DriveE2E: An Infrastructure-Grounded Ego-Closed-Loop Replay Benchmark for End-to-End Autonomous Driving

Anonymous authors
Paper under double-blind review

Abstract

Closed-loop evaluation is important for end-to-end autonomous driving, but existing CARLA-based benchmarks often rely on manually designed scenarios whose traffic patterns may differ from real-world urban driving. We present DriveE2E, an infrastructure-grounded ego-closed-loop replay benchmark for evaluating end-to-end autonomous driving models in reconstructed real-world intersection scenarios. DriveE2E uses high-mounted infrastructure sensors to extract traffic trajectories from 100 hours of urban intersection data, constructs CARLA-compatible digital twins for 15 real intersections, and imports 800 curated traffic scenarios into simulation. In DriveE2E, the tested model controls the ego vehicle and receives simulation-generated observations from its current simulated state, while non-ego agents replay trajectories extracted from real-world traffic. This protocol does not model fully reactive multi-agent behavior; instead, it provides a reproducible intermediate regime between open-loop log replay and fully reactive simulation. We instantiate the benchmark with representative E2EAD baselines and analyze their open-loop and ego-closed-loop replay performance across behavior categories. The results suggest that DriveE2E can expose differences between open-loop trajectory accuracy and policy behavior under ego feedback in dense intersection scenarios. Code is included in the supplementary material, and will be publicly released upon acceptance.

1 Introduction

End-to-end autonomous driving (E2EAD) models aim to map sensor observations and ego states directly to planned trajectories or control actions, and have become a central paradigm for learning-based autonomous driving Hu et al. (2023b); Chen et al. (2024a); Jia et al. (2023); Shao et al. (2024). A persistent challenge for evaluating such models is that different evaluation protocols measure different aspects of driving behavior. Open-loop evaluation on recorded logs, as commonly used in nuScenes-style planning evaluation Caesar et al. (2020); Hu et al. (2023b); Jiang et al. (2023), is scalable and reproducible, but the tested model does not influence future observations or the subsequent ego state. As a result, open-loop trajectory metrics can miss failure modes caused by feedback between policy outputs and future ego states Zhai et al. (2023); Li et al. (2024). Fully reactive simulation, in contrast, allows the ego policy to interact with a simulated environment, but requires non-ego traffic behavior models whose realism and stability are difficult to guarantee, especially in dense urban intersections. CARLA-based benchmarks such as CARLA Leaderboard, Longest6, and Bench2Drive Carla Contributors (2024); Chitta et al. (2023); Jia et al. (2024) provide valuable reactive or scripted simulation environments, yet many scenarios are manually designed or procedurally specified rather than reconstructed from observed real-world traffic traces.

Recent non-reactive and pseudo-simulation benchmarks further highlight the need for intermediate evaluation regimes. NAVSIM v1 Dauner et al. (2024) evaluates planned trajectories with simulation-based metrics in a data-driven non-reactive simulator, providing a scalable alternative to displacement-only open-loop metrics. NAVSIM v2 Cao et al. (2025) extends this direction through pseudo-simulation, combining real observations with pre-rendered synthetic observations to approximate closed-loop effects efficiently. These benchmarks are closely related to DriveE2E in their motivation to move beyond open-loop trajectory error while preserving reproducibility. However, they evaluate policies from fixed scene states or pseudo-simulation states rather

Table 1: Positioning of DriveE2E among common E2EAD evaluation regimes. The table distinguishes evaluation protocols and their scope boundaries rather than ranking benchmark quality.

Evaluation regime	**Ego control**	**Non-ego agents**	**Protocol boundary**
Open-loop replay	No policy feedback	Logged trajectories	No ego-state or observation feedback from policy actions
Non-reactive / pseudo-simulation	Planned trajectory or pseudo-rollout	Logged or pseudo non-reactive agents	No online ego-observation feedback from executed actions
Fully reactive simulation	Policy-controlled	Scripted or learned reactive agents	Traffic behavior realism and stability depend on the agent model
DriveE2E	Policy-controlled	Real-trajectory log replay	Non-ego agents preserve logged trajectories rather than reacting to ego actions

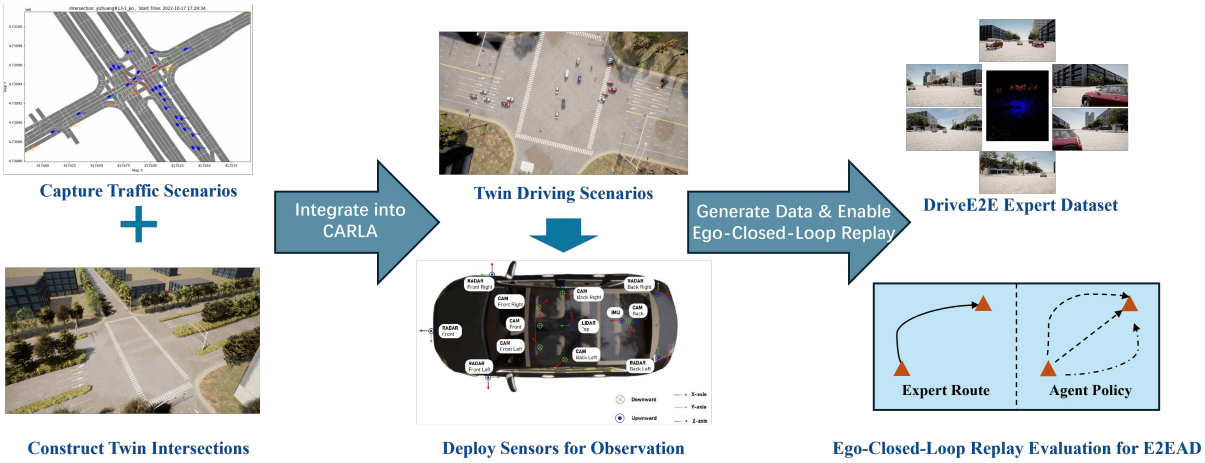


Figure 1: **Overview of DriveE2E.** Infrastructure videos are processed into estimated traffic trajectories and static digital-twin intersections. These reconstructed real-world intersection scenarios are imported into CARLA. During evaluation, the tested policy controls the ego vehicle and receives simulation-generated observations from its current simulated state, while non-ego agents follow real-world trajectories through log replay.

than providing a sequence of new observations generated from the ego vehicle's updated simulated state after executing previous actions. This leaves a complementary space for benchmarks that preserve real-world traffic traces while still allowing the ego policy to update its own state and subsequent observations in simulation.

We introduce DriveE2E, an infrastructure-grounded ego-closed-loop replay benchmark for E2EAD evaluation in reconstructed real-world intersection scenarios. DriveE2E extracts estimated traffic traces from infrastructure data, reconstructs CARLA-compatible digital-twin intersections, and evaluates ego policies in simulation. It occupies an intermediate regime between open-loop replay, NAVSIM-style non-reactive or pseudo-simulation benchmarks Dauner et al. (2024); Cao et al. (2025), and fully reactive simulation. Unlike open-loop replay, the tested model controls the ego vehicle and receives observations rendered from its updated simulated state. Unlike NAVSIM-style fixed-state or pseudo-simulation evaluation, DriveE2E provides a sequential ego-feedback observation loop in CARLA. Unlike fully reactive simulation, non-ego agents follow extracted real-world trajectories and do not react to the ego vehicle. This design trades non-ego reactivity for real-trace grounding, reproducibility, stable evaluation, and dense intersection reconstruction. Table 1 summarizes this evaluation positioning.

Infrastructure-view sensing is central to DriveE2E's construction. Because traffic is captured from elevated infrastructure viewpoints rather than an instrumented ego vehicle, DriveE2E can designate different visible vehicles as candidate ego vehicles and reconstruct dense surrounding traffic from multi-view observations. These estimated trajectories and static digital-twin intersections are imported into CARLA for expert-data collection and ego-closed-loop replay evaluation, as illustrated in Figure 1. This infrastructure-grounded construction is complementary to vehicle-view Real2Sim settings such as MetaDrive and ScenarioNet Li et al. (2022a; 2023), where scenarios are commonly derived from vehicle-view driving logs. The scope of DriveE2E is intentionally focused: it contains 800 curated scenarios selected from 100 hours of infrastructure video at 15 signalized intersections within one metropolitan deployment region. This concentration enables consistent calibration, trajectory extraction, and digital-twin construction, but limits claims about cross-city or cross-country generalization. We instantiate DriveE2E with representative E2EAD baselines to study how open-loop trajectory accuracy relates to ego-closed-loop replay behavior. Thus, the paper is positioned as a benchmark/resource and measurement study, not as a new algorithm or an exhaustive leaderboard.

Our contributions are summarized as follows.

- **Infrastructure-grounded Real2Sim construction.** We reconstruct CARLA-compatible digital-twin intersections and traffic scenarios from infrastructure-view urban traffic data, yielding 800 curated scenarios across 15 intersections within one deployment region.
- **Ego-closed-loop replay evaluation protocol.** We define an evaluation protocol in which the ego policy controls the vehicle and receives simulation-generated observations, while non-ego agents replay real trajectories. We explicitly position this protocol as an intermediate regime between open-loop replay, NAVSIM-style non-reactive or pseudo-simulation, and fully reactive simulation.
- **Benchmark evaluation with representative E2EAD baselines.** We instantiate DriveE2E with representative E2EAD baselines and analyze their open-loop and ego-closed-loop replay performance across behavior categories and reconstruction settings.

2 Related Work

End-to-End Autonomous Driving. End-to-end (E2E) autonomous driving approaches integrate perception, prediction, and planning into a unified model Hu et al. (2023b); Chen et al. (2024a); Chib & Singh (2024); Hao et al. (2026), often learning to map sensor observations and ego state to planned trajectories or low-level control Jia et al. (2023); Shao et al. (2024). Imitation-learning methods use expert demonstrations Codevilla et al. (2018); Prakash et al. (2021); Wu et al. (2022), while reinforcement-learning methods learn through interaction with an environment Liang et al. (2018); Kendall et al. (2019); Jia et al. (2023). Recent E2EAD systems include transformer-based models Prakash et al. (2021); Chitta et al. (2023); Shao et al. (2023a); Jaeger et al. (2023); Shao et al. (2023b), language-augmented models Pan et al. (2024); Chen et al. (2024b); Xu et al. (2024); Fu et al. (2024); Sima et al. (2024), and world-model-based approaches Zheng et al. (2024); Li et al. (2025); Wang et al. (2024). DriveE2E does not introduce a new E2EAD model; instead, it provides an evaluation setting for studying how representative models behave under ego feedback in reconstructed intersection scenarios.

