners follow these routes and prepare for upcoming maneuvers, we introduce SD Route Compliance and Pre-Lane Change Score. These assessments are complemented by *RouteDS*, which extends the Driving Score with penalties for SD-route deviations and failed lane preparation. We adapt state-of-the-art planners, including vision-language-action (VLA) models, and evaluate their navigation performance using these metrics. Odyssey raises questions about route representation and integration in E2E driving stacks. Benchmark code and adapted baselines will be released publicly.

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.

1 INTRODUCTION

A central challenge in autonomous driving is reliable navigation through complex traffic environments that ensure safety and comfort, as well as efficient progress toward an intended destination. With the growing availability of large-scale driving data, planning-oriented end-to-end (E2E) paradigms have gained prominence (Hu et al., 2023; Liao et al., 2025; Kim et al., 2026; Kirby et al., 2026), in which driving systems directly map onboard sensor observations to driving decisions. Such planners should be evaluated by their ability to sustain safe driving as the vehicle interacts with its environment over time. Since open-loop evaluation independently compares decisions at individual time steps against recorded driving logs, it cannot capture their downstream effects. Accounting for downstream effects via closed-loop evaluation is essential to assess driving performance under evolving traffic conditions.

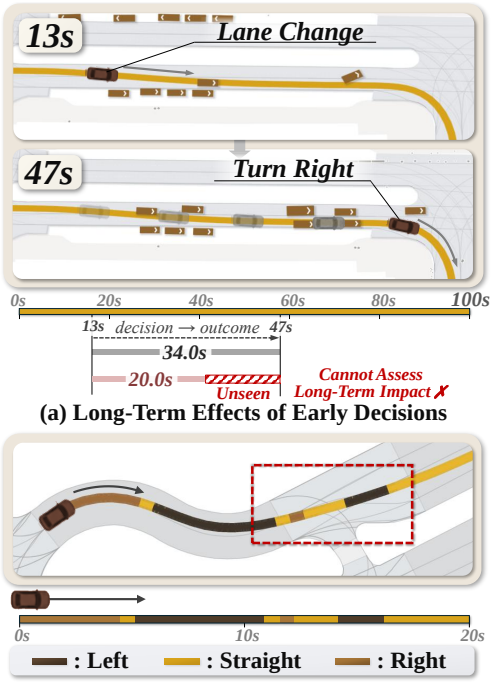


Figure 2: **Limitations of existing evaluation.** (a) Short rollouts cannot assess whether a prior lane change enables a later right turn. (b) Inconsistent command flips obscure navigation intent along a continuous trajectory.

For E2E planners, closed-loop evaluation requires sensor observations that reflect the consequences of the ego vehicle's actions. Neural rendering provides this feedback by reconstructing 3D real-world scenes from driving logs and rendering new observations from the simulated ego pose. Early approaches relied on neural radiance fields (NeRFs) (Yang et al., 2023; Tonderski et al., 2024; Yang et al., 2024), whose volumetric rendering incurs substantial computational cost. To reduce this cost, recent methods adopt 3D Gaussian Splatting (3DGS) (Zhou et al., 2024b; Hess et al., 2025; Zhou et al., 2024a; Chen et al., 2025) for faster rendering while maintaining high visual fidelity. These developments have supported growing research on neural-rendering-based closed-loop benchmarks (Ljungbergh et al., 2024; Zhou et al., 2025; NVIDIA et al., 2025) for E2E planners. However, existing benchmarks are limited in assessing a planner's ability to handle long-term driving tasks (see Figure 1 (a)).

One limitation of existing closed-loop benchmarks (Ljungbergh et al., 2024; Zhou et al., 2025) is their short evaluation horizon, typically limited to approximately 20 seconds. Such short rollouts are insufficient to fully assess the navigation capabilities of E2E planners, particularly when navigation requires a sequence of coordinated maneuvers. For example, as shown in Figure 2 (a), completing an upcoming right turn may require changing lanes in advance. However, the evaluation may terminate before the turn occurs, making it hard to assess the planner's ability to complete the maneuver sequence. Furthermore, high-level meta-action commands used in many E2E benchmarks (Caesar et al., 2020; Dauner et al., 2024; Cao et al., 2025; Zhou et al., 2025; Xu et al., 2026) provide ambiguous guidance for navigation in real-world driving scenarios. As illustrated in Figure 1 (a), a directional command such as "turn right" may correspond to multiple possible maneuvers. For example, if two right exits occur in succession, the command alone does not indicate which exit the planner should take. Annotating these commands in real-world driving datasets may not accurately reflect the driver's intended maneuver (OpenScene Contributors, 2023), as shown in Figure 2 (b). Reliable long-horizon evaluation therefore requires more explicit navigation guidance than high-level meta-action commands alone can provide.

Another challenge is maintaining high rendering quality throughout long-horizon closed-loop simulations. When the ego vehicle deviates from the trajectory recorded during data collection, rendering quality may deteriorate because novel viewpoints lack sufficient observational support. The resulting artifacts can distort the visual input to E2E planners, influence their subsequent decisions, and undermine the reliability of both planning and evaluation. In addition, existing evaluation setups do

not accurately model changes in traffic signals or pedestrian motion, further limiting their ability to represent realistic driving scenarios.

In this paper, we introduce **Odyssey**, a real-world closed-loop benchmark for evaluating E2E driving models over long-horizon rollouts. Odyssey uses neural rendering to reconstruct 3D scenes and support realistic closed-loop simulations. Each scenario includes an explicit navigation route represented on a standard-definition (SD) map. By combining long-horizon evaluation with explicit route guidance, Odyssey enables a comprehensive assessment of models' ability to navigate complex road structures safely—a capability that existing benchmarks have not adequately evaluated.

For these extended rollouts, Odyssey builds on the explicit geometric representation of 3D Gaussian Splatting (Kerbl et al., 2023), followed by diffusion-based refinement for improving visual fidelity. Using OmniRe (Chen et al., 2025), we reconstruct continuous 100-second driving environments with rigid vehicles and deformable pedestrians whose motion is controlled independently of the static background. We further apply its time-conditioned deformation field to traffic lights to model changes in their color states throughout the sequence. To reduce blur and recover missing detail at novel ego poses, we apply single-step diffusion refinement (Wu et al., 2025; NVIDIA, 2025b) to the rendered camera images without modifying the underlying scene representation or simulated state.

Odyssey provides navigation routes as explicit guidance for E2E planners. By specifying the intended sequence of road segments, they resolve the ambiguity inherent in directional commands and eliminate the need to determine when to issue individual high-level meta-action commands. For each scenario, we construct an SD-map route from trip data of the data-collection fleet. However, this road-level representation encodes only coarse road-level topology and does not specify lane-level structure or the precise goal location that a driving model should reach. E2E planners must therefore interpret the high-level route jointly with the scene perceived by onboard sensors and translate it into a concrete, scene-consistent navigation goal. This requires the planner to reason about how an abstract road-level route is translated to appropriate lane-level maneuvers within the current road geometry. Odyssey establishes a foundation for evaluating such reasoning capability of E2E planners.

We design the closed-loop benchmark to enable systematic comparison of different E2E planners. We curate 100 scenarios from the raw nuPlan data (Karnchanachari et al., 2024), covering sequences of complex navigation maneuvers and diverse road topologies. We further introduce two evaluation metrics, termed *Pre-Lane Change Score* and *RouteDS*, which assess whether the vehicle is appropriately positioned for upcoming navigation maneuvers, as well as driving safety and progress along the intended route. We evaluate a set of baseline planners adapted from recent state-of-the-art E2E and vision-language-action (VLA) models, incorporating navigation routes as additional input.

Our main contributions are as follows:

- We introduce **Odyssey**, a high-fidelity real-world closed-loop benchmark for evaluating whether E2E planners can sustain safe and goal-directed driving under explicit route guidance over long-horizon rollouts.
- We evaluate planners over continuous rollouts based on recorded 100-second sequences, with explicit SD-map routes that resolve ambiguities in high-level driving commands and enable consistent assessment of navigation performance across complex road topologies.
- We develop a closed-loop simulation framework that combines the structural consistency of 3DGS with diffusion-based refinement to improve rendering fidelity when ego-vehicle motion reveals poorly reconstructed or previously unobserved regions.
- We will release the benchmark, evaluation code, and baseline results to enable reproducible comparison and facilitate further research on long-horizon closed-loop evaluation for autonomous driving.

2 Related Work

Open-loop displacement metrics overlook how planning actions affect subsequent driving (Caesar et al., 2020; Hu et al., 2022; 2023). NAVSIM introduces simulation-based driving metrics (Dauner et al., 2024), and NAVSIM v2 adds pre-rendered feedback (Cao et al., 2025). Neural simulators render observations from evolving ego poses in reconstructed environments (Yang et al., 2023; Tonderski et al., 2024; Ljungbergh et al., 2024; Zhou et al., 2025; NVIDIA et al., 2025), but short

Table 1: Comparison of driving benchmarks and simulation frameworks. ✓, △, ×, and − denote support, partial support, no documented support, and not applicable, respectively.

Benchmark	Real-log scenarios	Sensor closed-loop	Realistic images	Navigation input	Recorded sequence duration	Rendering type	Traffic-light state	Pedestrian motion
nuScenes	✓	×	✓	Commands	-	×	-	-
NAVSIM	✓	×	✓	Commands	-	×	-	-
NAVSIMv2	✓	×	✓	Commands	-	Offline 3DGS	-	-
WOD-E2E	✓	×	✓	Commands	-	×	-	-
nuPlan	✓	×	×	HD route	15 s	×	-	-
Waymax	✓	×	×	HD route	9 s	×	-	-
MetaDrive	△	✓	×	Target points	User-set	Rasterization	✓	✓
CARLA Leaderboard	×	✓	×	Target points	∼ 3–5 min	Rasterization	✓	✓
Bench2Drive	×	✓	×	Target points	∼ 15–30 s	Rasterization	✓	✓
NeuroNCAP	✓	✓	✓	Commands	20 s	NeRF	×	×
HUGSIM	✓	✓	✓	Commands	≤ 20 s	3DGS	×	×
DriveArena	△	✓	✓	Commands	20 s	Diffusion	×	×
Odyssey (ours)	✓	✓	✓	**SD route**	**100 s**	**3DGS + diffusion**	✓	✓

evaluation sequences limit assessment across successive maneuvers (Ljungbergh et al., 2024; Zhou et al., 2025). Long-horizon evaluation also requires an explicit route. Directional commands can leave consecutive branches ambiguous (Hu et al., 2023; Jiang et al., 2023; Dauner et al., 2024; Zhou et al., 2025), and nearby target points provide limited route context (Dosovitskiy et al., 2017; Jia et al., 2024).

