Challenger: Affordable Adversarial Driving Video Generation

Zhiyuan Xu[1,2*] **Bohan Li**[3,4*] **Huan-ang Gao**[1] **Mingju Gao**[1] **Yong Chen**[6]
Ming Liu[6] **Chenxu Yan**[6] **Xin Jin**[4] **Hang Zhao**[7] **Shuo Feng**[8] **Hao Zhao**[1,5†]
[1]AIR, Tsinghua [2]UCAS [3]SJTU [4]EIT, Ningbo [5] BAAI
[6] Geely Auto [7]IIIS, Tsinghua [8] Department of Automation, Tsinghua

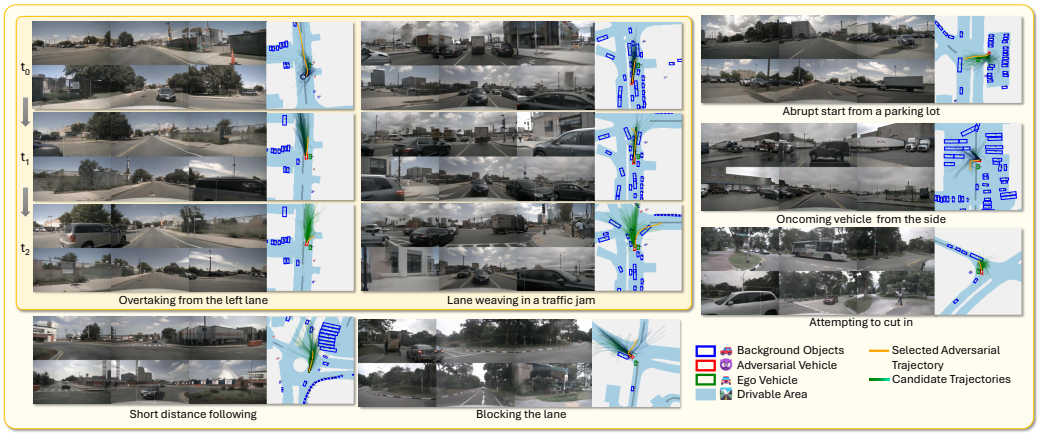


Figure 1: **Photorealistic adversarial driving videos generated by *Challenger*.** Each scenario includes an adversarial vehicle—highlighted with a white 3D bounding box in camera views and depicted as a red rectangle in the bird's-eye view (BEV) map—that is intentionally designed to challenge the ego vehicle through aggressive or unexpected maneuvers. *Challenger* autonomously produces these videos using a diffusion-based trajectory generator, physics-aware planning, and multiview neural rendering, with an affordable computation budget. Readers are suggested to zoom in on BEV maps and camera views for detailed inspection. Please visit our project page for videos.

Abstract: Generating photorealistic driving videos has seen significant progress recently, but current methods largely focus on natural scenarios. Meanwhile, efforts to generate adversarial driving scenarios often operate on abstract trajectory or BEV representations, falling short of delivering realistic sensor data that can truly stress-test autonomous driving (AD) systems. In this work, we introduce *Challenger*, a framework that produces physically plausible yet photorealistic adversarial driving videos. Generating such videos poses a fundamental challenge: it requires jointly optimizing over the space of traffic interactions and high-fidelity sensor observations. *Challenger* makes this affordable through: (1) a physics-aware multi-round trajectory refinement process that narrows down candidate adversarial maneuvers, and (2) a tailored trajectory scoring function that encourages realistic yet adversarial behavior while maintaining compatibility with downstream video synthesis. As tested on the nuScenes dataset, *Challenger* generates a diverse range of aggressive driving scenarios—including cut-ins, sudden lane changes, tailgating, and blind spot intrusions—and renders them into multiview photorealistic videos. Extensive evaluations show that these scenarios significantly increase the collision rate of state-of-the-art end-to-end AD models (UniAD, VAD, SparseDrive, and DiffusionDrive), and importantly, adversarial behaviors discovered for one model often transfer to others. Our code, models, and dataset can be found at `https://pixtella.github.io/Challenger/`.

Keywords: Autonomous Driving, Adversarial Driving Environments, Diffusion Models, Photorealistic Video Generation

---

*Equal contribution.
†Corresponding author.

Workshop on Safe and Robust Robot Learning for Operation in the Real World at CoRL 2025.

1 Introduction

Autonomous driving (AD) systems are typically evaluated using large-scale datasets such as nuScenes [1], OpenScene [2], NAVSIM [3] and Bench2Drive [4], which capture a wide range of real-world urban driving behaviors. However, these datasets primarily consist of natural traffic flows, and rarely include deliberately challenging interactions that put AD systems under pressure. This lack of adversarial or safety-critical scenarios makes it difficult to systematically evaluate how well these systems handle rare but high-risk events.

To address this gap, researchers have proposed various methods for adversarial scenario generation. For example, Feng et al. [5] explore adversarial behavior modeling in highway environments using a closed-loop simulator, enabling analysis of specific risk patterns under controlled conditions. Ding et al. [6] propose synthesizing rare-event distributions but limit interactions to two vehicles and do not produce corresponding sensor data. Other works like NeuroNCAP [7] enable closed-loop photorealistic testing but rely on manually constructed scenarios confined to a few predefined adversarial categories. At the same time, recent methods for photorealistic driving video generation have focused on synthesizing normal, everyday behaviors, and thus do not expose critical weaknesses in AD models. In short, there remains a significant gap: no existing method can automatically generate diverse, physically plausible, and photorealistic adversarial driving videos within real-world scenes—a capability that is essential for robust and systematic AD evaluation.

To fill this gap, we introduce *Challenger*, a framework that synthesizes multiview, high-fidelity driving videos where an adversarial agent actively challenges the ego vehicle within realistic urban environments. To illustrate the capabilities of *Challenger*, we present a selection of generated adversarial scenarios in Fig.1. Each scenario includes an ego vehicle (green) and an adversarial vehicle (red), embedded within realistic traffic scenes replayed from the nuScenes dataset. The adversarial agent executes physically plausible yet aggressive maneuvers—such as sudden cut-ins, blind-spot overtakes, tailgating, and lane blocking—that are specifically designed to interfere with the ego vehicle's driving decisions. While these scenarios are challenging, they are designed to be solvable, ensuring meaningful tests of AD models. They encompass a broad spectrum of urban conditions, including multi-lane traffic, intersections, roundabouts, and parking areas. Going beyond handcrafted corner cases, these scenarios provide a scalable and cost-effective automated approach for assessing how autonomous driving models respond to unexpected and potentially unsafe behaviors.

In summary, Our key contributions are as follows:

- We introduce the task of adversarial driving video generation and propose *Challenger*, a unified framework that makes it affordable via multi-round physics-aware trajectory refinement and renderer-compatible adversarial scoring.
- We validate *Challenger* on the nuScenes dataset, generating diverse and photorealistic adversarial maneuvers—such as cut-ins, tailgating, and blocking—across a wide range of dynamic traffic contexts including intersections, roundabouts, and parking lots.
- *Challenger*-generated videos significantly degrade the performance of state-of-the-art AD models (UniAD [8], VAD [9], SparseDrive [10], DiffusionDrive [11]) and reveal transferable failure patterns, providing insights into model vulnerabilities.

2 Related Work

2.1 Generating Driving Trajectories

Driving trajectory generation, prediction, and adversarial manipulation are central to the safety validation of autonomous driving (AD) systems. Recent research has proposed a range of techniques [5, 12, 13, 14, 15, 16, 17, 18, 19] to enhance scenario diversity, realism, and controllability. Feng et al. [5] proposed an intelligent testing framework that integrates naturalistic and adversarial behaviors, reducing evaluation mileage without sacrificing fairness. D2RL [12] further densifies safety-critical events from real data to accelerate validation. To increase realism, NeuralNDE [13] models multi-agent interactions and rare-event conflicts via learned critics and safety mapping networks. Diffusion-based planners have also emerged as effective tools for trajectory generation.

Diffusion-ES [14] employs gradient-free optimization for instruction-conditioned trajectory synthesis, while Zheng et al. [15] introduce a transformer-based variant to capture multi-modal behavior and ensure strong generalization across diverse driving styles. In parallel, language-conditioned methods enable intuitive scenario customization through natural language input. LCTGen [16], InteractTraj [17], and ChatSUMO [18] employ LLMs for traffic scene and trajectory generation with controllable semantics. VLP [19] integrates language and planning for robust scene understanding in long-tail scenarios. These works typically lack photorealistic sensor data or explicit adversarial intent. Our Challenger bridges this gap by combining diffusion-based trajectory generation, physics-aware planning, and multiview rendering to produce photorealistic adversarial driving videos that systematically stress-test AD models.

