CriticalScenes

Scalable Photorealistic Critical-Scenario Generation for Autonomous Driving Training and Evaluation

Master Thesis

Alexandros Theocharous

Delft University of Technology



TU Delft

CriticalScenes

Scalable Photorealistic Critical-Scenario Generation for Autonomous Driving Training and Evaluation

by

Alexandros Theocharous

Student Name	Student Number
Alexandros Theocharous	5930901

To obtain the degree of Master of Science in Robotics at the Delft University of Technology. Day of public defence, Thursday, September 3rd, 2026

Instructor and Committee Chair:	Holger Caesar
Defence Committee:	Hadi J. Rad
Defence Committee:	Z. (Ziqi) Zhang
Faculty:	Faculty of Mechanical Engineering, Delft

TU Delft

Acknowledgements

The present document is the report of the Master Thesis Assignment conducted by the author, for the needs of the MSc Robotics at TU Delft.

I would like to take this opportunity to thank all the people in my family and those close to me who have supported me throughout this journey; without them, I believe this result would not have been possible. Additionally, I would also like to thank my supervisor, who assisted me throughout this very demanding process.

Alexandros Theocharous
Delft, September 2026

CriticalScenes: Scalable Photorealistic Critical-Scenario Generation for Autonomous Driving Training and Evaluation

Alexandros Theocharous Holger Caesar

Delft University of Technology
Delft, The Netherlands

Abstract

Safety-critical interactions are rare in real-world driving data, making it difficult for end-to-end autonomous-driving planners to learn reliable collision-avoidance behaviour from standard training datasets alone. This work presents CriticalScenes, a reconstruction-based framework for automatically transforming ordinary recorded driving scenes into controllable, photorealistic critical scenarios for both evaluation and training. The framework operates on editable 3D Gaussian Splatting reconstructions of real driving environments, enabling the generation of collision-course threat interactions and corresponding safe ego responses while preserving compatible surrounding traffic and the original scene's appearance. To improve visual fidelity under modified viewpoints and actor placements, the rendered observations are subsequently processed with a diffusion-based image restoration model to reduce novel-view artefacts. The generated scenarios expose substantial weaknesses in UniAD, a structured, end-to-end autonomous-driving planner, particularly in interactions requiring strong evasive behaviour. Fine-tuning UniAD on the generated critical scenarios substantially improves performance on unseen generated interactions while preserving performance on conventional nuScenes driving. Evaluation on NeuroNCAP, an independent photorealistic closed-loop safety benchmark, further shows that the learned behaviour transfers beyond the proposed generation pipeline, although the improvement is interaction-dependent and considerably smaller than in open-loop evaluation. Overall, the results demonstrate that reconstruction-based critical-scenario generation can provide useful and scalable supervision for safety-critical autonomous-driving behaviour.

1. Introduction

End-to-end autonomous-driving systems increasingly integrate perception, prediction, and planning within a single learning framework, allowing driving decisions to be optimised directly toward the final planning objective. Structured end-to-end approaches, such as UniAD [20], have demonstrated strong performance on large-scale real-world autonomous-driving benchmarks, including nuScenes [5]. However, these models are predominantly trained by imitating recorded human driving, in which imminent collisions and emergency avoidance manoeuvres occur only rarely. Consequently, strong performance on conventional open-loop benchmarks does not necessarily imply that a planner has learned how to respond when rapid corrective action is required.

Closed-loop safety benchmarks such as NeuroNCAP [25] expose this limitation by placing autonomous-driving models in photorealistic reconstructions of real driving scenes and introducing actors on predefined collision courses. Rather than evaluating only the deviation between a predicted trajectory and a recorded expert trajectory, closed-loop evaluation allows the planner's decisions to influence the future scene state and directly measures whether the resulting collision is avoided. Such benchmarks demonstrate that safety-critical behaviour cannot be assessed solely through conventional open-loop evaluation. At the same time, they highlight a broader data-generation problem: constructing controlled, photorealistic critical interactions typically requires carefully selected scenes and actors, manually designed trajectories, and substantial supervision to ensure that the resulting scenario remains visually and physically valid.

Synthetic data generation offers a way to provide the missing critical-scenario supervision without deliberately collecting dangerous interactions in the real world. Conventional simulators such as CARLA [14] and AirSim [34] provide strong controllability but introduce a visual sim-to-

real gap. Generative video methods [24, 39] can produce diverse and realistic driving observations, but maintaining temporal and cross-view consistency while providing precise control over scene evolution remains challenging. World models [3, 19, 31] offer powerful learned representations of driving dynamics, but can require substantial computational resources and are generally formulated around learned scene evolution rather than explicit geometric specification of individual actor trajectories. Reconstruction-based approaches provide an attractive alternative by modifying recorded real-world environments while retaining their original appearance and geometry. Methods such as NeuRAD [36] and StreetGaussians [42] additionally separate dynamic traffic participants from the static environment, enabling controlled actor manipulation.

This work builds on the reconstruction-based formulation and introduces an automated framework to convert ordinary nuScenes recordings into photorealistic, safety-critical scenarios. The framework reconstructs an editable real-world scene, automatically constructs a controlled collision-course interaction together with a corresponding safe ego response, and renders the modified interaction while preserving compatible surrounding traffic. The resulting observations and annotations are converted into a nuScenes-compatible representation that can be used directly for end-to-end planner evaluation or training.

The framework is evaluated from both a data-generation and planner-adaptation perspective. We first investigate whether the manipulated threat actors remain perceptually usable, whether the generated interactions provide challenging yet plausible planner test cases, and whether multiple critical variants can be produced efficiently from reconstructed scenes. The generated scenarios are then used to fine-tune UniAD, with evaluation on unseen-source synthetic scenarios and on ordinary nuScenes driving scenarios. Finally, NeuroNCAP provides an independent closed-loop test of whether the learned behavioural changes transfer beyond the proposed StreetGaussians generation pipeline.

Based on this formulation, this work addresses the following research questions:

1. Can vehicles rendered using 3D Gaussian Splatting be used to create critical scenarios that remain perceptually usable by an end-to-end planner?
2. Can the generated scenarios provide challenging yet physically plausible test cases for end-to-end planners?
3. Can the proposed framework efficiently generate diverse critical scenarios from reconstructed real-world driving scenes?
4. Does fine-tuning on the generated critical scenarios improve collision-avoidance behaviour while preserving ordinary driving performance?

In addressing these questions, this work makes the following contributions:

- We introduce *CriticalScenes*, a scalable reconstruction-based framework that transforms ordinary recorded driving scenes into diverse, photorealistic critical scenarios for direct use by end-to-end driving planners. The generated scenarios include both controlled collision-course interactions and corresponding safe ego responses, allowing the same reconstructed environment to support multiple critical variants.
- We demonstrate that the generated scenarios provide both targeted safety evaluation and useful supervision for planner adaptation. Using UniAD as the downstream planner, the scenarios expose substantial weaknesses in collision avoidance during safety-critical interactions that are rare in ordinary driving data and enable targeted fine-tuning of the corresponding behaviour.
- We identify and analyse an open–closed-loop generalisation gap in the fine-tuned UniAD planner: strong improvements on unseen-source-generated critical scenarios do not uniformly translate to closed-loop collision avoidance, revealing limitations in the initiation and maintenance of evasive behaviour under replanning.

2. Related Work

2.1. End-to-End Autonomous Driving

End-to-end autonomous driving seeks to optimise driving decisions directly from sensor observations rather than relying on independently designed perception, prediction, planning, and control modules [1, 32]. Direct approaches such as TransFuser [9] and TCP [41] map sensor observations to trajectories or vehicle controls with limited explicit intermediate structure. Structured end-to-end methods instead retain interpretable intermediate representations while jointly optimising the system toward planning.

UniAD [20] is a representative structured end-to-end architecture that integrates object tracking, online mapping, motion forecasting, occupancy prediction, and ego planning within a unified framework. Subsequent approaches have explored alternative scene representations and planning structures. VAD [21] represents surrounding agents and map elements using vectorised queries, while SparseDrive [35] employs sparse scene representations to reduce the computational cost associated with dense BEV features. More recent methods, such as BridgeAD [43], additionally incorporate historical prediction and planning information to improve temporal consistency between successive decisions. Despite these architectural developments, driving behaviour is still largely learned from recorded real-world trajectories. Consequently, the ability of such models to perform rare collision-avoidance manoeuvres remains closely related to the availability of corresponding examples in the training distribution.

The scale of the training data used by representative end-to-end driving methods further illustrates this dependency. As summarised in Table 1, established structured planners are typically trained on hundreds to thousands of driving scenes, while more recent VLA-based approaches [8] increasingly rely on tens of thousands of curated samples or substantially larger collections of driving video. At the same time, dedicated safety-critical or long-tail interactions are absent from the standard training data of many widely used planners. Providing such behaviour through manually constructed scenarios would therefore be difficult to scale to the volume required for modern end-to-end training, motivating automated methods for generating controlled critical interactions.

2.2. Critical-Scenario Data Generation

Safety-critical interactions form a long-tailed component of real-world driving data and are difficult to deliberately collect because reproducing imminent collisions is unsafe. Synthetic generation, therefore, provides a practical means of introducing controlled critical interactions into training and evaluation. Conventional simulators such as CARLA [14] and AirSim [34] provide explicit control over the environment and traffic participants, but the visual discrepancy between simulated and real-world observations can introduce a sim-to-real domain gap [30]. Generative approaches offer greater visual diversity, but introduce different challenges. Diffusion-based driving generators explicitly identify temporal and cross-view consistency and controllability as key difficulties [24, 39]. World models such as GAIA-1 [19] can model complex scene evolution and provide conditioning over ego behaviour and scene properties, but large generative world models can require substantial computational resources for training and high-dimensional video generation. Moreover, they are formulated around learned scene evolution rather than explicit geometric construction of individual actor trajectories.

Reconstruction-based rendering provides an alternative by constructing editable representations directly from recorded real-world environments. NeRF [27] and 3D Gaussian Splatting [22] enable novel views of reconstructed scenes to be rendered while preserving much of their original appearance and geometry. Driving-specific methods such as NeuRAD [36] and StreetGaussians [42] adapt reconstruction-based rendering to dynamic road scenes by modelling the static environment and traffic participants separately. However, reconstruction quality is inherently limited by the viewpoints observed in the original driving sequence. Unlike many novel-view-synthesis settings in which images can be deliberately captured from diverse viewpoints around a scene or object, autonomous-driving datasets are recorded primarily along the vehicle's natural ego trajectory and therefore provide comparatively narrow viewpoint coverage. As a result, repositioning a traffic actor may expose surfaces that were never observed during recording, leading to incomplete or distorted actor appearances. Similarly, rendering the scene from a substantially modified ego trajectory can reveal previously unseen regions of the environment, producing missing geometry or degraded background reconstruction. These limitations make actor selection, trajectory design, and restoration of novel-view artefacts important when constructing modified critical interactions.

NeuroNCAP demonstrates the potential of reconstruction-based rendering for safety-critical evaluation. Using NeuRAD reconstructions of real nuScenes sequences, the benchmark places selected traffic actors on collision courses with the ego vehicle and evaluates whether an autonomous-driving system can avoid the resulting interaction in a closed loop. Its scenarios, however, rely on manually selected source sequences and actors together with predefined collision trajectories. SceneCrafter [16] extends reconstruction-based simulation toward training-data generation by introducing interactive traffic into 3D Gaussian Splatting reconstructions and using an expert planner to generate corresponding ego supervision. Augmenting real training data with these synthetic driving logs improves both open- and closed-loop performance of VAD. However, the expert primarily follows the recorded ego route while adapting its speed according to the surrounding traffic, whereas our framework explicitly constructs collision-course interactions together with corresponding evasive ego trajectories. Its manipulated traffic participants primarily follow predefined behaviours such as lane-following, turning, and lane-changing.

These works establish reconstruction-based rendering as a promising basis for both safety evaluation and synthetic training. However, automatically constructing planner-ready collision-course interactions remains comparatively underexplored. In particular, a scalable critical-scenario generation pipeline must identify suitable actors, generate both the dangerous interaction and an appropriate avoidance response, maintain physically plausible motion and surrounding traffic context, and remain robust to artefacts from novel-view reconstruction. Our work targets this gap by combining StreetGaussians-based scene reconstruction with automated threat and ego trajectory generation, preservation of compatible recorded traffic, and Fixer [40]-based restoration, enabling reconstructed real-world scenes to be converted into controllable critical scenarios for both planner evaluation and supervision.

2.3. Closed-Loop Safety Evaluation and Adaptation

Open-loop planning evaluation measures agreement between a predicted trajectory and a recorded expert trajectory, but does not capture how executing the planner's own

Table 1. Training data and supervision used by the representative end-to-end autonomous-driving planners. Reported data quantities follow the units used by the respective works and are therefore not directly comparable. The final column indicates whether the targeted critical or long-tail data are used for training, used only for evaluation, or are not specifically reported.

Method	Family	Training	Dataset	Reported scale	Driving supervision	Critical-data use
P3 [33]	Partial structured	Multi-task IL	Proprietary	6,100 scenarios (25 s)	Scene + ego planning	No
MP3 [6]	Partial structured	Multi-task IL	Urban Expert	5,000 scenarios (25 s)	Map + perception/prediction + planning	No
UniAD [20]	Structured	Multi-task IL	nuScenes	700 scenes (20 s)	Perception + motion + planning	No
VAD [21]	Structured	Multi-task IL	nuScenes	700 scenes (20 s)	Agent/map + planning	No
BridgeAD [43]	Structured	Multi-task IL	nuScenes	700 scenes (20 s)	Perception + motion + planning	Eval. only
VaVAM [2]	Direct	Video pretrain. + IL	OpenDV + nuPlan + nuScenes	1,700+ h pretrain.; 94 h + 5.5 h IL	Ego trajectory	Eval. only
Impromptu VLA [8]	VLA	VLA fine-tuning	8 public datasets	80k+ curated clips	Trajectory + planning-language supervision	Training
Reasoning-VLA [44]	VLA	SFT + RL	8 public datasets	Unified 8-dataset corpus	Trajectory + reasoning	No
Alpamayo-R1 [29]	VLA	SFT + RL	NVIDIA proprietary	80k h driving + 700k CoC clips	Trajectory + CoC reasoning	Training

decisions changes its future observations. Moreover, multiple trajectories may safely resolve the same critical interaction, making deviation from a single recorded trajectory an incomplete measure of collision-avoidance capability. Closed-loop evaluation instead allows the planner to influence the scene's future state and directly assess the outcomes of its decisions.

NeuroNCAP provides such an evaluation using photorealistic nuScenes reconstructions containing stationary, frontal, and lateral collision threats. Its original experiments showed substantial closed-loop failure rates for both UniAD and VAD despite their strong conventional open-loop performance. NeuroNCAP has subsequently been used as a complementary safety benchmark by methods including BridgeAD [43] and VaVAM [2]. Recent work has also begun to investigate the role of training data in such evaluations. Impromptu VLA [8], for example, curates challenging long-tail driving examples from real autonomous driving datasets and reports improvements in both open-loop planning and closed-loop NeuroNCAP performance after targeted training.

These results suggest that closed-loop safety depends not only on planner architecture but also on whether appropriate critical behaviour is represented during training. Rather than introducing a new planning architecture, this work investigates whether automatically generated photorealistic critical scenarios can provide such supervision. The resulting planner is evaluated on held-out synthetic data, standard nuScenes driving data, and independently rendered NeuroNCAP scenarios, while UniAD's existing occupancy-based collision optimiser is treated separately as an inference-time ablation.

3. Methodology

This section presents the proposed critical-scenario generation framework, covering editable scene reconstruction, automated critical-interaction generation, rendering and quality control, and the use of the resulting scenarios for end-to-end planner adaptation.

3.1. System Overview

The proposed framework transforms ordinary recorded driving sequences into photorealistic, planner-ready safety-critical scenarios. As illustrated in Fig. 1, the pipeline starts from a real driving scene and progressively converts it into an editable reconstruction, constructs a controlled critical interaction, renders the modified scene, and performs quality control and dataset conversion to produce a planner-compatible scenario.

Each selected scene is first reconstructed using an object-centric 3D Gaussian Splatting method, providing an editable representation of the static environment and its dynamic traffic participants. In the implementation used in this work, StreetGaussians [42] provides this reconstruction. A conflict pose along the nominal ego trajectory is then defined, together with a suitable reconstructed threat actor and starting pose. Hybrid A* is used to construct the threat trajectory and the corresponding ego collision-avoidance response, after which the interaction is temporally aligned with the recorded scene. Surrounding traffic is preserved whenever its recorded motion remains compatible with the modified interaction.

The resulting ego and actor trajectories are converted into the pose sequences required for novel-view rendering. The modified scene is rendered from the generated ego tra-

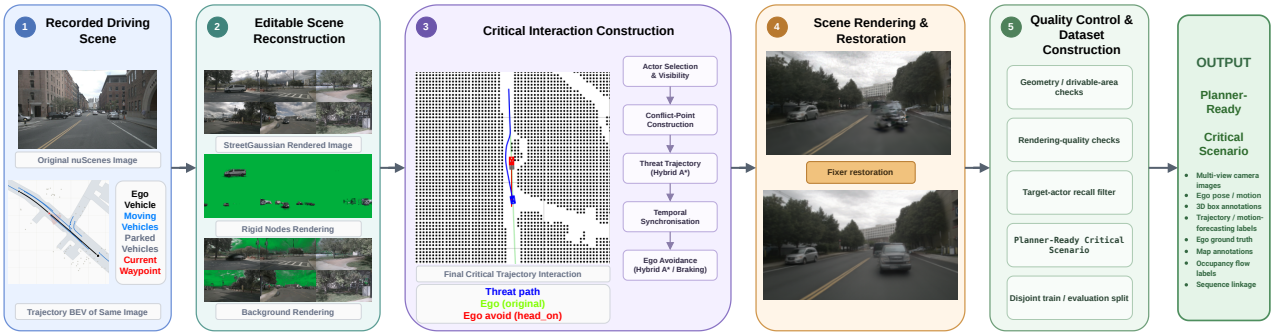


Figure 1. Overview of the proposed critical-scenario generation framework. Starting from an ordinary recorded driving scene, the framework reconstructs an editable 3D representation, selects a suitable traffic actor and conflict pose, and constructs a collision-course threat trajectory together with a corresponding ego collision-avoidance trajectory. Compatible surrounding traffic is preserved before the modified scene is rendered and restored. Following geometric, perceptual, and rendering-quality checks, accepted scenarios are converted into a planner-compatible representation for evaluation or training.

jectory, and a diffusion-based image-restoration model is then applied to reduce visual artefacts arising from viewpoints not observed during the original recording. In this work, Fixer [40] is used for this restoration stage. Finally, geometric, perceptual, and rendering-quality checks are applied before accepted scenarios are converted into a planner-compatible representation containing the observations, annotations, and planning supervision required by the downstream model.