3 ODYSSEY

Odyssey provides a closed-loop simulation platform for evaluating long-horizon navigation along explicit routes on an SD map. In each 100-second scenario, the planner predicts a local driving trajectory based on camera observations and route guidance. The simulator advances the ego vehicle along the predicted trajectory and uses a neural rendering engine to generate new camera observations from its updated pose. We evaluate planner performance in terms of driving safety, route progress, adherence to the road-level route, and lane selection before intersections.

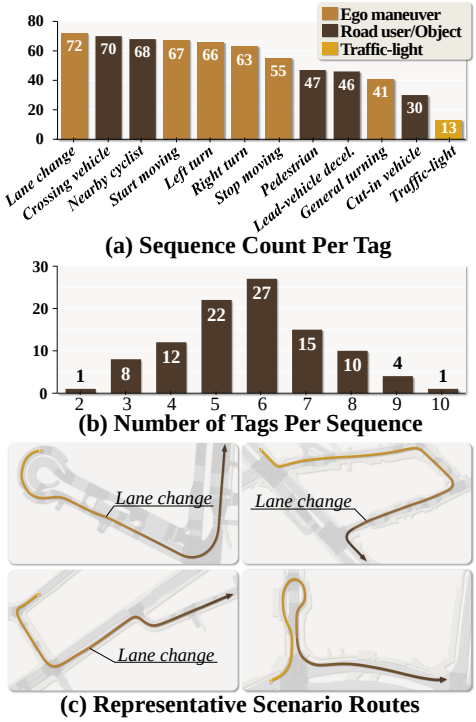
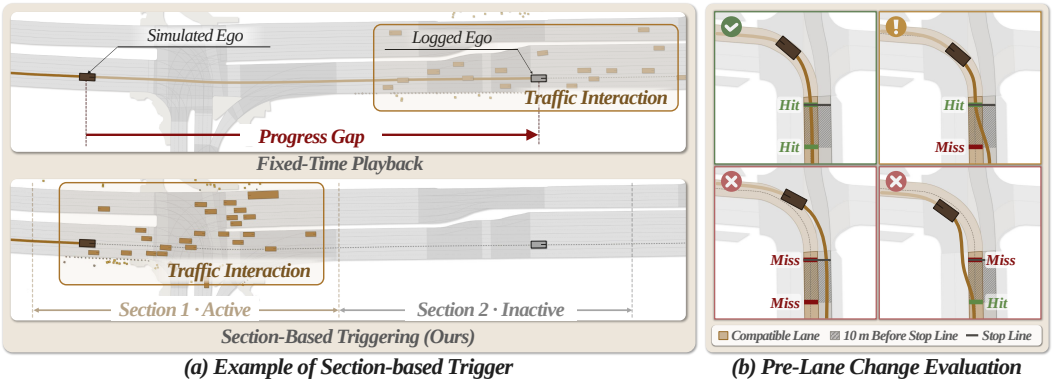


Figure 3: Scenario composition and example routes in Odyssey.

3.1 SCENARIO CURATION

Odyssey comprises 100 driving scenarios curated from nuPlan (Karnchanachari et al., 2024). To capture diverse driving situations over extended sequences, we combine RefAV (Davidson et al., 2026)-based tagging with human review of navigation maneuvers and traffic interactions. The distribution of individual tags is reported in Figure 3 (a). The selected scenarios contain multiple maneuver and interaction tags, with the number of tags per scenario shown in Figure 3 (b). The selected set includes 13 scenarios involving traffic lights and 47 involving pedestrians, covering both signal compliance and safety around vulnerable road users.

Within this set, we prioritize scenarios with complex road layouts and sequences of navigation maneuvers. Figure 3 (c) shows representative routes involving successive turns, branching roads, and lane changes. In particular, scenarios with a lane change followed by a turn allow us to assess whether the planner changes lanes in preparation for the turn. The set includes 49 such scenarios, 16 of which have more than 20 seconds between the start of the lane change and the start of the turn. A single 20-second evaluation window cannot capture the onset of both maneuvers in these cases.



(a) Example of Section-based Trigger *(b) Pre-Lane Change Evaluation*

Figure 4: **Traffic triggering and pre-lane change evaluation.** (a) When the simulated ego is slower than the logged ego, fixed-time playback (top) places traffic interactions ahead of it. By activating traffic when the ego enters a section and keeping subsequent sections inactive, section-based triggering (bottom) aligns traffic interactions with ego progress. (b) Pre-Lane Change requires both driving in route-compatible lanes throughout the final 10 m and crossing the stop line in a compatible lane. The four examples illustrate cases that satisfy both conditions, only one, or neither.

3.2 SD Map Routing

SD maps represent the road network through coarse polylines and connectivity at intersections and branches. A navigation route specifies the sequence of roads to follow. To construct each route, we match the full logged ego trajectory to OpenStreetMap (OSM) using HMM-based map matching (Newson & Krumm, 2009), yielding a single polyline from the scenario start to the destination. We manually check the matched roads and route continuity against the logged trajectory and SD map. During closed-loop simulation, we update the route segment provided to the planner according to the ego vehicle's current position.

3.3 Closed-Loop Simulations

3D Gaussian Scene Reconstruction. Odyssey reconstructs each 3D sequence using OmniRe (Chen et al., 2025), with rigid transformations for vehicles, SMPL-based deformation (Loper et al., 2015) for pedestrians, and a learned deformation field for other actors. Traffic lights have fixed geometry and position, with time-dependent spherical-harmonic coefficients and opacity capturing signal changes. To stabilize road geometry, we incorporate *ground modeling* inspired by HUGSIM (Zhou et al., 2025), representing the road surface separately using thin Gaussian disks initialized from LiDAR. Their heights and normals follow a fixed LiDAR-derived height field as we optimize their in-plane geometry and appearance. Opacity regularization suppresses overlapping background Gaussians near the road surface.

When training these scene representations over long sequences, uniform image sampling concentrates updates in regions where the recording vehicle stops or moves slowly. We therefore apply *coverage-aware sampling*, weighting each view by aggregating inverse observation counts over its visible scene voxels. This increases sampling probabilities for views covering less-observed regions.

Reconstruction-Conditioned Visual Refinement. Rendering from ego poses beyond the logged viewpoints can expose blur and missing detail in poorly observed regions. We reduce these artifacts using Fixer (NVIDIA, 2025b), which adapts the diffusion-based refinement of Difix3D+ (Wu et al., 2025) to a Cosmos backbone (NVIDIA, 2025a). Fixer refines each image in a single denoising step at a fixed timestep without adding random noise. We fine-tune it offline on paired degraded renderings and recorded images, following the data preparation strategy of Difix. During simulation, the model refines rendered observations before they are passed to the planner.

Ego Vehicle Control. The ego vehicle tracks the planner's trajectory at 10 Hz using a linear quadratic regulator (LQR) (Anderson & Moore, 2007). At each step, the LQR computes acceleration and steering rate that a bicycle model (Kong et al., 2015) uses to update the ego state.

3.4 Traffic Simulation

Traffic Configuration. Following nuPlan (Caesar et al., 2021), we evaluate planners under non-reactive traffic, where other actors replay recorded motion, and reactive traffic, where surround-

ing vehicles use IDM-based longitudinal control (Treiber et al., 2000). Vehicles enter the simulation as they appear in the logs. At unprotected intersections, newly introduced vehicles can create conflicts that longitudinal control alone cannot resolve. We therefore add intersection-handling logic inspired by CARLA's Traffic Manager (Dosovitskiy et al., 2017) to prevent deadlocks.

SECTION-BASED TRAFFIC TRIGGERING. When the ego vehicle travels faster or slower than the logged vehicle, traffic playback on the original timeline can cause it to miss recorded interactions. We therefore introduce section-based traffic triggering to align traffic activation and traffic light signal timing with ego progress in both non-reactive and reactive settings. Since traffic interactions are concentrated around intersections, we place section boundaries between intersections. As the ego vehicle approaches a section, we activate its traffic using a fixed time offset determined by the difference between the logged and simulated ego entry times. The same offset is applied to the section's traffic light signals, which then advance through their recorded states at the original rate. Figure 4 (a) compares this approach with traffic playback on the original timeline.

3.5 EVALUATION PROTOCOL

We introduce *SD Route Compliance*, *Pre-Lane Change Accuracy*, *Pre-Lane Change Score*, and *RouteDS* to assess navigation performance. We also report Efficiency and Comfortness.

SD ROUTE COMPLIANCE (SDC). We map-match the executed ego trajectory to assess adherence to the provided SD-map route. A rollout scores 1 if all matched road segments belong to the route and 0 otherwise, even if the vehicle stays on drivable roads. SDC is the mean score across rollouts.

PRE-LANE CHANGE ACCURACY (PLCA) & SCORE (PLCS). Long-horizon navigation requires timely lane positioning to follow the route through successive intersections. We assess this ability via PLCA and PLCS at intersections where continuing along the route requires specific approach lanes. PLCA averages scores over evaluated intersections encountered in a rollout. PLCS averages scores over all evaluation intersections along the full scenario route, counting unencountered intersections as 0 to account for incomplete rollouts. At each stop line, the vehicle scores 1 if it crosses in a route-compatible lane and has maintained compatible lane occupancy for the preceding 10 m. A later entry scores 0.5, and crossing outside a compatible lane scores 0 (Fig. 4 (b)).