Open-loop, Non-reactive, and Reactive Simulation Benchmarks. Open-loop datasets and log-based evaluations, such as nuScenes-style planning evaluation Caesar et al. (2020), are reproducible and scalable, and have been widely used to assess planning or trajectory prediction modules Hu et al. (2023b); Jiang et al. (2023). However, because the model does not influence future observations, open-loop metrics can miss failure modes caused by feedback between policy outputs and subsequent ego states Zhai et al. (2023); Li et al. (2024). NAVSIM v1 Dauner et al. (2024) targets an intermediate regime between open-loop log evaluation and fully interactive closed-loop simulation by evaluating planned trajectories with simulation-based metrics in a data-driven non-reactive simulator. NAVSIM v2 Cao et al. (2025) further develops this direction through pseudo-simulation, combining real observations with pre-rendered synthetic observations to approximate closed-loop effects in a scalable evaluation framework. These NAVSIM-style benchmarks are closely related to DriveE2E in their motivation to move beyond displacement-only open-loop metrics while

Table 2: Comparison with related benchmark settings. Unlike Table 1, which compares evaluation protocols, this table distinguishes benchmark construction choices rather than ranking benchmark quality. *Scene recon.* indicates whether real-world scene structure is reconstructed into a simulation-ready environment. *Ego-feedback obs.* indicates whether the tested policy receives new observations generated from the ego vehicle's updated simulated state after executing previous actions. DriveE2E emphasizes infrastructure-view-derived scenario construction, CARLA-compatible scene reconstruction, non-ego log replay, and ego-feedback observations.

Benchmark	Environment	Scenario Source	Non-Ego Agents	Scene Recon.	Ego-Feedback Obs.
Longest6 Chitta et al. (2023)	CARLA	Designed	Scripted/reactive	✗	✓
CARLA LB V2 Carla Contributors (2024)	CARLA	Designed	Scripted/reactive	✗	✓
Bench2Drive Jia et al. (2024)	CARLA	Designed	Scripted/reactive	✗	✓
SafeBench Xu et al. (2022)	CARLA	Designed/generated	Scripted/reactive	✗	✓
MetaDrive Li et al. (2022a)	MetaDrive	Vehicle-log derived	Replay/scripted/reactive	✓	✓
ScenarioNet Li et al. (2023)	MetaDrive	Vehicle-log derived	Replay/interactive	✓	✓
NAVSIM v1 Dauner et al. (2024)	BEV sim.	Vehicle-log derived	Non-reactive replay	✗	✗
NAVSIM v2 Cao et al. (2025)	Pseudo-sim.	Vehicle/synthetic-log derived	Pseudo/non-reactive	✗	✗
DriveE2E (Ours)	CARLA	Infrastructure-log derived	Non-ego log replay	✓	✓

preserving reproducibility. However, they evaluate policies from fixed scene states or pseudo-simulation states rather than providing a sequence of new observations generated from the ego vehicle's updated simulated state after executing previous actions. Reactive CARLA benchmarks, including CARLA Leaderboard, Longest6, and Bench2Drive Carla Contributors (2024); Chitta et al. (2023); Jia et al. (2024), evaluate policies in simulation with scripted or behavior-model-driven traffic. These benchmarks are valuable for testing reactive behavior and rule compliance, but their scenarios are typically designed or procedurally specified rather than reconstructed from observed real-world traffic traces. DriveE2E is complementary to both NAVSIM-style non-reactive benchmarks and reactive CARLA benchmarks: it runs the ego policy in CARLA so that executed ego actions update the ego state and subsequent rendered observations, while preserving non-ego log replay and infrastructure-view reconstructed scenes for reproducibility.

Real2Sim, Scenario Replay, and Infrastructure-view Data. Real2Sim and scenario-replay methods import real-world datasets into simulators to obtain repeatable evaluation scenarios. Works such as MetaDrive Li et al. (2022a) and ScenarioNet Li et al. (2023) can load dataset-derived scenarios into simulation, often using vehicle-view logs and lightweight rendering platforms such as Panda3D Goslin & Mine (2004). Generative reconstruction methods based on diffusion, GPT, NeRF, or 3D Gaussian Splatting Yang et al. (2025); Hu et al. (2023a); Tonderski et al. (2024); Cao et al. (2025) provide complementary tools for visual synthesis, but closed-loop evaluation still requires a protocol for agent motion, ego control, and metric computation. Infrastructure-view and V2X datasets provide a different source of information: elevated sensors can observe intersections more completely than a single vehicle-view platform Yu et al. (2022); Hao et al. (2025). DriveE2E uses this property to reconstruct dense intersection traffic traces and CARLA-compatible digital twins. Its scope is not fully reactive multi-agent simulation; rather, it occupies an intermediate regime between open-loop replay and reactive simulation by combining non-ego log replay with ego-closed-loop control.

3 DriveE2E

DriveE2E is an infrastructure-grounded Real2Sim benchmark for ego-closed-loop replay evaluation. As illustrated in Figure 2, the benchmark construction starts from multi-view infrastructure data, extracts estimated traffic trajectories, reconstructs static digital-twin intersections, assigns an ego vehicle and sensor suite, and imports the reconstructed scenarios into CARLA for expert-data collection and ego-closed-loop replay evaluation. The pipeline consists of four main components: dynamic traffic scenario acquisition from infrastructure sensors (Sec. 3.1); static intersection asset construction (Sec. 3.2); ego vehicle assignment and sensor configuration (Sec. 3.3); and the ego-closed-loop replay protocol in CARLA (Sec. 3.5). We also describe the expert dataset used for imitation learning (Sec. 3.4) and summarize the scenario statistics (Sec. 3.6).

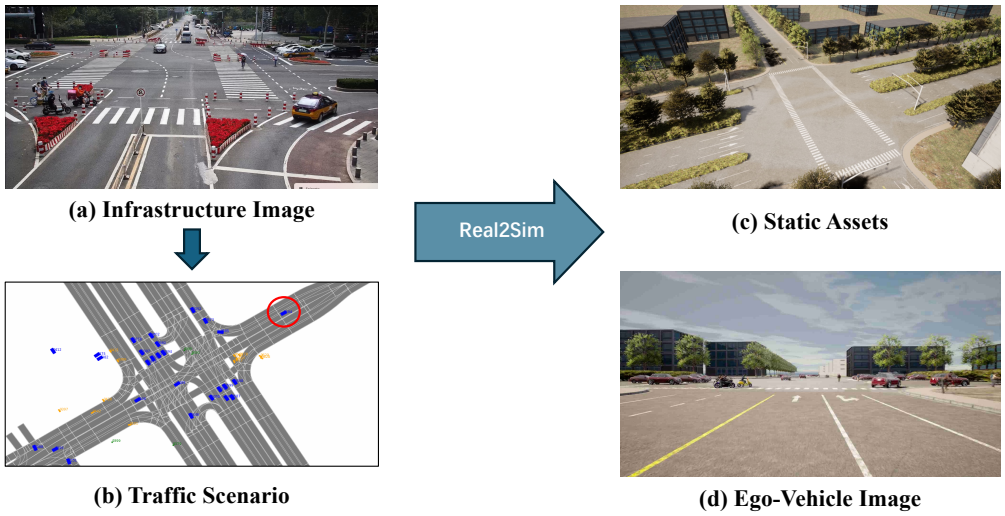
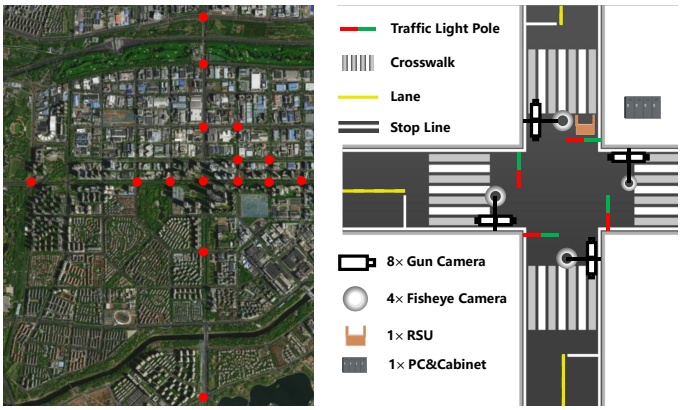


Figure 2: **Infrastructure-grounded Real2Sim construction in DriveE2E.** The figure summarizes the construction components in Secs. 3.1–3.3. (a) Multi-view infrastructure cameras observe the intersection. (b) Traffic participants are extracted as estimated trajectories, with the designated ego vehicle indicated. (c) Static digital-twin assets are reconstructed and aligned with the scene. (d) CARLA renders ego-vehicle observations from the assigned ego state. During ego-closed-loop replay evaluation, non-ego agents follow logged trajectories and do not react to the ego vehicle.

3.1 Dynamic Traffic Scenario Acquisition

Infrastructure sensor setup. Infrastructure-mounted sensors, installed at elevated positions, provide a broader view of signalized intersections than a single vehicle-mounted platform Yu et al. (2022); Hao et al. (2024). We use 15 intersections from one metropolitan deployment region. At each intersection, multiple roadside cameras and additional blind-spot cameras are calibrated to a shared coordinate system and configured to record traffic light state. The geographic concentration is intentional: it enables consistent calibration and digital-twin construction, but the benchmark should be interpreted as an intersection-centric urban driving benchmark rather than a cross-city generalization benchmark.

Data collection, annotation, and curation. We collect 100 hours of multi-view infrastructure video together with traffic-light state and weather/time metadata. The videos are processed with trained 3D object detection Rukhovich et al. (2022), multi-object tracking Weng et al. (2020), and multi-view fusion. The calibration assumption is that each camera has known intrinsic and extrinsic parameters with respect to the intersection coordinate frame, allowing detections from different views to be associated and fused into a common 3D trajectory representation. Each trajectory record contains a timestamp, track ID, object category, 3D box center, size, orientation, velocity when available, and visibility/quality indicators. Agent categories include car, truck, bus, van, motorcycle, tricycle, pedestrian, and cyclist.