For the experiments in this work, nuScenes provides the recorded source scenes, and UniAD is used as the downstream end-to-end planner.

3.2. Scene Reconstruction and Traffic Representation

The proposed framework assumes an editable, object-centric 3D Gaussian Splatting reconstruction in which the static environment and foreground traffic participants are represented separately. In this work, the reconstruction is obtained using StreetGaussians [42] on recorded nuScenes sequences. Rather than representing the complete driving scene as a single static reconstruction, StreetGaussians explicitly separates the static environment from dynamic foreground traffic participants. Similar object-centric decompositions have also been adopted by other 3D Gaussian Splatting methods for driving scenes, such as DrivingGaussian [46] and OmniRe [7], which separately represent the environment and dynamic actors to support reconstruction and manipulation of complex traffic scenes.

This decomposition is particularly useful for the proposed critical-scenario generation framework, since the Gaussian representation of each reconstructed vehicle is associated with that individual object, while the remaining Gaussians represent the static environment. Consequently, a vehicle can be manipulated independently of the background: its recorded poses can be replaced by those of a newly generated trajectory, or the actor can be suppressed from the rendered scene without modifying the remainder of the reconstruction. Figure 2 illustrates this separation between the reconstructed static environment and the individually represented rigid traffic objects.

The identities and recorded trajectories of the reconstructed traffic participants are retained as part of the editable scene representation. This allows the proposed framework to manipulate selected actors while preserving the remaining traffic context whenever possible, to increase scene realism and diversity. The reconstruction configuration, rigid-object association, and implementation of object transformation and suppression are detailed in Appendix A.

3.3. Conflict-Pose Construction and Threat-Actor Selection

The construction of a critical interaction begins by selecting where and when to challenge the nominal ego trajectory. A conflict pose $\mathbf{q}_c = (x_c, y_c, \psi_c)$ is sampled directly from the recorded future ego trajectory between 5 and 15 s after the beginning of the sequence. The corresponding position defines the interaction location, while its timestamp determines the time available for the threat actor to reach it. A feasible threat starting pose $\mathbf{q}_s = (x_s, y_s, \psi_s)$ is then sampled from the surrounding drivable area. Candidate starts are restricted to the same connected drivable region as the conflict point, must allow the threat to reach the conflict point within the available time at a plausible urban speed, and must produce an approach direction sufficiently different from the ego trajectory to suppress approximately parallel rear-end interactions. The sampled pose must additionally be collision-free. The complete sampling constraints and implementation are provided in Appendix B.1.

Geometric suitability alone is insufficient for selecting



(a) Static environment.

(b) Dynamic rigid objects.

Figure 2. StreetGaussians scene decomposition used by the proposed framework. The static environment is represented separately from foreground traffic participants, allowing individual reconstructed vehicles to be independently manipulated when constructing a critical scenario.

the reconstructed vehicle used as the threat, since rendering quality depends on whether the viewpoints required by the generated interaction were sufficiently observed during reconstruction. The framework, therefore, estimates an actor-relative visibility profile for each reconstructed vehicle and compares it with the viewpoints required by the generated threat trajectory. As illustrated in Fig. 3, different actors can provide substantially different viewpoint coverage. Candidate actors with compatible viewpoint coverage are retained, after which the actor whose original observation conditions most closely match those required by the generated interaction is selected. The complete visibility-profile construction and matching procedures are detailed in Appendix B.2, while the final candidate-ranking procedure is provided in Appendix B.3.

3.4. Critical-Scenario Generation

Given the conflict pose and threat starting pose defined in Sec. 3.3, the framework constructs a collision-course trajectory for the threat actor and a corresponding collision-avoidance response for the ego vehicle. Hybrid A* is used to generate kinematically feasible trajectories within the available drivable region, accounting for the non-holonomic motion of car-like vehicles. This formulation has been widely adopted for autonomous vehicles and non-holonomic wheeled robots, where feasible steering and orientation must be considered during path generation [4, 11, 12, 28]. For threat generation, alternative terminal orientations support both crossing and oncoming interactions, and the lowest-cost valid forward trajectory is retained. If no feasible moving trajectory can be generated, the selected actor is retained as a stationary threat. Rear-end configurations are excluded because the framework focuses on interactions that require an active ego-avoidance response. The complete Hybrid A* formulation, terminal-pose construction, collision model, and planner configuration are provided in Appendix C.1.

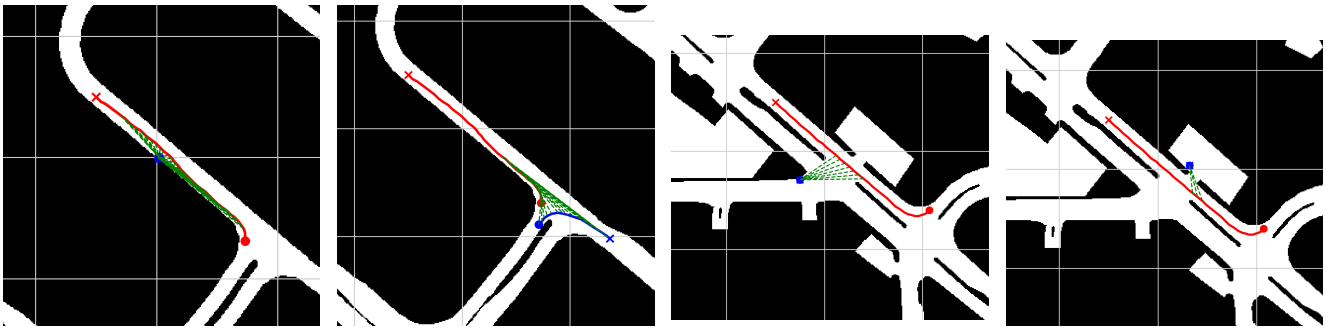
For moving threats, the generated spatial trajectory is temporally synchronised with the recorded ego trajectory such that the threat and nominal ego reach the conflict region at approximately the same time. Hybrid A* therefore determines the spatial geometry of the threat motion, while temporal synchronisation determines when the critical interaction occurs. The complete trajectory resampling and timing procedure is described in Appendix C.2.

A corresponding safe ego response is then generated. A steering-based evasive manoeuvre is first sought using Hybrid A*, allowing the ego vehicle to depart from the nominal trajectory, avoid the threat, and subsequently recover towards the recorded route. Moving head-on interactions require such lateral avoidance and are discarded if no feasible steering response can be generated. For interactions in which stopping before the conflict region provides a valid response, a braking trajectory is generated instead by modifying the ego vehicle's longitudinal progression along its recorded route. The original recorded ego trajectory is retained as the nominal collision trajectory, while the generated response provides corresponding supervision for safe planning. The complete reaction-point selection, steering and braking construction, and trajectory post-processing are detailed in Appendix C.3.

Representative generated threat and ego-avoidance trajectories are shown in Fig. 4, illustrating lateral-crossing, head-on steering, and head-on braking interactions. Additional moving- and stationary-threat examples are provided in Appendix C.4.

3.5. Rendering and Restoration

Once the threat and ego trajectories have been generated, they are applied to the editable 3D Gaussian Splatting reconstruction to produce the modified critical interaction. The generated ego trajectory determines the camera poses used for novel-view rendering, while the selected reconstructed vehicle is repositioned according to the generated threat trajectory. Surrounding traffic participants retain their recorded trajectories whenever these remain compatible with the modified interaction, preserving as much of the original scene structure and traffic context as possible.



(a) (b) (c) (d)

Figure 3. Examples of the actor-visibility analysis produced directly by the visibility module for different reconstructed rigid actors. White regions denote the drivable area used by the geometric visibility test, while black regions lie outside the drivable area. The red curve represents the recorded ego trajectory, and the blue curve or marker represents the recorded trajectory of the candidate actor. Each green dashed segment connects the ego and actor at a timestep for which the actor passes the line-of-sight visibility test; the orientations of these segments therefore indicate the relative viewpoints from which the vehicle was observed during the original sequence. The visible observations are subsequently classified into eight actor-relative viewing directions and used to construct the actor's visibility profile.

Despite the viewpoint-aware actor selection in Sec. 3.3, the modified ego and actor poses can still depart from the viewpoint distribution observed during reconstruction, resulting in incomplete geometry, missing image regions, or other novel-view rendering artefacts.

To improve the visual completeness of the generated observations, the raw novel-view renderings are subsequently processed with a diffusion-based image restoration model. In the implementation used in this work, this stage is performed with Fixer [40]. The restoration stage operates on the rendered images rather than modifying the underlying trajectories or reconstructed scene geometry, and is therefore used only to improve the visual appearance of the final camera observations. An example of the raw and restored output is shown in Fig. 5. Exact rendering and restoration settings and additional examples are provided in Appendix C.5.

3.6. Scenario Quality Control and Dataset Construction

Not every successfully planned and rendered interaction is suitable for training or evaluation, and generated scenarios therefore undergo a quality-control stage before being converted into the final dataset representation. Most quality-control steps are automated. Geometric checks reject interactions with invalid trajectory placement or collisions with incompatible scene geometry, while perception-based target-actor recall provides an automatic post-render measure of whether the manipulated threat remains recognisable to the downstream planner. Scenarios are additionally rejected when the threat actor is missing or incorrectly positioned. Manual inspection is retained only for visual failure modes that are difficult to capture reliably with the current automatic criteria, such as severe reconstruction artefacts, malformed actor appearance, or incomplete restoration.

Particular attention is given to the reconstructed threat actor, since it represents the central modification of each generated scenario. The actor must remain sufficiently observable during the critical part of the interaction, and its rendered position must remain consistent with the generated trajectory. These post-render checks complement the viewpoint-aware actor selection described in Sec. 3.3: actor selection reduces the likelihood of unsupported novel views before rendering, while quality control evaluates the resulting observations after the complete scene has been generated. The exact target-recall criterion and filtering procedure are described in Sec. 4.2.

Accepted scenarios are subsequently converted into the data representation required by the downstream planner, thereby associating rendered observations with generated ego poses, modified actor annotations, and collision-avoidance planning supervision. Variants originating from the same reconstructed source scene are retained within the same dataset split to prevent closely related versions of the same environment from appearing in both training and evaluation. In the implementation used in this work, the scenarios are converted into the nuScenes-compatible representation required by UniAD. The complete temporal-subsampling, image, pose, annotation, trajectory, sequence-linkage, and CAN-bus conversion procedure is provided in Appendix C.6.

3.7. UniAD Fine-Tuning

The generated critical scenarios are used as targeted supervision for adapting UniAD. Fine-tuning is initialised from the pretrained Stage-2 checkpoint rather than training the model from scratch.

For each accepted synthetic scenario, the rendered multi-

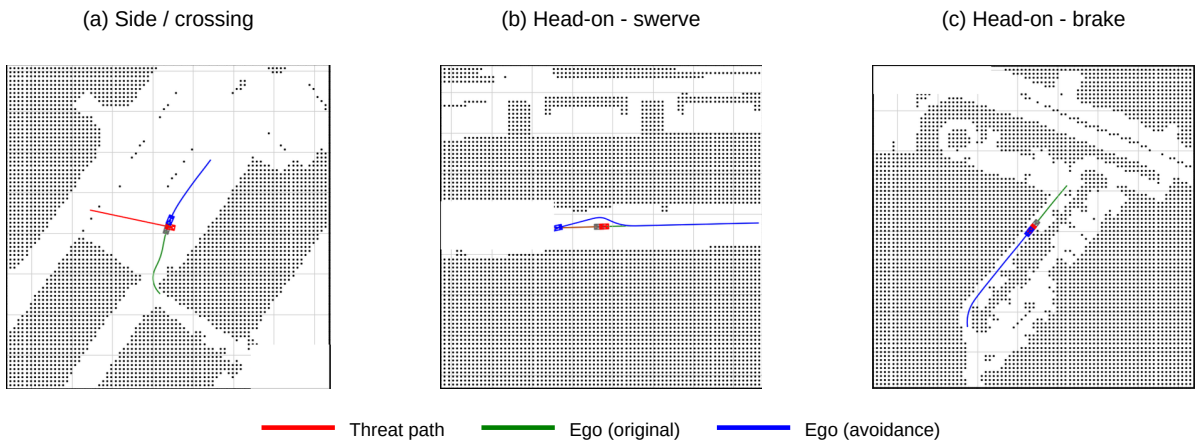


Figure 4. Representative critical interactions generated by the proposed trajectory-generation module. Red denotes the generated collision-course threat trajectory or stationary threat location, green the nominal recorded ego trajectory, and blue the generated collision-avoidance response, which in steering-based cases is constructed to return to the nominal route after passing the conflict.



Figure 5. Example of the rendering and restoration stage. Left: raw novel-view render produced by StreetGaussians after modifying the ego and threat trajectories. Right: the corresponding image after Fixer restoration. The restoration stage reduces visual artefacts arising from viewpoints that differ from those used during scene reconstruction.

camera observations are paired with the generated collision-avoidance ego trajectory described in Sec. 3.4. This trajectory replaces the original collision-course ego motion as the planning supervision for the modified interaction, thereby providing an explicit example of the desired response to the introduced threat. Synthetic critical scenarios are combined with samples from the original nuScenes training set to reduce the risk that adaptation to the comparatively small critical-scenario set degrades ordinary driving behaviour. A similar fine-tuning strategy has been followed in Scencrafter [16] to avoid catastrophic forgetting.

Following the Stage-2 training procedure, the perception-related representation is kept frozen, and fine-tuning is concentrated on the task-specific components responsible for downstream prediction and planning. This focuses adaptation on the components relevant to the introduced critical behaviour while preserving the pretrained perception representation. The exact data mixture, optimisation settings, module-freezing configuration, and training duration are reported in Sec. 4.1 and Appendix D.2.

4. Experiments

The experiments evaluate the proposed framework from both a data-generation and planner-adaptation perspective. We first assess the perceptual quality, trajectory feasibility, and scalability of the generated critical scenarios. We then use the scenarios as an open-loop stress test for pretrained UniAD, before evaluating whether targeted fine-tuning improves critical-scenario behaviour while preserving performance on ordinary nuScenes driving and transferring to the independently rendered NeuroNCAP closed-loop benchmark.

4.1. Experimental Protocol

After trajectory generation, rendering, and initial quality-control filtering, 188 candidate critical scenarios remain. From this pool, 100 scenarios are selected for fine-tuning, and a separate set of 20 scenarios is selected for synthetic evaluation. To ensure independence between the two sets, all variants originating from the same reconstructed source scene are assigned to a single split; consequently, none of the source scenes used to generate the evaluation scenarios contributes any scenarios to the fine-tuning set. We refer to this as the *unseen-source evaluation set*, since it measures generalisation to critical interactions generated from previously unseen reconstructed environments. The 100 fine-tuning scenarios comprise 39 head-on steering, 36 head-on braking, and 25 side-interaction cases, while the held-out set contains 10, 5, and 5 scenarios of each type. Finally, the synthetic scenarios are mixed with original nuScenes data rather than used in isolation, so that the rare collision-avoidance behaviour receives substantial training weight while the model remains exposed to ordinary driving behaviour.

All UniAD experiments are initialised from the publicly released Stage-2 checkpoint. Fine-tuning is performed according to the procedure described in Sec. 3.7, while keeping the pretrained perception representation fixed and adapting the downstream task-specific components. Similar selective-training strategies have been used in autonomous driving to preserve pretrained perception representations while adapting downstream planning or control modules [13, 45]. The complete fine-tuning configuration, including the data mixture, optimisation settings, and module-freezing configuration, is provided in Appendix D.2.

Planner performance is evaluated on three complementary datasets. The held-out synthetic set measures performance on critical interactions generated by the proposed framework, while the standard nuScenes validation set measures retention of ordinary driving performance. Finally, NeuroNCAP provides an independent closed-loop safety evaluation based on NeuRAD reconstructions and is not used during either scenario generation or fine-tuning. It tests whether behaviour learned from the generated StreetGaussians scenarios transfers to critical interactions outside the proposed rendering pipeline.

All main UniAD evaluations use the standard occupancy-based collision optimiser with identical parameters for the original and fine-tuned checkpoints. Sensitivity to alternative collision-optimiser settings is analysed separately in Appendix D.5.

4.2. Scenario Generation Quality and Scalability

The proposed pipeline is applied to 53 reconstructed nuScenes source scenes. Across these reconstructions, 1,060 critical-interaction planning attempts are performed, of which 979 produce feasible threat trajectories. From these, 259 interactions are selected while preserving as many distinct variants per source scene as possible. An initial geometric filtering stage removes trajectories that collide with scene objects not included in the drivable-area map used by Hybrid A*, such as road barriers or vehicles not reconstructed as rigid objects. The remaining scenarios are rendered with StreetGaussians and restored with Fixer, then undergo a final manual check for missing threat actors or severe artefacts that Fixer cannot correct, resulting in 188 candidate scenarios that are subsequently used for perceptual and trajectory-quality evaluation. Subsets are then selected for fine-tuning and held-out evaluation. We evaluate this generated scenario pool along three dimensions. First, we assess whether the repositioned threat actor remains perceptually usable to UniAD during the critical interaction. Second, we evaluate whether the generated trajectories reliably construct the intended collision-course interaction while remaining geometrically and kinematically plausible. Finally, we analyse the success rate and computational cost of the generation pipeline to characterise its scalability.

Table 2. UniAD target-actor detection recall at different ego–target distances. NeuroNCAP results are reproduced from [25].