ROUTEDS. We extend the Driving Score of Bench2Drive (Jia et al., 2024) with penalties for SD-route deviations and failed pre-lane change evaluations. These additions complement route completion and penalties for at-fault collisions (Dauner et al., 2024), lane deviation, and red-light violations. For each scenario, the route completion ratio is multiplied by a cumulative penalty factor initialized to 1, with multipliers in $[0, 1]$ applied for infractions. Each failed pre-lane change evaluation incurs a separate penalty. We report the mean RouteDS across scenarios. Detailed metric definitions and penalty values are provided in the Appendix.

4 IMPLEMENTATION DETAILS

4.1 SIMULATOR IMPLEMENTATION DETAILS

We reconstruct the entire 100-second sequence as a single scene using OmniRe (Chen et al., 2025), initializing Gaussians from LiDAR point clouds and accelerating optimization with FastGS (Ren et al., 2026). Each scene contains 8,000 images captured at 10 Hz across eight cameras and is optimized for 140,000 iterations. Following MTGS (Li et al., 2025a), we additionally apply per-image affine color corrections to compensate for exposure variations. We fine-tune pretrained Fixer (NVIDIA, 2025b) on paired degraded renderings and corresponding recorded images, following the data preparation strategy of Difix (Wu et al., 2025). Details of training data generation and fine-tuning are provided in the Appendix.

4.2 SD-ROUTE-CONDITIONED BASELINES

We adapt existing planners to SD-route guidance and compare their closed-loop driving performance over long-horizon rollouts. The baselines comprise TransFuser (Chitta et al., 2023), DiffusionDrive (Liao et al., 2025), DrivoR (Kirby et al., 2026), SafeDrive (Kim et al., 2026), and ReCogDrive (Li et al., 2026b). All baselines predict a 4-second future trajectory sampled at 2 Hz and are trained on the NAVSIM training set (Dauner et al., 2024).

Table 2: Ablation study of reconstruction components on Odyssey. KID_{in} and KID_{ex} evaluate interpolated and extrapolated views, respectively.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	KID_{in} ↓	KID_{ex} ↓	Latency (ms/image) ↓
OmniRe (Baseline)	24.34	0.788	0.399	29.5	52.2	**17.8**
+ Color Correction	24.98	0.793	0.392	25.4	47.0	18.1
+ Coverage-aware Sampling	25.22	0.796	0.382	24.1	42.8	18.1
+ Ground Modeling	25.20	0.794	0.376	20.1	32.8	32.4
+ Pretrained Fixer	25.00	**0.801**	0.330	14.6	17.6	39.1
+ Fixer Fine-tuning **(Ours)**	**25.34**	0.786	**0.233**	**5.6**	**8.9**	39.1

Table 3: Effect of Fixer refinement on perception and driving performance.

Observation	Open-loop	Closed-loop	
	mIoU ↑	mIoU ↑	*RouteDS* ↑
GT Image	35.4	–	–
3D GS Only	31.5	26.0	53.80
+ Diffusion	31.9	28.2	52.62
+ Fine-tuning	**33.7**	**29.4**	**54.43**

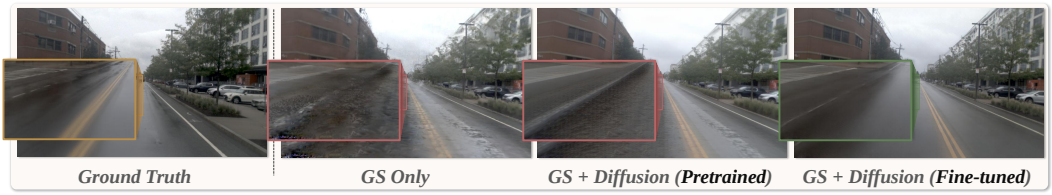


Figure 5: **Effect of diffusion refinement at a novel viewpoint.** Renderings are generated from a viewpoint shifted 2 m to the left of the ground-truth camera position.

At each planning step, we sample 120 route points at 1-meter intervals ahead of the ego's projection onto the SD-route polyline. Their positions, directions, and headings are expressed in the ego frame to provide navigation guidance in place of driving commands. For vision-based planners, a linear projection and max pooling over groups of five consecutive points produce 24 route tokens. Fourier embeddings of group starting poses and learned order embeddings encode their spatial placement and sequence. Each planning decoder layer attends to these tokens as keys and values through an additional cross-attention module before its feed-forward network. For ReCogDrive, we summarize the route into a single token added to the ego-status features and supply a route-derived navigation sentence to the VLM. Model-specific adaptations are detailed in the Appendix.

Dedicated reasoning mechanisms that relate SD-map routes to the perceived scene and required driving maneuvers are left to future work.

5 Experimental Evaluation

5.1 Scene Reconstruction Quality

We evaluate reconstruction quality using PSNR, SSIM, LPIPS, and KID (Bińkowski et al., 2018). Following HUGSIM (Zhou et al., 2025), KID measures squared maximum mean discrepancy between rendered and recorded Inception features using a polynomial kernel, requiring no paired ground truth at novel viewpoints. Table 2 shows cumulative LPIPS and KID reductions from color correction, coverage-aware sampling, ground modeling, and diffusion refinement. The reported per-image latency was measured on an NVIDIA B200 GPU. Fixer fine-tuning further reduces blur and artifacts (Fig. 5). Our appearance model captures traffic-light changes missed by OmniRe (Fig. 6), enabling evaluation of planner responses.

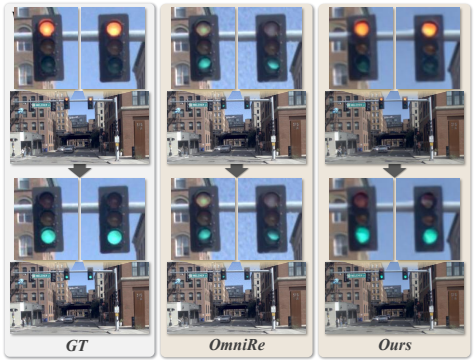


Figure 6: **Traffic-light modeling.** Our deformation field reproduces the red-to-green transition, which OmniRe fails to recover.

5.2 Effect of Diffusion Refinement on Closed-Loop Planning

Table 3 evaluates visual refinement with the reconstructed scene fixed, using the same SafeDrive checkpoint with IL scoring and safety filtering. Open-loop mIoU uses the logged trajectory, while closed-loop mIoU compares all variants at identical camera poses from the rollout with fine-tuned refinement. Diffusion alone improves mIoU but slightly lowers *RouteDS*. Odyssey-specific fine-tuning achieves the best results in both metrics, improving scene interpretation and driving performance.

Table 4: Comparison of baseline planners with and without SD-route guidance under non-reactive and reactive traffic. Eff. and Comf. denote Efficiency and Comfortness, respectively. Bold and underlined values indicate the best and second-best results in each metric, respectively.

Method	SD-Route Guidance	Odyssey Non-reactive						Odyssey Reactive						NAVSIM Open Loop
		RouteDS	SDC	PLCA	PLCS	Eff.	Comf.	*RouteDS*	SDC	PLCA	PLCS	Eff.	Comf.	PDMS
Vision-based planners														
LTF	×	16.2	51	66.0	50.3	63.9	**100**	16.3	54	63.5	49.1	76.1	99	83.8
	✓	20.1	95	57.1	56.1	68.2	96	23.0	93	53.6	52.6	78.0	93	85.6
DiffusionDrive	×	25.6	59	78.8	63.7	61.4	**100**	24.2	59	77.0	63.1	75.7	**100**	85.8
	✓	35.9	**100**	**82.0**	**80.5**	63.4	99	41.0	**100**	**81.2**	**80.2**	71.5	99	86.9
DrivoR	×	20.6	50	75.5	62.8	**104.7**	44	25.5	53	76.6	62.8	**133.4**	47	**93.1**
	✓	36.5	90	72.8	66.0	84.5	60	41.4	89	71.8	66.0	107.4	64	92.2
SafeDrive	×	27.5	41	79.3	63.7	69.4	47	30.4	45	78.8	62.8	86.3	40	90.8
	✓	**44.4**	82	75.0	69.8	68.3	44	**48.9**	79	73.8	69.5	81.3	49	90.6
VLA-based planners														
ReCogDrive	×	15.4	71	75.7	59.0	76.9	**100**	20.3	62	77.1	58.7	93.7	99	90.8
	✓	21.5	83	74.4	55.8	59.3	98	26.3	84	75.6	56.7	75.5	98	90.4

Table 5: Effects of safety scoring and reinforcement learning on driving performance under non-reactive and reactive traffic and on NAVSIM. All models use SD-map route guidance.

Configuration	Odyssey Non-reactive						Odyssey Reactive						NAVSIM
	RouteDS	SDC	PLCA	PLCS	Eff.	Comf.	*RouteDS*	SDC	PLCA	PLCS	Eff.	Comf.	PDMS
SafeDrive													
Safety Scoring	44.4	82	75.0	69.8	**68.3**	44	48.9	79	73.8	69.5	**81.3**	49	**90.6**
IL Scoring	53.9	95	**77.1**	**74.4**	66.5	**100**	55.3	94	**78.0**	**74.4**	75.7	**100**	89.5
IL Scoring + Safety Filtering	**54.4**	**96**	76.7	72.7	65.3	**100**	**56.1**	**96**	77.7	74.1	78.3	**100**	89.5
ReCogDrive													
IL + GRPO Fine-tuning	21.5	83	74.4	55.8	**59.3**	98	26.3	84	75.6	56.7	**75.5**	98	**90.4**
IL only	**37.7**	**97**	**79.4**	**73.8**	58.9	**100**	**42.3**	**96**	**79.6**	**75.0**	70.4	**100**	86.8

5.3 Baseline Analysis

Table 4 compares planners with and without SD-route guidance under non-reactive and reactive traffic. SD-route guidance consistently improves SDC across planners, indicating better adherence to the intended route. This improvement is illustrated in Figure 7, where SafeDrive deviates from the intended route when the driving command flips but follows it with SD-route guidance. The table also shows that higher NAVSIM open-loop PDMS does not consistently correspond to higher RouteDS. This mismatch highlights the need for continuous closed-loop evaluation, where planning decisions affect subsequent observations and outcomes.