(a) Location Distribution for Data Acquisition (b) Roadside Sensor Deployment Settings

Figure 3: Dynamic traffic scenario acquisition from infrastructure-view video.

Candidate clips are produced from continuous tracks and filtered with a deterministic curation procedure. Completeness measures whether the candidate ego and relevant nearby agents remain observable for the clip duration, have sufficiently long and continuous tracks, and avoid large temporal gaps. Traffic-rule compliance checks whether the candidate ego route is consistent with lane direction, signal state, drivable area, and basic intersection rules. We also remove physically implausible trajectories, unstable tracks, and clips whose start or destination cannot be mapped reliably into CARLA. The resulting process is: 100 hours of video → candidate trajectories → candidate clips → curated scenarios → 800 imported scenarios. Some steps, such as detection, tracking, fusion, and metadata extraction, are automatic; ego selection, scenario validation, map alignment, and a subset of asset corrections are manually reviewed. We do not claim perfect trajectory annotations. Instead, DriveE2E uses this deterministic curation procedure to remove incomplete, physically implausible, or traffic-rule-inconsistent clips before simulation import.

3.2 Static Intersection Asset Construction

We construct CARLA-compatible static assets for the 15 selected intersections. HD maps provide vectorized lane centerlines, crosswalks, stop lines, and drivable regions in a format similar to Argoverse Chang et al. (2019). The maps are imported into RoadRunner MathWorks (2023)[1], where road geometry is manually refined using map evidence and image references. Surrounding elements such as buildings are derived from OpenStreetMap OSM contributors (2023)[2] when available and then adjusted for spatial consistency. Traffic lights, poles, lane markings, crosswalks, signs, and adjacent structures are integrated in Blender Blender Studio (2023) and exported to CARLA-compatible assets. The goal is not photorealistic reconstruction, but a spatially consistent digital-twin environment that can render ego observations and support repeatable policy evaluation.

3.3 Ego Vehicle Assignment and Sensor Configuration

Because the original data are collected from infrastructure sensors rather than an instrumented vehicle, DriveE2E explicitly assigns an ego vehicle in each scenario. The ego is selected according to full-track visibility, route validity, sufficient route length, compatibility with CARLA spawning and destination points, behavior diversity, and representativeness within the candidate scenario set. The assigned ego is equipped in simulation with a standard sensor suite similar to nuScenes Caesar et al. (2020), summarized in Table 3. This procedure makes the evaluation target explicit, but it also means that the original ego route is a reference route rather than a guarantee that all policy deviations remain feasible under non-ego replay.

Table 3: Key sensor specifications for the ego vehicle.

Sensor	Details
1x LiDAR	64 channels, 85-meter range, 360° horizontal FOV, +10° to -30° vertical FOV
6x Camera	Surround coverage, RGB, 900x1600 resolution, JPEG compressed
5x Radar	100-meter range
1x IMU&GPS	Position, heading, speed, acceleration, and angular velocity

3.4 Reference Dataset Collection

For imitation-learning baselines, we collect reference demonstrations in the reconstructed CARLA scenarios. We use the term reference demonstrations to denote trajectories obtained by replaying the assigned ego vehicle along its original route. For each selected scene, the corresponding static intersection asset is loaded, non-ego traffic participants are instantiated from the curated trajectories, and the assigned ego vehicle follows its original route. The exported reference dataset is organized around synchronized sensor observations and annotation files. The full benchmark records include multi-view RGB images, LiDAR point clouds, radar data, ego-state and control annotations, route commands or target waypoints, weather metadata, sensor metadata, and 3D boxes mapped to CARLA actors.

[1]RoadRunner MathWorks (2023): a 3D environment editing tool for road and traffic-scene simulation assets.
[2]OpenStreetMap OSM contributors (2023): an open, user-contributed map database.

3.5 Ego-Closed-Loop Replay Protocol

DriveE2E uses an ego-closed-loop replay protocol. The ego vehicle is controlled by the tested policy and receives observations rendered from its current simulated state. Non-ego agents follow trajectories extracted from infrastructure logs and do not react to the ego vehicle. This design is intentional: it preserves real-world traffic traces and reproducibility, while not attempting to model fully reactive multi-agent behavior.

At the start of each scenario, the ego vehicle is spawned at a source location (x_{src}, y_{src}) and given a destination (x_{dst}, y_{dst}) on the reference route. At each simulation step, the E2EAD system receives sensor observations, ego state, GPS coordinates, and downsampled route waypoints. The policy outputs either low-level controls or future waypoints that are converted to CARLA control commands. The ego state then updates in CARLA, and the next observation is rendered from the resulting state. Meanwhile, non-ego agents advance according to their logged trajectories.

Evaluation metrics. We adopt Success Rate (SR), the fraction of routes completed without collision or major traffic violation, and Driving Score (DS), which combines route completion with infraction penalties following CARLA Leaderboard V2 Carla Contributors (2024) and Bench2Drive Jia et al. (2024). Detailed formulae are provided in Appendix A.

Protocol scope and design trade-offs. The closed-loop component in DriveE2E is the ego policy feedback: the ego's actions change its state and future observations. The replayed component is the non-ego traffic: other participants preserve their recorded trajectories. This makes the benchmark useful for studying policy behavior under ego feedback in real-trace-grounded dense intersections, but it cannot evaluate mutual negotiation, yielding by other agents, or reactive collision avoidance by surrounding traffic. If the ego vehicle deviates strongly from the recorded route, some collisions may be artifacts of non-reactive replay rather than failures that would necessarily occur with responsive human drivers. For this reason, SR and DS should be interpreted as metrics under the ego-closed-loop replay protocol, not as estimates of fully reactive real-world driving performance. Table 1 summarizes this distinction by comparing DriveE2E with open-loop replay, NAVSIM-style non-reactive or pseudo-simulation, and fully reactive simulation.

3.6 Scenario Statistics

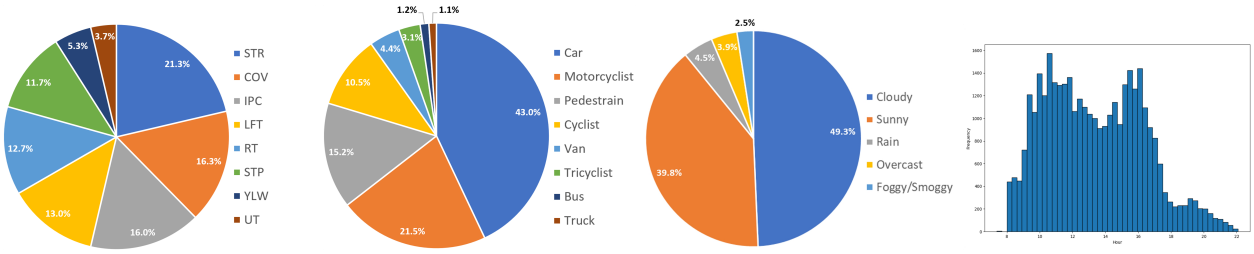
DriveE2E includes 800 curated scenarios at 15 signalized urban intersections from one metropolitan deployment region. The benchmark is focused on dense intersection-centric urban driving rather than broad geographic coverage.

Static twin intersection assets. The selected intersections contain different lane layouts, crosswalks, stop lines, traffic lights, poles, signs, and adjacent structures. Visualizations of the 15 assets are provided in the Appendix.

Driving behaviors. DriveE2E categorizes the curated scenarios into eight behavior groups: Interaction with Pedestrians and Cyclists (IPC), Competing with Other Vehicles (COV), Passing through during Yellow Lights (YLW), Making a U-turn (UT), Stopping at Red Lights (STP), Going Straight through Intersection (STR), Making a Left Turn (LFT), and Making a Right Turn (RT). The distribution is shown in Figure 4(a). These categories are intended for stratified analysis within the collected region, not as an exhaustive taxonomy of urban driving.

Traffic agents. As shown in Figure 4(b), DriveE2E contains eight traffic-agent categories. Infrastructure views provide broader observations for estimating traffic trajectories than a single vehicle-view sensor, especially around occlusions, but the estimated trajectories may still contain detection, tracking, and fusion errors.

Weather and lighting. The distributions of weather and time conditions are shown in Figure 4(c–d). These metadata allow CARLA scenes to approximate the collected conditions, while CARLA rendering remains a limited proxy for real-world perception.



(a) Behavior category (b) Agent category (c) Scenario weather (d) Scenario time

Figure 4: Scenario distributions within the collected deployment region.

4 Experiments

4.1 Baselines and Implementation Details

We instantiate DriveE2E with representative End-to-End Autonomous Driving (E2EAD) baselines to characterize the protocol and examine how open-loop trajectory accuracy relates to ego-closed-loop replay behavior. The baselines are not intended to form an exhaustive leaderboard. The 800 expert data clips are divided into training, validation, and test sets using a 2:1:1 split. Model training is conducted via imitation learning on the 400 training clips. Both open-loop and ego-closed-loop replay evaluations are performed on the validation split. Open-loop performance is reported as L2 Error (m) at 1s and 2s horizons. The evaluated models include UniAD Hu et al. (2023b), VAD Jiang et al. (2023), AD-MLP Zhai et al. (2023), TCP Wu et al. (2022), and MomAD Song et al. (2025). Additional baseline details are provided in Appendix.

4.2 Main Results

Open-loop evaluation results. As shown in Table 4, the evaluated baselines show different open-loop trajectory errors on the validation split. AD-MLP has the largest average L2 error among the evaluated methods, while VAD, UniAD, and MomAD obtain lower average L2 errors. VAD obtains the lowest average L2 error among the evaluated baselines in this split. These results suggest that open-loop trajectory error remains informative for this collection of dense intersection scenarios, but it should not be interpreted as a complete measure of policy behavior under ego feedback.

Ego-closed-loop replay results. In the ego-closed-loop replay setting, AD-MLP and TCP obtain lower SR/DS than the transformer-based baselines in this split. UniAD obtains the highest SR/DS among the evaluated baselines, with 47.00 SR and 77.62 DS. VAD obtains the lowest open-loop L2 error but lower SR/DS than UniAD, indicating that open-loop trajectory accuracy and ego-feedback behavior are related but not equivalent under this protocol. We also report the average evaluation time per scene for each model.