Scenario set	5–15 m	15–25 m	25–35 m
NeuroNCAP – Stationary	**0.970**	**0.980**	0.890
NeuroNCAP – Frontal	0.830	0.970	**0.900**
NeuroNCAP – Side	0.920	0.960	0.650
Ours – StreetGaussians	0.864	0.718	0.554

Following the target-actor evaluation introduced by NeuroNCAP [25], a predicted vehicle is considered a true positive when its predicted bounding box has non-zero overlap with the target annotation or when the distance between their centres is below 2 m. Recall is evaluated over the same ego–target distance intervals used by NeuroNCAP.

For dataset selection, we additionally define a critical-window recall score using the same target-matching criterion. For each generated scenario, the score is computed only over frames in which the threat is within 15 m of the ego vehicle and up to the frame of closest approach. This focuses the ranking on whether the threat remains reliably detectable during the approach to the critical interaction, while excluding post-interaction observations after the vehicles have passed. The 188 StreetGaussians-rendered candidate scenarios are ranked using this score, and the highest-quality candidates are prioritised for the 100-scenario fine-tuning set, while scenarios overlapping with NeuroNCAP source scenes are excluded to preserve the independence of the closed-loop evaluation.

For the proposed scenarios, particular emphasis is placed on the 5–15 m interval, since this corresponds to the close-range portion of the interaction in which the ego vehicle must recognise the introduced threat and initiate an avoidance response. UniAD achieves a target-actor recall of 0.864 in this range. This lies within the 0.83–0.97 range reported by NeuroNCAP for its stationary, frontal, and side collision scenarios. Despite repositioning the reconstructed vehicle and introducing modified ego and threat trajectories, the threat actor therefore generally remains perceptually usable during the most safety-critical part of the interaction.

Recall decreases to 0.718 at 15–25 m and 0.554 at 25–35 m. These longer-range measurements are less central to the intended avoidance interaction and are also more sensitive to the smaller image footprint of distant actors and, potentially, to occlusions caused by the surrounding traffic retained in the generated scenes. Close-range recall is therefore treated as the primary indicator of perceptual usability, while the longer-range measurements are reported for completeness.

The 979 feasible planned interactions are first examined to determine whether the generated threat produces the in-

tended collision-course geometry. In all 979 cases, the generated threat approaches the original recorded ego trajectory to within 2 m during the interaction, confirming that the generated path intersects the nominal ego motion rather than merely occupying the same road region. The generated moving threats travel at an average speed of approximately 22.8 km/h, consistent with the 15–30 km/h urban operating range targeted during start-pose construction.

The robustness of the trajectory planner is additionally evaluated on an independently sampled subset of 100 generated interactions, for which Hybrid A* recovers a valid threat trajectory in 99 cases. These threat trajectories exhibit a relatively aggressive dynamic tail, with a median effective lateral acceleration of 3.59 m/s^2 and 21.2% exceeding 8 m/s^2. This is less critical than the dynamic feasibility of the generated ego trajectories, which provide the actual planning supervision: none of the braking trajectories exceeds 8 m/s^2, while only 9.0% of steering-based avoidance trajectories do, with a median maximum lateral acceleration of 3.94 m/s^2. The generated ego responses are therefore generally dynamically plausible, although the most aggressive swerving manoeuvres approach the vehicle's grip limits. Hybrid A* itself enforces kinematic rather than full dynamic feasibility. A more detailed geometric and dynamic feasibility analysis, including the generated threat speed, curvature constraints, and effective lateral-acceleration distributions, is provided in Appendix D.1.

The 979 successful trajectories from 1,060 planning attempts correspond to a trajectory-generation success rate of 92.4%. From this feasible pool, 259 interactions are selected for further processing while preserving diversity across the reconstructed source scenes, leaving 720 feasible planned interactions that are not considered further in the present experiments. After geometric filtering, the remaining interactions are rendered and restored, with subsequent post-render quality control reducing the selected set to 188 candidate scenarios, as described above. The trajectory-generation stage, therefore, can produce feasible critical-interaction trajectories in large numbers, with 979 valid trajectories obtained from 1,060 planning attempts. Although subsequent geometric and visual quality filtering remains necessary, 188 of the 259 interactions selected for further processing are retained in the final candidate pool, corresponding to an acceptance rate of approximately 72.6%. This indicates that the framework can generate critical interactions at scale while maintaining a high downstream scenario-retention rate.

StreetGaussians reconstruction represents the dominant one-time computational cost, with a median reconstruction time of 59.2 min per source scene. Once a reconstruction is available, additional critical variants are considerably cheaper to produce: Hybrid A* planning requires approximately 15 s per attempted interaction, StreetGaussians rendering approximately 35 s per generated entry, and Fixer restoration approximately 120 s. Since the same reconstructed environment can support multiple independently generated critical interactions, the reconstruction cost can be amortised across scenario variants, providing the main source of scalability in the proposed framework.

Table 3. Open-loop planning performance of pretrained UniAD on the full nuScenes validation set (6,019 samples) and the 100 generated critical scenarios used for the pre-fine-tuning stress test.

Scenario set	Metric	1 s	2 s	3 s
nuScenes	L2 [m]	0.4421	0.8786	1.5177
nuScenes	Object collision	0.0000	0.0005	0.0012
nuScenes	Object-box collision	0.0013	0.0020	0.0053
Critical	L2 [m]	6.01	10.26	14.02
Critical	Object collision	0.0171	0.0422	0.0438
Critical	Object-box collision	0.1519	0.2160	0.2128

4.3. Critical-Scenario Planning Performance

We next evaluate whether the generated scenarios provide a meaningful planning challenge for the pretrained UniAD model. This experiment is performed before any synthetic fine-tuning and therefore measures the behaviour of the original Stage-2 checkpoint on the generated critical interactions. The 100 synthetic training scenarios are used here as a stress-test set, while the 20 held-out scenarios are reserved for evaluating planner adaptation in Sec. 4.4.

Table 3 compares UniAD's open-loop planning performance on the generated critical scenarios with its performance on conventional nuScenes validation data. On the generated scenarios, the L2 planning error reaches 6.01 m, 10.26 m, and 14.02 m at horizons of 1, 2, and 3 s, respectively. In comparison, the corresponding errors on nuScenes are 0.44, 0.88, and 1.52 m, thus the generated critical interactions increase the planning error by approximately 13.7×, 11.7×, and 9.2× at the respective horizons.

The degradation is also reflected in the collision-based metrics. At a 3 s horizon, the object-collision metric increases from 0.0012 on conventional nuScenes validation data to 0.0438 on the generated critical scenarios, while the object-box collision metric increases from 0.0053 to 0.2128. The latter represents an approximately 40× increase, and together with the substantially larger L2 errors, these results indicate that the generated interactions expose safety-relevant planning failures rather than merely causing the predicted trajectory to deviate from the reference trajectory.

The difficulty of the generated scenarios also depends on the required avoidance behaviour. As shown in Table 4, head-on interactions requiring a steering-based evasive manoeuvre produce the largest planning errors, reach-

Table 4. Pretrained UniAD planning error for different generated critical-interaction types.

Interaction type	L2 @ 1 s [m]	L2 @ 3 s [m]
Head-on – swerve	8.93	24.19
Head-on – brake	5.02	12.81
Side	4.26	10.05

ing 8.93 m after 1 s and 24.19 m after 3 s. Head-on interactions resolved through braking result in errors of 5.02 m and 12.81 m at the same horizons, while lateral interactions reach 4.26 m and 10.05 m. Although the number and geometry of scenarios differ between categories, the consistently larger errors for steering-based head-on avoidance suggest that interactions requiring a substantial lateral departure from the nominal trajectory are particularly challenging for the pretrained planner.

Overall, the generated scenarios expose a pronounced weakness in the pretrained planner, particularly when the required response differs substantially from ordinary lane-following behaviour. The results, therefore, establish the generated scenarios as a meaningful stress test for UniAD and motivate the targeted fine-tuning evaluation in Sec. 4.4.

4.4. Fine-Tuning and Safety Evaluation

We finally evaluate whether the generated critical scenarios provide useful supervision for adapting UniAD. The analysis considers three complementary settings: generalisation to unseen generated critical scenarios, retention of conventional nuScenes performance, and transfer to the independently constructed NeuroNCAP closed-loop benchmark. All results use the standard UniAD inference pipeline, including its default occupancy-based collision optimiser, with the same configuration applied to the original and fine-tuned checkpoints.

Synthetic generalisation is evaluated on the 20-scenario unseen-source evaluation set introduced in Sec. 4.1, comprising 715 frames. These scenarios are selected from the same quality-controlled candidate pool as the fine-tuning data, but their underlying nuScenes source scenes do not contribute any generated variants to the training set. The evaluation, therefore, measures whether the learned collision-avoidance behaviour generalises beyond the reconstructed environments observed during fine-tuning.

As shown in Table 5, fine-tuning substantially reduces planning error across all evaluated horizons. At 1 s, the L2 error decreases from 7.19 m to 0.84 m, while at 3 s it decreases from 16.92 m to 2.46 m, corresponding to an approximately 6.9× reduction. Object-collision rates also improve strongly at short horizons: at 1 s, the collision rate decreases from 0.0215 to 0.0025. The difference is smaller at the 3 s horizon, where the rate changes from 0.0203 to 0.0190. This longer-horizon open-loop metric should be interpreted differently from an executed closed-loop collision outcome, since in closed-loop operation UniAD replans before committing to the complete 3 s predicted trajectory. The strong reduction in short-horizon collision rate is therefore particularly relevant to the immediate avoidance response, while the substantial reduction in L2 error across the full horizon indicates that fine-tuning also improves the overall shape and timing of the predicted evasive manoeuvre.

Table 5. Open-loop performance of the original and fine-tuned UniAD checkpoints on the 20-scenario unseen-source evaluation set of generated critical scenarios.

Metric	1 s	2 s	3 s
Base L2 [m]	7.1860	12.4322	16.9187
Tuned L2 [m]	**0.8416**	**1.6876**	**2.4644**
Base obj. col.	0.0215	0.0241	0.0203
Tuned obj. col.	**0.0025**	**0.0177**	**0.0190**

Table 6. Planning-performance retention on the 6,019-sample nuScenes validation set.

Model	L2 @ 1 s	L2 @ 2 s	L2 @ 3 s	Obj. col. @ 3 s
Original UniAD	0.4421	0.8786	1.5177	**0.0015**
Fine-tuned UniAD	**0.4339**	**0.8594**	**1.5042**	**0.0015**

The fine-tuned checkpoint is next evaluated on all 6,019 samples of the standard nuScenes validation set to determine whether adaptation to critical interactions degrades nominal driving capability. Since the perception representation remains frozen during fine-tuning, detection, tracking, and mapping performance remains unchanged: both checkpoints achieve an NDS of 0.4502, an mAP of 0.3810, AMOTA/AMOTP of 0.3857/1.3171, and drivable-area/lane IoU of 0.6921/0.3137. Motion-prediction metrics are similarly unchanged across the evaluated classes.

Planning performance is also preserved, as shown in Table 6. The fine-tuned checkpoint achieves a slightly lower L2 error across all reported horizons, with the 3 s error decreasing from 1.5177 m to 1.5042 m. Object-collision rates remain effectively unchanged, reaching 0.0015 for both checkpoints at 3 s. The large improvement in unseen-source critical scenarios, therefore, does not result in a measurable degradation of conventional nuScenes performance.

The final experiment evaluates whether the behaviour learned from the generated StreetGaussians scenarios transfers to an independently constructed closed-loop environment. NeuroNCAP [25] is selected for this purpose because it provides a photorealistic closed-loop safety benchmark based on NeuRAD reconstructions and has also been adopted by several recent autonomous-driving methods, including BridgeAD [43], VaVAM [2], and Impromptu VLA [8], making it a useful common reference for evaluat-

Table 7. Corrected closed-loop NeuroNCAP results using the standard UniAD inference configuration. Overall collision rate follows the official category-level aggregation. A lower collision rate and a higher NeuroNCAP score are better.

Model	Frontal	Side	Stationary	Overall	Score
Original UniAD	**90.2**	**76.8**	33.1	66.70	1.824
Fine-tuned UniAD	94.0	78.4	**23.1**	**65.17**	**1.937**

ing collision-avoidance behaviour. Its reconstructed scenes and predefined collision trajectories are not used during either scenario generation or fine-tuning in this work, providing an evaluation setting independent of the proposed StreetGaussians pipeline. The objective is not to optimise specifically for NeuroNCAP, but to determine whether training an existing end-to-end planner with the proposed synthetic data produces measurable behavioural changes in an established closed-loop benchmark outside the generation pipeline.

Both checkpoints are evaluated using the same standard UniAD inference configuration, including its default occupancy-based collision optimiser. Following the official NeuroNCAP aggregation procedure, collision rates are first averaged within the frontal, side, and stationary categories, after which the category-level results are averaged to obtain the overall result. During verification of the evaluation pipeline, a collision-handling issue affecting several scenes that contained an excavator or a barrier was identified and corrected. All results reported here use the corrected evaluation; the issue and its verification are documented in Appendix D.3.

Fine-tuning reduces the official overall collision rate from 66.70% to 65.17% and increases the NeuroNCAP score from 1.824 to 1.937. The aggregate improvement is modest compared with the large open-loop gains on the unseen-source set, and the category-level results show that the effect is not uniform. In particular, stationary collisions decrease substantially from 33.1% to 23.1%, whereas frontal and lateral collision rates increase slightly. The generated supervision, therefore, transfers to closed-loop behaviour, but the learned avoidance strategy does not generalise equally across interaction geometries.

To investigate this asymmetry, we compare the same-seed closed-loop trajectories for which the two checkpoints produce different outcomes while keeping the complete inference pipeline fixed. The strongest discriminator is the magnitude of the lateral avoidance response. In `stationary-0278`, where the fine-tuned checkpoint succeeds when the baseline fails, the fine-tuned model produces the larger lateral manoeuvre in all 44 matched runs, with a mean peak lateral offset of $4.87\,\mathrm{m}$ compared with $1.15\,\mathrm{m}$ for the baseline. Conversely, across 58 matched frontal and lateral runs in which the fine-tuned checkpoint performs worse, the baseline produces the larger lateral displacement in 54 cases (93%). The fine-tuned model also brakes more frequently in many of these interactions, although this behaviour is not universal. Insufficient lateral commitment, therefore, emerges as the dominant failure mode in the difficult frontal and lateral cases. The complete trajectory-level analysis is provided in Appendix D.4.

5. Discussion, Limitations, and Future Work

Despite the significant improvement in unseen-source-generated scenarios, the corresponding closed-loop NeuroNCAP gain is considerably smaller and varies across interaction types. This discrepancy highlights an important distinction between open- and closed-loop evaluation. In open-loop evaluation, UniAD predicts a future trajectory from a fixed point in the expert trajectory and is evaluated against the generated collision-avoidance supervision. In closed-loop operation, only part of each predicted trajectory is executed before new observations are obtained and the planner is invoked again. Each decision, therefore, changes the subsequent ego state and sensor observations, allowing small differences in the initial response to propagate through successive replanning steps. Accurately predicting the desired avoidance trajectory in open loop does not consequently guarantee that the same behaviour will remain effective throughout a closed-loop interaction.

The trajectory-level analysis in Sec. 4.4 and Appendix D.4 confirms that this discrepancy reflects a genuine change in avoidance strategy rather than a failure of the frozen perception representation. Lateral commitment emerges as the principal behavioural difference between the checkpoints, while increased braking accompanies some, but not all, fine-tuned responses. The generated supervision therefore changes the learned planning behaviour, but the resulting strategy does not generalise uniformly across interaction geometries.

A further factor is that open-loop evaluation can partially mask this lateral-commitment failure. Predictions are generated from states along the expert avoidance trajectory, meaning that at later evaluation frames, the ego vehicle may already be laterally displaced and committed to the manoeuvre. In those cases, the planner is primarily required to continue an avoidance behaviour that has already begun, rather than decide when to initiate it. Closed-loop evaluation is more demanding because the planner must independently trigger and maintain the manoeuvre from its own evolving state. Consequently, insufficient or delayed lateral commitment can have a much larger effect in closed loop than is reflected by open-loop trajectory error alone.

The remaining open–closed-loop gap is unlikely to be explained solely by missing interaction types in the synthetic training data, since the fine-tuning set already contains frontal and lateral collision scenarios together with

corresponding steering- and braking-based avoidance trajectories. Instead, the discrepancy highlights the greater difficulty of closed-loop generalisation. The amount of synthetic supervision used in this work is comparatively limited: only 100 critical scenarios are used for fine-tuning, and the adaptation procedure deliberately makes relatively small changes to a pretrained UniAD model to preserve its nominal driving capability. The remaining closed-loop limitations may therefore reflect a combination of the planner architecture, the quantity and diversity of adaptation data, and the selected fine-tuning configuration rather than an inability of the proposed framework to generate the required avoidance behaviours. Isolating the contribution of each factor would require controlled ablations over dataset size, data composition, fine-tuning strategy, and downstream planner.

NeuroNCAP should therefore be interpreted as the most demanding generalisation test evaluated in this work. Unlike the unseen-source synthetic evaluation, which uses previously unseen environments but follows the same scenario-generation process as the training data, NeuroNCAP introduces both an independent NeuRAD-based rendering pipeline and a closed-loop feedback process in which the planner's actions determine its future observations. The purpose of this evaluation is not to optimise UniAD specifically or to establish a new state-of-the-art NeuroNCAP result, but to test whether supervision produced by the proposed framework changes the behaviour of an existing structured end-to-end planner outside the generation pipeline. More recent planning architectures and training paradigms have begun to address limitations relevant to closed-loop driving through improved use of temporal information [43], large-scale video pretraining [2], and targeted training on challenging long-tail scenarios [8]. Evaluating the same generated scenarios with stronger planners, therefore, represents an important next step and would help separate limitations of the generated data from limitations of the particular downstream model used in this work.