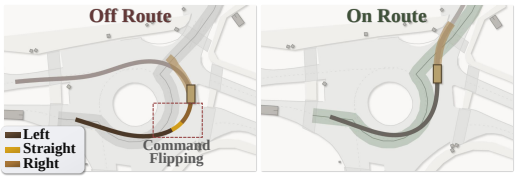
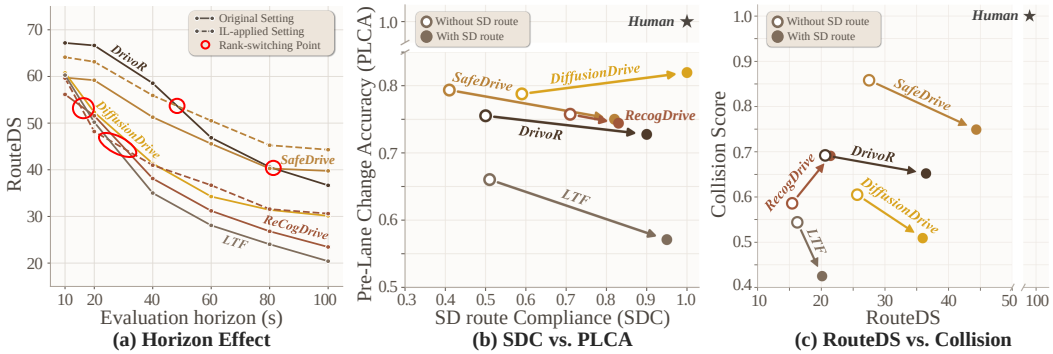


Figure 7: **Driving commands vs. SD-route guidance.** With driving commands, SafeDrive deviates from the intended route when the command flips. With SD-route guidance, it follows the intended route.

5.4 Do Open-Loop Gains Transfer to Closed-Loop Driving?

Recent research has explored trajectory scoring and reinforcement learning as ways to improve driving safety. In NAVSIM-based training, trajectory scorers learn to predict simulation-based driving scores, while reinforcement learning uses these scores as rewards. Both objectives evaluate individual trajectories in an open-loop setting, without directly optimizing how the planner responds to new observations after execution.

Table 5 shows the limits of this approach. For SafeDrive, IL scoring outperforms safety scoring despite similar PDMS values. Safety filtering further improves performance by selecting the highest IL-scored trajectory among candidates that pass safety checks. For ReCogDrive, GRPO improves PDMS but lowers closed-loop performance relative to IL-only training. These results suggest that



(a) Horizon Effect **(b) SDC vs. PLCA** **(c) RouteDS vs. Collision**

Figure 8: **Effects of evaluation horizon and route guidance under non-reactive traffic.** (a) RouteDS decreases as the evaluation horizon grows, and model rankings change at the highlighted crossings. Solid and dashed curves show the original and IL-based settings, respectively. (b) SD-route guidance improves route compliance, but does not consistently improve timely lane positioning. (c) Higher RouteDS with SD-route guidance does not consistently coincide with higher collision scores, where higher scores indicate fewer collision penalties.

optimizing individual trajectories does not ensure better continuous driving. Odyssey closes this gap by evaluating planners on the consequences of their own decisions, not on isolated trajectories.

5.5 Why Evaluate over Long Horizons?

Short-horizon evaluation tests whether a planner can handle an immediate driving situation, but may not reveal whether it can sustain good performance over successive maneuvers. Figure 8(a) plots each model's RouteDS against the evaluation horizon. RouteDS decreases as the horizon grows for all planners, reflecting the difficulty of sustaining performance over longer rollouts. More importantly, the curves cross: DrivoR leads at short horizons but falls behind SafeDrive as driving continues. Short-horizon results therefore do not fully capture a planner's ability to sustain safe, route-following behavior, which long-horizon evaluation reveals by testing successive decisions and the situations they create.

5.6 What Is Needed Beyond Route Guidance?

Figure 8(b) shows that replacing driving commands with SD-route guidance substantially improves SDC, whereas PLCA improves only slightly, remains similar, or decreases. Knowing the intended roads thus helps models follow the route, but not position themselves in the lanes needed for upcoming maneuvers. Figure 8(c) reveals a similar gap in safety. Although SD-route guidance raises RouteDS for every planner, the collision score decreases for four of the five. Route guidance alone therefore ensures neither lane-level preparation nor safe interaction with surrounding traffic, and closing both gaps remains an open direction for future research on Odyssey.

6 Future Works

Surrounding vehicles can appear blurred or distorted when viewed from angles not covered by the driving logs. Using diffusion guidance during 3D Gaussian Splatting optimization could improve reconstruction in these regions. Future work could also improve the temporal consistency of diffusion refinement. Beyond visual quality, adversarial agents could introduce more challenging traffic interactions for safety evaluation.

7 Conclusion

We introduced Odyssey, a closed-loop benchmark for long-horizon driving in environments reconstructed from real-world logs. By pairing continuous 100-second scenarios with explicit SD-map routes, Odyssey evaluates how well planners sustain safe, route-following driving through RouteDS, SDC, and PLCA. Using this benchmark, we find that planner rankings shift as the horizon grows and that open-loop gains do not necessarily carry over to closed-loop driving. We further find that SD-route guidance improves route adherence but does not ensure timely lane positioning or safe interaction with traffic. These findings suggest that driving ability emerges from successive decisions, and we hope Odyssey serves as a testbed for planners that can sustain it.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and the code implementation of our research methods. We did not use these tools to develop research concepts, propose or refine hypotheses, design experiments, or interpret experimental results. Tasks involving the development or writing of mathematical proofs are not applicable to this work. Additionally, we used generative AI tools for language editing and code debugging. We have reviewed all AI-assisted work. Specifically, we checked the translations and edits to ensure that they preserved the intended meaning, and manually inspected, executed, and tested the AI-assisted code. 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 introduces Odyssey, a benchmark for evaluating the safety and route-following performance of autonomous driving models in long-horizon closed-loop simulations using scenarios curated from the existing nuPlan dataset. Performance on this benchmark should not be interpreted as a guarantee of safety in real-world driving. We encourage the responsible use of Odyssey for research and evaluation, with additional validation required before deploying autonomous driving systems on public roads.

REPRODUCIBILITY STATEMENT

We support reproducibility by providing detailed descriptions of the benchmark construction, simulation framework, evaluation metrics, baseline adaptations, and experimental settings in the main paper and appendix. The benchmark code, evaluation code, and adapted baselines will be released upon publication.

REFERENCES

Brian D. O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. Dover Publications, 2007.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11618–11628. IEEE, 2020.

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

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, 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.

Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *International Journal of Computer Vision*, 2026.

Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo De Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. In *International Conference on Learning Representations*, volume 2025, pp. 85508–85527, 2025.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1290–1299, 2022.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

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, 2024.

Cainan Davidson, Deva Ramanan, and Neehar Peri. Refav: Towards planning-centric scenario mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21537–21548, 2026.

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, pp. 1–16, 2017.

Leon Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *Advances in Neural Information Processing Systems*, volume 36, pp. 7730–7742, 2023.

Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11982–11992. IEEE, 2025.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pp. 533–549. Springer, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023.

Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21983–21994, 2023.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. DriveTransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, 2025.

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*, pp. 8340–8350, 2023.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636. IEEE, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

Jungho Kim, Jiyong Oh, Seunghoon Yu, Hongjae Shin, Donghyuk Kwak, and Jun Won Choi. Safedrive: Fine-grained safety reasoning for end-to-end driving in a sparse world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, et al. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026.

Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1094–1099, 2015. doi: 10.1109/IVS.2015.7225830.

Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2Drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in CARLA-V2). In *European Conference on Computer Vision*, pp. 142–158. Springer, 2024a.

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*, 45(3):3461–3475, 2022.

Tianyu Li, Yihang Qiu, Zhenhua Wu, Carl Lindström, Peng Su, Matthias Nießner, and Hongyang Li. MTGS: Multi-traversal gaussian splatting. *arXiv preprint arXiv:2503.12552*, 2025a.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27137–27146, 2025b.

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*, 2026a.

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*, 2026b.

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. *arXiv preprint arXiv:2406.06978*, 2024b.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047. IEEE, 2025.

Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, and Wenming Yang. VastGaussian: Vast 3D gaussians for large scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

William Ljungbergh, Adam Tonderski, Joakim Johnander, Holger Caesar, Kalle Åström, Michael Felsberg, and Christoffer Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *European Conference on Computer Vision*, pp. 161–177. Springer, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248:1–248:16, 2015.

Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4990–4999, 2017.

Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 336–343, 2009. doi: 10.1145/1653771.1653818.

NVIDIA. Cosmos-Predict2-0.6B-Text2Image. `https://huggingface.co/nvidia/Cosmos-Predict2-0.6B-Text2Image`, 2025a.

NVIDIA. Fixer: Improving 3D reconstructions with single-step diffusion models. `https://github.com/nv-tlabs/Fixer`, 2025b.

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

OpenScene Contributors. OpenScene: The largest up-to-date 3D occupancy prediction benchmark in autonomous driving. `https://github.com/OpenDriveLab/OpenScene`, 2023.

Shiwei Ren, Tianci Wen, Yongchun Fang, and Biao Lu. Fastgs: Training 3d gaussian splatting in 100 seconds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26094–26103, 2026.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. SparseDrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8795–8801. IEEE, 2025.

Yingqi Tang, Zhuoran Xu, Zhaotie Meng, and Erkang Cheng. HiP-AD: Hierarchical and multi-granularity planning with deformable attention for autonomous driving in a single decoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14895–14904. IEEE, 2024.

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.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. PARA-Drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15449–15458, 2024.