Table 4: Open-loop and ego-closed-loop replay evaluation results of different baseline models in DriveE2E. The average time cost for evaluating each model per scene in the ego-closed-loop replay setting is also reported.

Methods	L2 Error (m) ↓			Ego-Closed-Loop Replay		
	1s	2s	Avg.	SR (%) ↑	DS (%) ↑	Test Time (s/scene)
AD-MLP Zhai et al. (2023)	4.98	11.75	8.36	1.0	29.01	35.11
TCP Wu et al. (2022)	1.67	3.45	2.56	10.00	48.47	32.93
TCP-ctr Wu et al. (2022)	-	-	-	3.00	26.73	29.55
TCP-traj Wu et al. (2022)	-	-	-	25.50	61.52	31.28
VAD Jiang et al. (2023)	0.62	1.16	**0.89**	35.00	62.29	79.91
UniAD Hu et al. (2023b)	0.69	1.47	1.08	**47.00**	**77.62**	103.06
MomAD Song et al. (2025)	0.68	1.28	0.98	29.64	60.98	104.03

Relationship between open-loop and ego-closed-loop replay results. The results indicate partial agreement between open-loop and ego-closed-loop replay metrics. For example, AD-MLP has the largest L2 error and also low SR/DS, while VAD and UniAD perform relatively well under both views. At the same time, the ranking is not identical: VAD has the lowest L2 error, whereas UniAD obtains the highest SR/DS. This suggests that ego-feedback evaluation provides complementary information to open-loop trajectory error, even when non-ego agents are replayed rather than reactive.

4.3 Performance on Different Behavioral Scenarios

We also evaluate trained E2EAD models across the eight behavior categories in DriveE2E, with results presented in Table 5. Some categories, such as IPC and COV, obtain lower success rates than STP for several baselines. These categories contain denser surrounding traffic and more potential conflicts under non-ego log replay, whereas STP scenarios are often more constrained. The behavior-wise results are intended as stratified measurements within the collected split rather than claims about universal category difficulty.

Table 5: Ego-closed-loop replay evaluation by different behavioral scenarios.

Models	Success Rate (%) for Different Behavior Categories ↑							
	COV	IPC	UT	YLW	STR	LFT	RT	STP
AD-MLP Zhai et al. (2023)	0.00	0.00	0.00	5.88	0.00	0.00	0.00	4.55
TCP Wu et al. (2022)	16.67	2.94	40.00	5.88	2.78	3.85	12.50	22.73
TCP-ctr Wu et al. (2022)	5.56	2.94	0.00	0.00	0.00	7.69	0.00	4.55
TCP-traj Wu et al. (2022)	25.00	28.57	20.00	40.00	21.21	8.70	14.29	88.89
VAD Jiang et al. (2023)	38.89	32.35	20.00	23.53	36.11	46.15	41.67	22.73
UniAD Hu et al. (2023b)	**40.63**	**46.43**	**60.00**	**53.33**	39.39	**65.22**	**52.38**	**100.00**
MomAD Song et al. (2025)	19.44	23.53	20.00	47.06	**41.67**	42.31	20.83	18.18

4.4 Occlusion-Based Agent Filtering Analysis

To analyze the effect of infrastructure-view versus vehicle-view reconstruction in the ego-closed-loop replay protocol, we use the existing scenarios to study occlusion-induced missing information in vehicle-view variants.

Implementation. Infrastructure-view sensing can enable more complete reconstruction of intersection traffic than vehicle-view sensing, especially for occluded agents. To analyze reconstruction completeness, we construct filtered variants by removing traffic participants that do not appear in the ego vehicle's multi-view camera images within the expert data. We reuse the same assigned ego vehicle and expert route. One variant filters only vehicle agents while retaining pedestrians and cyclists; another filters all occluded agent types. We then reload the modified scenarios and re-evaluate the same baselines.

Analysis. As shown in Table 6, filtering occluded agents changes DS, but the effect is model-dependent. AD-MLP and VAD obtain higher DS when vehicle agents are removed (AD-MLP: 29.01→29.87; VAD: 62.29→64.17). When all occluded agents are filtered, AD-MLP, VAD, and UniAD obtain higher DS than in the complete reconstruction, while TCP obtains lower DS. These results suggest that removing occluded agents can simplify the evaluation for some models by reducing non-ego traffic complexity, while the effect varies across methods. Table 6 should therefore be interpreted as an analysis of reconstruction completeness, not as definitive proof that infrastructure-view reconstruction always changes rankings dramatically.

5 Scope and Limitations

DriveE2E is designed as an infrastructure-grounded ego-closed-loop replay benchmark, and its claims should be interpreted within that scope.

Table 6: Effect of occlusion-based agent filtering on ego-closed-loop replay Driving Score. 'Occ.' denotes occlusion.

Models	DS in Different Benchmarks ↑		
	Complete	Occ. Filtering (Vehicle)	Occ. Filtering (All)
AD-MLP Zhai et al. (2023)	29.01	29.87	29.57 (+0.56)
TCP Wu et al. (2022)	48.47	47.53	46.66 (-1.79)
VAD Jiang et al. (2023)	62.29	64.17	65.89 (+3.60)
UniAD Hu et al. (2023b)	77.62	76.80	78.49 (+0.87)

Protocol boundary: non-ego log replay. DriveE2E intentionally preserves non-ego agents through log replay: surrounding vehicles, pedestrians, and cyclists follow trajectories extracted from infrastructure logs rather than being replaced by learned or scripted reactive traffic models. This preserves the timing, spatial arrangement, and motion patterns of the collected traffic scenes, but it also means that DriveE2E should not be interpreted as a benchmark for fully reactive multi-agent negotiation.

Non-cooperative replay setting. Under this replay protocol, non-ego agents preserve their recorded trajectories regardless of the ego vehicle's behavior. This creates a non-cooperative evaluation setting: the ego policy cannot rely on surrounding agents to yield, brake, or otherwise compensate for ego deviations. As a result, the protocol places additional demand on the ego policy to maintain safe progress while staying compatible with the logged traffic flow. If the ego vehicle deviates strongly from the recorded route, some collisions may be unavoidable under replay. Such cases are meaningful for evaluating behavior under a real-trace-preserving, non-cooperative replay protocol, but they should not be interpreted as direct predictions of what would happen with responsive human drivers.

Geographic concentration. The 15 intersections are collected from one metropolitan deployment region. This concentration supports consistent calibration and digital-twin construction, but DriveE2E does not make claims about cross-city or cross-country generalization.

Dataset scale. The 800 scenarios form a focused benchmark of dense urban signalized intersection scenarios. They are representative of the collected region and curation criteria, but they are not intended to exhaustively cover urban driving.

Visual fidelity. DriveE2E uses CARLA rendering to generate ego observations. The benchmark should not be interpreted as replacing real-world perception evaluation, because CARLA does not fully reproduce real sensor noise, photorealism, or all appearance variation.

Annotation noise. Trajectories come from detection, tracking, and multi-view fusion. Although the curation pipeline removes incomplete, physically implausible, and traffic-rule-inconsistent clips, remaining errors in boxes, identities, velocities, or temporal alignment may affect evaluation.

6 Conclusion

DriveE2E provides a reproducible ego-closed-loop replay benchmark grounded in infrastructure-captured real traffic traces. It complements open-loop datasets and reactive CARLA benchmarks by enabling policy-controlled ego evaluation in reconstructed dense intersection scenarios while preserving non-ego real-trajectory replay. The current version intentionally uses non-ego log replay, and its conclusions should be interpreted within this scope. Our results with representative E2EAD baselines suggest that open-loop trajectory error and ego-feedback behavior provide related but non-identical views of model performance under the DriveE2E protocol.

References

Blender Studio. Blender, 2023. URL `https://www.blender.org/`.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *Conference on Robot Learning (CoRL)*, 2025.

Carla Contributors. Carla autonomous driving leaderboard, 2024. URL `https://leaderboard.carla.org/`.

Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8748–8757, 2019.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10164–10183, 2024a.

Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *IEEE International Conference on Robotics and Automation*, pp. 14093–14100, 2024b.

Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 9:103–118, 2024.

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*, 45(11):12878–12895, 2023.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *IEEE international conference on robotics and automation*, 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, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pp. 1–16, 2017.

Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 910–919, 2024.

Mike Goslin and Mark R Mine. The panda3d graphics engine. *Computer*, 37(10):112–114, 2004.

Ruiyang Hao, Siqi Fan, Yingru Dai, Zhenlin Zhang, Chenxi Li, Yuntian Wang, Haibao Yu, Wenxian Yang, Jirui Yuan, and Zaiqing Nie. Rcooper: A real-world large-scale dataset for roadside cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22347–22357, 2024.

Ruiyang Hao, Haibao Yu, Jiaru Zhong, Chuanye Wang, Jiahao Wang, Yiming Kan, Wenxian Yang, Siqi Fan, Huilin Yin, Jianing Qiu, et al. Research challenges and progress in the end-to-end v2x cooperative autonomous driving competition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 1828–1839, 2025.

Ruiyang Hao, Bowen Jing, Haibao Yu, and Zaiqing Nie. Styledrive: Towards driving-style aware benchmarking of end-to-end autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4627–4635, 2026.

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. *arXiv preprint arXiv:2309.17080*, 2023a.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17853–17862, 2023b.

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*, pp. 8240–8249, 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. In *Advances in Neural Information Processing Systems*, volume 37, pp. 819–844, 2024.

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.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *International Conference on Robotics and Automation*, pp. 8248–8254. IEEE, 2019.

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, 2022a.

Quanyi Li, Zhenghao Mark Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in neural information processing systems*, 36:3894–3920, 2023.

Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. In *International Conference on Learning Representations*, volume 2025, pp. 42942–42959, 2025.

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the European conference on computer vision*, pp. 1–18. Springer, 2022b.

Zhiqi Li, Zhiding Yu, Shiyi Lan, Jiahan Li, Jan Kautz, Tong Lu, and Jose M Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14864–14873, 2024.

Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European conference on computer vision*, pp. 584–599, 2018.

MathWorks. Roadrunner, 2023. URL https://www.mathworks.com/products/roadrunner.html.

OSM contributors. Openstreetmap: The free wiki world map, 2023. URL https://www.openstreetmap.org.

Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro G Allievi, Senem Velipasalar, and Liu Ren. Vlp: Vision language planning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14760–14769, 2024.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7077–7087, 2021.

Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2397–2406, 2022.

Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pp. 726–737, 2023a.

Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13723–13733, 2023b.

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

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *Proceedings of the European conference on computer vision*, pp. 256–274. Springer, 2024.

Ziying Song, Caiyan Jia, Lin Liu, Hongyu Pan, Yongchang Zhang, Junming Wang, Xingyu Zhang, Shaoqing Xu, Lei Yang, and Yadan Luo. Don't shake the wheel: Momentum-aware planning in end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22432–22441, 2025.

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

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *Proceedings of the European conference on computer vision*, pp. 55–72. Springer, 2024.

Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. Ab3dmot: A baseline for 3d multi-object tracking and new evaluation metrics. *arXiv preprint arXiv:2008.08063*, 2020.

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.

Chejian Xu, Wenhao Ding, Weijie Lyu, Zuxin Liu, Shuai Wang, Yihan He, Hanjiang Hu, Ding Zhao, and Bo Li. Safebench: A benchmarking platform for safety evaluation of autonomous vehicles. *Advances in Neural Information Processing Systems*, 35:25667–25682, 2022.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, pp. 1–8, 2024.

Xuemeng Yang, Licheng Wen, Tiantian Wei, Yukai Ma, Jianbiao Mei, Xin Li, Wenjie Lei, Daocheng Fu, Pinlong Cai, Min Dou, et al. Drivearena: A closed-loop generative simulation platform for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26933–26943, 2025.

Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21361–21370, 2022.

Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *Proceedings of the European conference on computer vision*, pp. 87–104. Springer, 2024.

A Evaluation Metrics

We adopt two metrics to evaluate the performance of the E2EAD system:

- **Success Rate (SR).** This metric measures the percentage of successfully completed routes within a certain time without collisions or traffic violations (e.g., leaving the drivable area).

- **Driving Score (DS).** This metric measures the driving performance while taking the route completion RC_i and infraction penalty of i-route into account as Eq. 1.

$$DS = \frac{1}{n_{total}} \sum_{i=1}^{n_{total}} RC_i \prod_{j=1}^{inf_i} (p_i^j), \tag{1}$$

  where n_{total} denotes the total number of routes, inf_i means a set of infraction that the ego vehicle triggered in ith-route, and p_i^j denotes the infraction penalty coefficient. For more details about infraction types and coefficients, refer to CARLA LB V2 Carla Contributors (2024).

B Artifact Availability and Reproducibility Statement

Review-time artifacts. Because the submission is under double-anonymous review, we provide the review artifacts as anonymized supplementary material rather than a public repository. The supplementary package contains three main components. First, `1.ExpertDatasetExample/` provides one anonymized reference-data example exported from the Real2Sim–CARLA pipeline. Due to supplementary material size constraints, this example is included only to illustrate the generated data format and file organization; it contains camera, LiDAR, radar, and annotation files, rather than the full released benchmark dataset. Second, `2.DynamicTrafficScenarios/` provides example dynamic traffic scenarios, including trajectory files and CARLA scenario XML files for the validation split. Third, `3.Code/` provides representative code for scenario loading, non-ego trajectory replay, ego-closed-loop CARLA evaluation, baseline interfaces, and metric computation. The top-level `README.md` describes the package scope, release constraints, and recommended inspection path.

The package is intended for review-time inspection of the data format and evaluation protocol, not as a complete public benchmark release.

Scenario and trajectory schema. Each processed scenario record contains anonymized scenario ID, split ID, intersection ID, behavior label, weather/time metadata, route source/destination, ego track ID, CARLA asset identifier, and a list of agent tracks. Each agent track contains timestamp, track ID, category, 3D box center, size, yaw, velocity when available, visibility flag, quality flag, and replay state. Metric files contain per-scenario route completion, infractions, SR, DS, collision indicators, behavior category, and evaluation time.

Real2Sim construction pseudocode. The procedure of DriveE2E's Real2Sim construction is summarized as follows:

1. Calibrate infrastructure cameras into a shared intersection coordinate frame.
2. Run 3D detection, multi-object tracking, and multi-view fusion on synchronized infrastructure video.
3. Convert fused tracks into trajectory records with category, 3D box, timestamp, velocity, and quality fields.
4. Segment candidate clips and score them for completeness, route validity, physical plausibility, and traffic-rule consistency.
5. Select an ego vehicle using full visibility, route validity, behavior coverage, and scenario representativeness.
6. Import the matched static intersection asset into CARLA and instantiate non-ego agents from replay trajectories.
7. Spawn the ego vehicle with the configured sensor suite and run either expert-data collection or ego-closed-loop replay evaluation.
8. Compute SR, DS, collision indicators, and behavior-category aggregates from the resulting logs.

C Baseline Models

We train and evaluate the following representative models in DriveE2E:

- UniAD Hu et al. (2023b) employs queries to integrate key tasks such as perception, mapping, prediction, and planning. The standard training process for UniAD typically involves three stages. To accelerate training and reduce GPU resource consumption, we bypassed the initial stages by directly training the stage-2 model using the BEVFormer Li et al. (2022b) model provided by Bench2Drive Jia et al. (2024) as a pre-trained model. We trained UniAD for one epoch. It is important to note that these settings may lead to a reduction in UniAD's accuracy.
- VAD Jiang et al. (2023) employs Transformer queries while enhancing efficiency through a vectorized scene representation. We trained the VAD model for two epochs, using a pre-trained model provided by Bench2Drive Jia et al. (2024) as a pretrain.
- AD-MLP Zhai et al. (2023) adopts a simple strategy by using the ego-vehicle past states into an MLP to generate future trajectory predictions.
- TCP Wu et al. (2022) predicts both trajectories and control signals. It only uses front-facing cameras and the ego state as inputs. Note that we did not train an expert model and did not use expert feature distillation during TCP training.
- MomAD Song et al. (2025) introduces trajectory momentum and perception momentum to stabilize and refine trajectory predictions, finally enhance the planning performance.

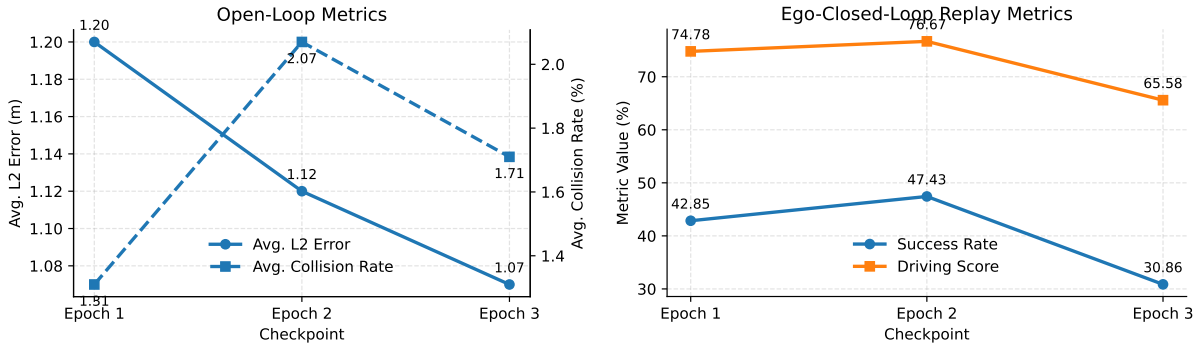


Figure 5: **UniAD checkpoint trends under open-loop and ego-closed-loop replay evaluation.** Left: open-loop metrics, including average L2 error and average collision rate. Right: ego-closed-loop replay metrics, including SR and DS. SR, DS, and collision rate are reported as percentages. The trends illustrate that lower open-loop L2 error in these checkpoints does not necessarily correspond to higher SR or DS under ego-feedback evaluation.

D Additional Results: Open-Loop vs. Ego-Closed-Loop Replay

In this section, we evaluate different UniAD checkpoints using both open-loop and ego-closed-loop replay metrics. Specifically, we save intermediate checkpoints during UniAD training and assess these checkpoints with two evaluation views. For open-loop evaluation, we report average L2 error and average collision rate. For ego-closed-loop replay evaluation, we report Success Rate (SR) and Driving Score (DS). The checkpoint trends are shown in Figure 5.

Figure 5 shows that the open-loop L2 error decreases from Epoch 1 to Epoch 3, while SR and DS peak at Epoch 2 and decrease at Epoch 3 under ego-closed-loop replay. This indicates that improved open-loop trajectory accuracy does not necessarily translate into better ego-feedback behavior under the replay protocol. The open-loop collision-rate trend is also not identical to the SR/DS trend, suggesting that the two evaluation views capture related but non-equivalent aspects of policy behavior.

E Static Intersection Construction Details

E.1 Construction Process

We provide more details to illustrate how to construct the static intersection assets in Figure 6. Here RoadRunner MathWorks (2023) is a 3D environment editing tool used for designing and editing road and traffic scenes for simulation and testing of autonomous driving systems. OpenStreetMap OSM contributors (2023) is a global, user-contributed, open-source map database. Blender Blender Studio (2023) is an open-source 3D creation suite for modeling, animation, and rendering.

E.2 Intersection Assets Visualization

DriveE2E presents 15 digital twins of urban intersections, each carefully designed to incorporate detailed roadside and road features, including traffic light poles, signage, lanes, crosswalks, stop lines, and surrounding buildings. These constructed twin intersections are presented in Figure 7.