Rendering quality remains an important limitation of the current framework. Although viewpoint-aware actor selection and Fixer restoration are used to mitigate novel-view artefacts, the decrease in target-actor recall at larger distances and the need for post-generation quality control show that reconstruction quality still limits which generated variants can be retained. A particularly important limitation is the viewpoint dependence of the reconstructed threat actors. Each rigid vehicle is reconstructed only from the viewpoints available in the original nuScenes sequence, and its appearance can therefore become incomplete when the generated interaction exposes previously unobserved sides of the vehicle. The viewpoint-aware actor-selection procedure mitigates this problem by matching the viewing requirements of the generated trajectory to the actor's observed visibility profile, but this also restricts which scene-native actors can be used for a particular interaction geometry. Consequently, the available viewpoint coverage of the recorded traffic limits both the visual fidelity and the diversity of critical scenarios that can be generated from a given source scene.

A promising extension is to supplement scene-native rigid actors with independently reconstructed vehicle assets when the required viewpoints are not sufficiently covered. Several existing works provide such 3DGS vehicle resources [15, 23, 38], and SceneCrafter [16] already demonstrates the use of external 3DRealCar vehicles within reconstructed driving scenes. A hybrid strategy could therefore prioritise native actors and use external vehicle assets only as a fallback when viewpoint coverage is insufficient.

More recent reconstruction methods could additionally reduce the underlying viewpoint limitation. For example, OmniRe [7] extends Gaussian-based driving-scene reconstruction beyond rigid vehicles to pedestrians, cyclists, and other non-rigid actors while achieving improved reconstruction quality, which could both increase novel-view fidelity and broaden the range of critical interactions supported by the framework. Likewise, newer generative reconstruction methods such as ArtiFixer [10] improve extrapolation into poorly observed or entirely unseen regions and could provide a stronger alternative to the current Fixer-based restoration stage. Domain-specific adaptation of the restoration model may also be beneficial: the automotive version of Fixer is trained on an in-house real-driving-scene dataset rather than nuScenes, so fine-tuning it directly on nuScenes reconstruction artefacts could improve restoration of the camera observations encountered by the proposed pipeline. Combining stronger reconstruction backbones, complete external vehicle assets, and driving-domain-specific restoration therefore represents a promising direction for improving visual quality, increasing the usable scenario yield, and extending the framework to additional road-user classes.

The current quality-control stage already automates geometric validation, viewpoint-based actor filtering, and perception-based target verification, but manual inspection is still used to identify a small subset of severe rendering artefacts and scene inconsistencies. A natural extension is to remove this remaining manual stage by combining the existing checks with automatic perceptual-quality measures, actor-projection consistency tests, and learned artefact detection. This would enable fully automatic filtering and further improve the scalability of the generation pipeline.

More generally, the present work establishes that automatically generated critical scenarios can provide effective supervision for modifying an end-to-end planner, but it does not attempt to optimise every component of the resulting training system. The logical next step is therefore to jointly investigate the planner, the fine-tuning strategy, the gener-

ated data distribution, and the renderer. A stronger planner could be adapted by using a larger and more diverse set of generated critical interactions, while improved reconstruction could increase visual fidelity and reduce the need for post-generation filtering. Fine-tuning itself could be studied more systematically by varying learning rates, training durations, data mixing ratios, module freezing strategies, and curriculum designs.

Ultimately, the generation framework could be incorporated into an iterative, failure-driven training loop in which synthetic critical scenarios are used not only for training but also to discover planner weaknesses. By generating and evaluating a broad range of interaction geometries across different reconstructed scenes, threat actors, conflict locations, approach directions, and timings, the framework could provide a substantially wider closed-loop search space than fixed sets of hand-crafted benchmark scenarios such as those used by NeuroNCAP. Failures observed during this synthetic closed-loop evaluation could then be grouped according to the underlying interaction geometry or behavioural deficiency, after which additional scenarios targeting the identified weakness could be generated for retraining. Repeating this process would allow both the evaluation distribution and the training data to evolve in response to observed safety failures, extending the proposed framework from static critical-scenario generation to an iterative system for discovering and addressing weaknesses in autonomous-driving planners.

6. Conclusion

This work presented a reconstruction-based framework to convert ordinary nuScenes recordings into photorealistic, safety-critical scenarios for end-to-end driving evaluation and training. The framework combines editable scene reconstruction, automated threat-actor selection, controllable collision-course and avoidance-trajectory generation, novel-view rendering and restoration, quality control, and conversion to a UniAD-compatible representation.

The experiments provide largely positive answers to the four research questions posed in Sec. 1. First, the framework can generate critical interactions in which manipulated vehicles rendered using 3D Gaussian Splatting remain perceptually usable during the safety-critical portion of the scenario, although reconstruction quality remains dependent on the available viewpoints. Second, the generated scenarios provide challenging, largely geometrically plausible, and kinematically feasible test cases that expose substantial failure modes in pretrained UniAD. Third, the framework can efficiently generate multiple critical-scenario variants from reconstructed real-world scenes, covering different interaction geometries and avoidance behaviours while requiring comparatively little additional computation once the scene reconstruction is available. Finally, fine-tuning on the generated scenarios substantially improves performance on unseen generated critical interactions while preserving conventional nuScenes driving performance. The independent closed-loop evaluation further reveals a substantial open–closed-loop generalisation gap: although fine-tuning produces strong improvements on held-out generated scenarios, these gains transfer only partially to closed-loop collision avoidance. The trajectory-level analysis indicates that successfully initiating and maintaining sufficiently strong evasive manoeuvres under repeated replanning remains a key limitation.

Overall, the results demonstrate that reconstruction-based critical-scenario generation can provide useful and controllable supervision for autonomous-driving models while also enabling systematic safety evaluation beyond fixed sets of manually constructed scenarios. Future work should combine larger and more diverse generated datasets with improved fine-tuning strategies, stronger downstream planners, more capable rendering methods, and fully automated quality control. More broadly, integrating the framework into an iterative, closed-loop process in which generated scenarios are used to identify planner weaknesses and subsequently produce targeted training data represents a promising direction toward more systematic improvement in autonomous driving safety.

References

[1] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst, 2018. 2

[2] Florent Bartoccioni, Elias Ramzi, Victor Besnier, Shashanka Venkataramanan, Tuan-Hung Vu, Yihong Xu, Loïck Chambon, Spyros Gidaris, Serkan Odabas, David Hurych, Renaud Marlet, Alexandre Boulch, Mickaël Chen, Éloi Zablocki, Andrei Bursuc, Eduardo Valle, and Matthieu Cord. VaViM and VaVAM: Autonomous driving through video generative modeling. *arXiv preprint arXiv:2502.15672*, 2025. 4, 11, 13

[3] Daniel Bogdoll, Yitian Yang, Tim Joseph, Melih Yazgan, and J. Marius Zollner. Muvo: A multimodal generative world model for autonomous driving with geometric representations. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, page 2243–2250. IEEE, 2025. 2

[4] Alessandro Bonetti, Simone Guidetti, and Lorenzo Sabattini. Improved path planning algorithms for non-holonomic autonomous vehicles in industrial environments with narrow corridors: Roadmap hybrid a* and waypoints hybrid a*. In *2023 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2023. 6

[5] 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, 2020. 1, 17

[6] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan, 2021. 4

[7] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, Li Song, and Yue Wang. Omnire: Omni urban scene reconstruction, 2025. 5, 13, 17

[8] Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, Leichen Wang, Xingtao Hu, Hao Sun, Hang Zhao, and Hao Zhao. Impromptu vla: Open weights and open data for driving vision-language-action models. In *Advances in Neural Information Processing Systems*, 2025. 3, 4, 11, 13

[9] 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. 2

[10] Riccardo de Lutio, Tobias Fischer, Yen-Yu Chang, Yuxuan Zhang, Jay Zhangjie Wu, Xuanchi Ren, Tianchang Shen, Katarina Tothova, Zan Gojcic, and Haithem Turki. Artifixer: Enhancing and extending 3d reconstruction with auto-regressive diffusion models, 2026. 13

[11] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Practical search techniques in path planning for autonomous driving. In *Proceedings of the First International Symposium on Search Techniques in Artificial Intelligence and Robotics (STAIR-08)*, pages 32–37, Chicago, IL, USA, 2008. AAAI Press. 6

[12] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, 29(5):485–501, 2010. 6

[13] Zeyu Dong, Yimin Zhu, Yu Wu, and Yu Sun. Frost-drive: Scalable and efficient end-to-end driving with a frozen vision encoder. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 1729–1738, 2026. 9

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017. 1, 3

[15] Xiaobiao Du, Yida Wang, Haiyang Sun, Zhuojie Wu, Hongwei Sheng, Shuyun Wang, Jiaying Ying, Ming Lu, Tianqing Zhu, Kun Zhan, and Xin Yu. 3drealcar: An in-the-wild rgb-d car dataset with 360-degree views, 2025. 13

[16] Junhao Ge, Zuhong Liu, Longteng Fan, Yifan Jiang, Jiaqi Su, Yiming Li, Zhejun Zhang, and Siheng Chen. Unraveling the effects of synthetic data on end-to-end autonomous driving, 2025. 3, 8, 13, 29

[17] Huasong Han, Kaixuan Zhou, Xiaoxiao Long, Yusen Wang, and Chunxia Xiao. Ggs: Generalizable gaussian splatting for lane switching in autonomous driving, 2024. 19

[18] Xiangyu Han, Zhen Jia, Boyi Li, Yan Wang, Boris Ivanovic, Yurong You, Lingjie Liu, Yue Wang, Marco Pavone, Chen Feng, and Yiming Li. Extrapolated urban view synthesis benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28718–28728, 2025. 19

[19] 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, 2023. 2, 3

[20] 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 (CVPR)*, pages 17853–17862, 2023. 1, 2, 4

[21] 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 (ICCV)*, pages 8340–8350, 2023. 2, 4

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 3

[23] Congcong Li, Jin Wang, Xiaomeng Wang, Xingchen Zhou, Wei Wu, Yuzhi Zhang, and Tongyi Cao. Car-gs: Addressing reflective and transparent surface challenges in 3d car reconstruction, 2025. 13

[24] Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. DrivingDiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. In *European Conference on Computer Vision (ECCV)*, pages 469–485. Springer, 2024. 2, 3

[25] 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 (ECCV)*, pages 161–177, 2024. 1, 9, 11

[26] Xin Ma, Jiguang Zhang, Peng Lu, Shibiao Xu, and Chengwei Pan. Novel view synthesis under large-deviation viewpoint for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6000–6008, 2025. 19

[27] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 3

[28] Dušan Nemec, Michal Gregor, Emília Bubeníková, Marián Hruboš, and Rastislav Pirník. Improving the hybrid a* method for a non-holonomic wheeled robot. *International Journal of Advanced Robotic Systems*, 16(1), 2019. 6

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

action prediction for generalizable autonomous driving in the long tail, 2026. 4

[30] Stefanos Pasios and Nikos Nikolaidis. Carla2real: A tool for reducing the sim2real appearance gap in carla simulator. *IEEE Transactions on Intelligent Transportation Systems*, 26 (11):18747–18761, 2025. 3

[31] Xuanchi Ren, Yifan Lu, Tianshi Cao, Ruiyuan Gao, Shengyu Huang, Amirmojtaba Sabour, Tianchang Shen, Tobias Pfaff, Jay Zhangjie Wu, Runjian Chen, Seung Wook Kim, Jun Gao, Laura Leal-Taixe, Mike Chen, Sanja Fidler, and Huan Ling. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models, 2025. 2

[32] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control, 2019. 2

[33] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations, 2020. 4

[34] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles, 2017. 1, 3

[35] Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation, 2024. 2

[36] 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 (CVPR)*, pages 14895–14904, 2024. 2, 3

[37] United Nations Economic Commission for Europe. UN Regulation No. 13-H: Uniform Provisions Concerning the Approval of Passenger Cars with Regard to Braking. UN Regulation 13-H, United Nations Economic Commission for Europe. 22

[38] Muzaffer Arda Uslu and Selcan Kaplan Berkaya. A multistage pipeline for 3d reconstruction of vehicles from heterogeneous multi-view images. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi*, 42(2), 2026. 13

[39] Yuqing Wen, Yucheng Zhao, Yingfei Liu, Fan Jia, Yanhui Wang, Chong Luo, Chi Zhang, Tiancai Wang, Xiaoyan Sun, and Xiangyu Zhang. Panacea: Panoramic and controllable video generation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6902–6912, 2024. 2, 3

[40] 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, 2025. 3, 5, 7, 23

[41] 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*, pages 6119–6132, 2022. 2

[42] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *European Conference on Computer Vision (ECCV)*, pages 139–155. Springer, 2024. 2, 3, 4, 5

[43] Bozhou Zhang, Nan Song, Xin Jin, and Li Zhang. Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6854–6863, 2025. 2, 4, 11, 13

[44] Dapeng Zhang, Zhenlong Yuan, Zhangquan Chen, Chih-Ting Liao, Yinda Chen, Fei Shen, Qingguo Zhou, and Tat-Seng Chua. Reasoning-VLA: An efficient and spatial-guided general vision-language-action reasoning model for autonomous driving. In *Proceedings of the 43rd International Conference on Machine Learning*. PMLR, 2026. 4

[45] Yinuo Zhao, Kun Wu, Zhiyuan Xu, Zhengping Che, Qi Lu, Jian Tang, and Chi Harold Liu. Cadre: A cascade deep reinforcement learning framework for vision-based autonomous urban driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3481–3489, 2022. 9

[46] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes, 2024. 5

Table 8. Main StreetGaussians reconstruction settings used in this work.

Setting	Value
Training iterations	30,000
Input frequency	10 Hz
Cameras	6 surround-view
Image downscale	$3\times$
Effective image resolution	$\approx 533 \times 300$
Dataset split	nuScenes trainval
SMPL reconstruction	Disabled
Moving-only RigidNodes	Disabled
Maximum points per instance	5,000
Rigid trajectory threshold	1.0
SH degree	3

A. StreetGaussians Reconstruction and Object Manipulation

A.1. Reconstruction Configuration

Each selected nuScenes scene is reconstructed independently using the StreetGaussians implementation provided by DriveStudio [7]. The reconstruction uses all six surround-view cameras and the 10 Hz processed nuScenes sequences. The native 1600×900 camera images are loaded with a factor-three downscale, resulting in an effective training resolution of approximately 533×300 pixels. The complete temporal extent of each selected scene is used for reconstruction.

The resolved configuration used throughout this work is summarised in Table 8. Most reconstruction and optimisation parameters follow the current DriveStudio StreetGaussians configuration. The main dataset-level changes are the use of the nuScenes `trainval` split, disabling SMPL-based non-rigid reconstruction, since we do not reconstruct pedestrians, and allowing both moving and stationary rigid vehicles to be represented as RigidNodes to allow more candidates in the threat actor selection process.

A.2. Rigid-Object Representation

StreetGaussians represents the static environment separately from rigid traffic objects. Each reconstructed rigid instance is represented by an individual RigidNode containing its Gaussian geometry and appearance together with its pose over time. In contrast to the default DriveStudio configuration, `only_moving=false` is used in this work, allowing stationary as well as moving eligible vehicles to be reconstructed as individually manipulable objects.

Internally, the Gaussians belonging to the rigid-object representation carry an object identifier through the `point_ids` tensor. All Gaussians associated with the same RigidNode therefore share the same instance identifier, allowing the Gaussian subset belonging to a particular vehicle to be isolated independently of other objects and the static background.

For the proposed pipeline, the original nuScenes `instance_token` is additionally associated with the corresponding RigidNode index during preprocessing. This provides the connection between actors selected from the original nuScenes annotations and their Gaussian representation in the reconstructed scene, so the visibility of the available rigid objects can be assessed directly from the nuScenes API without requiring the reconstruction of the scene.

A.3. Actor Manipulation

Once the RigidNode corresponding to a selected traffic participant has been identified, its recorded pose sequence can be replaced by poses generated by the critical-scenario planner. The Gaussian geometry and appearance of the vehicle remain unchanged; only its placement and orientation over time are modified. The conversion from the planar planner output to the 3D pose sequence required by the renderer is described in Sec. A.4.

Actors that should not appear in the generated interaction are suppressed rather than permanently removed from the trained reconstruction. The opacity of the Gaussian subset associated with the corresponding RigidNode is reduced to effectively zero during rendering. The identities of suppressed actors are retained so that their annotations can also be excluded when constructing the corresponding UniAD-compatible ground truth, maintaining consistency between the rendered observations and their labels.

A.4. Trajectory-to-Render Pose Construction

Because the number of poses returned by the critical-scenario planner generally differs from the number of frames in the rendering interval, the generated planar trajectory (x, y, ψ) is resampled to the required frame count. The original planner poses and the required rendering-frame poses are each parameterised uniformly over the normalised interval $[0, 1]$, and (x, y, ψ) are interpolated piecewise linearly at the rendering-frame locations. This operation changes only the sampling density of the already generated path rather than introducing additional path geometry. Before interpolation, the yaw sequence is unwrapped to eliminate discontinuities caused by its periodic 2π representation.

Since the generated trajectory contains no vertical coordinate, a vertical position is estimated independently for every generated (x, y) location so that the actor follows changes in the road elevation. The nuScenes Velodyne HDL-32E contains 32 beams with approximately 1080 returns per ring, corresponding to roughly 34,500 points in a raw LiDAR sweep [5]. For each trajectory location, LiDAR points within a radius of 1.5 m are selected and used

to estimate the local road surface. If fewer than 20 points are available within this neighbourhood, the local surface is considered insufficiently sampled; such sparse observations typically occur at larger distances or in partially occluded regions. In this case, the neighbourhood is expanded by selecting the 150 nearest LiDAR points in the horizontal plane. The candidate heights are sorted, the lowest 20% are retained, and the median of this subset is used as a robust estimate of the local road height, reducing the influence of returns from vehicles, vegetation, and other structures above the road surface. The vertical position of the actor centre is then obtained by adding a center-to-ground offset estimated from the actor's original recorded pose when available, with half of the annotated bounding-box height used as a lower bound. Repeating this procedure along the trajectory produces a sequence (x, y, z, ψ) that follows the local road elevation and defines the 3D poses subsequently applied to the corresponding RigidNode.