2.2 Driving Video Generation

Recent advancements in driving video generation have enhanced the realism and controllability of autonomous driving simulations [20, 21, 22, 23, 24, 25, 26, 27]. DriveDreamer [28] improves video and action prediction accuracy by enforcing real-world traffic constraints. Its successor, DriveDreamer-2 [29], integrates large language models to enable user-defined video generation with enhanced temporal coherence. MagicDrive [20] achieves high-fidelity street-view synthesis via tailored encoding and cross-view attention mechanisms. Extensions like MagicDriveDiT [23] and MagicDrive3D [30] address scalability through DiT architectures and deformable Gaussian splatting with monocular depth initialization, respectively. UniScene [31] unifies multi-modal data generation (e.g., occupancy, RGB, LiDAR) using progressive processes, while DreamDrive [26] synthesizes 4D scenes via hybrid Gaussian representations for balanced visual quality and generalizability. Drive-WM [25] pioneers multiview video forecasting by decoupling spatial-temporal dynamics with view factorization. Stag-1 [27] advances 4D simulation through spatial-temporal decoupling and point cloud reconstruction, achieving photo-realistic, viewpoint-agnostic scene evolution. Meanwhile, StreetCrafter [32] and FreeVS [33] leverage diffusion models for controllable novel-view synthesis, expanding capabilities for urban scene generation. These methods primarily simulate natural traffic flows. In contrast, Challenger specifically targets adversarial videos, combining diffusion-based trajectory optimization with multiview rendering to expose model vulnerabilities under physically plausible yet deliberately challenging conditions.

2.3 End-to-End Autonomous Driving

End-to-end autonomous driving (E2E-AD) [34, 35, 36, 37, 8, 38, 39, 40] integrates perception, decision-making, and control into a unified, holistically optimized framework. Pioneering work by Bojarski et al. [34] demonstrated direct mapping from raw visual inputs to steering commands using convolutional neural networks, bypassing modular pipelines. Subsequent research enhanced robustness through uncertainty quantification [35] and adversarial training [41], with the latter introducing a two-stage paradigm where privileged agents guide vision-based policy learning, achieving state-of-the-art results on CARLA/NoCrash benchmarks. Recent advances emphasize multimodal trajectory prediction and interaction modeling. Methods like R-Pred [42], EigenTrajectory [43], and Singular-Trajectory [44] exemplify this trend. Safety validation techniques, including adversarial 3D object optimization [45] and traffic condition manipulation [46], generate challenging scenarios via closed-loop simulations. Neural rendering [47] further extends adversarial testing by perturbing scene textures. Scalability and uncertainty management are addressed through coverage-driven testing [48] and probabilistic trajectory prediction [49]. Architectural innovations, such as SparseAD [50] and GenAD [36], improve multi-task efficiency. Attention mechanisms [37] and multimodal fusion frameworks [51, 39] enhance interpretability and vehicle-to-everything (V2X) coordination, with UniAD [8] achieving top performance via task-aware planning. Despite progress, E2E-AD systems remain vulnerable to adversarial attacks [52]. Mitigation strategies include reinforcement learning for sensor failure resilience [53] and large language model integration for interpretable control [38]. OpenEMMA [40] promotes accessibility through multimodal toolkits. These models are evaluated under benign conditions and remain vulnerable to rare or adversarial events. Challenger directly addresses this limitation by synthesizing photorealistic adversarial videos that expose failure cases in E2E-AD systems, offering a practical and scalable complement to existing evaluation pipelines.

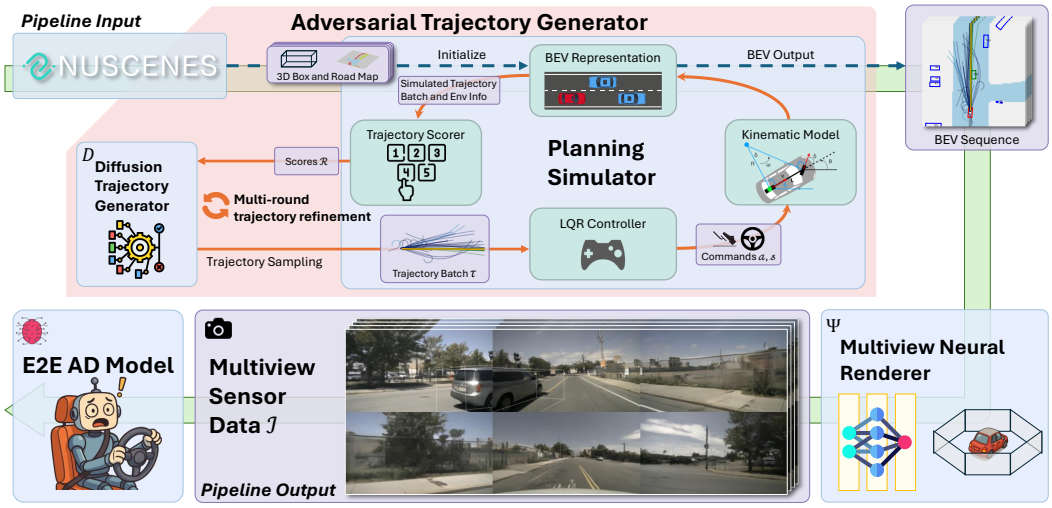


Figure 2: ***Challenger* Overview.** *Challenger* first ingests 3D bounding boxes and BEV road maps from a real-world dataset (e.g., nuScenes) to initialize a driving scene. It then randomly selects a background vehicle to act as the adversarial agent, while leaving all other participants unchanged. At fixed keyframes, *Challenger* plans the adversarial vehicle's trajectory and executes it continuously between keyframes. At each planning keyframe, a multi-round refinement process is used to explore the trajectory space efficiently and generate an adversarial one. In the first round, a batch of candidate trajectories $\tau \in R^{B\times T\times 2}$ is sampled from a diffusion-based generator. To ensure physical feasibility, a physics-aware planning simulator with an LQR controller and a kinematic model is used to simulate the motion of the adversarial vehicle. The simulated trajectories are then scored, and the top-performing trajectories are resampled (with replacement), perturbed by noise, denoised via the diffusion model, and fed back into the planning simulator for subsequent rounds. This iterative process continues for a fixed number of rounds in order to progressively narrow down the trajectory space in an efficient manner, after which the best simulated trajectory is selected as the final adversarial driving plan for that keyframe. The generated trajectory modifies the original driving scene, and this process is repeated across all keyframes. Finally, a multiview neural renderer synthesizes photorealistic video outputs of the final adversarial scenario.

3 Method

Our proposed *Challenger* comprises three key components: a diffusion-based trajectory generator (Section 3.1) that enables multi-round trajectory refinement (Section 3.4), a physics-aware planning simulator (Section 3.2) with a trajectory scorer (Section 3.3), and a multiview neural renderer (Section 3.5). Figure 2 provides an overview while Appendix A contains implementation details.

3.1 Diffusion Trajectory Generator

Diffusion Models. Diffusion models [54, 55, 56] are a class of generative models that learn to synthesize data by reversing a gradual noising process. During training, a forward diffusion process incrementally adds Gaussian noise to clean data over T steps, transforming an initial sample $\tau_0 \sim p(\tau_0)$ into a noise distribution $\tau_T \sim \mathcal{N}(0, I)$. The model is trained to approximate the reverse process by learning a denoising function $p_\theta(\tau_{t-1}|\tau_t)$, which estimates the distribution of the previous step given the current noisy input. To generate new data, the sampling process begins with pure noise τ_T and iteratively applies the learned denoising function over T steps, ultimately producing a sample τ_0 from the target data distribution. In the case of trajectory generation, diffusion models are promising because they can learn the distribution of driving trajectories in a data-driven manner, and are capable of generating diverse and complex driving behaviors.

Inside *Challenger*, before the adversarial vehicle can learn to drive adversarially, it must first acquire the fundamental ability to generate realistic driving behavior. To this end, we train an *unconditional* diffusion model that captures the distribution $p(\tau)$ of naturalistic driving trajectories, where each trajectory $\tau \in R^{T\times 2}$ is represented as a fixed-length sequence of 2D waypoints in the ego vehicle's local coordinate system. The model is trained on the nuPlan dataset [57], which provides a larger

and more diverse set of real-world trajectories than nuScenes. This enables the adversarial agent to implicitly learn the underlying distribution of human-like motion patterns. Once trained, the diffusion model can generate plausible driving trajectories from noise via iterative denoising. It can also refine existing trajectories through a truncated reverse process—adding partial noise and re-denoising—yielding trajectory variants with controlled stochasticity. This capability forms the foundation of the adversarial vehicle's learned behavior, enabling it to explore and execute diverse yet physically plausible maneuvers in downstream planning.

3.2 Physics-aware Planning Simulator

While a trained diffusion model enables *Challenger* to sample diverse driving plans, they are not guaranteed to be physically feasible for the selected background vehicle, nor inherently adversarial to the ego vehicle. This is due to: (1) the selected background vehicle may have different size and dynamics compared to the vehicles in the nuPlan dataset, where the model is trained; and (2) the generated trajectories are sampled without explicit awareness of the surrounding driving context.

Therefore, we introduce a physics-aware planning simulator. It first uses a linear-quadratic regulator (LQR) controller to track each sampled trajectory by generating control commands (steering angle and acceleration) for the adversarial vehicle, and then simulates the motion of the adversarial vehicle using a kinematic bicycle model [58], which captures essential vehicle dynamics and produces the resulting 3D bounding boxes over time. Parameters of the model are customized to match the selected adversarial vehicle's characteristics.