F Simulation-to-Real (Sim2Real) Discussion

DriveE2E is designed as a reproducible ego-closed-loop replay benchmark in reconstructed CARLA scenes. It should not be interpreted as a substitute for real-world closed-loop testing or as evidence of direct simulation-

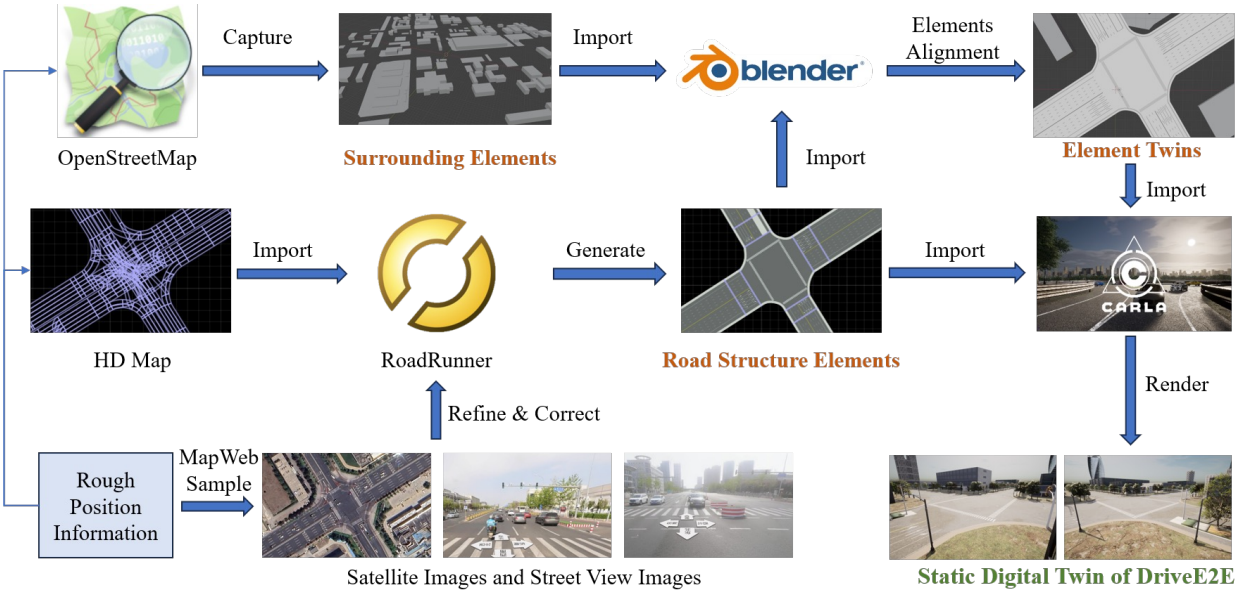


Figure 6: **Static Intersection Asset Construction Process**: We first obtain HD maps for each intersection and refine the road structures in RoadRunner. Surrounding elements are collected from OpenStreetMap. Finally, we integrate all components—including traffic light poles and signals—using Blender to create static intersection assets compatible with CARLA Dosovitskiy et al. (2017). These assets can be rendered directly in the CARLA simulator.

to-real transfer. This appendix documents the main Sim2Real gaps of DriveE2E and clarifies how the benchmark should be used.

F.1 Reconstruction Scope and Visual Fidelity

DriveE2E reconstructs both static intersection structure and dynamic traffic motion from infrastructure-view data. The static assets include road geometry, lane markings, crosswalks, stop lines, traffic lights, poles, signs, and surrounding structures. These elements are reconstructed from HD maps, image references, OpenStreetMap, RoadRunner, Blender, and CARLA-compatible asset libraries. The goal is to obtain a spatially consistent digital-twin environment for repeatable ego-policy evaluation, not a photorealistic replica of the collected intersections.

Dynamic agents follow estimated trajectories extracted from infrastructure videos. This preserves trace-level traffic motion under the non-ego log-replay protocol, but does not reproduce the full appearance, intent, or reactive behavior of real traffic participants. Traffic-light states and coarse weather/time metadata are imported when available, while environmental conditions are approximated using CARLA's built-in rendering and weather controls. Consequently, DriveE2E can support controlled ego-feedback evaluation in dense reconstructed intersections, but the rendered observations remain an approximation of real sensor observations.

F.2 Sensor and Perception Domain Gap

DriveE2E renders ego observations in CARLA using a configured ego sensor suite. Although this makes evaluation reproducible and allows the ego observation to change with the simulated ego state, the generated observations differ from real-world sensor data. Possible domain gaps include camera optics, lens distortion, image signal processing, compression, resolution, frame rate, LiDAR intensity patterns, radar characteristics, sensor noise, weather effects, lighting, material appearance, and object texture.

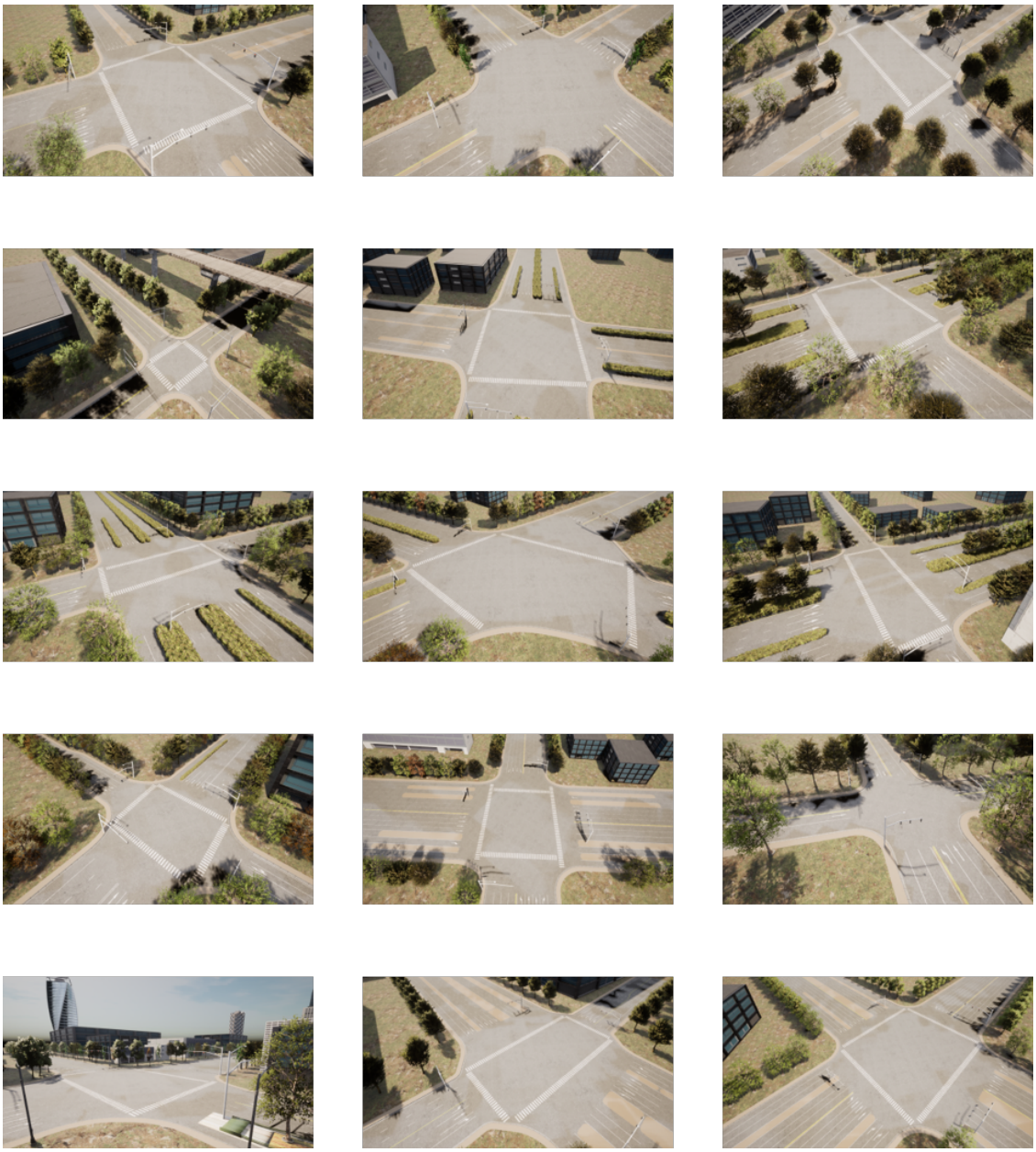


Figure 7: **Visualization of 15 Twin Intersection Assets.** These twins encompass intricate roadside elements, including traffic light poles and nearby buildings, along with diverse road features such as signage, lanes, crosswalks, and stop lines.

These gaps affect perception-heavy E2EAD models and should be considered when interpreting benchmark results. The reported performance in DriveE2E reflects behavior under the benchmark's fixed reconstruction, rendering, and sensor configuration. It should not be read as a claim that the same model performance transfers directly to arbitrary real vehicles, sensor suites, or cities. In this sense, DriveE2E is best understood as a focused evaluation proxy for policy behavior under ego-feedback observations in reconstructed dense intersections.

F.3 Rationale for Non-ego Log Replay

DriveE2E intentionally preserves non-ego agents through log replay rather than replacing them with learned or scripted reactive traffic models. The purpose is to retain the original traffic context captured by infrastructure sensors: surrounding vehicles, pedestrians, and cyclists follow their extracted trajectories, preserving the timing, spatial arrangement, and motion patterns of the collected dense intersection scenes. This follows the rationale of data-driven non-reactive evaluation used in NAVSIM v1 Dauner et al. (2024) and NAVSIM v2 Cao et al. (2025), where avoiding reactive behavior models improves reproducibility and prevents benchmark results from being dominated by assumptions or artifacts of a learned or rule-based traffic simulator. DriveE2E extends this philosophy by implementing the evaluation in CARLA: while non-ego agents retain their real-world trajectories, the ego policy is executed online, its actions update the ego state, and subsequent sensor observations are rendered from the updated ego viewpoint. Thus, DriveE2E combines real-trajectory-preserving non-ego replay with an ego-feedback observation loop for E2EAD policies. The resulting protocol should therefore be interpreted as a real-trace-preserving ego-closed-loop replay evaluation, rather than as a fully reactive traffic simulation.

F.4 Real-world Closed-loop Testing

We have not conducted controlled real-world closed-loop E2EAD evaluation. Such testing is difficult for two reasons. First, real-world intersections are not repeatable experimental environments: traffic participants, trajectories, timing, weather, and interactions vary across trials, making fair model comparison difficult. Artificially controlling surrounding traffic would also introduce safety and ethical concerns. Second, deploying a trained model from DriveE2E into a real vehicle requires substantial engineering, including real-time onboard integration, safety monitoring, sensor calibration, hardware synchronization, actuation-interface adaptation, and fallback mechanisms.