A.5. Coordinate Convention

The reconstruction involves four relevant coordinate frames: the global nuScenes frame $\mathcal{W}$, the local DriveStudio reconstruction frame $\mathcal{L}$, the ego-vehicle frame $\mathcal{E}_t$, and the individual camera frames $\mathcal{C}_i$. The generated ego and threat trajectories are defined in the global nuScenes frame, whereas the trained Gaussian scene and its RigidNodes are expressed in the local reconstruction frame $\mathcal{L}$, whose origin is defined by the initial front-camera pose. Consequently, generated poses must be transformed from $\mathcal{W}$ into $\mathcal{L}$ before being applied to the reconstruction. For ego rendering, the fixed extrinsic transform between each camera $\mathcal{C}_i$ and the ego vehicle $\mathcal{E}$ is retained, such that all six cameras follow the generated ego trajectory while preserving their original relative positions and orientations. For camera i, the corresponding local rendering pose is computed as

$$\mathbf{T}^{\mathcal{L}}_{\mathcal{C}_i,t} = \left(\mathbf{T}^{\mathcal{W}}_{\mathcal{L}}\right)^{-1} \mathbf{T}^{\mathcal{W}}_{\mathcal{E},t} \mathbf{T}^{\mathcal{E}}_{\mathcal{C}_i}, \tag{1}$$

where $\mathbf{T}^{\mathcal{W}}_{\mathcal{E},t}$ is constructed from the generated ego pose (x_t, y_t, ψ_t), $\mathbf{T}^{\mathcal{E}}_{\mathcal{C}_i}$ is the fixed camera-to-ego extrinsic transform, and $\mathbf{T}^{\mathcal{W}}_{\mathcal{L}}$ denotes the world pose of the initial front camera that defines the reconstruction origin. Generated threat-actor poses are similarly represented in the local reconstruction frame before being applied to the corresponding RigidNode.

B. Conflict-Pose Construction and Threat-Actor Selection Details

B.1. Conflict-Pose and Start-Pose Sampling

The conflict pose is sampled directly from the recorded future ego trajectory. A nuScenes keyframe index k_c is drawn uniformly from the interval $[10, 30]$, corresponding to 5–15 s after the beginning of the scenario at the 2 Hz nuScenes annotation frequency, in order for the ego to have a meaningful time window to react within the scene limits. The ego pose at this keyframe defines the conflict pose $\mathbf{q}_c = (x_c, y_c, \psi_c)$, while the corresponding time-to-conflict is $t_c = 0.5k_c$. Selecting the conflict location from the recorded ego trajectory ensures that the generated interaction is constructed around a position that the nominal ego vehicle actually reaches at the nuScenes annotation frequency.

The local map is rasterised into an 80×80 binary drivable-area grid, corresponding to a spatial resolution of 2.5 m per pixel. This resolution is chosen as a compromise between map fidelity and computational efficiency: it preserves the coarse road topology required for feasible trajectory generation while reducing the Hybrid A* search space. The drivable-area mask is partitioned using 8-connected component labelling, and candidate threat starts are restricted to the component containing the conflict point. This ensures that the sampled start position is connected to the conflict region through the drivable area rather than lying on a disconnected road segment. Non-drivable cells and the boundary of the local map patch are represented as obstacles for the subsequent planner.

Candidate threat positions are randomly sampled from the valid drivable-area points, with at most 1,000 candidates evaluated per scenario. Let $\mathbf{p}_s = (x_s, y_s)$ and $\mathbf{p}_c = (x_c, y_c)$ denote the sampled start and conflict positions. To ensure that the threat can reach the conflict point within the available time at a plausible urban speed, the Euclidean start-to-conflict distance must satisfy

$$v_{\min} t_c \le \|\mathbf{p}_c - \mathbf{p}_s\|_2 \le v_{\max} t_c, \tag{2}$$

where $v_{\min} = 15$ km/h and $v_{\max} = 30$ km/h. This range is chosen to remain representative of the low-speed urban driving conditions in nuScenes, for which reported mean ego speeds are approximately 16–18 km/h, while allowing sufficient variation to construct more dynamic critical interactions. The initial threat heading is directed towards the conflict point,

$$\psi_s = \operatorname{atan2}(y_c - y_s,\, x_c - x_s). \tag{3}$$

To suppress approximately parallel, same-direction interactions, which are more likely to result in rear-end rather than ego-avoidance scenarios, the difference between this approach direction and the ego heading ψ_c at the conflict point must be at least 70°. Finally, the complete start pose $\mathbf{q}_s = (x_s, y_s, \psi_s)$ is checked using the same vehicle-footprint collision model employed by the trajectory planner and is rejected if it already overlaps an obstacle. The first sampled pose satisfying all criteria is retained; if no valid pose is found after 1,000 attempts, the scenario is discarded.

B.2. Visibility-Profile Construction and Matching

Because the reconstructed appearance of each RigidNode depends on the viewpoints from which it was observed in the original sequence, the framework estimates the viewpoint coverage available for every candidate actor. For each frame in which the ego vehicle and candidate actor are simultaneously present, a geometric line-of-sight test is performed between their positions. The line segment is rasterised onto the local drivable-area mask, and the observation is considered visible only when all traversed cells remain within the drivable region and the line of sight is not occluded by another annotated vehicle. For the occlusion test, the ground-plane footprints of surrounding vehicles are expanded by $0.5\,\mathrm{m}$ before the intersection is evaluated to take into account misalignments of the nuScenes bounding boxes. This procedure provides a lightweight geometric approximation of viewpoint availability rather than an image-based visibility estimate.

For every visible observation, the relative bearing from the candidate actor towards the ego vehicle is expressed in the actor's local heading frame. The resulting angle is assigned to one of eight discrete viewing directions: front, front-left, left, rear-left, rear, rear-right, right, and front-right. Accumulating these observations across the sequence produces an eight-bin visibility histogram for each reconstructed actor. The same procedure is subsequently applied to the generated ego–threat interaction, producing a second histogram that describes the viewpoints required to render the selected threat trajectory.

To compare the available and required viewpoint coverage, viewing directions accounting collectively for the dominant observations of the generated threat trajectory are first identified. A direction is considered required when it contributes at least 25% of the synthetic visibility observations. Candidate actors that do not contain observations from every required direction are rejected. For the remaining actors, the candidate and synthetic histograms are normalised over the required directions, and their difference is measured using the L_1 distance

$$d_{\mathrm{vis}} = \sum_{b \in \mathcal{B}_{\mathrm{req}}} \left| h_b^{\mathrm{syn}} - h_b^{\mathrm{cand}} \right|, \tag{4}$$

where $\mathcal{B}_{\mathrm{req}}$ denotes the set of required viewing directions and h_b^{syn} and h_b^{cand} are the corresponding normalised histogram values. A visibility similarity score is then defined as

$$S_{\mathrm{vis}} = \frac{1}{1 + d_{\mathrm{vis}}}. \tag{5}$$

Candidates with $S_{\mathrm{vis}} \geq 0.80$ are retained for the final threat-actor ranking described in Appendix B.3. If no candidate satisfies this threshold, the actor with the highest visibility similarity is retained as a fallback.

B.3. Final Threat-Actor Ranking

The visibility matching described in Appendix B.2 can retain multiple reconstructed actors with sufficiently compatible viewpoint coverage. A final ranking is therefore applied to select the actor whose original observation conditions most closely resemble those required by the generated critical interaction. This design is motivated by the limited extrapolation capability of driving-scene novel-view synthesis methods, whose rendering quality can deteriorate when the requested viewpoint differs substantially from the observations available during reconstruction [17, 18, 26]. Consequently, the ranking favours actors that were originally observed at distances and viewing directions similar to those required during the synthetic interaction.

The closest approach between the generated ego and threat trajectories is used as the reference for the ranking. This represents the most demanding part of the interaction because the threat is rendered at close range and therefore occupies a larger portion of the image. Let d^* denote the ego–threat distance at closest approach. For each candidate actor i, the minimum distance at which it was observed from the ego in the original sequence is denoted by d_i. Their difference is defined as

$$\Delta d_i = |d_i - d^*|. \tag{6}$$

Since the generated scenarios intentionally create near-collision interactions, d^* is generally small. The distance term therefore primarily favours reconstructed actors that were also observed at close range in the original sequence, reducing the amount of scale and viewpoint extrapolation required when the actor is rendered near the ego vehicle.

The relative bearing at closest approach, β^*, is additionally compared with the bearings from which each candidate actor was observed in the original sequence. Let $\mathcal{B}_i$ denote the set of recorded ego-relative bearings for candidate i. The bearing discrepancy is defined as

$$\Delta\beta_i = \min_{\beta \in \mathcal{B}_i} \left| \mathrm{wrap}(\beta - \beta^*) \right|, \tag{7}$$

where the angular difference is wrapped to account for the periodicity of the bearing representation. A small $\Delta\beta_i$ therefore indicates that the actor was previously observed from a direction similar to that required during the closest part of the generated interaction.

The final candidate score combines these two geometric discrepancies with the actor's visibility quality and number of available observations,

$$J_i = 2.0\,\Delta d_i + 0.05\,\Delta\beta_i - 0.5\,\bar{V}_i - 0.05\,\min(N_i, 40), \tag{8}$$

where $\bar{V}_i$ denotes the candidate's average visibility measure and N_i its number of valid observations. Lower scores are preferred. The coefficients are heuristic weighting factors

chosen to encode the desired priority of the ranking rather than parameters adopted from prior work. Distance agreement is assigned the greatest influence, since the generated interaction requires the selected actor to be rendered at close range. Bearing agreement provides a secondary preference for similar viewing direction, while the visibility and observation-count terms favour actors supported by stronger reconstruction evidence. The observation contribution is capped at 40 frames so that actors with long tracks do not dominate the ranking solely because they appear more frequently.

Since this ranking is applied only after the visibility-compatibility stage of Appendix B.2, it is used to distinguish between actors that already provide suitable viewpoint coverage rather than to determine actor suitability by itself. The candidate with the lowest score is selected as the final threat actor.

C. Critical-Scenario Generation Details

C.1. Hybrid A* Configuration

The Hybrid A* implementation used for all generated scenarios discretises the continuous vehicle state for search while propagating each motion primitive using a car-like kinematic model. Each search state is represented by the vehicle pose (x, y, ψ) together with its current driving direction and steering command. Candidate successors are generated by applying a discrete set of steering inputs in both forward and reverse motion. During the search, an analytic Reeds-Shepp expansion is additionally attempted from the current state to the goal pose. If the resulting connection satisfies the vehicle's turning constraints and is collision-free, it is accepted as the terminal portion of the trajectory and the search terminates; otherwise, Hybrid A* continues expanding kinematically feasible motion primitives. This follows the structure of the open-source Hybrid A* implementation used as the basis of the planner, which combines kinematic and Reeds–Shepp expansions with holonomic and non-holonomic search costs.

Search nodes are prioritised according to

$$f(n) = g(n) + w_h h_{\mathrm{hol}}(n), \tag{9}$$

where $g(n)$ is the accumulated motion cost and $h_{\mathrm{hol}}(n)$ is an obstacle-aware holonomic cost-to-go from the current position to the goal. The latter is obtained from a two-dimensional A* search over the static obstacle map and therefore guides the vehicle around obstacles without itself imposing the non-holonomic constraints considered by the Hybrid A* state expansion.

For a motion primitive of length Δs, steering angle δ_k, and driving direction $d_k \in \{-1, +1\}$, the incremental motion cost can be written as

$$\Delta g = w_d \Delta s + w_{\mathrm{gear}} \mathbb{I}[d_k \neq d_{k-1}] + w_\delta |\delta_k| + w_{\Delta\delta} |\delta_k - \delta_{k-1}|, \tag{10}$$

where $w_d = 1$ for forward motion and $w_d = w_{\mathrm{rev}}$ for reverse motion. The remaining terms penalise changes in driving direction, large steering angles, and changes in steering between successive primitives. The resulting objective therefore favours short forward-moving trajectories with few gear changes and comparatively smooth steering, while the holonomic heuristic encourages efficient progress towards the goal through the available drivable region.

Table 9. Hybrid A* configuration used for critical-scenario generation.

Parameter	Value	Role
XY_{reso}	2.0 m	Position discretisation
ψ_{reso}	15°	Heading discretisation
Move step	0.4 m	Primitive propagation step
Steering samples	10	Steering choices per expansion
Collision step	5	Collision-check interval
Search margin	1 cell	Grid-bound extension
Wheelbase	3.5 m	Vehicle kinematics
Front extent	4.5 m	Rear axle to front bumper
Rear extent	1.0 m	Rear axle to rear bumper
Vehicle width	3.0 m	Collision footprint
Max. steering	0.6 rad	Steering constraint
Gear-change cost	200	Direction-change penalty
Reverse cost	100	Reverse-motion multiplier
Steering-change cost	50	Steering variation penalty
Steering-angle cost	20	Steering magnitude penalty
Heuristic weight	5	Holonomic heuristic weight
Planning timeout	15 s	Maximum search time

The configuration used by the batch-generation pipeline is summarised in Table 9. These values correspond to the planner configuration that produced the trajectories evaluated in this work.

The first group of parameters controls search resolution and computational efficiency. The spatial and angular discretisation, motion-primitive step, collision-check interval, and number of steering samples were selected to provide sufficient trajectory fidelity while keeping individual planning attempts short, which is important for scalable scenario generation. The second group specifies the vehicle footprint used during collision checking. The footprint is deliberately conservative relative to a typical passenger vehicle, introducing additional clearance around obstacles and reducing the likelihood that paths passing very close to surrounding geometry are accepted as feasible. The final group contains the cost weights appearing in Eqs. 9 and 10. In particular, the high reverse and gear-change penalties strongly favour direct forward-driving trajectories and suppress parking-like manoeuvres, while the steering-related penalties discourage unnecessarily sharp or oscillatory motion.

For threat-trajectory generation, the sampled starting

pose $\mathbf{q}_s = (x_s, y_s, \psi_s)$ is connected to the fixed conflict position (x_c, y_c). Two alternative terminal orientations are considered,

$$\psi_g^{(1)} = \psi_s, \qquad \psi_g^{(2)} = \psi_c - \pi, \tag{11}$$

where ψ_c is the nominal ego heading at the conflict pose. The first hypothesis allows the planner to construct crossing-type interactions while approximately preserving the threat's approach orientation, whereas the second explicitly encourages an oncoming configuration by orienting the threat opposite to the ego vehicle. Hybrid A* is executed independently for both terminal poses. Solutions that reach the conflict region in reverse motion are rejected, and when both hypotheses produce valid forward trajectories, the lower-cost path is retained. Rear-end interaction geometries are subsequently excluded because the framework focuses on scenarios that require an active ego-avoidance response.

Each of the two goal-pose hypotheses is given a maximum planning time of 15 s. If no valid solution is found within this budget, the corresponding search is terminated rather than allowing a difficult planning instance to delay the wider scenario-generation process. If neither hypothesis produces a feasible moving threat trajectory, the sampled interaction is not immediately discarded; instead, the selected threat actor is retained as a stationary threat and the pipeline proceeds to ego collision-avoidance generation. This fallback increases the diversity of generated interactions while allowing otherwise suitable scene and actor configurations to remain usable when a kinematically feasible moving threat path cannot be constructed.

C.2. Threat Timing and Temporal Synchronisation

Hybrid A* produces a geometric path without an explicit temporal parameterisation. To construct a time-indexed threat trajectory, the selected path is resampled according to the conflict time defined by the recorded ego trajectory, let k_c denote the nuScenes keyframe containing the selected conflict pose. Since nuScenes keyframes are spaced by 0.5 s, the nominal time-to-conflict is

$$t_c = 0.5\, k_c. \tag{12}$$

The generated threat trajectory is represented at 10 Hz to match the temporal resolution used during scene reconstruction and rendering. The number of output samples is set to

$$N = 5(k_c + 1). \tag{13}$$

Rather than interpolating directly over the indices of the Hybrid A* output, the trajectory is parameterised by cumulative planar arc length. For a raw path containing positions $\mathbf{p}_i = (x_i, y_i)$, the cumulative distance is defined as

$$s_0 = 0, \qquad s_i = \sum_{k=1}^{i} \|\mathbf{p}_k - \mathbf{p}_{k-1}\|_2 . \tag{14}$$

The resampled trajectory is evaluated at N uniformly spaced arc-length locations,

$$s'_j = \frac{j}{N-1} s_{\text{end}}, \qquad j = 0, \ldots, N-1, \tag{15}$$

where s_{end} denotes the total length of the Hybrid A* path. The x and y coordinates are linearly interpolated as functions of cumulative arc length. Vehicle yaw is first unwrapped to avoid discontinuities at $\pm\pi$ and is subsequently interpolated using the same arc-length parameterisation.

Because successive output frames are uniformly spaced both in time and along cumulative path length, this procedure imposes an approximately constant path-length rate over the generated threat trajectory. In other words, Hybrid A* determines the spatial geometry of the interaction, while the subsequent resampling assigns a nominally constant speed such that the trajectory occupies the time interval associated with the selected conflict keyframe. The start-position constraints described in Appendix B.1 ensure that this implied speed remains within the intended urban operating range of 15–30 km/h.

The implementation uses $N = 5(k_c + 1)$ samples. At a 10 Hz sampling rate, the interval between the first and final samples is therefore

$$T_{\text{threat}} = (N-1)0.1 = t_c + 0.4\ \text{s}. \tag{16}$$

Consequently, the generated threat reaches the end of its planned path with a fixed 0.4 s offset relative to the nominal conflict timestamp. This discretisation effect is small relative to the 5–15 s interaction horizon and is explicitly evaluated in the trajectory-feasibility experiments.

This arc-length-based temporal construction is distinct from the later renderer-specific resampling described in Appendix A. The procedure here assigns the threat's progression along its generated path, whereas the subsequent step only converts an already generated trajectory into the pose sequence required by the rendering pipeline.

C.3. Ego Collision-Avoidance Generation

Once the moving or stationary threat configuration has been established, a corresponding safe ego response is generated. A steering-based collision-avoidance trajectory is first sought using Hybrid A*. If a valid steering trajectory is obtained, it is retained as the ego response. If steering-based avoidance cannot be generated, braking is considered only when stopping before the conflict region provides a valid collision-avoidance response. Moving oncoming head-on interactions require lateral avoidance, since stopping on the nominal route would leave the ego vehicle in the path of

the approaching threat; such interactions are therefore discarded if no valid steering-based response can be generated.