3.3 Trajectory Scorer

In order to incorporate contextual information and find the most adversarial trajectory, we design a scoring function. Ideally, it needs to distinguish contextually feasible trajectories from those that are not, while also identifying the most adversarial ones. Although it is theoretically possible to render the resulting driving scenes for every trajectory in the batch and use a surrogate AD model to find which trajectory is the most adversarial (resulting in the highest failure rate of the AD model), this approach is computationally expensive and impractical. Instead, we propose a scoring function that operates at an abstract level and thus can be computed efficiently. Specifically, a composite score is calculated based on three criteria (detailed in Appendix A.1): (1) drivable area compliance, which penalizes deviation from the legal road surface; (2) collision rate, which penalizes contact with static or dynamic objects, including the ego vehicle; and (3) adversarial challenge, which rewards close encounters with the ego vehicle.

3.4 Multi-round Trajectory Refinement

In order to find high-quality adversarial trajectories, a search through the trajectory space is needed. However, naively sampling a huge number of trajectories and picking the best one is computationally expensive. Instead, *Challenger* employs a multi-round refinement process inspired by recent advancements in diffusion-based planning [14]. This iterative approach balances exploration and exploitation, enabling the adversarial vehicle to find strategically challenging trajectories in an affordable manner, without the need of a prohibitively large batch size.

In the initial round, a batch of candidate trajectories $\tau \in R^{B\times T\times 2}$ is sampled from the diffusion model, simulated using the physics-aware planning simulator, and then scored. Subsequently, a resampling step selects trajectories from the current batch with replacement, assigning higher probabilities to those with superior scores. This results in a new batch $\tilde{\tau} \in R^{B\times T\times 2}$ that emphasizes promising candidates. To introduce diversity and facilitate exploration, Gaussian noise is added to these trajectories, and the diffusion model denoises them, producing a mutated batch $\hat{\tau}$ for the next round.

This iterative process is conducted over a predetermined number of rounds. In each iteration, the adversarial vehicle effectively learns to generate trajectories that are increasingly tailored to challenge the ego vehicle, all while adhering to environmental constraints. Please refer to Figure 6 for an intuitive illustration. The trajectory yielding the highest final score is selected as the optimal adversarial plan. *Challenger* also verifies that this trajectory avoids collisions with any objects, and that the ego vehicle can safely navigate the scene by following its original path in the dataset. If

Dataset	AD Model	1s	2s	3s	Avg	Degradation	AD Model	1s	2s	3s	Avg	Degradation
nuScenes-val	UniAD	0.10%	0.15%	0.61%	0.29%		SparseDrive	0.020%	0.063%	0.238%	0.107%	
nuScenes-val-R		0.07%	0.24%	0.64%	0.32%			0.010%	0.090%	0.370%	0.156%	
Adv-nuSc		**0.80%**	**4.10%**	**6.96%**	**3.95%**	12.6× ↑		**0.029%**	**0.618%**	**2.430%**	**1.026%**	8.6× ↑
nuScenes-val	VAD	0.11%	0.24%	0.42%	0.26%		DiffusionDrive	0.059%	0.068%	0.169%	0.099%	
nuScenes-val-R		0.19%	0.32%	0.59%	0.37%			0.030%	0.055%	0.403%	0.163%	
Adv-nuSc		**4.46%**	**7.59%**	**9.08%**	**7.05%**	26.1× ↑		**0.068%**	**1.299%**	**3.646%**	**1.671%**	15.9× ↑

Table 1: **Performance of End-to-End Autonomous Driving Systems Across Different Datasets.** We report the collision rate over 3 seconds and the average collision rate for each dataset. Tested E2E AD models perform much worse on adversarial driving scenarios generated by *Challenger*.

these conditions are not met, *Challenger* **discards** the scene entirely. This is to guarantee that every generated scenario presents a **challenging yet solvable** situation for the ego vehicle.

Together, the diffusion-based trajectory generator, physics-aware planning simulator, and multi-round refinement process empower *Challenger* to produce diverse and complex adversarial driving scenarios in an efficient and affordable manner, in the sense that: (1) the diffusion model captures a wide spectrum of driving behaviors, enabling the generation of varied trajectories that challenge the ego vehicle in multiple ways; (2) adversarial trajectory planning operates at an abstract level, without involving costly video rendering; and (3) the multi-round refinement process progressively narrows and enhances candidate trajectories, facilitating the affordable creation of high-quality adversarial scenarios without human supervision.

3.5 Multiview Neural Renderer

To synthesize photorealistic, multiview driving videos for our generated scenarios, we need a neural renderer that takes BEV maps and 3D bounding boxes of objects as conditional input, and generate photorealistic videos from six cameras surrounding the ego vehicle. It turns out that MagicDriveDiT [23] is a good candidate. It is a diffusion-based model that is trained on the nuScenes dataset, capable of generating high-fidelity videos that are long enough to cover the entire driving scene. Quantitative and qualitative results in Table 1, Table 3, and Figure 5 also validate this choice.

4 Experiments

4.1 Experiment Setup

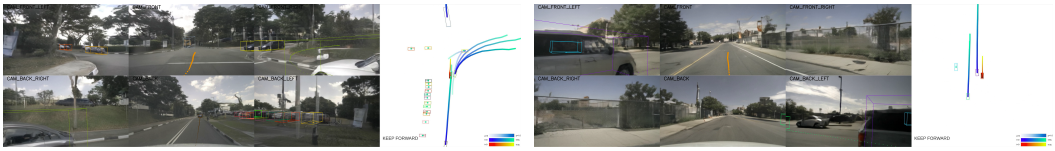
Datasets. We adopt the widely used nuScenes dataset [1] as the foundation for our proposed *Challenger* framework, enabling seamless integration with existing end-to-end autonomous driving (AD) models. It is a large-scale dataset containing 1,000 scenes where each scene is 20 seconds long. Each scene is annotated with 3D bounding boxes of objects, including vehicles, pedestrians, and obstacles, at a frequency of 2Hz, and each annotated frame can serve as a sample for an E2E AD system. Specifically, we use the nuScenes validation set, which contains 150 scenes (6,019 samples). To investigate potential distributional differences between the original camera data and our rendered outputs, we additionally re-render the original nuScenes validation set using our multiview neural renderer. We refer to this rendered version as *nuScenes-val-R*. To evaluate adversarial robustness, we generate a new dataset, *Adv-nuSc*, using *Challenger*. It contains 156 scenes (6,115 samples) and is specifically crafted to challenge the ego vehicle by introducing adversarial traffic participants.

Autonomous Driving Models. We evaluate four state-of-the-art vision-based end-to-end AD models: UniAD [8], VAD [9], SparseDrive [10], and DiffusionDrive [11]. For each method, we use official pretrained models and use the default settings for evaluation. All models take multiview camera images as input and predict the ego vehicle's motion trajectory for the next 3 seconds.

4.2 Results on the *Adv-nuSc* Dataset

We evaluate the performance of four end-to-end autonomous driving systems on three datasets: the original nuScenes validation set, the re-rendered *nuScenes-val-R*, and our *Adv-nuSc* dataset. Results are summarized in Table 1. We report the collision rate over 3 seconds—defined as the proportion of evaluated samples in which the planned trajectory collides with any object.

As shown in Table 1, all models perform comparably on the original nuScenes dataset and its re-rendered counterpart *nuScenes-val-R* with limited degradation, indicating that distribution shifts between these datasets do exist but are minimal. In contrast, all models exhibit a substantial increase



(a) Surpassing from the right. (b) Cut-in from the left.

Figure 3: Failure cases of an E2E AD model in adversarial scenarios generated by *Challenger*.

Settings	Collision Rate
Challenger	6.17%
- MTR	3.49%
- TS	0.13%
- MTR - TS	0.80%

Table 2: **Ablation study.** We show the effectiveness of the multi-round trajectory refinement process (MTR) and trajectory scoring process (TS).

Dataset	SC	IQ
nuScenes-val	0.8349	0.3801
nuScenes-val-R	0.8173	0.3471
Adv-nuSc	0.8093	0.3469

Table 3: **Quantitative evaluation of video quality.** We report the subject consistency (SC) and imaging quality (IQ) of the datasets.

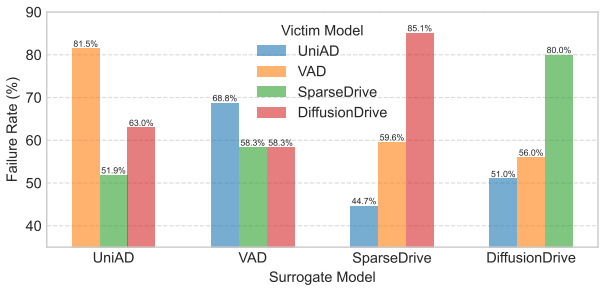


Figure 4: **Attack Transferability.** This figure illustrates the failure rates of victim models when evaluated on adversarial driving scenarios filtered based on the performance of a surrogate model. A failure is recorded when the victim model collides with another object in at least one sample of a given scenario. The failure rate is computed as the number of such scenarios divided by the total number of filtered scenarios.

in collision rates on the *Adv-nuSc* dataset. These results clearly demonstrate that our *Challenger* framework generates adversarial scenarios that effectively expose the failure modes of existing autonomous driving systems, highlighting critical gaps in their robustness and generalization. We also provide some qualitative examples in Section 4.3 to analyze how these adversarial driving scenarios challenge E2E AD systems.