DriveE2E therefore targets a different use case. It provides a reproducible benchmark for comparing E2EAD models under a clearly defined ego-closed-loop replay protocol. Real-world deployment still requires additional validation under real sensors, responsive traffic, safety constraints, and operational design-domain requirements.

F.5 Summary

DriveE2E provides infrastructure-grounded digital twins and real-trajectory replay scenarios for reproducible ego-feedback evaluation. Its main strengths are controllability, repeatability, dense intersection reconstruction, and the ability to update ego observations according to the executed ego actions. Its main Sim2Real boundaries are CARLA visual fidelity, sensor-domain gaps, trajectory-estimation noise, the protocol boundary around non-ego reactivity, and the absence of real-world closed-loop deployment. Accordingly, DriveE2E should be used as a focused benchmark for studying policy behavior in reconstructed dense intersections, while its results should be interpreted within the scope of the ego-closed-loop replay protocol.

G Driving Scenarios Visualization

Driving Behavior Illustration. DriveE2E identifies and categorizes eight behavior groups from 800 real-world traffic clips, covering common behavior categories within the collected intersection scenarios. These scenarios include Interaction with Pedestrians and Cyclists (IPC), Competing with Other Vehicles (COV), Passing Through During Yellow Lights (YLW), Making a U-turn (UT), Stopping at Red Lights (STP), Going Straight Through Intersections (STR), Making a Left Turn (LFT), and Making a Right Turn (RT).

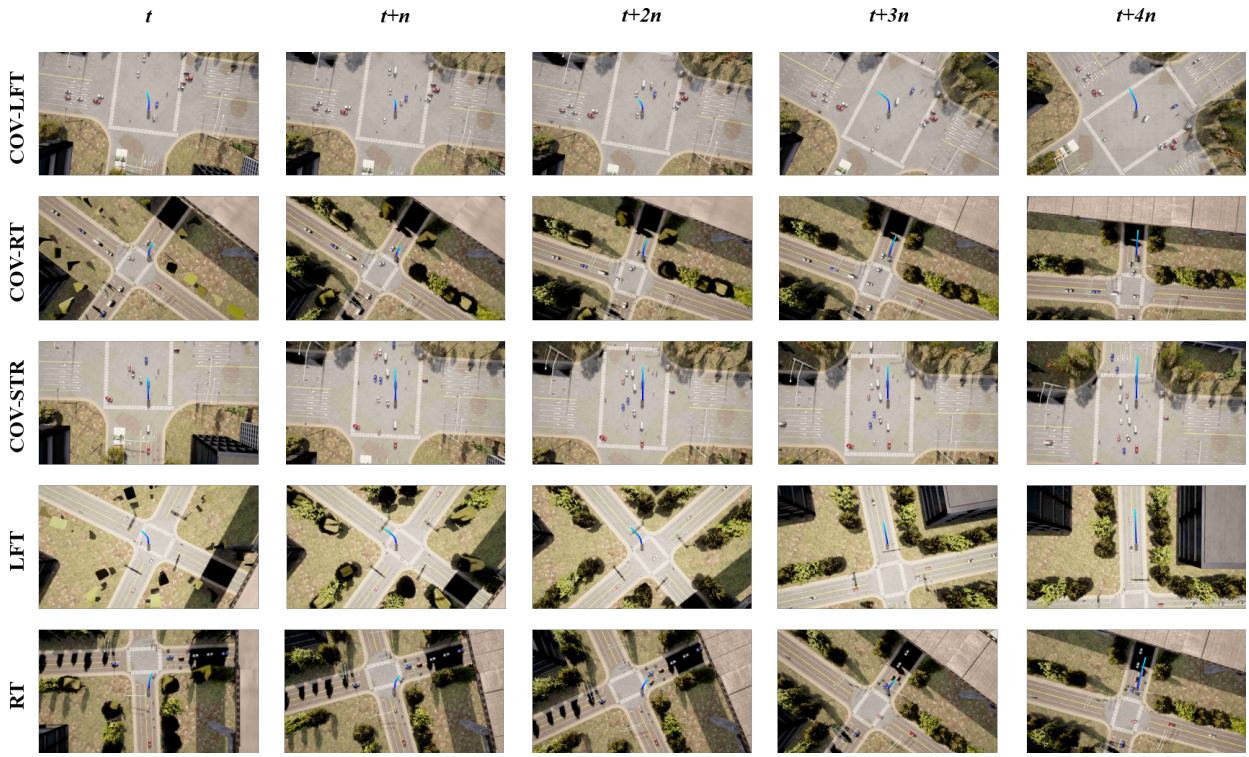


Figure 8: **Five Sub-scenario Driving Behavior Visualization:** This visualization encompasses five driving scenarios: competing with other vehicles while turning left (COV-LET), turning right (COV-RT), and going straight (COV-STR), as well as normal left turns (LFT) and right turns (RT). Frames are sampled at intervals t, $t+n$, $t+2n$, $t+3n$, and $t+4n$ from the driving sequences to depict the vehicle's behavior over time. Each image is presented from a **top-down view**, with the ego vehicle (depicted in gray) centrally positioned. The vehicle's motion direction is represented by a purple trajectory line.

- **IPC**: *Interaction with Pedestrians and Cyclists* involves safely navigating around or yielding to pedestrians and cyclists.
- **COV**: *Competing with Other Vehicles* refers to scenarios where the vehicle asserts its position in traffic, such as during merges or unprotected left turns.
- **YLW**: *Passing through during Yellow Lights* describes the decision-making process of whether to stop or start when the light turns yellow, balancing safety and timing.
- **UT**: *Making a U-turn* involves turning the vehicle to reverse its direction, either partially or fully, at an intersection or designated point.
- **STP**: *Stopping at Red Lights* involves halting the vehicle to comply with traffic signals.
- **STR, LFT, RT**: *Going Straight through Intersection, Making a Left Turn, and Making a Right Turn* are the most common driving behaviors at intersections, not specifically categorized under the other types.

These eight scenarios are **further refined into 14 specific sub-scenarios** based on turning conditions and anomalies. We illustrate these sub-scenarios in Figure 8, Figure 9 and Figure 10.

Twin Weather and Lighting Conditions. We document weather and lighting metadata for each driving scenario and approximate these conditions with CARLA's built-in weather system. The resulting scenes are intended to preserve coarse environmental variation rather than precisely reproduce all real-world visual effects. Figure 11 illustrates reconstructed scenes under different weather and lighting conditions.

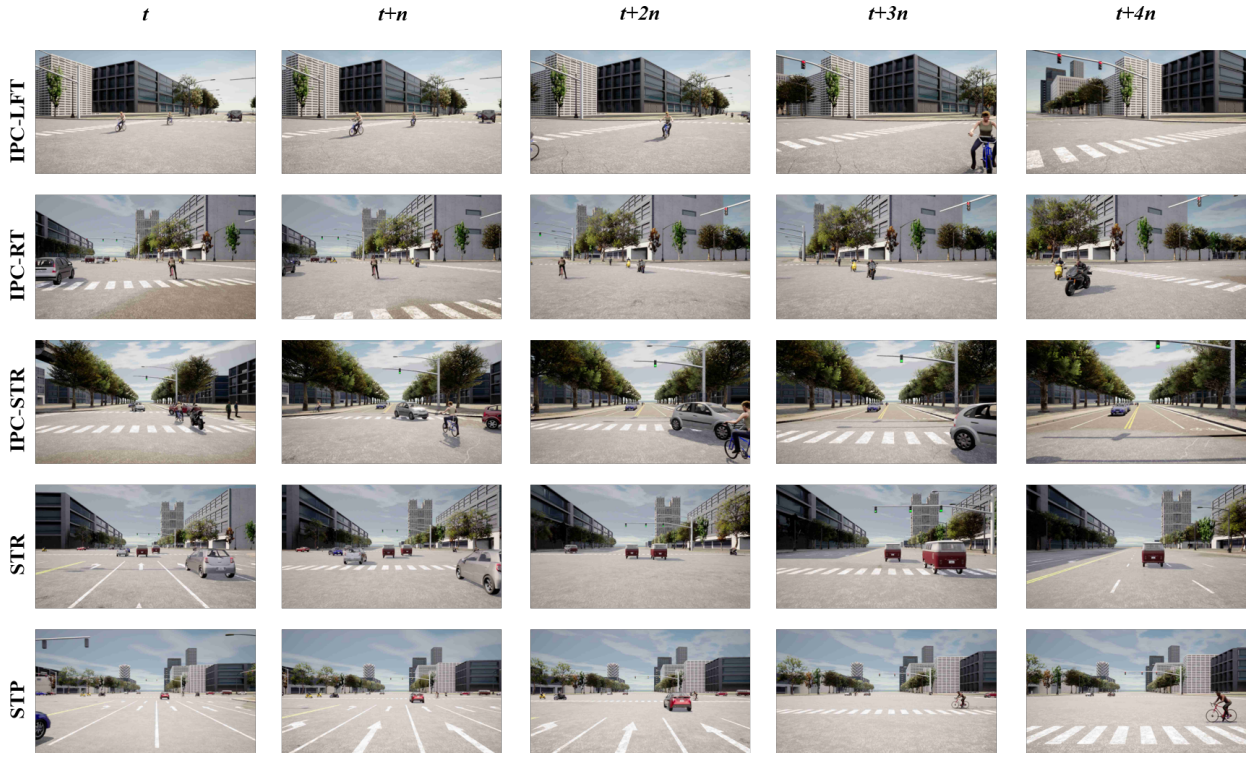


Figure 9: **Five Sub-scenario Driving Behavior Visualization:** This visualization encompasses five driving scenarios: Interaction with pedestrians and cyclists while turning left (IPC-LET), turning right (IPC-RT), and going straight (IPC-STR), along with normal straight driving (STR) and stopping at red lights (STP). Each image is presented from a **front view**.

H Visualization of Occlusion-Based Agent Filtering

We also present a visualization comparing traffic scenarios before and after occlusion-based agent filtering in Figure 12 and Figure 13.