The original nuScenes ego trajectory is retained before the response begins, and the reaction point is determined spatially with respect to the selected conflict location rather than from a fixed time offset. Let $\mathbf{p}_c = (x_c, y_c)$ denote the ego position at the conflict keyframe and $\mathbf{p}_k = (x_k, y_k)$ the recorded ego position at frame k. The reaction frame is selected as

$$k_r = \max(0, k-1), \qquad k = \min\{j : \|\mathbf{p}_j - \mathbf{p}_c\|_2 < d_r\}, \tag{17}$$

where the reaction distance is $d_r = 25\,\text{m}$ for steering-based avoidance and $d_r = 15\,\text{m}$ for braking.

The different reaction distances reflect the distinct spatial requirements of the two response types. For braking, a reaction threshold of $15\,\text{m}$ is used together with a stopping threshold of $5\,\text{m}$. The latter identifies the point along the recorded ego trajectory at which the generated response should have reached zero speed, thereby maintaining a spatial margin before the conflict location. On a locally straight trajectory, these thresholds correspond to approximately $10\,\text{m}$ of available braking distance; the exact distance used by the implementation is subsequently computed from the cumulative arc length of the recorded ego trajectory. Under constant deceleration, the magnitude required to stop over distance D is

$$|a_{\text{req}}| = \frac{v_0^2}{2D}. \tag{18}$$

At urban ego speeds of approximately 16–18 km/h, this corresponds to approximately 0.99–$1.25\,\text{m/s}^2$ for $D = 10\,\text{m}$, while even at $30\,\text{km/h}$ the required deceleration is approximately $3.47\,\text{m/s}^2$. These values are within the braking capability expected of conventional passenger vehicles; for reference, UN Regulation No. 13-H specifies a mean fully developed deceleration of at least $6.43\,\text{m/s}^2$ for the Type-0 service-brake test [37]. The selected braking distance therefore provides a physically feasible stopping response over the urban-speed range considered in this work.

Steering-based avoidance is assigned a larger reaction distance because lateral displacement must be developed progressively before the conflict point. Providing additional distance before the interaction gives the planner more opportunity to construct a collision-free departure from the nominal route and reduces the need for abrupt steering close to the conflict region. The $25\,\text{m}$ steering threshold is therefore used as a practical design value rather than a universal safety limit.

For steering-based avoidance, a new Hybrid A* search is performed from the recorded ego pose at k_r. The terminal pose is deliberately selected on the original recorded ego trajectory after the conflict rather than at an independently sampled location. Starting immediately after the conflict keyframe, arc length is accumulated along the recorded trajectory until $25\,\text{m}$ has been travelled beyond the conflict location. The corresponding recorded position and heading define the rejoining goal pose. If fewer than $25\,\text{m}$ of recorded trajectory remain, the final available ego pose is used instead.

This rejoining constraint is an intentional part of the generated supervision. Rather than terminating the trajectory once the immediate collision has been avoided, the generated manoeuvre leaves the nominal route before the critical interaction and subsequently returns to the same recorded path after passing the threat. The supervision therefore contains both the evasive manoeuvre and the subsequent recovery phase, allowing the ego vehicle to return towards nominal driving and continue along its original route. The resulting trajectory captures the complete response from manoeuvre initiation through collision avoidance to post-interaction recovery.

To force the Hybrid A* search to avoid the threat with sufficient clearance, an additional rectangular keep-out region is inserted around the threat pose at the critical interaction. In a coordinate frame aligned with the threat heading, the region extends $1\,\text{m}$ ahead of the threat, $6\,\text{m}$ behind it, and $0.5\,\text{m}$ to either side. The asymmetric longitudinal extent provides a larger exclusion region behind the threat and reduces the likelihood that the generated ego path passes closely behind the collision-course vehicle. No explicit left- or right-swerve direction is prescribed; Hybrid A* selects whichever collision-free route satisfies the vehicle constraints and the combined static-obstacle and threat keep-out map.

The resulting steering-based path is parameterised at the $10\,\text{Hz}$ trajectory rate using a constant assigned speed. Because the discrete Hybrid A* motion primitives can introduce abrupt steering transitions, only the post-reaction portion of this trajectory is subsequently smoothed. The x and y coordinates and unwrapped yaw sequence are independently filtered using a Gaussian filter with $\sigma = 4$ frames, corresponding to $0.4\,\text{s}$ at $10\,\text{Hz}$. The first smoothed sample is reset to its original value,

$$\mathbf{q}_{k_r}^{\text{smooth}} = \mathbf{q}_{k_r}, \tag{19}$$

to preserve continuity with the unchanged recorded trajectory preceding the reaction point. The pre-reaction segment is not filtered.

When a valid steering-based avoidance trajectory cannot be obtained and the interaction can instead be safely resolved by stopping before the conflict region, a braking response is generated. Braking does not generate a new spatial path. Instead, the original ego trajectory is reparameterised in time so that the vehicle decelerates to a stop before entering the conflict region. In addition to the $15\,\text{m}$

reaction threshold, a stopping threshold of 5 m is used to identify the first recorded frame k_s sufficiently close to the conflict point. If no valid reaction frame is found before k_s, the reaction point falls back to

$$k_r = \max(0, k_s - 1). \tag{20}$$

Let $s(k)$ denote the cumulative arc length of the original ego trajectory, with

$$s_r = s(k_r), \qquad s_s = s(k_s), \qquad D = s_s - s_r. \tag{21}$$

The implementation estimates the initial motion from the recorded arc-length increment immediately before braking,

$$\Delta s_0 = s(k_r) - s(k_r - 1). \tag{22}$$

Since the trajectory is represented at 10 Hz, Δs_0 is the travelled distance per frame, corresponding to a physical initial speed of

$$v_0 = \frac{\Delta s_0}{\Delta t}, \qquad \Delta t = 0.1\,\mathrm{s}. \tag{23}$$

The number of frames required to stop under constant deceleration is then chosen as

$$N_{\mathrm{stop}} = \mathrm{round}\left(\frac{2D}{\Delta s_0}\right). \tag{24}$$

For frame offset $i = 1, \ldots, N_{\mathrm{stop}}$, the new arc-length position is

$$s_i = s_r + \Delta s_0 i - \frac{1}{2}\frac{\Delta s_0}{N_{\mathrm{stop}}} i^2. \tag{25}$$

This is the discrete-frame form of a constant-deceleration trajectory. Once N_{stop} frames have elapsed, the arc-length position is held fixed for the remainder of the sequence.

The new arc-length schedule is finally mapped back onto the original recorded trajectory by interpolating its x, y, and yaw values as functions of cumulative arc length. Braking therefore changes only the ego vehicle's longitudinal progression along the nominal route and introduces no new lateral geometry, in contrast to the steering-based response, which performs a new Hybrid A* search. No Gaussian smoothing is consequently required for the braking trajectory.

The resulting collision-avoidance trajectory is retained as the safe planning target associated with the generated interaction, while the original recorded ego trajectory defines the corresponding nominal collision course. The generated supervision therefore represents not only the immediate collision-avoidance action but, for steering-based cases, also the recovery towards ordinary driving after the critical interaction.

C.4. Representative Generated Trajectories

Figures 6 and 7 present additional examples of the critical interactions produced by the proposed trajectory-generation pipeline. The examples complement those shown in Fig. 4 and illustrate variation in both the generated threat configuration and the corresponding ego collision-avoidance response.

Stationary threats can be resolved through either steering or braking, depending on the surrounding road geometry. For steering-based responses, Hybrid A* determines the collision-free direction of the evasive manoeuvre within the available drivable region and subsequently steers the ego vehicle back onto its recorded route. Braking responses instead retain the original spatial path while modifying the ego vehicle's longitudinal progression so that it stops before the conflict region.

Moving threats impose additional constraints on the admissible set of responses. Moving head-on interactions require lateral avoidance because stopping on the nominal route would leave the ego vehicle in the path of the approaching threat; such scenarios are therefore retained only when a valid steering manoeuvre can be generated. For laterally moving threats, braking can be an effective response by preventing the ego vehicle from entering the conflict region before the threat crosses its path.

C.5. Rendering and Restoration Details

The generated ego and threat trajectories are applied to the reconstructed StreetGaussians scene using the pose-construction, coordinate-transformation, and actor-manipulation procedures described in Appendix A. The modified interaction is then rendered from all six surround-view cameras, and the resulting images are subsequently processed by Fixer to reduce novel-view reconstruction artefacts.

The StreetGaussians reconstructions used in this work are trained with a factor-three image downscale, as already described. The resulting rendered camera observations therefore have a spatial resolution of approximately 533×300 pixels, compared with the native nuScenes resolution of 1600×900. The modified ego and threat poses can additionally depart substantially from the viewpoints observed during reconstruction. A steering-based ego response may expose previously unseen regions of the static environment, while repositioning the threat actor can reveal vehicle surfaces that were weakly observed or entirely unobserved in the original sequence. Consequently, the raw StreetGaussians outputs can contain incomplete geometry, degraded object surfaces, blurred regions, or other novel-view artefacts.

To reduce these artefacts, the rendered observations are subsequently processed using the pretrained Fixer model released with Difix3D+ [40]. The checkpoint used in this

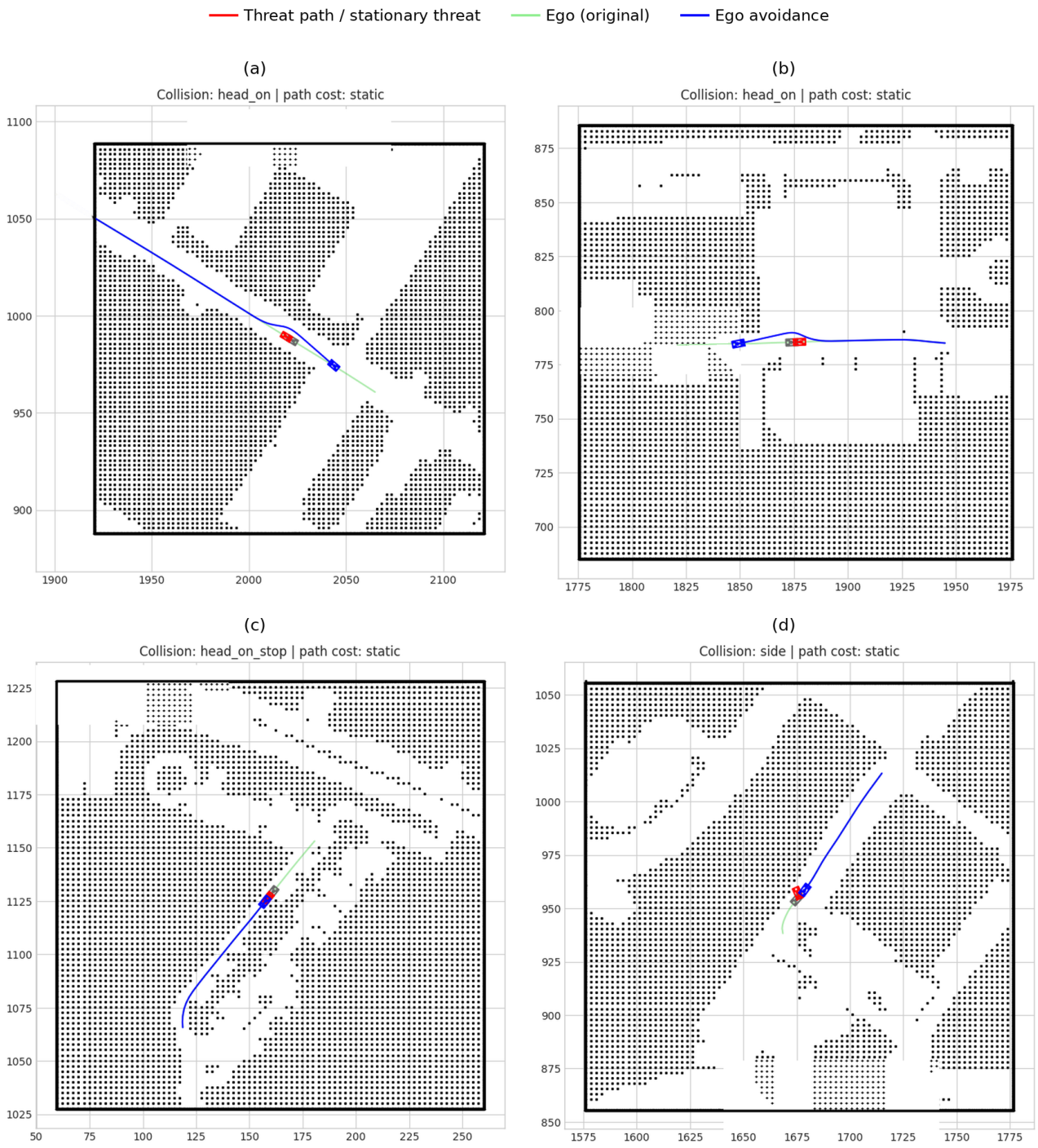


Figure 6. Representative interactions involving stationary threats. Red denotes the stationary threat location, green the nominal recorded ego trajectory, and blue the generated collision-avoidance response. Panels (a)–(b) show steering-based avoidance, while panels (c)–(d) show braking-based responses.

work is `pretrained_fixer.pkl` and it is loaded on top of the Cosmos Predict2 0.6B image-generation pipeline using the fast-tokeniser VAE, with the pretrained Fixer checkpoint replacing the corresponding UNet and VAE param-

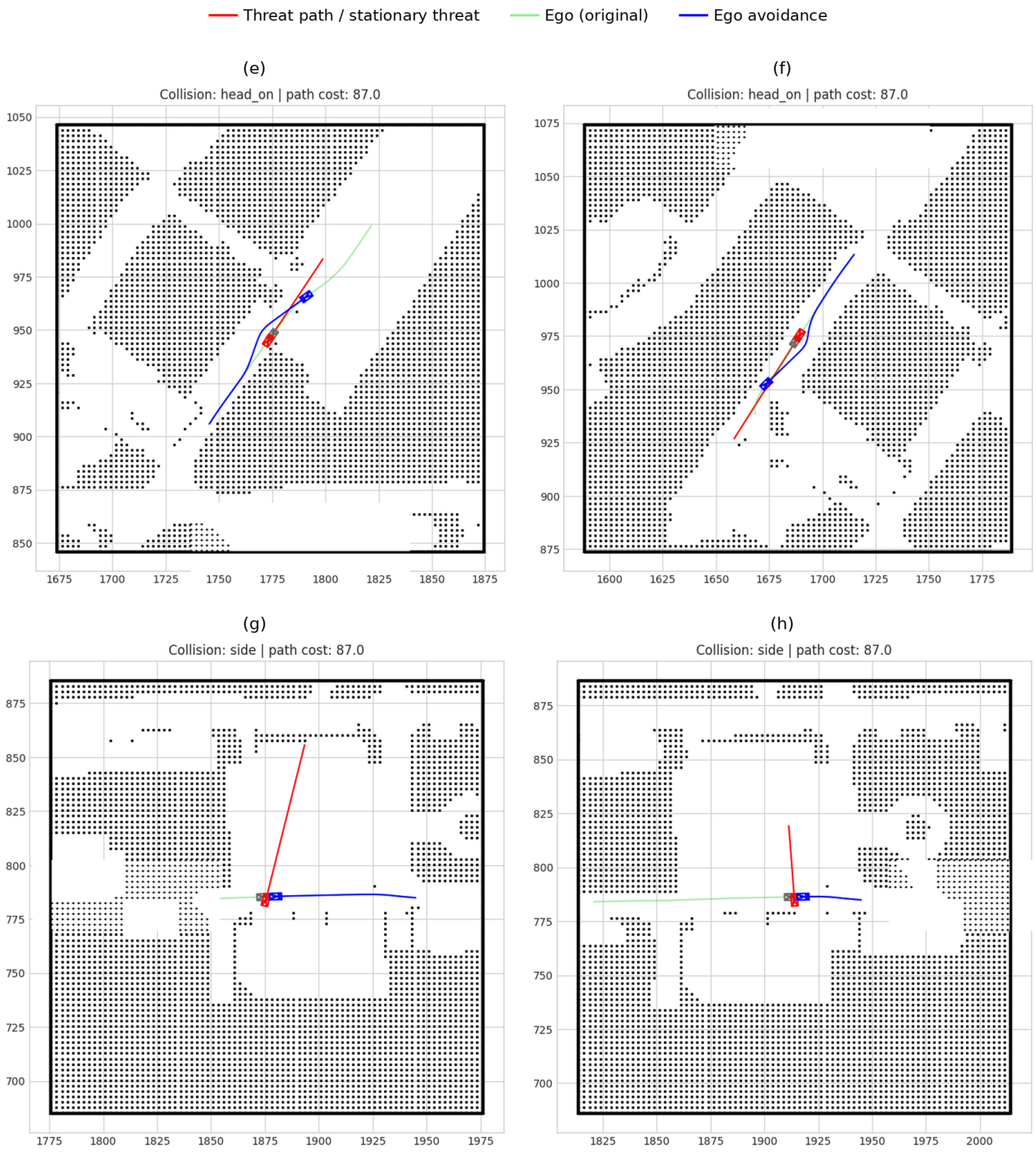


Figure 7. Representative interactions involving moving threats. Red denotes the generated threat trajectory, green the nominal recorded ego trajectory, and blue the generated collision-avoidance response. Panels (e)–(f) show moving head-on interactions resolved through steering-based avoidance, while panels (g)–(h) show lateral moving threats resolved through braking.

eters. The no-conditioning variant is used during inference. No text prompt or scene-specific textual conditioning is therefore supplied; the restoration is determined entirely from the rendered input image.

Fixer uses a single-step diffusion formulation rather than iterative sampling. The inference timestep is set to 250, corresponding to $\sigma = 0.25$, and a single denoising prediction is performed. VAE skip connections are disabled, and inference is performed with a batch size of one using `bfloat16` precision on CUDA. The model is wrapped using `torch.compile`. Before processing the scenario images, 50 dummy warm-up iterations are executed to initialise the compiled inference path; these iterations affect execution efficiency rather than the restoration result.