4.3 Qualitative Examples of Adversarial Driving Scenarios

We visualize the behavior of the ego vehicle when controlled by a representative E2E AD model, UniAD, in two adversarial scenarios generated by *Challenger*, as shown in Figure 3.

In Figure 3a, the adversarial vehicle—outlined with a green bounding box—is positioned to the right of the ego vehicle. Its actual intention is to surpass the ego vehicle from the right lane. However, the UniAD model incorrectly predicts that the vehicle intends to turn right and thus fails to yield. This misjudgment likely stems from dataset bias, as vehicles in the right lane typically perform right turns during training, leading the model to generalize incorrectly.

In Figure 3b, the adversarial vehicle—highlighted in a purple bounding box—initiates a sudden cut-in from the left lane. The UniAD model, expecting the vehicle to maintain a straight path in its lane, continues forward without adjusting its trajectory. This behavior reflects the rarity of such aggressive maneuvers in the training data, limiting the model's ability to generalize to these out-of-distribution scenarios.

4.4 Attack Transferability

We further demonstrate that our proposed *Challenger* framework can be reformulated to perform black-box adversarial attacks on E2E AD models with significant success rates. To explore this, we integrate a surrogate model into the *Challenger* pipeline, generate multiple adversarial driving scenes and filter them by only keeping those in which the surrogate model fails to navigate safely in at least one sample.

This filtering strategy is designed to select scenarios that are sufficiently challenging and thus more likely to induce failure in other AD models. We subsequently evaluate the remaining victim models

Figure 5: **Representative video frames from datasets.** The re-rendered and adversarial datasets maintain photorealistic quality comparable to the original nuScenes dataset.

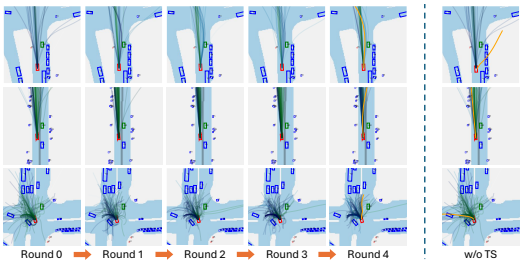


Figure 6: **Qualitative example of MTR and TS**. Please refer to Figure 1 for a legend.

on this curated subset. An attack is considered *transferable* if a victim model also experiences a collision in at least one sample of a scenario that was selected based on surrogate failure.

To quantify this effect, we conduct pairwise evaluations across the four tested E2E AD models, using each model in turn as a surrogate and assessing the failure rate of the remaining three as victims. Results, shown in Figure 4, reveal that a substantial portion of the filtered scenarios lead to failure in models other than the one used for filtering. This highlights *Challenger*'s ability to generate transferable adversarial driving scenes, even in a black-box setting. These findings raise critical safety implications, suggesting that current E2E AD systems may share common vulnerabilities and remain susceptible to transferable adversarial attacks.

4.5 Ablation Study.

To assess the contributions of the multi-round trajectory refinement and trajectory scoring processes within *Challenger*, we conduct an ablation study. We randomly select 20 scenes from the nuScenes validation set as initializations and run *Challenger* under various configurations. For each configuration, we report the percentage of samples that *Challenger* could successfully generate while the representative UniAD model failed to navigate safely. The results are summarized in Table 2. Additionally, we provide qualitative examples in Figure 6. Intuitively, with multi-round trajectory refinement, the candidate trajectories gradually converge to better trajectories. Without trajectory scoring, context information cannot be gathered, and the generated trajectory is not challenging.

4.6 Video Quality Evaluation

To evaluate the visual realism of the datasets used in our experiments, we conduct both qualitative and quantitative assessments. Representative examples are provided in Figure 5, covering various conditions such as daytime, evening, and rainy weather. For quantitative evaluation, we adopt two metrics: *subject consistency* and *imaging quality*. The detailed evaluation protocol is described in Appendix C, and results are summarized in Table 3.

Our findings show that both the re-rendered *nuScenes-val-R* and the adversarial *Adv-nuSc* datasets exhibit photorealistic quality that is visually comparable to the original nuScenes dataset. However, a slight distributional gap is observed between nuScenes and *nuScenes-val-R* in terms of quantitative metrics, reflecting minor artifacts introduced by neural rendering. Additionally, the *Adv-nuSc* dataset shows slightly lower subject consistency than *nuScenes-val-R*. We attribute this to the more aggressive and complex behaviors exhibited by adversarial vehicles in *Adv-nuSc*, which pose greater challenges for the neural renderer.

5 Conclusion and Future Work

In this work, we introduced *Challenger*, a framework for generating diverse, photorealistic driving scenarios to test end-to-end autonomous driving (E2E AD) systems. By combining a diffusion-based trajectory generator, a physics-aware simulator, and a neural renderer, *Challenger* creates sensor-level video data that exposes vulnerabilities in E2E AD models. Our results show increased failure rates and transferable adversarial behaviors across models, highlighting shared weaknesses in current paradigms. While effective, *Challenger* has limitations, including minor rendering discrepancies and a focus on open-loop, vision-based models. Future work includes extending to closed-loop systems and other modalities like LiDAR and HD maps.

References

[1] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[2] O. Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023.

[3] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, A. Geiger, and K. Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, Dec. 2024.

[4] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. Nov. 2024. URL https://openreview.net/forum?id=y09S5rdaWY#discussion.

[5] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):748, 2021.

[6] W. Ding, B. Chen, M. Xu, and D. Zhao. Learning to collide: An adaptive safety-critical scenarios generating method. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 2243–2250, Oct. 2020. doi:10.1109/IROS45743.2020.9340696. URL https://ieeexplore.ieee.org/abstract/document/9340696.

[7] W. Ljungbergh, A. Tonderski, J. Johnander, H. Caesar, K. Åström, M. Felsberg, and C. Petersson. *NeuroNCAP: Photorealistic Closed-Loop Safety Testing for Autonomous Driving*, volume 15088 of *Lecture Notes in Computer Science*, page 161–177. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-73403-8. doi:10.1007/978-3-031-73404-5_10. URL https://link.springer.com/10.1007/978-3-031-73404-5_10.

[8] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[9] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, page 8306–8316, Oct. 2023. doi:10.1109/ICCV51070.2023.00766. URL https://ieeexplore.ieee.org/document/10377337/?arnumber=10377337.

[10] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. (arXiv:2405.19620), May 2024. doi:10.48550/arXiv.2405.19620. URL http://arxiv.org/abs/2405.19620. arXiv:2405.19620.

[11] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, and X. Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. (arXiv:2411.15139), Nov. 2024. doi:10.48550/arXiv.2411.15139. URL http://arxiv.org/abs/2411.15139. arXiv:2411.15139 [cs].

[12] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature*, 615(7953):620–627, 2023.

[13] X. Yan, Z. Zou, S. Feng, H. Zhu, H. Sun, and H. X. Liu. Learning naturalistic driving environment with statistical realism. *Nature communications*, 14(1):2037, 2023.

[14] B. Yang, H. Su, N. Gkanatsios, T.-W. Ke, A. Jain, J. Schneider, and K. Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15342–15353, 2024.

[15] Y. Zheng, R. Liang, K. Zheng, J. Zheng, L. Mao, J. Li, W. Gu, R. Ai, S. E. Li, X. Zhan, et al. Diffusion-based planning for autonomous driving with flexible guidance. *arXiv preprint arXiv:2501.15564*, 2025.

[16] S. Tan, B. Ivanovic, X. Weng, M. Pavone, and P. Kraehenbuehl. Language conditioned traffic generation. *arXiv preprint arXiv:2307.07947*, 2023.

[17] J. Xia, C. Xu, Q. Xu, Y. Wang, and S. Chen. Language-driven interactive traffic trajectory generation. *Advances in Neural Information Processing Systems*, 37:77831–77859, 2024.

[18] S. Li, T. Azfar, and R. Ke. Chatsumo: Large language model for automating traffic scenario generation in simulation of urban mobility. *IEEE Transactions on Intelligent Vehicles*, 2024.

[19] C. Pan, B. Yaman, T. Nesti, A. Mallik, A. G. Allievi, S. Velipasalar, and L. Ren. Vlp: Vision language planning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14760–14769, 2024.

[20] R. Gao, K. Chen, E. Xie, L. Hong, Z. Li, D.-Y. Yeung, and Q. Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *ICLR*, 2024.

[21] S. Gao, J. Yang, L. Chen, K. Chitta, Y. Qiu, A. Geiger, J. Zhang, and H. Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems*, 37:91560–91596, 2025.

[22] B. Li, J. Deng, W. Zhang, Z. Liang, D. Du, X. Jin, and W. Zeng. Hierarchical temporal context learning for camera-based semantic scene completion. In *European Conference on Computer Vision*, pages 131–148. Springer, 2024.

[23] R. Gao, K. Chen, B. Xiao, L. Hong, Z. Li, and Q. Xu. Magicdrivedit: High-resolution long video generation for autonomous driving with adaptive control. *arXiv preprint arXiv:2411.13807*, 2024.

[24] B. Li, Y. Sun, Z. Liang, D. Du, Z. Zhang, X. Wang, Y. Wang, X. Jin, and W. Zeng. Bridging stereo geometry and bev representation with reliable mutual interaction for semantic scene completion. *arXiv preprint arXiv:2303.13959*, 2023.

[25] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, and Z. Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14749–14759, 2024.