Jay Zhangjie Wu, Yuxuan Zhang, Haithem Turki, Xuanchi Ren, Jun Gao, Mike Zheng Shou, Sanja Fidler, Zan Gojcic, and Huan Ling. DIFIX3D+: Improving 3d reconstructions with single-step diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26024–26035, 2025.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6119–6132, 2022.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. In *International Conference on Learning Representations*, volume 2024, pp. 16739–16766, 2024.

Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1389–1399. IEEE, 2023.

Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M. Alvarez, and Zuxuan Wu. DriveSuprim: Towards precise trajectory selection for end-to-end planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026. doi: 10.1609/aaai.v40i14.38178.

Zhenlong Yuan, Chengxuan Qian, Jing Tang, Rui Chen, Zijian Song, Lei Sun, Xiangxiang Chu, Yujun Cai, Dapeng Zhang, and Shuo Li. AutoDrive-R^2: Incentivizing reasoning and self-reflection capacity for VLA model in autonomous driving. In *International Conference on Learning Representations*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21336–21345. IEEE, 2024a.

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*, 2025.

Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21634–21643. IEEE, 2024b.

APPENDIX FOR ODYSSEY

Contents

A Related Work **15**

A.1 End-to-End Driving Models 15

A.2 Evaluation of E2E Planning 16

A.3 Driving Intention for End-to-End Models 16

B Implementation Details of Closed-loop Simulator **16**

B.1 3D Gaussian Reconstruction 16

B.2 Fixer fine-tuning 18

B.3 Section-based Traffic Triggering 19

C Evaluation Metric Details **20**

C.1 Rollout Termination Criteria 20

C.2 Pre-Lane Change Metrics 21

C.3 RouteDS Components 21

C.4 Additional Driving Metrics 21

D SD-Route Conditioning of Baseline Planners **22**

E Additional Analysis **22**

E.1 Comparison with SOTA 3DGS methods 23

E.2 Additional Qualitative Analysis 23

E.3 Detailed RouteDS Analysis 23

A RELATED WORK

A.1 END-TO-END DRIVING MODELS

Planning-oriented E2E driving integrates perception, prediction, and planning (Hu et al., 2022; 2023), with subsequent work improving scene representation and task interaction (Jiang et al., 2023; Sun et al., 2025; Weng et al., 2024; Jia et al., 2025). Building on these representations, planners jointly predict trajectories and controls (Wu et al., 2022; Jia et al., 2023) or separate path geometry from speed prediction (Jaeger et al., 2023; Tang et al., 2025). To handle multiple feasible actions, diffusion models generate diverse trajectories (Liao et al., 2025), and candidate-based planners learn to score alternatives (Li et al., 2024b; Yao et al., 2026). World models further inform planning through predicted future states and action risks (Li et al., 2025b; Kim et al., 2026), and provide imagined interactions for reinforcement learning (Li et al., 2024a).

Vision-language-action models extend sensor-based planning with language understanding and reasoning. SimLingo aligns language instructions with driving actions (Renz et al., 2025), and DriveVLA-W0 incorporates future-image prediction to enrich visual learning beyond action supervision (Li et al., 2026a). ReCogDrive and AutoDrive-R^2 further combine reasoning with reinforcement learning to improve driving decisions (Li et al., 2026b; Yuan et al., 2026). Odyssey adapts vision-based and VLA planners to SD-route guidance for evaluating sustained navigation.

A.2 Evaluation of E2E Planning

Evaluating E2E planners requires assessing the consequences of their actions, which open-loop displacement based metrics overlook (Caesar et al., 2020; Hu et al., 2022; 2023). NAVSIM (Dauner et al., 2024) moves beyond imitation accuracy through simulation-based driving metrics, while NAVSIM v2 (Cao et al., 2025) approximates subsequent feedback using pre-rendered observations. However, continuous replanning requires observations reflecting ego motion. Graphics-based simulators provide this feedback but introduce a visual domain gap (Dosovitskiy et al., 2017; Li et al., 2022; Jia et al., 2024), whereas planning simulators using real logs lack sensor rendering (Caesar et al., 2021; Karnchanachari et al., 2024; Gulino et al., 2023).

Neural-rendering-based simulators bring sensor feedback to recorded driving environments by reconstructing scenes and rendering observations in response to ego actions (Yang et al., 2023; Tonderski et al., 2024; NVIDIA et al., 2025). Within this paradigm, NeuroNCAP (Ljungbergh et al., 2024) examines collision avoidance in NeRF- reconstructed scenarios, and HUGSIM (Zhou et al., 2025) provides closed-loop evaluation through 3DGS rendering and controllable traffic. However, deviations from the logged trajectory can expose poorly reconstructed regions (Zhou et al., 2024b; 2025; Hess et al., 2025; Chen et al., 2025), producing artifacts that compromise the reliability of sensor feedback. Odyssey addresses this issue by refining 3DGS-rendered observations with diffusion to reduce artifacts at novel ego poses.

A.3 Driving Intention for End-to-End Models

Navigation guidance specifies the intended path when multiple driving maneuvers are feasible (Codevilla et al., 2018). Benchmarks based on real driving logs (Hu et al., 2023; Jiang et al., 2023; Dauner et al., 2024; Cao et al., 2025; Ljungbergh et al., 2024; Zhou et al., 2025; Xu et al., 2026) commonly express this intent through driving commands, but this compact representation cannot uniquely distinguish consecutive branches sharing the same direction. CARLA-based benchmarks (Dosovitskiy et al., 2017), including Bench2Drive (Jia et al., 2024), provide explicit guidance through target points or route waypoints in synthetic environments. However, guidance limited to nearby targets provides insufficient route context for planning across multiple maneuvers over longer horizons. Odyssey uses OSM-derived SD-map routes to provide this guidance in reconstructed real-world environments. These routes specify the intended road sequence throughout the rollout and reduce reliance on detailed HD maps as planner inputs. Local driving decisions remain based on sensor observations.

B Implementation Details of Closed-loop Simulator

B.1 3D Gaussian Reconstruction

Reconstruction Framework. We build on OmniRe (Chen et al., 2025) to jointly reconstruct the static background and dynamic actors, and model traffic lights separately to reproduce signal changes. Pose noise in nuPlan logs degrades 3D Gaussian reconstruction quality, motivating pose refinement with COLMAP (Schönberger & Frahm, 2016). We keep the logged x and y coordinates fixed to preserve alignment with the HD map used for drivable-area evaluation, and optimize only z, roll, pitch, and yaw. To separate scene components, we generate sky and dynamic-object masks using SegFormer (Xie et al., 2021). We initialize background Gaussians by back-projecting MoGe-2 pseudo-depth (Wang et al., 2025) after excluding dynamic pixels. Rigid and deformable actors are initialized from accumulated LiDAR points, and pedestrians from SMPL templates (Loper et al., 2015). For traffic-light modeling, we obtain head masks using Mask2Former (Cheng et al., 2022) pretrained on Mapillary Vistas v2 (Neuhold et al., 2017).

Ground Modeling. Inspired by HUGSIM (Zhou et al., 2025), we represent the road surface with a separate set of thin Gaussian disks. We first accumulate LiDAR points using the refined ego poses and build a road height field on a 0.125 m grid. Ground Gaussians are placed on this height field within the HD-map road regions; heights of unobserved road cells are filled by harmonic interpolation from the observed cells. Each disk has a fixed thickness of 0.01 m and stays attached to the surface: its height and normal are periodically re-sampled from the height field at its current horizontal position. The remaining attributes are optimized, namely the horizontal position, the in-plane

Table 6: Gaussian densification, pruning, and optimization settings.

Parameter	Background	Rigid actors
Densification		
Start / end iteration	2,500 / 93,500	2,500 / 93,500
Interval (iterations)	500	500
Views per scoring step	120	120
Normalized error threshold	0.1	0.1
Required score	> 5	> 5
Clone gradient threshold	2×10^{-4}	2×10^{-4}
Split absolute-gradient threshold	1×10^{-4}	3×10^{-4}
Clone / split scale boundary	0.0562 m	0.09 m
Children per split	2	2
Pruning and opacity control		
Minimum views for pruning	5	2
Early opacity threshold	0.005	0.005
Early scale threshold	5.617 m	5.617 m
Opacity cap	0.8	0.8
Opacity reset interval	7,000	14,500
Opacity reset value	0.01	0.01
Post-pruning start / end	70,000 / 140,000	93,500 / 140,000
Post-pruning interval	4,500	4,500
Post-pruning opacity threshold	0.05	0.05
Post-pruning score threshold	0.9	0.9
Background update intervals (iterations)		
Training iterations	**Geometry / opacity**	**SH**
Up to 70,000	1	16
70,001–93,334	32	32
After 93,334	64	64

scale, the rotation about the surface normal, the opacity, and the spherical-harmonic coefficients. To keep the background from also representing the road, we remove background points within 0.3 m of the surface at initialization. During training, we also penalize the mean opacity of background Gaussians that lie between 0.3 m below and 1.0 m above the surface.

COVERAGE-AWARE SAMPLING. Sampling images uniformly over a long driving sequence concentrates training updates on places where the recording vehicle stops or moves slowly. To balance spatial coverage, we back-project the LiDAR depth of each training image, excluding dynamic objects, into 0.5 m voxels and count how many training views observe each voxel. The weight of an image is the mean inverse observation count of its visible voxels, normalized by the 90th percentile over all images and clipped to $[0.1, 1]$. Images that mainly see less-observed regions therefore receive higher weights, and frequently observed regions are revisited less often. Rather than drawing images at random, we build a sampling cycle in which each image appears a number of times proportional to its weight, so that every training view is used at least once per cycle. Images containing dynamic objects are given a higher minimum weight of 0.25 to preserve the temporal coverage of moving actors. Each reconstruction update uses a single image.