Each input image is first converted to RGB. Although the rendered StreetGaussians image has a resolution of approximately 533×300, Fixer internally operates at a canonical 1024×576 resolution. The input is therefore bilinearly resized to 1024×576 before normalisation and inference. Following the single denoising step, the output is denormalised, clamped to the valid image range, converted back to an RGB image, and bilinearly resized to the dimensions of the original input frame. Consequently, Fixer does not permanently upscale the generated observations: both the raw StreetGaussians image and the saved restored image remain at approximately 533×300. The 1024×576 representation exists only internally during the restoration process.

Fixer modifies only the rendered RGB observations. It does not alter the generated ego trajectory, threat trajectory, camera calibration, RigidNode poses, or the underlying StreetGaussians reconstruction. The geometric and temporal construction of the critical interaction therefore remains exactly that produced by the preceding scenario-generation stages; restoration is applied solely as a visual post-processing operation.

The complete restoration configuration used in the experiments is summarised in Table 10.

Figure 8 provides additional qualitative examples of the restoration stage. The examples illustrate three representative types of novel-view degradation encountered in the generated scenarios: background artefacts caused by modified ego viewpoints, degradation of a repositioned threat actor observed from the front, and degradation of the same type of manipulated actor from a lateral viewpoint. In each case, the raw StreetGaussians rendering is shown alongside the corresponding Fixer-restored output. Since both images retain the same external resolution, the visual differences result from the restoration model rather than from spatial upsampling.

C.6. UniAD-Compatible Dataset Construction

After rendering and restoration, each accepted synthetic interaction is converted into the information-file representation consumed by the UniAD nuScenes data pipeline. Rather than modifying UniAD's input interface, the conversion constructs a new sample-information dictionary for each synthetic keyframe by retaining compatible metadata from the corresponding original nuScenes sample and replacing the fields affected by the generated ego and threat motion. The resulting samples can therefore be loaded using the same dataset and training pipeline as the original nuScenes data.

Table 10. Fixer restoration configuration used for the generated StreetGaussians observations.

Setting	Value
Checkpoint	`pretrained_fixer.pkl`
Base model	Cosmos Predict2 0.6B
Tokenizer / VAE	Fast tokenizer
Conditioning	None
Diffusion steps	Single step
Inference timestep	250
σ	0.25
Rendered input resolution	$\approx 533 \times 300$
Internal Fixer resolution	1024×576
Saved output resolution	$\approx 533 \times 300$
Resize interpolation	Bilinear
VAE skip connection	Disabled
Batch size	1
Precision	`bfloat16`
Compilation	`torch.compile`
Warm-up iterations	50
Camera filtering	None
Frame filtering	None within supplied input
Guardrail	Disabled

The trajectory-generation and rendering stages operate on a 10 Hz representation, whereas UniAD uses the 2 Hz nuScenes keyframe rate. The generated ego-avoidance and threat trajectories are therefore subsampled using a common stride of five,

$$\mathbf{q}_i^{2\mathrm{Hz}} = \mathbf{q}_{5i}^{10\mathrm{Hz}}, \tag{26}$$

ensuring that the generated ego and threat states remain temporally aligned. Camera observations are not independently downsampled during information-file construction; instead, the conversion directly references the already generated 2 Hz images stored for each camera. Similarly, the annotations of unmodified traffic participants originate from the original nuScenes keyframes and are therefore already represented at 2 Hz.

For each synthetic sample, the original per-sample token, temporal links, frame index, and timestamp are retained. A new synthetic scene token is constructed so that different generated variants originating from the same recorded scene remain distinct temporal sequences. The original nuScenes scene token is stored separately as `orig_scene_token`. This original token is used whenever the nuScenes API or HD-map information must be queried, while the synthetic scene token is preserved for dataset-level temporal grouping. During data loading, the scene token is temporarily replaced by `orig_scene_token` for the parent nuScenes lookup and subsequently restored to the synthetic value. The occupancy temporal-window information is then re-



Figure 8. Additional examples of the rendering and restoration pipeline. The left column shows raw StreetGaussians novel-view renderings, and the right column shows the corresponding outputs after Fixer restoration. The first row illustrates restoration of background artefacts introduced by a modified ego viewpoint, while the second and third rows show restoration of a repositioned threat actor observed from frontal and lateral viewpoints, respectively. Both raw and restored images retain the same spatial resolution of approximately 533×300; the 1024×576 representation used by Fixer exists only internally during inference.

computed using the restored synthetic token so that neighbouring frames belonging to the same generated scenario remain correctly associated.

The generated ego trajectory replaces the original `ego2global_translation` and `ego2global_rotation` fields. The fixed LiDAR-to-ego calibration is retained, since the physical sensor extrinsics are unchanged. The CAN-bus representation is partially recomputed from the generated ego trajectory. Specifically, entries 7–15 are replaced with the ego-frame acceleration, rotational-rate, and velocity components obtained by finite differences of the 2 Hz generated trajectory. World-frame velocity and acceleration are rotated into the instantaneous ego frame, the vertical velocity and acceleration components are set to zero, and the rotational-rate vector contains only the generated yaw rate. The remaining CAN-bus entries are retained from the corresponding original sample during information-file construction, while the generated position and orientation are represented explicitly by the updated ego-to-global pose fields.

The rendered observations are associated with all six camera entries of the synthetic sample. StreetGaussians and Fixer operate on images at approximately 533×300 resolution, corresponding to a factor-three downscale relative to the native nuScenes camera resolution of 1600×900. After Fixer restoration, the images are upscaled to the native 1600×900 nuScenes resolution before being used in the final UniAD-compatible dataset. Consequently, the original nuScenes camera calibration and intrinsic parameters can be retained in the final dataset representation, since the images presented to the UniAD data pipeline again follow the native nuScenes image dimensions. The camera extrinsics are likewise preserved, as the physical relative transformations between the ego vehicle and the six cameras remain unchanged.

Object annotations are initialised from the corresponding original nuScenes keyframe. Actors suppressed during rendering are identified through the stored suppression-token list and removed consistently from all parallel annotation arrays. The selected threat actor is then inserted or updated using its generated world-frame position and heading. Its pose is transformed into the LiDAR coordinate system used by UniAD, while its velocity is estimated from consecutive generated threat positions and rotated into the same frame. If the selected instance already appears in the original annotations, its box and motion information are replaced; otherwise, a new vehicle annotation is appended. Compatible surrounding actors retain their original annotations and recorded motion.

Future and past motion labels are additionally constructed for the modified interaction. Fresh past and future trajectories are generated directly from the synthetic threat trajectory, whereas the original precomputed motion labels are retained for surrounding actors whose trajectories have not been modified.

The generated ego avoidance trajectory provides the planning supervision used during UniAD fine-tuning. For each keyframe, the next six planning steps are transformed into the initial-frame LiDAR coordinate system following the same transformation chain used by UniAD for the original nuScenes planning labels. The corresponding planning mask indicates which future steps are available. A high-level driving command is inferred from the final valid planning waypoint: a sufficiently positive or negative lateral displacement produces a left or right command, respectively, while smaller lateral deviations are classified as forward motion. The generated ego future trajectory is stored separately from this planning-specific representation so that the synthetic sample provides both the standard ego-motion target and the planner-specific supervision expected by UniAD.

Table 11 summarises the principal transformations applied during construction of the synthetic information files.

Table 11. Principal transformations used to construct the UniAD-compatible synthetic dataset representation.

Component	Synthetic representation
Temporal rate	Generated trajectories: 10 Hz $\rightarrow$ 2 Hz
Camera observations	Six restored 2 Hz views
Image resolution	Restored at $\approx 533 \times 300$, then upscaled to 1600×900
Camera intrinsics	Original nuScenes intrinsics retained
Ego pose	Replaced by generated avoidance pose
Sensor extrinsics	Original calibration retained
CAN bus	Accel., yaw rate, velocity recomputed
Threat annotation	Generated box, heading, and velocity
Other actors	Original annotations retained when compatible
Suppressed actors	Removed from all annotation arrays
Threat motion labels	Recomputed from generated trajectory
Other motion labels	Original labels retained
Planning supervision	Generated safe ego trajectory
Scene identity	Synthetic scene token + original lookup token

D. Experimental Details and Additional Analysis

D.1. Trajectory-Feasibility Analysis

The trajectory-generation module is based on Hybrid A* and therefore directly enforces geometric and non-holonomic feasibility, but it does not optimise a full vehicle-dynamics model. We consequently perform an additional post-generation analysis to determine whether the resulting threat and ego trajectories exhibit reasonable geometric and dynamic behaviour. The analysis considers three complementary properties: whether the generated threat produces the intended collision-course interaction, whether the trajectories remain compatible with the planner's kinematic constraints, and whether the accelerations induced by the generated motion remain within plausible ranges.

Across the 1,060 threat-planning attempts reported in Sec. 4.2, 979 produce a feasible trajectory, corresponding to a planning success rate of 92.4%. For every successful interaction, the minimum distance between the generated threat trajectory and the original recorded ego trajectory is additionally evaluated. All 979 trajectories approach the nominal ego path to within 2 m, confirming that the planner consistently constructs an actual conflict with the recorded ego motion rather than merely placing the threat elsewhere within the same drivable region. Moving threats travel at an average speed of approximately 22.8 km/h, which lies within the 15–30 km/h operating range used during threat-start construction and temporal synchronisation.

As an additional robustness check, 100 generated interactions are independently sampled and replanned using the same Hybrid A* configuration. A valid threat trajectory is recovered in 99 of the 100 cases.

Dynamic quantities are evaluated from the time-parameterised trajectories after trajectory generation and post-processing. For a trajectory with planar position $\mathbf{p}_t = (x_t, y_t)$, heading ψ_t, and sampling interval Δt, translational

Table 12. Distribution of maximum effective lateral acceleration for generated trajectories and ordinary nuScenes motion. Percentiles are computed across trajectories.

Trajectory set	P50	P90	P99	> 7	> 8
	[m/s^2]			[%]	
nuScenes reference	0.29	2.04	3.88	–	–
Generated threat	3.59	9.54	11.72	30.3	21.2
Ego braking	0.17	1.96	3.88	0.0	0.0
Ego steering	3.94	7.73	10.56	15.7	9.0

speed and yaw rate are estimated from successive states. Curvature is evaluated from the change in unwrapped vehicle heading relative to travelled arc length,

$$\kappa_t \approx \frac{\Delta\psi_t}{\Delta s_t}, \qquad \Delta s_t = \|\mathbf{p}_{t+1} - \mathbf{p}_t\|_2 , \tag{27}$$

and the corresponding effective lateral acceleration is estimated as

$$a_{\mathrm{lat},t} = v_t^2 |\kappa_t|. \tag{28}$$

For each trajectory, the maximum absolute effective lateral acceleration over the generated manoeuvre is retained, and the reported percentiles are computed across trajectories. This quantity is used as a diagnostic measure of manoeuvre aggressiveness rather than as evidence of full dynamic feasibility, since tyre forces, suspension behaviour, road friction, and vehicle-specific dynamics are not represented by Hybrid A*.

The steering geometry itself remains bounded by the vehicle model used during Hybrid A*. With wheelbase $L = 3.5\,\mathrm{m}$ and maximum steering angle $\delta_{\max} = 0.6\,\mathrm{rad}$, the corresponding kinematic curvature bound is

$$|\kappa|_{\max} = \frac{\tan(\delta_{\max})}{L} \approx 0.195\ \mathrm{m}^{-1}. \tag{29}$$

The generated Hybrid A* trajectories remain within this steering constraint by construction.

The independently replanned threat trajectories exhibit a substantially more aggressive dynamic tail than ordinary recorded nuScenes motion. Table 12 reports the distribution of the maximum effective lateral acceleration. For the 99 successfully replanned threat trajectories, the median is 3.59 m/s^2, increasing to 9.54 m/s^2 at the 90th percentile and 11.72 m/s^2 at the 99th percentile. Approximately 30.3% exceed 7 m/s^2 and 21.2% exceed 8 m/s^2. For comparison, the corresponding values measured from ordinary nuScenes ego motion are substantially lower, with a median of 0.29 m/s^2, a 90th percentile of 2.04 m/s^2, and a 99th percentile of 3.88 m/s^2.

These results show that collision-course threat generation occasionally produces motions that are considerably more aggressive than those in ordinary recorded driving. This is primarily a consequence of using a kinematic planner to satisfy a constrained start pose, a conflict pose, a heading, and an arrival time without explicitly optimising vehicle dynamics. The threat trajectories should therefore be interpreted as kinematically feasible collision-course motions rather than as trajectories guaranteed to satisfy a complete vehicle-dynamics model. This limitation is less consequential for the training objective of this work because the threat trajectory defines the hazardous interaction, whereas the generated ego response provides the planning supervision that UniAD learns.

The generated ego responses exhibit substantially milder dynamics overall. Braking trajectories retain the recorded route and modify only longitudinal progression, and consequently exhibit low lateral acceleration: the median maximum value is 0.17 m/s^2, the 90th percentile is 1.96 m/s^2, and the 99th percentile is 3.88 m/s^2. None of the evaluated braking responses exceeds either 7 or 8 m/s^2.

Steering-based avoidance is necessarily more aggressive because the ego vehicle must depart from the recorded route, achieve sufficient lateral displacement to clear the threat, and subsequently recover towards the nominal trajectory. The final steering trajectories have a median maximum effective lateral acceleration of 3.94 m/s^2, with 90th- and 99th-percentile values of 7.73 and 10.56 m/s^2, respectively. Approximately 15.7% exceed 7 m/s^2 and 9.0% exceed 8 m/s^2. The majority of the generated steering supervision, therefore, remains below these high-acceleration thresholds, although a small tail contains aggressive evasive manoeuvres approaching practical vehicle-grip limits.

The feasibility analysis, therefore, indicates a distinction between the generated threat and ego trajectories. The threat planner successfully constructs the intended collision geometry with high reliability, but its most aggressive trajectories can exceed the dynamics typically observed in nuScenes, whereas the ego trajectories used for learning supervision are considerably better behaved.

D.2. UniAD Fine-Tuning Configuration

All fine-tuning experiments are initialised from the publicly released UniAD Stage-2 checkpoint, `uniad_base_e2e.pth`. The training data are implemented as a `ConcatDataset` containing 100 original nuScenes training scenes and the 100 selected synthetic critical scenarios, corresponding to 4,031 and 3,789 keyframes, respectively. No source-specific reweighting or oversampling is applied, and all samples are seen once per epoch in shuffled order.

Synthetic scenarios are mixed with original driving data rather than used in isolation. This follows the general data-augmentation strategy demonstrated by SceneCrafter [16], which fine-tunes end-to-end planners using synthetic driving logs together with real-world training data; its closed-

loop experiments augment 32k real samples with 6k simulated samples generated from 60 reconstructed training scenes. In our setting, an approximately balanced mixture is used instead because the objective is specifically to adapt the planner to rare collision-avoidance behaviour. The similar numbers of real and synthetic keyframes give the generated critical supervision substantial influence during the short fine-tuning stage while maintaining continued exposure to ordinary driving and reducing the risk of catastrophic forgetting. SceneCrafter therefore provides precedent for the real–synthetic mixing strategy, while the approximately balanced mixture used here is a design choice specific to the targeted adaptation objective of this work.

The image backbone remains frozen at the model level following the Stage-2 configuration, including its BatchNorm parameters. Additional perception-related components are frozen through zero learning-rate multipliers in the optimiser configuration: `pts_bbox_head`, `seg_head`, `query_embedding`, `reference_points`, `query_interact`, and `memory_bank`. The `motion_head`, `occ_head`, and `planning_head` remain trainable. Fine-tuning is therefore concentrated on the downstream motion, occupancy, and planning components while preserving the pretrained image, detection, tracking, and mapping representations.

Training is performed for five epochs using AdamW with a peak learning rate of 2×10^{-5} and weight decay of 0.01. The cosine-annealing schedule inherited from the Stage-2 configuration is retained, with a minimum learning-rate ratio of 10^{-3}. Linear warm-up is applied for the first 500 iterations using a warm-up ratio of $1/3$. Training uses one sample per GPU on four Tesla V100-SXM2-32GB GPUs, resulting in an effective batch size of four.

No validation-based checkpoint selection or early stopping is performed. The evaluation hook is executed at the end of the fifth epoch, while intermediate checkpoints are stored every 1,000 optimisation iterations for recovery purposes. All fine-tuned results reported in Sec. 4.4 use the final `epoch_5.pth` checkpoint.

D.3. NeuroNCAP Evaluation Verification

During verification of the NeuroNCAP closed-loop evaluation, we identified a collision-registration issue affecting a small number of reconstructed scenes containing obstacles such as an excavator or road barriers. In these cases, the closed-loop rollout could visibly progress into and through the obstacle region without the interaction being registered as a collision by the original evaluation path. Since NeuroNCAP collision rate is determined from the outcome of the executed closed-loop trajectory, such missed contacts would incorrectly count the corresponding rollout as collision-free and therefore bias the reported safety metric.

Table 13. Detailed UniAD fine-tuning configuration.

Setting	Value
Initial checkpoint	`uniad_base_e2e.pth`
Synthetic scenarios	100 (3,789 keyframes)
Original nuScenes scenes	100 (4,031 keyframes)
Dataset combination	Concatenation, no source reweighting
Optimizer	AdamW
Peak learning rate	2×10^{-5}
Weight decay	0.01
LR schedule	Cosine annealing
Minimum LR ratio	10^{-3}
Warm-up	Linear, 500 iterations, ratio $1/3$
Gradient clipping	Max L_2 norm 35
Training epochs	5
Batch size	1/GPU, 4 effective
GPUs	4× Tesla V100-SXM2-32GB
Frozen components	Image/perception components (see text)
Trainable components	Motion, occupancy, planning heads
Evaluation checkpoint	Final epoch (`epoch_5.pth`)

Figure 9 illustrates this behaviour for two representative interactions. Panels (a)–(c) show successive closed-loop frames from a scene containing an excavator. As the rollout progresses, the ego vehicle enters the obstacle region and continues past the expected contact point, rather than the evaluation terminating with a collision. Panels (d)–(f) show the same failure mode for an affected barrier interaction. The use of several successive frames is important here, since it confirms that the observation is not a single-frame visualisation artefact: the simulated ego state continues to advance through the occupied region over multiple closed-loop steps.