[26] J. Mao, B. Li, B. Ivanovic, Y. Chen, Y. Wang, Y. You, C. Xiao, D. Xu, M. Pavone, and Y. Wang. Dreamdrive: Generative 4d scene modeling from street view images. *arXiv preprint arXiv:2501.00601*, 2024.

[27] L. Wang, W. Zheng, D. Du, Y. Zhang, Y. Ren, H. Jiang, Z. Cui, H. Yu, J. Zhou, J. Lu, et al. Stag-1: Towards realistic 4d driving simulation with video generation model. *arXiv preprint arXiv:2412.05280*, 2024.

[28] X. Wang, Z. Zhu, G. Huang, X. Chen, and J. Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *ECCV*, 2024.

[29] G. Zhao, X. Wang, Z. Zhu, X. Chen, G. Huang, X. Bao, and X. Wang. Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. *arXiv preprint arXiv:2403.06845*, 2024.

[30] R. Gao, K. Chen, Z. Li, L. Hong, Z. Li, and Q. Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024.

[31] B. Li, J. Guo, H. Liu, Y. Zou, Y. Ding, X. Chen, H. Zhu, F. Tan, C. Zhang, T. Wang, et al. Uniscene: Unified occupancy-centric driving scene generation. *arXiv preprint arXiv:2412.05435*, 2024.

[32] Y. Yan, Z. Xu, H. Lin, H. Jin, H. Guo, Y. Wang, K. Zhan, X. Lang, H. Bao, X. Zhou, et al. Streetcrafter: Street view synthesis with controllable video diffusion models. *arXiv preprint arXiv:2412.13188*, 2024.

[33] Q. Wang, L. Fan, Y. Wang, Y. Chen, and Z. Zhang. Freevs: Generative view synthesis on free driving trajectory. *arXiv preprint arXiv:2410.18079*, 2024.

[34] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[35] R. Michelmore, M. Kwiatkowska, and Y. Gal. Evaluating uncertainty quantification in end-to-end autonomous driving control. *arXiv preprint arXiv:1811.06817*, 2018.

[36] W. Zheng, R. Song, X. Guo, and L. Chen. Genad: Generative end-to-end autonomous driving. *arXiv preprint arXiv:2402.11502*, 2024.

[37] K. Chitta, A. Prakash, and A. Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15793–15803, 2021.

[38] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 2024.

[39] H. Yu, W. Yang, J. Zhong, Z. Yang, S. Fan, P. Luo, and Z. Nie. End-to-end autonomous driving through v2x cooperation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9598–9606, 2025.

[40] S. Xing, C. Qian, Y. Wang, H. Hua, K. Tian, Y. Zhou, and Z. Tu. Openemma: Open-source multimodal model for end-to-end autonomous driving. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1001–1009, 2025.

[41] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating, 2019. URL https://arxiv.org/abs/1912.12294.

[42] S. Choi, J. Kim, J. Yun, and J. W. Choi. R-pred: Two-stage motion prediction via tube-query attention-based trajectory refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8525–8535, 2023.

[43] I. Bae, J. Oh, and H.-G. Jeon. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10017–10029, 2023.

[44] I. Bae, Y.-J. Park, and H.-G. Jeon. Singulartrajectory: Universal trajectory predictor using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17890–17901, 2024.

[45] J. Sarva, J. Wang, J. Tu, Y. Xiong, S. Manivasagam, and R. Urtasun. Adv3d: Generating safety-critical 3d objects through closed-loop simulation. *arXiv preprint arXiv:2311.01446*, 2023.

[46] L. Zhang, Z. Peng, Q. Li, and B. Zhou. Cat: Closed-loop adversarial training for safe end-to-end driving. In *Conference on Robot Learning*, pages 2357–2372. PMLR, 2023.

[47] Y. Abeysirigoonawardena, K. Xie, C. Chen, S. Hosseini, R. Chen, R. Wang, and F. Shkurti. Generating transferable adversarial simulation scenarios for self-driving via neural rendering. *arXiv preprint arXiv:2309.15770*, 2023.

[48] J. Tu, S. Suo, C. Zhang, K. Wong, and R. Urtasun. Towards scalable coverage-based testing of autonomous vehicles. In *Conference on Robot Learning*, pages 2611–2623. PMLR, 2023.

[49] B. Kim, C. M. Kang, S. H. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network, 2017. URL https://arxiv.org/abs/1704.07049.

[50] D. Zhang, G. Wang, R. Zhu, J. Zhao, X. Chen, S. Zhang, J. Gong, Q. Zhou, W. Zhang, N. Wang, et al. Sparsead: Sparse query-centric paradigm for efficient end-to-end autonomous driving. *arXiv preprint arXiv:2404.06892*, 2024.

[51] A. Prakash, K. Chitta, and A. Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7077–7087, 2021.

[52] A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang. Attacking vision-based perception in end-to-end autonomous driving models. *Journal of Systems Architecture*, 110:101766, 2020.

[53] Z. Huang, S. Sun, J. Zhao, and L. Mao. Multi-modal policy fusion for end-to-end autonomous driving. *Information Fusion*, 98:101834, 2023.

[54] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, page 8780–8794. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

[55] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, page 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[56] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. Oct. 2020. URL https://openreview.net/forum?id=St1giarCHLP.

[57] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. (arXiv:2106.11810), Feb. 2022. doi:10.48550/arXiv.2106.11810. URL http://arxiv.org/abs/2106.11810. arXiv:2106.11810 [cs].

[58] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 812–818, 2017. doi:10.1109/IVS.2017.7995816.

[59] Z. Huang, Y. He, J. Yu, F. Zhang, C. Si, Y. Jiang, Y. Zhang, T. Wu, Q. Jin, N. Chanpaisit, Y. Wang, X. Chen, L. Wang, D. Lin, Y. Qiao, and Z. Liu. Vbench: Comprehensive benchmark suite for video generative models. page 21807–21818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

[60] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[61] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Appendix

This appendix provides additional insights and technical details regarding our proposed *Challenger*. We begin with an expanded discussion of implementation specifics in Appendix A. We then present a runtime analysis of *Challenger* in Appendix B. To assess video quality, we describe our evaluation protocol in Appendix C. Next, we provide details on the construction and filtering strategy of the *Adv-nuSc* dataset in Appendix D. We also describe how our framework can be extended to support multiple adversarial vehicles in the same scene in Appendix E. Finally, we showcase additional qualitative examples to illustrate common failure modes of state-of-the-art AD systems on our dataset in Appendix F.

A Implementation Details

In this section, we provide additional details regarding the implementation of our method. Algorithm 1 outlines the procedure for *Challenger* to generate an adversarial driving scene, as a complementary to Figure 2 in the main paper. Appendix A.1 further elaborates on the trajectory scoring process. Appendix A.2 formulates the resampling process in the multi-round trajectory refinement. Appendix A.3 provides the empirical values of the hyperparameters used in *Challenger*.

Algorithm 1 Adversarial Scene Generation.
The algorithm takes 3D bounding boxes and a BEV map from a real-world scene as input, and makes adversarial driving plans for an adversarial vehicle periodically. On each planning key frame, a multi-round trajectory refinement process is performed, and the final trajectory is used to update the 3D bounding boxes. The algorithm then renders the scene using the updated bounding boxes and BEV map.

Input: $Boxes, BEV$ {3D bounding boxes and BEV map from a real-world scene}
$cur_ts \leftarrow 0$ {Current timestamp, will iterate until the end of the scene}
while $cur_ts < scene_length$ **do**
  $\epsilon \leftarrow \mathcal{N}(0, I)$ {Gaussian noise}
  $\tau \leftarrow \text{TrajectoryDiffusionDenoiser}(\epsilon, T)$ {Sample an initial batch of trajectories, where T is the total number of diffusion steps}
  for $i \leftarrow 0$ **to** $\#refinement_rounds$ **do**
    $a, s \leftarrow \text{LQRController}(\tau)$
    $\tau_{sim} \leftarrow \text{KinematicBicycleModel}(a, s)$ {Simulate the trajectory}
    $R \leftarrow \text{TrajectoryScorer}(\tau_{sim}, Boxes, BEV)$
    if i is not last round of refinement **then**
      $\tilde{\tau} \leftarrow \text{MultinomialSampling}(\tau, R)$
      $\epsilon \leftarrow \mathcal{N}(0, I)$
      $\tau_{noised} \leftarrow \sqrt{\alpha_{t_{trunc}}} \cdot \tau + \sqrt{1 - \alpha_{t_{trunc}}} \cdot \epsilon$ {Add noise. t_{trunc} controls the amount of noise added.}
      $\hat{\tau} \leftarrow \text{TrajectoryDiffusionDenoiser}(\tau_{noised}, t_{trunc})$ {Denoise}
      $\tau \leftarrow \hat{\tau}$ {For next round of refinement}
    end if
  end for
  $\tau_{final} \leftarrow \tau_{sim,\text{argmax}(R)}$ {Select the best trajectory}
  $Boxes \leftarrow \text{ApplyTrajectory}(Boxes, \tau_{final})$ {Update 3d bounding boxes with the best trajectory}
  $cur_ts \leftarrow cur_ts + planning_horizon$ {Move forward to next planning key frame}
end while
$\mathcal{I} \leftarrow \text{MultiviewRendering}(Boxes, BEV)$ {Render the scene}
return $\mathcal{I}$ {Generated video frames}

A.1 Trajectory Scoring

This section provides further details on the scoring process used to evaluate simulated trajectory candidates in *Challenger*. Each candidate receives a final score that is a weighted sum of three components: (1) Drivable Area Compliance, (2) Collision Rate, and (3) Adversarial Challenge. In our setup, *lower* scores are *better*.