TRAFFIC-LIGHT MODELING. To capture signal changes, we represent each traffic-light head as a separate deformable Gaussian node. Using head masks predicted by Mask2Former (Cheng et al., 2022) and depth estimated by MoGe-2 (Wang et al., 2025), we back-project masked pixels from calibrated views to initialize Gaussian positions and colors. Each node has a fixed installation position and orientation, and its geometry is kept fixed up to a small tolerance of 5 cm that absorbs residual calibration error. Signal changes are therefore modeled through appearance: an MLP takes Gaussian positions, time, and a learned embedding for each traffic light, and predicts temporal residuals to the spherical-harmonic coefficients and opacity. After reconstruction, we manually verify the signal schedules against the rendered traffic-light states and revise them where necessary.

OPTIMIZATION. We jointly optimize each complete 100-second sequence as a single scene for 140k iterations, sampling one training image per iteration. We use eight cameras at 10 Hz with

Table 7: Reconstruction losses and weights.

Loss	Objective	Weight (λ)
	Appearance supervision	
$\mathcal{L}_{\text{rgb}}$	RGB L1 after color correction	0.8
$\mathcal{L}_{\text{ssim}}$	SSIM before color correction	0.2
$\mathcal{L}_{\text{color}}$	Out-of-range color penalty	0.8
$\mathcal{L}_{\text{res}}$	Frame-dependent residual magnitude	0.01
$\mathcal{L}_{\text{temp}}$	Temporal variation of color residuals	0.1
	Geometry and opacity supervision	
$\mathcal{L}_{\text{sky}}$	Sky-opacity BCE	0.05
$\mathcal{L}_{\text{depth}}$	LiDAR inverse-depth supervision	0.1
$\mathcal{L}_{\text{ent}}$	Opacity entropy	0.05
$\mathcal{L}_{\text{smooth}}$	Edge-aware inverse-depth smoothness	0.001
$\mathcal{L}_{\text{los}}$	Line-of-sight opacity	0.1
$\mathcal{L}_{\text{box}}$	Actor bounding-box regularization	1.0
$\mathcal{L}_{\text{TL}}$	Traffic-light Gaussian displacement	1.0

1920×1080 images, starting at half resolution and switching to full resolution at iteration 1,166. To balance training across the sequence, we assign higher sampling weights to views covering less frequently observed 0.5 m spatial voxels, while every training view is still served at least once per sampling cycle. FastGS, GPU-resident caching of images and supervision signals, and a reduced update cadence for the background parameters of Gaussians reduce training overhead.

We apply FastGS density control (Ren et al., 2026) to background and rigid-actor Gaussians, computing scores from 120 views sampled across cameras and the full sequence. To account for uneven visibility over long sequences, we normalize densification scores by each Gaussian's visible-view count. Table 6 lists the settings.

TRAINING OBJECTIVES. We jointly optimize appearance and geometry using a weighted sum of the losses. Our appearance objective combines RGB L1 and SSIM losses with color-correction regularization. RGB L1 compares color-corrected renderings with recorded images, and SSIM supervises the uncorrected renderings to preserve image structure (Lin et al., 2024). The regularization penalizes colors outside the valid range, limits frame-dependent corrections, and discourages abrupt changes between adjacent frames. To account for exposure and white-balance differences, we adjust rendered RGB values using a learnable 3×4 affine transformation following MTGS (Li et al., 2025a). The first three columns transform the RGB channels, and the last column adds a per-channel offset. A shared matrix captures persistent differences between cameras, and a residual matrix for each camera and frame captures temporal variations.

Our geometry losses regularize scene structure and reduce reconstruction artifacts. LiDAR depth provides additional geometric supervision, and edge-aware smoothness reduces depth noise without smoothing across image boundaries (Chen et al., 2026). Sky-opacity BCE encourages transparent sky regions, and a line-of-sight loss suppresses opacity at least $0.8\,\text{m}$ in front of LiDAR returns (Hess et al., 2025). Opacity entropy further encourages values near zero or one. For actors, we apply bounding-box regularization to both rigid and deformable Gaussians, extending OmniRe's regularization to rigid actors. We separately penalize local displacement of traffic-light Gaussians to limit deformation around their fixed installations. Table 7 summarizes the loss terms and their weights.

B.2 FIXER FINE-TUNING

We refine rendered images using Fixer (NVIDIA, 2025b), a single-step diffusion model built on the 0.6B Cosmos-Predict2 backbone (NVIDIA, 2025a). The autoencoder encodes the degraded rendering directly into latent features without adding random noise. The degraded rendering therefore serves as the input to be restored, with reconstruction artifacts providing the corruption that the model learns to remove. A diffusion transformer refines the latent features in one step, and the decoder reconstructs the output image. To adapt the pretrained model to Odyssey, we construct

Table 8: Fixer inference acceleration.

Method	Time (ms/image) ↓	Speedup ↑
Baseline	25.0	1.0×
Optimized	**6.7**	**3.7**×

rendering–real image pairs from all 100 benchmark scenes using three degradation strategies following Difix3D+ (Wu et al., 2025).

- **UNDER-TRAINED RECONSTRUCTION (30%).** We use renderings from the $80\,\mathrm{k}$, $120\,\mathrm{k}$, and final $140\,\mathrm{k}$ checkpoints to cover different reconstruction quality levels. Final-checkpoint renderings discourage unnecessary corrections to high-quality inputs.
- **CYCLE RECONSTRUCTION (40%).** We render a trained Gaussian scene from a trajectory shifted laterally by $2\,\mathrm{m}$, train a second Gaussian model on these renderings, and render the original trajectory from the second model. These images are paired with the original real images to train refinement of novel-view artifacts.
- **CROSS-REFERENCE RECONSTRUCTION (30%).** We train a Gaussian model using six of the eight cameras, holding out `CAM_L2` and `CAM_R2`. Renderings from the two held-out cameras are paired with their real images. These viewpoints receive only indirect supervision from neighboring cameras, producing novel-view artifacts.

We randomly sample 20% of the frames from the $10\,\mathrm{Hz}$ sequences for training-pair generation. Under-trained, cycle, and cross-reference reconstruction account for 30%, 40%, and 30% of the training pairs, respectively. Pairs are generated at 576×1024 resolution without image-space augmentation.

TRAINING OBJECTIVES. Using these pairs, we train Fixer with pixel, perceptual, and texture supervision:

$$\mathcal{L}_{\mathrm{Fixer}} = \lambda_{\mathrm{L2}}\mathcal{L}_{\mathrm{L2}} + \lambda_{\mathrm{LPIPS}}\mathcal{L}_{\mathrm{LPIPS}} + \lambda_{\mathrm{Gram}}\mathcal{L}_{\mathrm{Gram}}. \tag{1}$$

Full-image L2 loss measures pixel error, LPIPS (Zhang et al., 2018) preserves perceptual content, and Gram loss (Gatys et al., 2015) matches texture statistics. LPIPS and Gram losses use VGG-16 features from spatially aligned crops of the refined and real images. LPIPS uses square crops with side lengths of 128–512 pixels, and Gram loss uses 512×512 crops.

We set $\lambda_{\mathrm{L2}} = \lambda_{\mathrm{LPIPS}} = 1$ throughout training. Fixer's public training recipe disables Gram loss (NVIDIA, 2025b), but we found that activating it during fine-tuning substantially reduced blur and improved texture detail. To avoid excessive sharpening, we keep $\lambda_{\mathrm{Gram}} = 0$ until the final $1{,}000$ steps, then set it to 0.02.

INFERENCE OPTIMIZATION. To improve inference speed, we remove cross-attention layers inactive under zero text conditioning and employ TensorRT to accelerate Fixer's encoder, diffusion transformer, and decoder. These optimizations achieve a $3.7\times$ speedup with negligible changes in restoration quality (Table 8).

B.3 SECTION-BASED TRAFFIC TRIGGERING

We provide implementation details for the triggering scheme introduced in Section 3.4. At each simulation step, we project the ego position onto the logged trajectory and retain the maximum arc-length progress reached so far. The first section is active at initialization; subsequent sections activate when this progress reaches 15 m before their start. Arrival at this activation point defines the entry time used to compute the section offset. In the reactive setting, vehicles are initialized to their first logged states and retained while within 50 m of the ego. Parked vehicles and static objects are excluded from triggering.

We assign traffic signals to sections by projecting their installation positions onto the logged ego trajectory, including signals for crossing and opposing traffic. Planner inputs, IDM control, signal rendering, and red-light violation evaluation use the same section offset. Before section activation or outside the recorded time range, signal states are treated as unavailable and excluded from planner

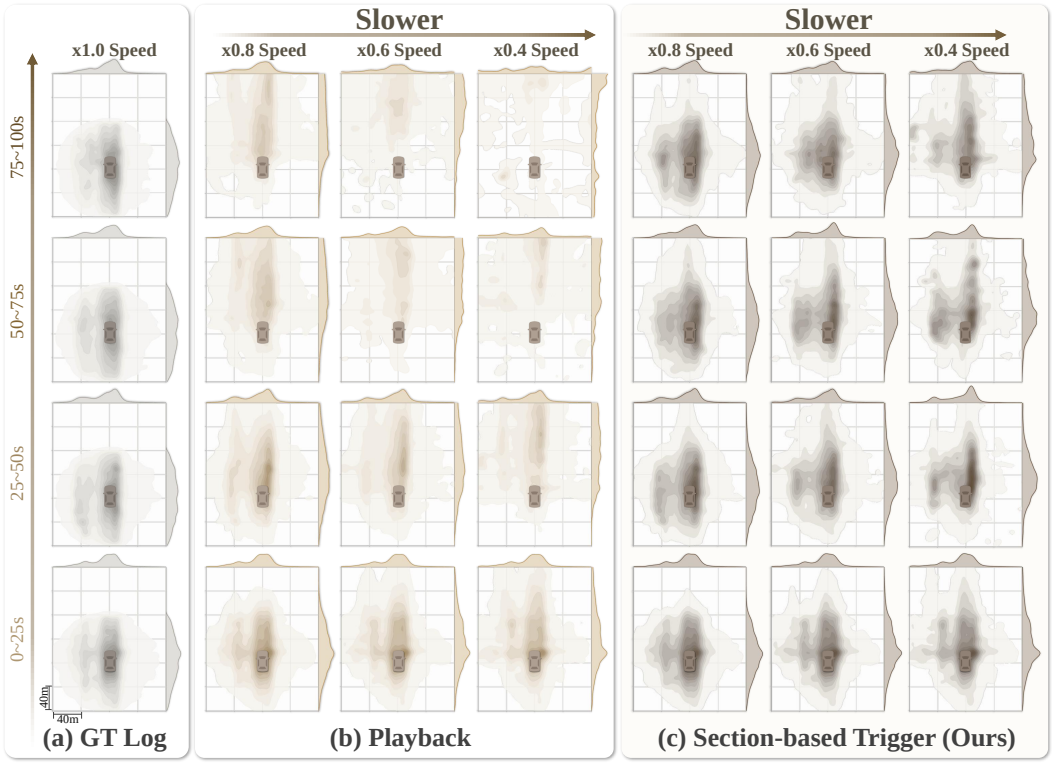


Figure 9: **Effect of Section-based Triggering on nearby agent density.** As ego speed decreases and simulation progresses, original-timeline playback (b) becomes increasingly sparse around the ego. Section-based Triggering (c) maintains density closer to the GT log (a) across the evaluated speeds and time intervals.

and IDM signal inputs and red-light violation evaluation. We do not extend the final recorded signal state, avoiding indefinite stops caused by a final red light.