The affected collision handling was corrected before producing the final results reported in this work. The evaluation was then repeated using the corrected implementation for both the original and fine-tuned UniAD checkpoints, while keeping the model checkpoints and inference configuration identical between the two evaluations. The corrected collision rates are therefore used consistently for all NeuroNCAP comparisons in Sec. 4.4.

D.4. Closed-Loop Avoidance-Strategy Analysis

To better understand the asymmetric NeuroNCAP results reported in Sec. 4.4, we analyse same-seed closed-loop runs in which the original and fine-tuned UniAD checkpoints produce different collision outcomes. Since both checkpoints are evaluated with identical scene initialisation, inference settings, and collision-optimiser configuration, these matched runs isolate differences in the resulting driving behaviour rather than differences in the evaluation setup.

The primary quantity used for this comparison is the peak lateral offset from the original recorded nuScenes ego trajectory. Let $\mathbf{p}_{\text{ego}}(t)$ denote the closed-loop ego position at time t, and let $\mathbf{p}_{\text{ref}}(t)$ and $\psi_{\text{ref}}(t)$ denote the position and heading of the recorded nuScenes ego trajectory interpolated to the same timestamp. The signed lateral deviation

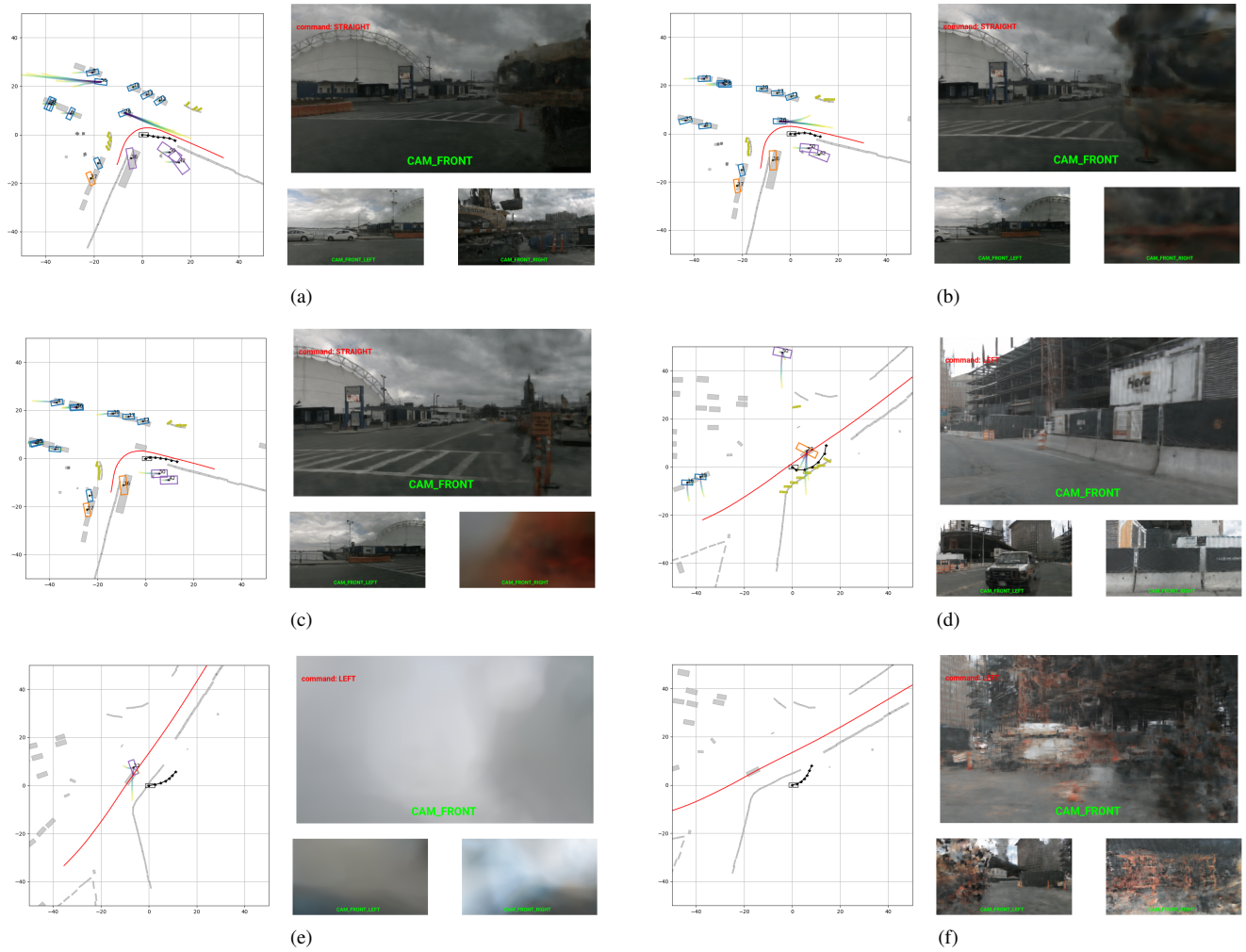


Figure 9. Representative examples of the collision-registration issue identified during NeuroNCAP evaluation. Panels (a)–(c) show successive frames from an interaction involving an excavator, while panels (d)–(f) show an affected barrier interaction. In both cases, the closed-loop rollout continues through the obstacle region over successive timesteps without the original collision-handling path registering the contact as a collision.

is computed as

$$d_{\text{lat}}(t) = (\mathbf{p}_{\text{ego}}(t) - \mathbf{p}_{\text{ref}}(t)) \cdot \mathbf{n}_{\text{ref}}(t), \tag{30}$$

where

$$\mathbf{n}_{\text{ref}}(t) = (-\sin\psi_{\text{ref}}(t), \cos\psi_{\text{ref}}(t)) \tag{31}$$

is the normal direction of the recorded route at that timestamp. The peak lateral offset for a rollout is then defined as

$$d_{\text{lat}}^{\max} = \max_t |d_{\text{lat}}(t)|\,. \tag{32}$$

This, therefore, measures the maximum cross-track deviation from the recorded route in the route's local time-varying frame, rather than relative to a fixed initial ego heading.

The clearest positive transfer is observed in the `stationary-0278` scenario. Across 44 matched runs for which the fine-tuned checkpoint avoids the obstacle while the original checkpoint collides, the fine-tuned model produces the larger peak lateral offset in all 44 cases. Its mean peak lateral offset is $4.87\,\text{m}$, compared with $1.15\,\text{m}$ for the original checkpoint. This indicates that successful avoidance in this scenario is associated with a substantially stronger lateral departure from the nominal route.

The opposite behaviour is observed in the frontal and lateral scenarios, for which fine-tuning degrades the closed-loop outcome. Across 58 matched runs from six such scenarios, the original checkpoint produces the larger peak lateral offset in 54 cases (93%). The dominant difference in these runs is therefore not an inability of either model to

Table 14. Same-seed closed-loop comparison for interactions in which the original and fine-tuned checkpoints produce different collision outcomes.

Interaction set	Runs	Larger lateral response	Frequency
`stationary-0278`	44	Fine-tuned	44/44 (100%)
Frontal/side failures	58	Original	54/58 (93%)

alter its longitudinal motion, but insufficient lateral commitment by the fine-tuned checkpoint in interactions that require a stronger steering response.

Figure 10 shows representative examples of both behaviours. The displayed runs are selected near the median lateral-response magnitude of the checkpoint exhibiting the larger avoidance manoeuvre, rather than as extreme examples. In `stationary-0278`, run 52, the original checkpoint remains close to the recorded route and collides, with a peak lateral offset of $0.89\,\mathrm{m}$. The fine-tuned checkpoint achieves a lateral offset of $4.65\,\mathrm{m}$ and successfully avoids the obstacle. Conversely, in `frontal-0346`, run 6, the original checkpoint produces a pronounced lateral manoeuvre with a peak offset of $5.27\,\mathrm{m}$ and avoids the threat, whereas the fine-tuned checkpoint remains much closer to the recorded route, reaching only $0.76\,\mathrm{m}$ before colliding.

Longitudinal behaviour provides a secondary distinction. Across the same 58 frontal and lateral matched runs, the fine-tuned checkpoint brakes in 45 cases (78%), compared with 5 cases (9%) for the original checkpoint. This pattern is not consistent across all scenarios, however, so the more reliable difference remains the reduced lateral commitment of the fine-tuned model.

Overall, the matched-run analysis indicates that fine-tuning changes the balance between longitudinal and lateral avoidance responses. The stronger lateral behaviour learned from the generated data is beneficial in stationary interactions, substantially improving collision avoidance. In several frontal and lateral interactions, however, the fine-tuned checkpoint instead exhibits insufficient lateral displacement and, frequently, a stronger braking response. This explains why the substantial improvement for stationary obstacles does not translate into uniform gains across all NeuroNCAP interaction geometries.

D.5. Collision-Optimiser Ablation

UniAD applies an occupancy-based nonlinear collision optimiser to the trajectory predicted by its planning head before execution. This module is deterministic and operates only during inference; it does not modify the learned model parameters and is not applied during training. All main NeuroNCAP experiments in Sec. 4.4 use the standard UniAD optimiser configuration for both checkpoints, so that the reported comparison isolates the effect of synthetic-data fine-tuning. We nevertheless conduct a separate ablation study to assess the extent to which this post-processing stage influences closed-loop collision avoidance.

The optimiser refines the trajectory predicted by the learned planner while penalising proximity to locations predicted as occupied. In simplified form, its objective can be written as

$$J = \alpha_{\mathrm{xy}} \sum_t \|\mathbf{p}_t - \hat{\mathbf{p}}_t\|_2^2 + \sum_t \sum_j \frac{\alpha_{\mathrm{col}}}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{p}_t - \mathbf{o}_{t,j}\|_2^2}{2\sigma^2}\right), \tag{33}$$

where $\mathbf{p}_t$ denotes the optimised ego position, $\hat{\mathbf{p}}_t$ is the trajectory originally predicted by the planning head, and $\mathbf{o}_{t,j}$ denotes a predicted occupied location. The first term encourages the optimiser to remain close to the learned trajectory, whereas the second introduces a Gaussian cost around predicted occupancy.

The standard UniAD configuration uses $\alpha_{\mathrm{xy}} = 1$, $\alpha_{\mathrm{col}} = 5$, and $\sigma = 1$. The parameter α_{col} controls the weight of the collision cost relative to trajectory preservation, while σ determines its spatial extent. A smaller σ produces a narrow repulsive region whose influence decreases rapidly with distance. Consequently, the optimiser may modify the trajectory substantially only once it is already relatively close to predicted occupancy. These parameters originate from the standard UniAD inference configuration evaluated on nuScenes and were not specifically selected for the aggressive collision-course interactions considered by NeuroNCAP.

For the ablation, we retain $\alpha_{\mathrm{xy}} = 1$ and increase the collision parameters to $\alpha_{\mathrm{col}} = 10$ and $\sigma = 2$. Importantly, because the Gaussian cost is normalised by σ, simultaneously doubling α_{col} and σ leaves its maximum value at zero distance approximately unchanged. The main effect is instead to broaden the repulsive region: the collision penalty decreases more slowly with distance, allowing predicted occupancy to influence the trajectory earlier. This behaviour is visible in Fig. 11, where the tuned optimiser begins deviating from the original predicted path substantially earlier than the default configuration.

Table 15 reports the resulting corrected NeuroNCAP collision rates and official NeuroNCAP scores. Strengthening the optimiser produces a large improvement for both checkpoints. For the original UniAD checkpoint, the collision rate decreases from 66.70% to 38.87%, while the NeuroNCAP score increases from 1.824 to 3.355. For the fine-tuned checkpoint, the collision rate decreases from 65.17% to 40.37%, while the score increases from 1.937 to 3.306.

The relative ordering of the checkpoints reverses under the tuned optimiser, although the remaining difference is small. With the default configuration, the fine-tuned checkpoint achieves a 1.53-percentage-point lower collision rate

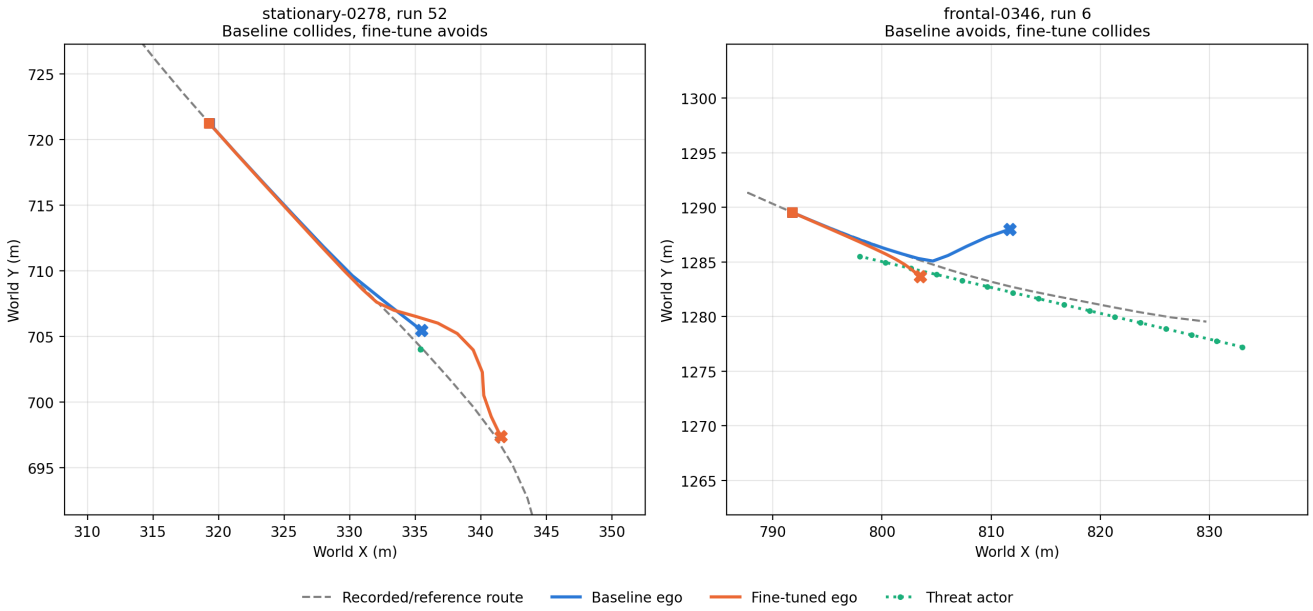


Figure 10. Representative same-seed closed-loop trajectories illustrating the dominant avoidance-strategy difference between the original and fine-tuned UniAD checkpoints. Left: `stationary-0278`, where the original checkpoint collides and the fine-tuned checkpoint avoids the threat through a substantially larger lateral manoeuvre. Right: `frontal-0346`, where the original checkpoint avoids the threat while the fine-tuned checkpoint remains close to the recorded route and collides. The dashed trajectory denotes the original recorded nuScenes ego route, while the dotted trajectory denotes the threat actor. The displayed runs are selected near the median lateral-response magnitude of the checkpoint that exhibits the larger manoeuvre, rather than as extreme examples.

Table 15. Effect of collision-optimiser tuning on corrected NeuroNCAP performance. Lower collision rate and higher NeuroNCAP score are better.

Optimiser	Checkpoint	Collision [%]	Score
Default	Original	66.70	1.824
Default	Fine-tuned	**65.17**	**1.937**
Tuned	Original	**38.87**	**3.355**
Tuned	Fine-tuned	40.37	3.306

and a 0.113-point higher NeuroNCAP score. With the tuned configuration, the original checkpoint instead achieves a 1.50-percentage-point lower collision rate and a 0.049-point higher score. The latter score difference corresponds to less than 1% of the full 0–5 NeuroNCAP scale and is small compared with the improvement produced by changing the optimiser itself.

This reversal is consistent with the trajectory-level analysis in Appendix D.4. In several difficult frontal and lateral interactions, the original checkpoint already produces a larger lateral response than the fine-tuned model. Since the optimiser refines the trajectory supplied by the learned planner rather than generating an independent avoidance manoeuvre, its effect depends on the trajectory and occupancy predictions produced by each checkpoint.

Figure 11 illustrates the qualitative effect in two representative scenarios. For each interaction, three successive frames are shown with the default configuration and three with the tuned configuration. Under the default optimiser, the avoidance correction occurs late and does not generate sufficient clearance from the threat, resulting in a collision. With the broader collision penalty produced by $\sigma = 2$, the trajectory begins moving away from the predicted occupied region earlier and the corresponding rollout successfully avoids the collision. These examples are consistent with the interpretation of σ as controlling the spatial range over which predicted occupancy influences trajectory refinement.

The magnitude of the improvement demonstrates that UniAD's deterministic collision optimiser has a substantial influence on closed-loop NeuroNCAP performance. This experiment, however, addresses a different question from the main contribution of this work. Changing α_{col} and σ does not improve the collision-avoidance behaviour learned from the generated data; it changes how an already predicted trajectory is modified during inference. The main experiments therefore retain the standard UniAD optimiser configuration, since the objective of this work is to determine whether the proposed critical-scenario generation pipeline provides useful training supervision. More broadly, the substantial performance improvement achieved by mod-



Figure 11. Representative effect of collision-optimiser tuning on closed-loop trajectory generation. Panels (a)–(c) and (d)–(f) show successive frames from the same interaction using the default and tuned optimiser, respectively, while panels (g)–(i) and (j)–(l) show a second interaction. With the default configuration, the avoidance correction occurs late and the rollout collides. Increasing the collision parameters to $\alpha_{\text{col}} = 10$ and $\sigma = 2$ broadens the occupancy-based repulsive region, resulting in an earlier trajectory correction and successful avoidance in both examples. Within each comparison, the model checkpoint is unchanged; only the deterministic collision-optimiser configuration differs.

ifying this deterministic post-processing stage highlights UniAD's dependence on its hand-designed collision optimiser for closed-loop safety. The fact that such a mechanism is required in the first place suggests that improving the learned planner itself, so that collision-aware behaviour is produced directly by the model rather than recovered through post-processing, is a more desirable direction for future work. This limitation is discussed further in Sec. 5.