Before scoring, all trajectory candidates are transformed into global coordinates, and the corresponding 3D bounding boxes of the adversarial vehicle are computed. To introduce mild stochasticity into the multi-round selection process, each trajectory is initialized with a random score sampled uniformly from $[0, \mathrm{InitScale}]$.

For detailed empirical values of the hyperparameters used in the scoring process, please refer to Table 4.

Drivable Area Compliance. In a driving scene, we expect the adversarial vehicle to stay within the drivable area. We approximate it using five key points: the four corners and the center of its bounding box. These points are projected into the bird's-eye view (BEV) map at each timestep, yielding five binary mask sequences that indicate whether the projected points lie within the drivable area.

We define two types of violations: (1) A severe violation occurs if the vehicle's center point is outside the drivable area; (2) A mild violation occurs if any corner is outside the drivable area. This reflects the occasional human behavior of slightly veering off-road (e.g., driving a wheel over the curb).

The drivable area compliance score is computed as:

$$\begin{aligned}\mathrm{DACScore} = {} & \mathrm{CenterViolation} \odot \mathrm{DACStepWeight} \\ & + \mathrm{CornerWeight} \times \mathrm{CornersViolation} \odot \mathrm{DACStepWeight}\end{aligned}$$

where $\mathrm{CenterViolation} \in \{0,1\}^H$ and $\mathrm{CornersViolation} \in \{0,1\}^H$ indicate violations at each timestep $t = 1, \ldots, H$; $\mathrm{DACStepWeight} \in \mathbb{R}^H$ is a monotonically decreasing weight vector that penalizes earlier violations more heavily; $\odot$ denotes element-wise multiplication followed by summation over time; and $\mathrm{CornerWeight}$ is a scalar controlling the penalty ratio between center and corner violations.

Please note that our DAC term does permit the adversarial vehicle to drive in the opposite direction. Although briefly entering an oncoming lane for an aggressive overtake may not be legal, such maneuvers are common among aggressive human drivers. Therefore, AD models should be able to handle these situations.

Collision Rate. Collisions are undesirable for several reasons: (1) our physics-based simulator does not support accurate post-collision dynamics, (2) our neural renderer is trained only on collision-free scenes and cannot realistically render crash outcomes, and (3) scenes with collisions may offer no feasible solution for the ego vehicle, making them unsuitable for evaluating autonomous driving policies.

To detect collisions, we first compute pairwise distances between the center of the adversarial vehicle and all other vehicles. If the distance is smaller than the sum of their respective short edges, we flag a collision. If the distance exceeds the sum of their long edges, we deem them non-colliding. For ambiguous cases, we use geometric intersection tests on their bounding boxes. Once collision statuses are determined for each pair at every timestep, the collision rate score is calculated as:

$$\mathrm{CRScore} = \mathrm{AnyCollision} \odot \mathrm{ColStepWeight}$$

where $\mathrm{AnyCollision} \in \{0,1\}^H$ indicates if a collision occurs with any vehicle at each timestep; and $\mathrm{ColStepWeight} \in \mathbb{R}^H$ is a decreasing weight vector, assigning higher penalty to earlier collisions.

Adversarial Challenge. To encourage adversarial behavior, we reward trajectories where the adversarial vehicle approaches the ego vehicle closely. We compute the center-to-center distance between

the adversarial and ego vehicles at each timestep and use it to define the challenge score:

$$\begin{aligned}\text{ACScore}' =& \text{Dist} \odot \text{ACStepWeight} \\ \text{ACScore} =& \frac{ACScore' - \min(ACScore')}{\max(ACScore') - \min(ACScore')}\end{aligned}$$

where $\text{Dist} \in \mathbb{R}^H$ is the distance vector between the adversarial and ego vehicles; $\text{ACStepWeight} \in \mathbb{R}^T$ is an increasing vector, which encourages the adversarial vehicle to perform challenging maneuvers toward the end of the planning horizon; and $\text{ACScore}'$ is normalized to the range of $[0, 1]$.

Final Score. The final score for each trajectory is computed as a weighted sum of the three components:

$$\text{FinalScore} = \text{DACWeight} \cdot \text{DACScore} + \text{CRWeight} \cdot \text{CRScore} + \text{ACWeight} \cdot \text{ACScore}$$

where DACWeight, CRWeight, and ACWeight are scalar hyperparameters controlling the relative importance of each term.

A.2 Multi-round Trajectory Refinement

This subsection details the resampling process used in the multi-round trajectory refinement stage. At each refinement round, a new set of trajectory candidates is sampled based on their scores using a multinomial distribution. The selection probability for each trajectory is computed as:

$$\text{Prob} = \text{Softmax}(-\text{FinalScores} \cdot \text{Temperature})$$

Here, FinalScores are the scores assigned during the trajectory scoring step, and Temperature is a hyperparameter that modulates the sampling behavior. A higher temperature makes the distribution peakier, favoring top-scoring trajectories (i.e., approaching a winner-takes-all scheme), while a lower temperature promotes diversity by allowing exploration of lower-scoring candidates.

The empirical values of all hyperparameters used in this process are listed in Table 4.

A.3 Hyperparameters

This section presents the empirical values of the hyperparameters used in *Challenger*. These values were chosen based on general observations rather than exhaustive tuning, and were not optimized for any specific scenario. The complete list of hyperparameters is provided in Table 4. We emphasize that no thorough hyperparameter search was conducted, as the search space is too large, and room for improvement may exist. Nevertheless, with the listed settings, *Challenger* is able to produce high-quality adversarial driving videos within an affordable runtime.

B Runtime Analysis

We provide a detailed runtime breakdown of *Challenger*, highlighting the computational cost of each component involved in generating one adversarial driving scene. All experiments are conducted on a cloud server equipped with dual Intel(R) Xeon(R) Platinum 8457C CPUs, 4 NVIDIA L20 GPUs (48GB each), and 512GB RAM.

During adversarial trajectory generation, only one GPU is used for sampling trajectories from the diffusion model, while the physics-based simulation and trajectory scoring are primarily executed on the CPU. The multiview neural rendering stage is distributed across all four GPUs. In Figure 7 and Table 5, we present the runtime statistics for generating a typical scenario consisting of approximately 240 frames and six camera views.

As shown in the figure and table, multiview neural rendering is the most time-consuming component, accounting for nearly half of the total runtime, which is expected since it involves rendering high-resolution videos from six camera views across hundreds of frames—resulting in a large overall pixel volume. This highlights the efficiency of our adversarial trajectory generation process, since

Hyperparameter	Value
InitScale	1e-3
CornerWeight	0.2
H (planning horizon)	12 timesteps
DACStepWeight	Quadratically decreasing from 1.0 to 0.85
ColStepWeight	Quadratically decreasing from 1.0 to 0.85
ACStepWeight	Linearly increasing from 0.7 to 1.0
DACWeight	5
CRWeight	10
ACWeight	1
Temperature	1.0
#Refinement Rounds	5

Table 4: **Hyperparameters Used in *Challenger*.** This table lists the empirical values of key hyperparameters used throughout our pipeline. These settings are selected based on general observations and remain fixed across all experiments (except for the ablation study), without task-specific tuning.

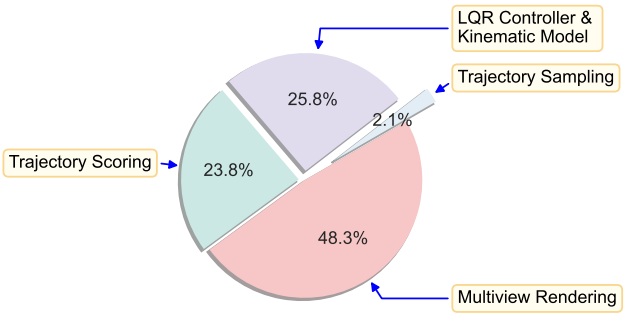


Figure 7: **Runtime Composition.** Percentage of total time spent on each major component in the *Challenger* pipeline.

Component	Runtime (s)
Traj. Sampling	17.36
Physics	211.36
Traj. Scoring	194.72
Rendering	395.18
Total	818.62

Table 5: **Runtime Breakdown.** Time spent on trajectory sampling, physics simulation (including the LQR controller and kinematic model), scoring, and multiview neural rendering.

the multi-round trajectory refinement and trajectory scoring process enables *Challenger* to generate high-quality adversarial driving plans without the need to render videos for every candidate trajectory. Without this design, evaluating each trajectory via rendering would incur a prohibitive computational cost, increasing the total runtime by orders of magnitude and making the approach infeasible.

The physics simulation and trajectory scoring components are also relatively time-consuming, primarily due to the batch-based nature of the refinement process and repeated simulations across multiple rounds. However, these steps are both necessary and cost-effective compared to rendering all trajectory candidates. Additionally, their current implementation is CPU-bound and has not been optimized for performance. We believe that with GPU acceleration or parallelized optimization, the runtime of these components can be significantly reduced.