Figure 9 shows how nearby agent density varies with ego speed and simulation time. Compared with GT replay (a), slowing the ego while retaining the original traffic timeline progressively reduces nearby agent density (b). Section-based triggering (c) maintains density close to GT replay across ego speeds and over time, demonstrating robustness to changes in ego speed.

Actors with limited observations due to distance from the logged ego or occlusion may be represented by too few Gaussians, resulting in poor rendering quality and insufficient visual cues for the planner. Such actors may nevertheless incur evaluation penalties. To mitigate this issue, experts carefully inspected actors individually and manually removed those with inadequate rendering quality.

C Evaluation Metric Details

This section defines the rollout termination criteria, pre-lane change metrics, RouteDS components, and additional driving metrics.

C.1 Rollout Termination Criteria

Each rollout evaluates progress toward a designated destination. Termination criteria bound its duration and prevent continued simulation far from the intended route. A rollout ends under one of three conditions:

- **Destination arrival.** The ego reaches the end of the designated SD route.

- **TIME LIMIT.** The rollout is limited to $200\,\mathrm{s}$, twice the duration of the recorded ground-truth driving sequence.
- **SD-ROUTE DEVIATION.** The rollout terminates as soon as the ego vehicle is more than $30\,\mathrm{m}$ from the designated SD route.

C.2 PRE-LANE CHANGE METRICS

We select evaluation intersections where the logged vehicle travels in route-compatible lanes throughout the $10\,\mathrm{m}$ before the stop line. To detect when vehicles approaching through oncoming lanes or off-road areas pass an evaluation intersection, we extend each stop line by $25\,\mathrm{m}$ at both ends and check for crossings. Crossings along these extensions, outside the original stop-line segment, are treated as failures to cross the stop line correctly.

C.3 ROUTEDS COMPONENTS

For each scenario, we compute

$$RouteDS = 100 \cdot RC_{SD} \cdot P_{\mathrm{PLC}} \cdot P_{\mathrm{SD}} \cdot P_{\mathrm{col}} \cdot P_{\mathrm{off}} \cdot P_{\mathrm{TL}}, \tag{2}$$

where $RC_{SD} \in [0, 1]$ denotes route completion and the remaining factors are penalties in $[0, 1]$, defined below. We report the mean ***RouteDS*** across scenarios.

- **ROUTE COMPLETION.** RC_{SD} is the fraction of the designated SD route completed by the ego vehicle. After each rollout, we apply HMM map matching (Newson & Krumm, 2009) to the executed trajectory to identify the SD-route segments traversed by the ego. If the ego deviates from the route, we determine where the ego left the route by repeating HMM matching from the start, extending the evaluated trajectory by $1\,\mathrm{m}$ each time. We count progress only up to the last point before the matching indicates that the ego is off route.
- **PRE-LANE CHANGE.** For each pre-lane change evaluation, we multiply P_{PLC} by 0.7 for a score of 0 and by 0.9 for a score of 0.5. Unencountered evaluation stop lines incur no pre-lane change penalty.
- **SD-ROUTE COMPLIANCE.** We set $P_{\mathrm{SD}} = 0$ if the rollout terminates because the ego exceeds the route-distance threshold defined above, or if post-rollout HMM matching identifies travel outside the designated route. If neither condition holds and matching succeeds, $P_{\mathrm{SD}} = 1$. Near the start and end of a rollout, limited trajectory coverage around a junction can make connected roads difficult to distinguish. We allow an $8\,\mathrm{m}$ tolerance for such boundary matches, excluding travel opposite to the route direction.
- **COLLISION PENALTY.** We penalize at-fault collisions with pedestrians, vehicles, and bicycles using nuPlan-style criteria (Caesar et al., 2021), excluding stationary-ego and active rear collisions. Front collisions and collisions with stationary agents count, with lateral collisions counted only when the ego spans multiple lanes or extends outside drivable areas. For each counted collision, we multiply P_{col} by 0.5 for pedestrians and 0.6 for vehicles or bicycles (Jia et al., 2024). If no collision is counted, $P_{\mathrm{col}} = 1$. The same object is counted again only after contact breaks and either displacement exceeds $5\,\mathrm{m}$ or elapsed time exceeds $5\,\mathrm{s}$.
- **OFF-ROAD PENALTY.** We compute $P_{\mathrm{off}} = 1 - D_{\mathrm{off}}/D_{\mathrm{total}}$, where D_{total} is the total traveled distance and D_{off} is the distance driven outside drivable areas or against the direction of traffic. Driving-direction violations are not counted within intersections.
- **TRAFFIC-LIGHT PENALTY.** Each red-light violation multiplies P_{TL} by 0.7. A violation occurs when the ego occupies an intersection lane whose traffic signal is red. Each violation is counted once, even if it persists across multiple evaluation steps.

C.4 ADDITIONAL DRIVING METRICS

We adapt Efficiency and Comfortness metrics from Bench2Drive (Jia et al., 2024) to our simulation setting, as detailed below.

EFFICIENCY. Efficiency measures whether the ego maintains driving speed relative to other traffic. We compare its speed with vehicles that exceed $1\,\mathrm{m/s}$ at least once during the rollout. At each evaluation step containing these vehicles, we compute their average speed. Efficiency is 100 times the sum of ego speeds divided by the sum of these average vehicle speeds over the same steps. A score of 100 indicates equal average speeds.

COMFORTNESS. Comfortness evaluates whether the ego drives smoothly without abrupt acceleration, braking, or steering. It is one if all limits in the table are satisfied throughout the trajectory, and zero otherwise. Except for longitudinal acceleration, the limits apply to absolute values.

Comfort Criterion	Limit
Longitudinal acceleration	$[-4.05,\ 2.40]\,\mathrm{m/s^2}$
Lateral acceleration	$4.89\,\mathrm{m/s^2}$
Yaw rate	$0.95\,\mathrm{rad/s}$
Yaw acceleration	$1.93\,\mathrm{rad/s^2}$
Longitudinal jerk	$4.13\,\mathrm{m/s^3}$
Derivative of acceleration magnitude	$8.37\,\mathrm{m/s^3}$

D SD-ROUTE CONDITIONING OF BASELINE PLANNERS

This section details the model-specific adaptations of the baselines in Section 4.2. We keep each original architecture and training recipe, and add only the route encoder and the modules that read it. Route point positions are expressed relative to the first route point, and headings are encoded by their cosine and sine. The added cross-attention has its own normalization and residual connection, and its output projections are initialized to zero, so each adapted planner starts training with the same output as its original architecture. A learned token that is always visible to the attention represents the absence of a route when all route points are masked.

- **TransFuser.** We use the camera-only latent variant. The trajectory query attends to the route in all three decoder layers, while the 30 agent queries do not. The ego status keeps only velocity and acceleration. Trained for 100 epochs.
- **DiffusionDrive.** The 20 anchor queries attend to the route in both decoder layers after the ego-query cross-attention. The route therefore conditions each of the two denoising steps. We use the camera-only latent variant. Trained for 100 epochs.
- **DrivoR.** The 64 proposal tokens attend to the route in all four refinement blocks. The ego history encoding is reduced from 11 to 7 channels (pose, velocity, and acceleration). Trained for 25 epochs.
- **SafeDrive.** Every layer of both planning decoders attends to the same route tokens from a shared encoder. The ego status is reduced from 8 to 4 channels. We use the camera-only variant with pedestrian detection.
- **ReCogDrive.** A two-stage point encoder with max pooling compresses the 120 route points into one token. A Fourier embedding of the route start is added to this token, and the result is added to the ego-status embedding of the diffusion planner. The navigation line of the VLM prompt is generated from the same route, naming the next maneuver with its distance and the one after it (e.g., "turn left in 30 m, then follow the road"). When the remaining route is too short to describe, the driving command is used instead.