I Planning Results Visualization

This section presents the visualization of planning results, showcasing both successful and failed cases of the VAD model across three scenarios: competing with other vehicles (COV), normal left turns (LFT), and going straight (STR). The corresponding visualizations are shown in Figure 14, Figure 15, and Figure 16, respectively.

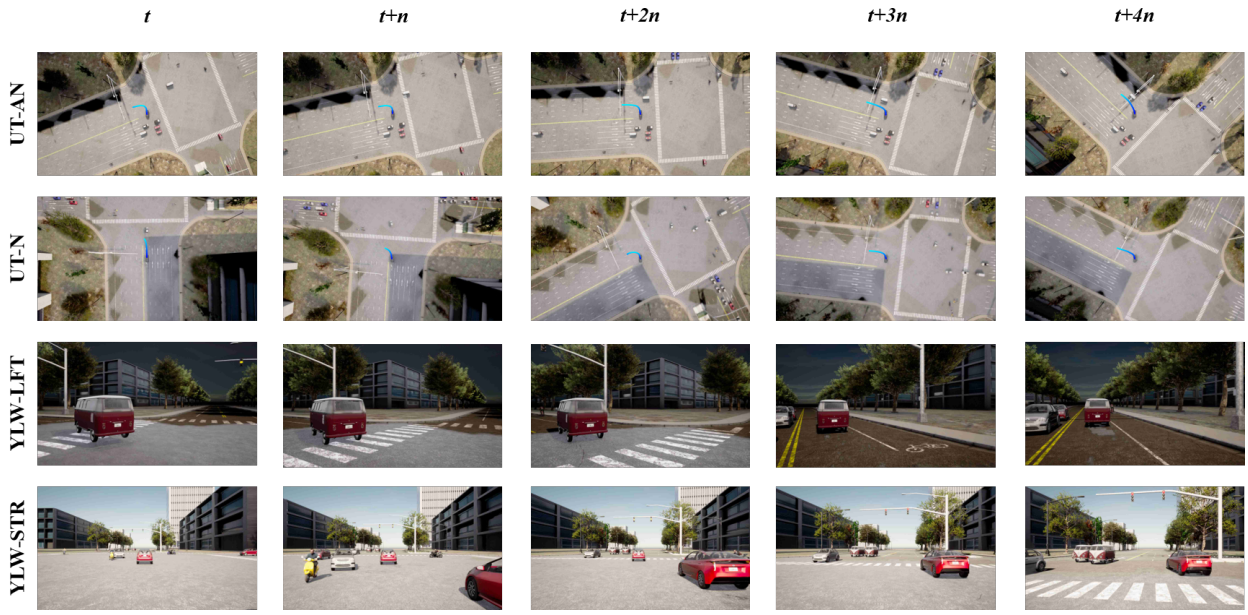


Figure 10: **Four Sub-scenario Driving Behavior Visualization:** This visualization encompasses four driving scenarios: U-turns in abnormal (UT-AN) and normal conditions (UT-N), and passing through yellow lights while turning left (YLW-LFT) or going straight (YLW-STR). Each image is presented from a **top-down view** or **front-head view**.

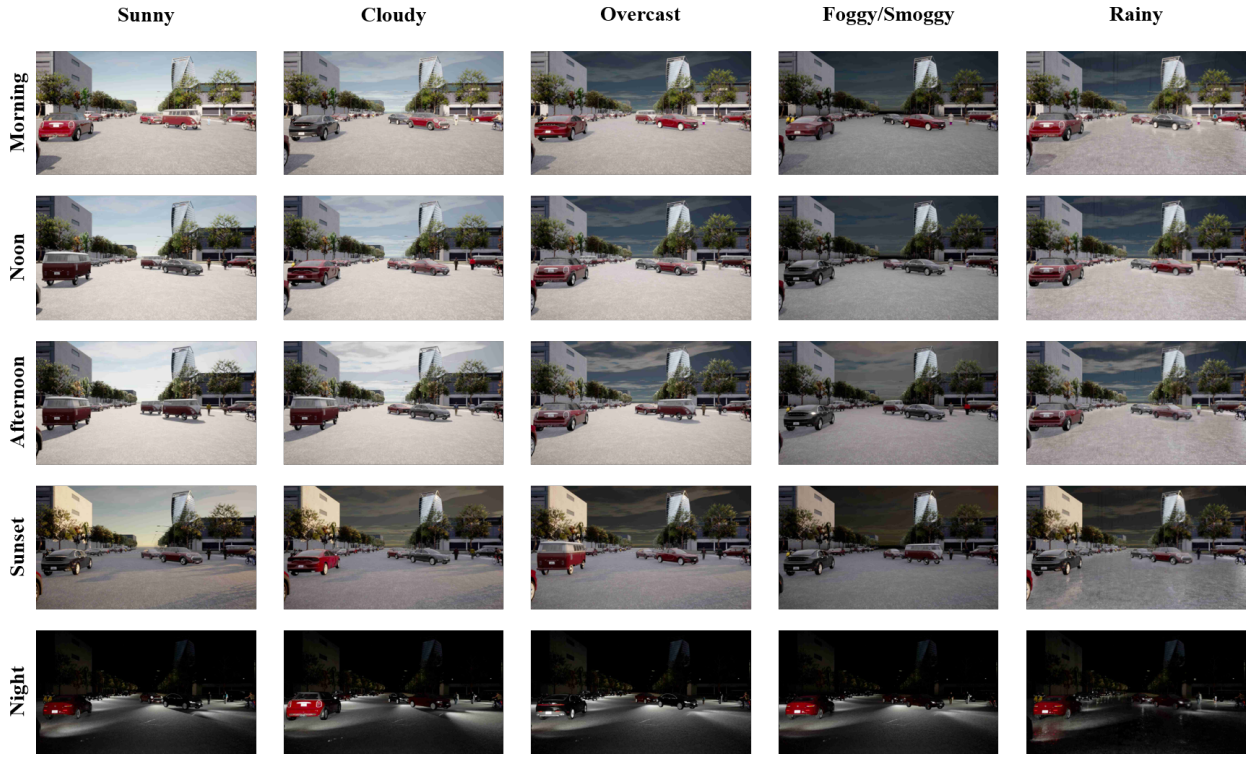


Figure 11: Weather and Lighting Conditions in Reconstructed Scenes.

(a) Traffic Scenario with Complete Agents

(b) Traffic Scenario with Incomplete Agents

Figure 12: **Traffic Scenario Comparison with Occlusion Filtering:** (a) Traffic scenario extracted from infrastructure sensor data, capturing all traffic agents. (b) Traffic scenario constructed by filtering out agents occluded from the ego-vehicle's sensor view. In this comparison, only vehicle agents are filtered, while all other agent types are retained.

(a) Traffic Scenario with Complete Agents

(b) Traffic Scenario with Incomplete Agents (Filter All)

Figure 13: **Traffic Scenario Comparison with All Occluded-Agents Filtering:** (a) Traffic scenario extracted from infrastructure sensor data, capturing all traffic agents. (b) Traffic scenario constructed by filtering out agents occluded from the ego-vehicle's sensor view. In this comparison, all agents are filtered.

Figure 14: **Successful and Failed Cases in the COV Scenario:** In the failed case, the ego vehicle controlled by the trained VAD model exhibited excessive caution while competing for the lane with another vehicle and collided with a replayed vehicle approaching from the right rear. Under the same replay protocol, the successful case maintained a higher speed and avoided collision.

Figure 15: **Successful and Failed Cases in the LFT Scenarios:** In the failed case, the VAD-controlled ego vehicle was cautious during a left turn and collided with replayed oncoming traffic. Under the same replay protocol, the successful case completed the turn without collision.

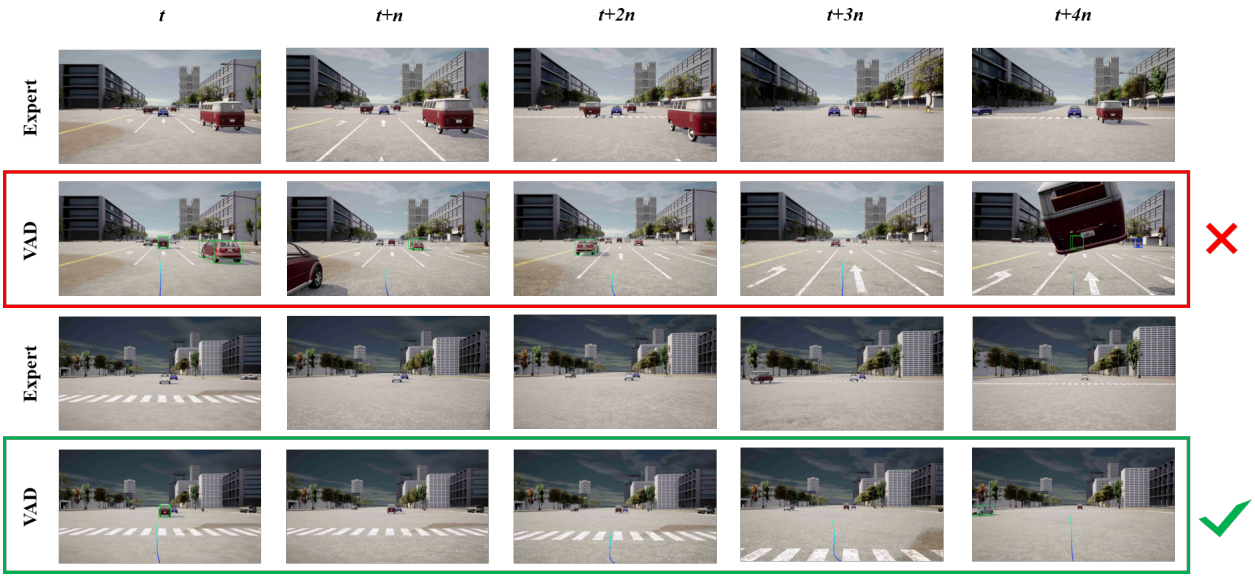


Figure 16: **Successful and Failed Cases in the STR Scenarios:** In the failed case, the ego vehicle controlled by the trained VAD model accelerated slowly while traveling straight and collided with a replayed trailing vehicle. Under the same replay protocol, the successful case traversed the intersection without collision.