C Video Quality Evaluation

This section describes the video quality evaluation protocol used in our experiments.

VBench [59] is a comprehensive benchmark suite designed to assess the quality of videos generated by general-purpose video generation models. However, it implicitly assumes a single dominant subject per video—typically one that persists throughout the sequence.

This assumption breaks down in the context of driving video generation, where scenes are much more complex. A driving scene often contains many objects (especially vehicles), and their presence within the frame can vary drastically over time—some vehicles may enter or exit the scene,

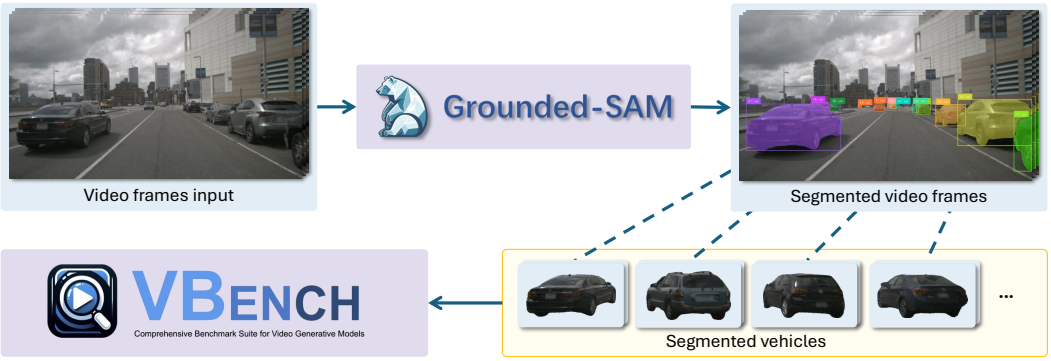


Figure 8: **Video Quality Evaluation Pipeline.** The video quality evaluation pipeline consists of two steps: (1) segmenting vehicles in the video frames using Grounded-SAM 2, and (2) computing the subject consistency metric and imaging quality metric from VBench [59].

occlude one another, or move in and out of view due to ego-motion. These characteristics violate the assumptions of VBench and require careful adaptation to ensure that quality metrics remain valid.

To address this, we adapt the VBench protocol using a two-stage process. Figure 8 illustrates the evaluation pipeline. First, we use Grounded-SAM 2 [60, 61] to segment all vehicles from each frame of the generated driving video using the text prompt "car." However, given the potential for over-segmentation or false positives, we refine the output using 3D annotations from the original dataset. Specifically, we project ground-truth 3D bounding boxes onto the image plane and discard any segmentation masks that do not overlap with known vehicle positions. This makes it more likely that the segmented vehicles are indeed present in the scene and relevant to the driving context.

Once valid vehicle instances are identified, we compute selected VBench metrics focused on these subjects. We evaluate imaging quality to assess spatial fidelity and subject consistency to measure temporal coherence across frames. Unlike the original VBench setting, our evaluation handles multiple dynamic subjects and allows for their appearance and disappearance across time—better reflecting the nature of driving scenarios. This adaptation enables a principled and scalable way to assess the realism and temporal quality of generated driving videos in complex multi-object settings.

D The *Adv-nuSc* Dataset

This section provides additional details on the construction of the *Adv-nuSc* dataset. It is built upon the validation split of the nuScenes dataset [1], which contains 150 scenes, each with 20 seconds of driving data. The dataset includes rich 3D annotations, multi-camera frames, and diverse adversarial instances designed to evaluate the robustness of autonomous driving systems.

For each scene in the validation split of nuScenes, we randomly select up to 10 background vehicles (if there are that many) that come close to the ego vehicle at any point in time and designate them as candidate adversarial agents. *Challenger* is then used to generate adversarial trajectories for these vehicles, creating diverse and challenging driving scenarios.

Before rendering, we apply a filtering step to discard any generated abstract scenes that meet any of the following conditions:

- The adversarial vehicle collides with another vehicle in the scene.
- The adversarial vehicle exits a 100m $\times$ 100m area centered around the ego vehicle.
- The adversarial vehicle never gets close to the ego vehicle (e.g., never appears directly in front, behind, or beside it at any time).

The remaining scenes are ranked by their adversarial challenge score, and the top-ranked scenarios are selected for rendering. To ensure quality, we also manually inspect a subset of randomly chosen scenes for sanity checking.

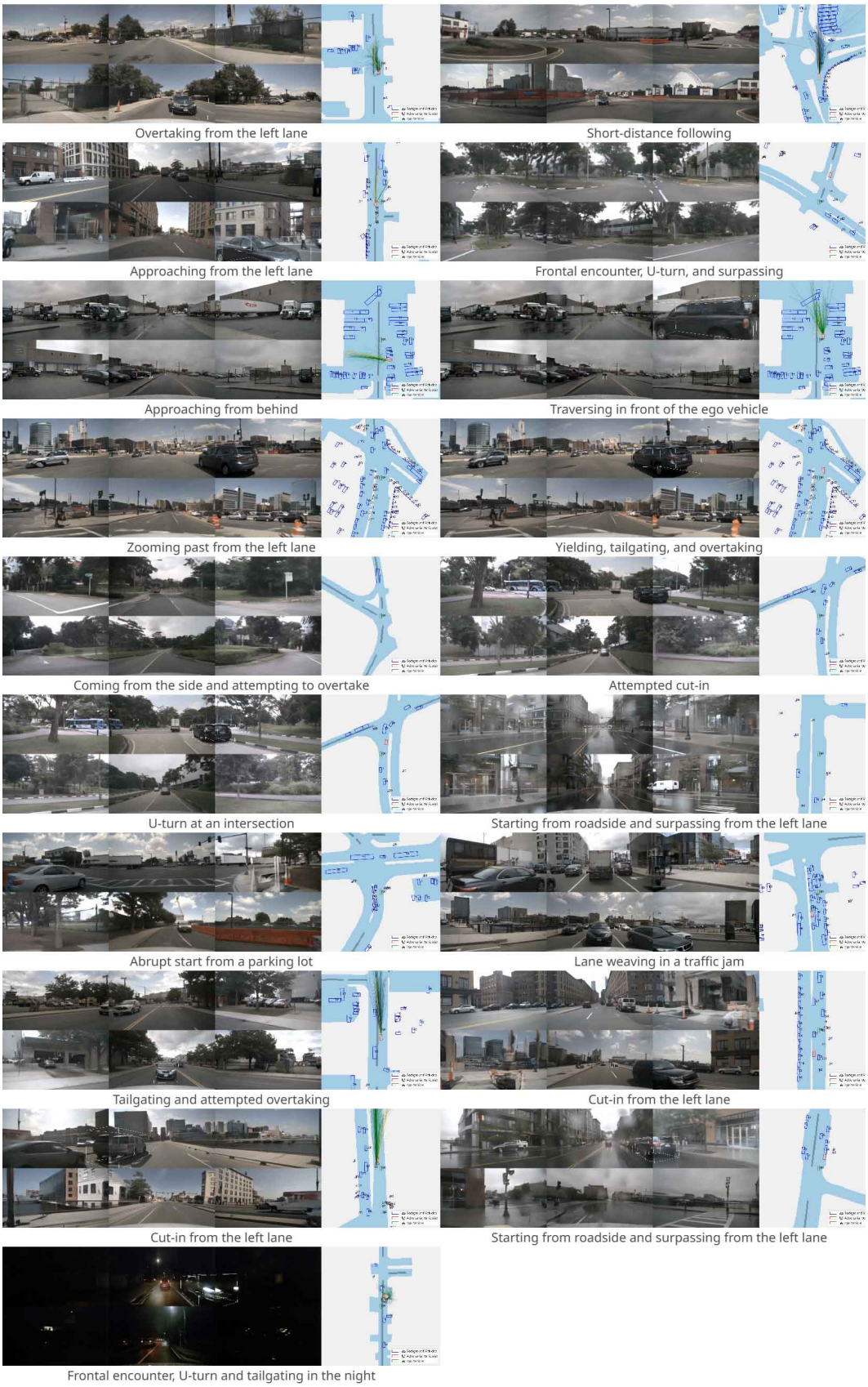


Figure 9: **Example Scenes from the *Adv-nuSc* Dataset.** We recommend the reader to have a look at our project page for dynamic videos.

Attribute	Count
# instances	12,858
# ego poses	254,436
# scenes	156
# samples	6,115
# sample data	254,436
# annotations	225,085

Table 6: **Overview of the *Adv-nuSc* Dataset.** Summary statistics for our adversarial driving dataset generated using the *Challenger* framework.

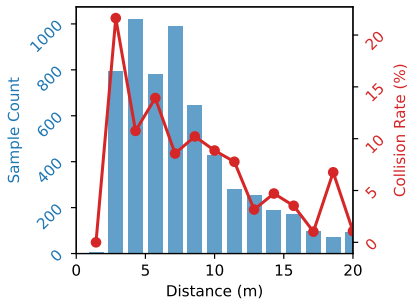


Figure 10: **Correlation between distance and collision rate.**

Table 6 presents the key statistics of the resulting *Adv-nuSc* dataset. Figure 10 shows the correlation between the distance to the adversarial vehicle and the collision rate of UniAD [8] on the generated scenarios. Figure 9 shows some example scenes from the dataset. We recommend that readers visit our project page for dynamic videos showcasing the generated scenarios.