E ADDITIONAL ANALYSIS

Table 9: The five nuPlan logs used for the comparison. Each scene is a 20 s window sampled at 2 Hz; frame indices are 2 Hz keyframe indices within each log.

	nuPlan log	City	Frames
S1	2021.09.16.19.27.01_veh-45_01749_03230	Pittsburgh	1539–1578
S2	2021.09.29.19.02.14_veh-28_00273_00514	Boston	119–158
S3	2021.09.09.19.10.24_veh-39_01125_01324	Boston	120–159
S4	2021.06.28.21.16.05_veh-14_00957_01198	Las Vegas	270–309
S5	2021.06.28.20.24.43_veh-38_03385_04952	Las Vegas	1561–1600

Table 10: Quantitative comparison with reconstruction baselines. All baselines are retrained under their official recipes on identical data, and all metrics are computed with a single implementation.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
OmniRe (Chen et al., 2025)	19.4	0.654	0.456
MTGS (Li et al., 2025a)	19.6	0.692	0.419
HUGSIM (Zhou et al., 2025)	20.1	0.701	0.436
Ours	**21.7**	**0.707**	**0.388**

E.1 Comparison with SOTA 3DGS methods

We compare reconstruction quality with OmniRe (Chen et al., 2025), MTGS (Li et al., 2025a), and HUGSIM (Zhou et al., 2025) using PSNR, SSIM, and LPIPS. We evaluate on five nuPlan (Caesar et al., 2021) scenes from different logs across three cities, listed in Table 9. Each scene comprises a 20 s window sampled at 2 Hz, using all eight surround-view cameras at 1920×1080 resolution. Images are undistorted using the original intrinsic matrices as the output camera intrinsics. We hold out every fourth keyframe across all eight cameras to evaluate novel-view synthesis at held-out camera poses. We train each baseline from scratch using its official training recipe and iteration count. All methods use the same camera poses, calibration, undistorted images, train/test split, and image resolution. As shown in Table 10, our method achieves the highest PSNR and SSIM and the lowest LPIPS among the compared methods, indicating improvements in both image fidelity and perceptual similarity.

E.2 Additional Qualitative Analysis

In addition to the qualitative results below, we provide a **supplementary video** showing long-horizon closed-loop rollouts, comparisons between directional commands and explicit SD route guidance, and planner behavior across diverse driving scenarios.

Novel-view refinement. Figure 10 shows novel views shifted by 3–5 m from the recorded camera positions, where limited observational coverage leads to pronounced reconstruction artifacts. Finetuned Fixer reduces blur, distortions, ghosting, and color and brightness inconsistencies across diverse scene elements, including lane markings, vehicles, pedestrians, and background structures.

Scenario diversity. Figure 11 overlays recorded ego trajectories and SD routes for scenarios involving lane changes before turns, multiple turns, and long straight routes. These examples span different levels of route complexity. In particular, lane changes before turns illustrate how navigation intent shapes driving decisions before the maneuver itself, motivating evaluation over extended sequences.

Traffic-light modeling. Figure 12 shows additional examples of reconstructed traffic-light states alongside their locations and signal schedules. The enlarged views highlight red and green states reproduced by the model, illustrating its ability to represent signal changes within the same reconstructed scene.

E.3 Detailed RouteDS Analysis

Table 11 reports route completion and individual penalty factors alongside *RouteDS*. SD-route guidance improves route compliance and *RouteDS* across planners, but improvements in individual penalty factors are not consistent. These results distinguish gains in route following from changes in other driving criteria.

Table 11: Detailed *RouteDS* results under non-reactive and reactive traffic.

Method	SD Route guidance	Non-reactive							Reactive						
		RouteDS	**RC**	$\mathbf{P}_{\mathrm{SD}}$	$\mathbf{P}_{\mathrm{col}}$	$\mathbf{P}_{\mathrm{off}}$	$\mathbf{P}_{\mathrm{TL}}$	$\mathbf{P}_{\mathrm{PLC}}$	***RouteDS***	**RC**	$\mathbf{P}_{\mathrm{SD}}$	$\mathbf{P}_{\mathrm{col}}$	$\mathbf{P}_{\mathrm{off}}$	$\mathbf{P}_{\mathrm{TL}}$	$\mathbf{P}_{\mathrm{PLC}}$
Vision-based planners															
LTF	×	16.24	0.72	0.51	0.54	0.67	0.98	0.88	16.27	0.75	0.54	0.59	0.67	0.98	0.86
	✓	20.13	**0.99**	0.95	0.43	0.63	0.99	0.81	23.01	**0.99**	0.93	0.48	0.63	0.98	0.80
DiffusionDrive	×	25.64	0.75	0.59	0.61	0.71	**0.99**	**0.92**	24.17	0.73	0.59	0.61	0.71	0.98	0.91
	✓	35.93	0.96	**1.00**	0.51	0.76	**0.99**	0.91	40.97	0.98	**1.00**	0.57	0.76	0.99	0.91
DrivoR	×	20.61	0.68	0.50	0.69	0.86	**0.99**	0.90	25.46	0.71	0.53	0.73	0.86	0.98	0.90
	✓	36.45	0.84	0.90	0.65	0.86	0.99	0.88	41.35	0.83	0.89	0.74	0.88	0.98	0.88
SafeDrive	×	27.52	0.61	0.41	**0.86**	**0.89**	0.99	**0.92**	30.42	0.64	0.45	**0.88**	0.89	0.98	**0.92**
	✓	**44.35**	0.85	0.82	0.75	0.89	0.99	0.89	**48.94**	0.84	0.79	0.85	**0.89**	0.98	0.88
VLA-based planners															
ReCogDrive	×	15.42	0.57	0.71	0.59	0.79	**0.99**	0.91	20.33	0.53	0.62	0.70	0.80	0.99	0.91
	✓	21.45	0.54	0.83	0.69	0.83	**0.99**	0.90	26.25	0.56	0.84	0.77	0.83	**0.99**	0.91



Figure 10: **Qualitative Comparison of Fixer Refinement.** We show challenging novel views with limited coverage from training observations, where 3DGS-only renderings exhibit pronounced artifacts. Fine-tuned Fixer reduces these artifacts and improves visual detail.

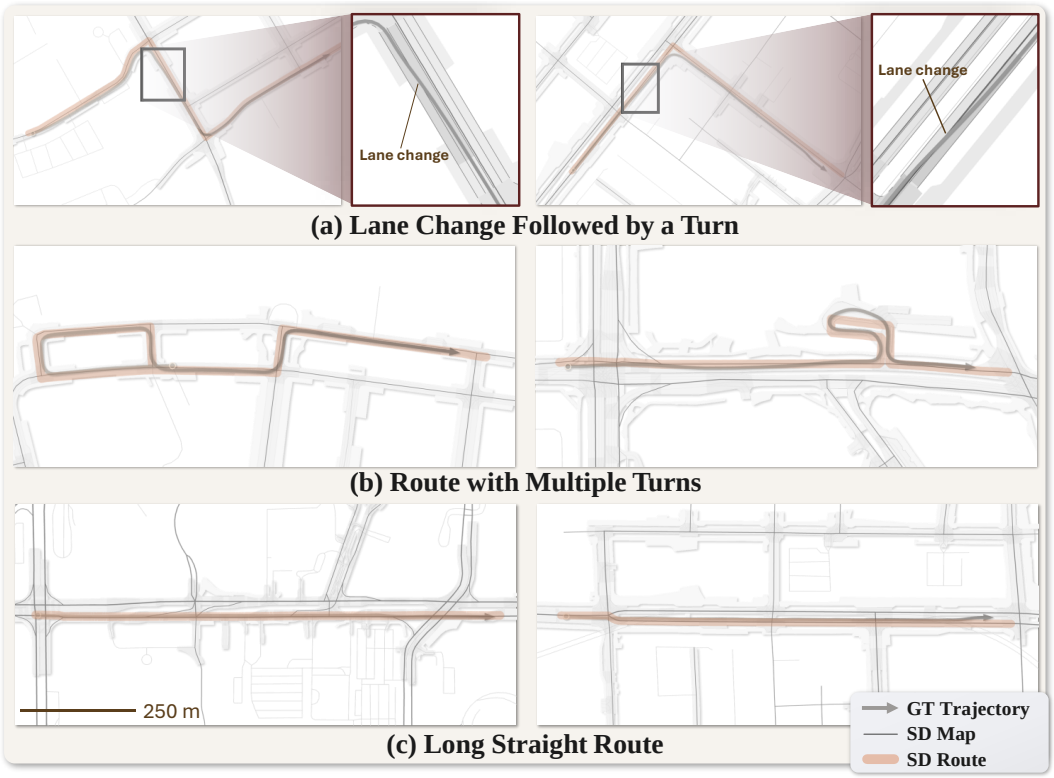


Figure 11: **Example trajectories and SD-Routes from Odyssey.** The benchmark includes lane changes in preparation for upcoming turns, routes with multiple turns, and long straight routes. These scenarios span varying route complexity and include anticipatory maneuvers that reflect the driver's navigation intent.

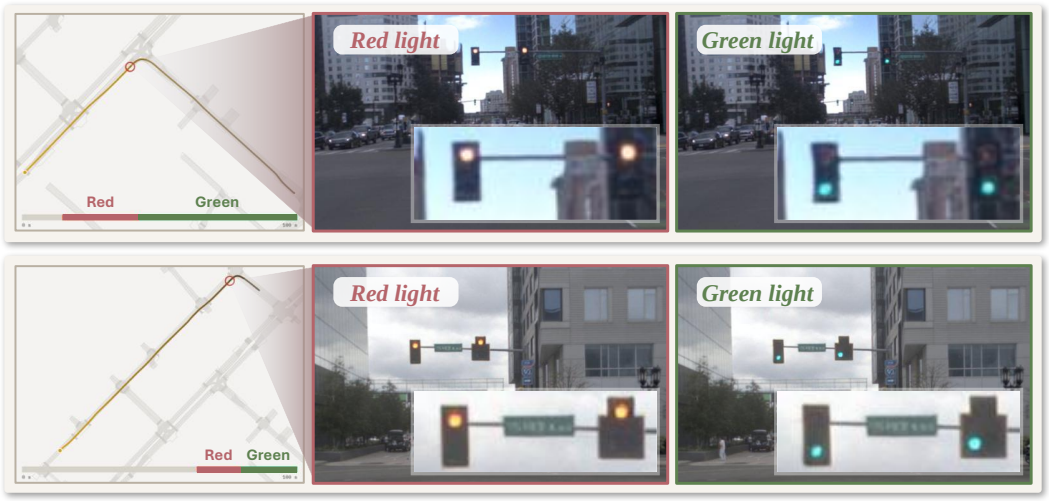


Figure 12: **Additional Qualitative Results of Traffic Light Modeling.** Our model reproduces red-to-green transitions in sync with the traffic-light schedule in each scene.