E Extending to Multiple Adversarial Vehicles

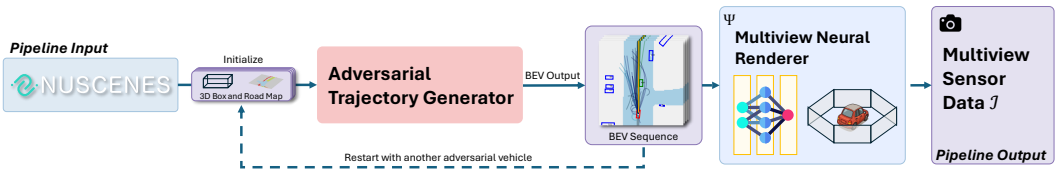


Figure 11: **Extending to Multiple Adversarial Vehicles.** The extension process is simple and modular. After generating a scenario with one adversarial vehicle and before rendering, we can simply feed the generated scene into *Challenger* again, and designate another vehicle as the adversarial vehicle.

This section outlines how *Challenger* can be extended to support multiple adversarial vehicles. While we do not currently model semantic interactions between adversarial agents, our framework allows multiple vehicles to challenge the ego vehicle within the same scene. Figure 11 illustrates the extension process.

The extension process is simple and modular. After generating a scenario with one adversarial vehicle and before initiating the rendering step, we re-feed the generated scene into *Challenger*, this time designating a different vehicle as the new adversarial agent. *Challenger* then generates adversarial trajectories for the newly selected vehicle, treating previously planned vehicles as part of the environment. This process can be repeated to add additional adversarial agents.

The final scene, with all adversarial vehicles included, is then rendered as usual. While each adversarial vehicle is optimized independently and coordination is not explicitly modeled, the sequential generation process ensures that later adversaries react to earlier ones. This setup enables the creation of complex, multi-threat scenarios that are useful for stress-testing autonomous driving systems.

Figure 12 shows some example scenes with multiple adversarial vehicles. The generated scenarios are complex and challenging, providing a rich set of conditions for evaluating the robustness of autonomous driving systems. We encourage readers to refer to our project page for example videos showcasing scenes with multiple adversarial vehicles.

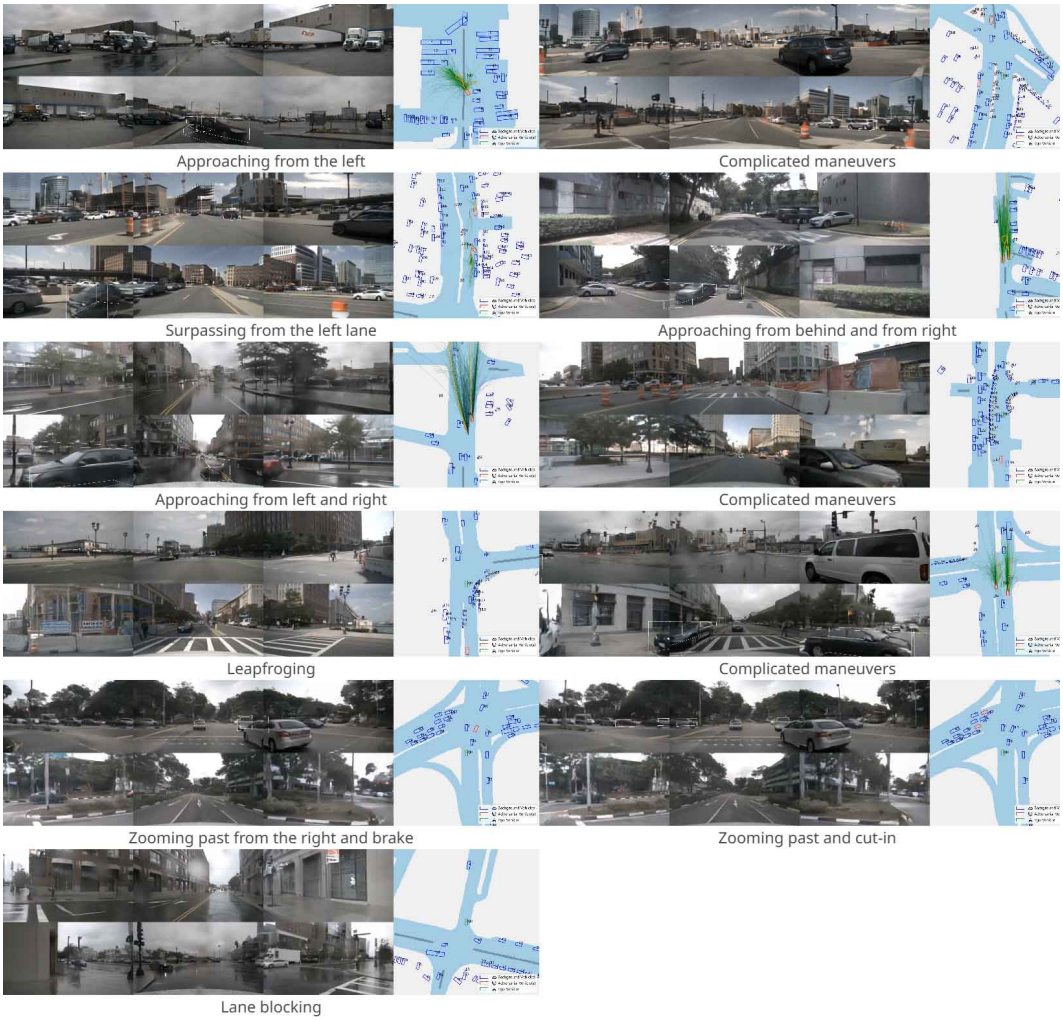


Figure 12: **Example Scenes with Multiple Adversarial Vehicles.** We recommend the reader to have a look at our project page for dynamic videos.

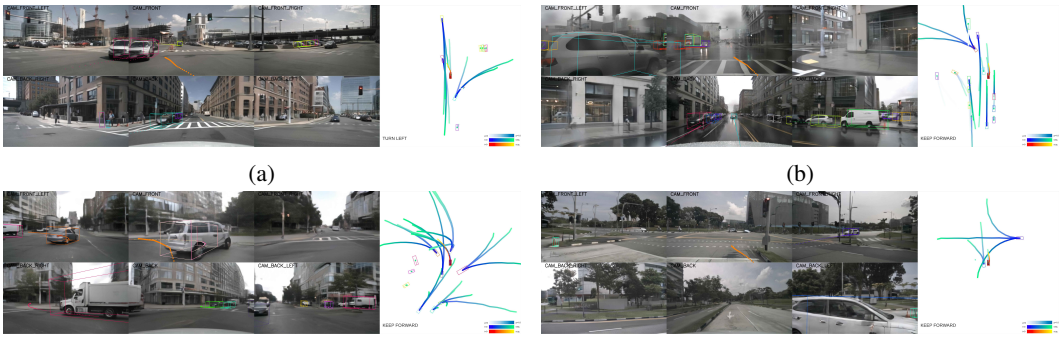
F Additional Qualitative Results

In this section, we provide additional analysis of the failure behaviors observed in end-to-end autonomous driving systems when evaluated on the *Adv-nuSc* dataset.

We manually reviewed a subset of failure cases involving two representative models—UniAD [8] and SparseDrive [10]—and identified two common failure modes: misforecasting and misplanning.

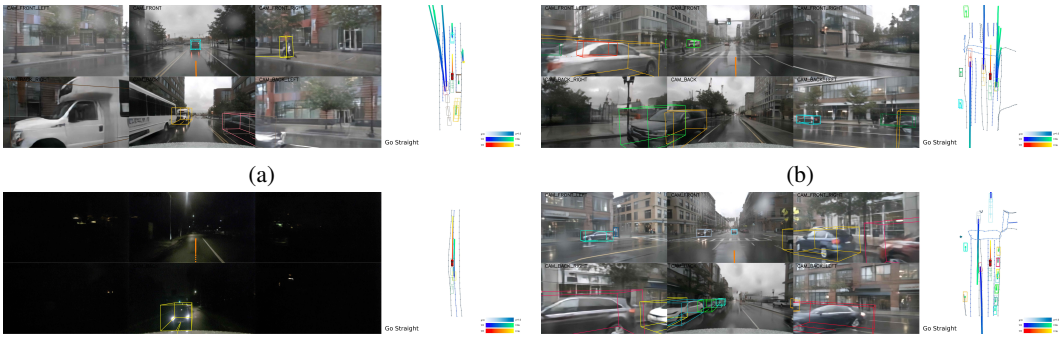
Misforecasting. The ego vehicle fails to accurately predict the future trajectory of the adversarial vehicle, resulting in a collision. We attribute this to the covariate shift between the naturalistic training data and the adversarial scenarios generated by *Challenger*. For example, UniAD [8] failed to anticipate that the white vehicle (magenta bounding box) would block its path in Figure 13a and Figure 13c, and failed to recognize upcoming cut-ins from the white vehicles in cyan and blue bounding boxes in Figure 13b and Figure 13d, respectively—leading to collisions. Similarly, SparseDrive [10] mispredicted the black vehicle (pink bounding box) in Figure 14a and the silver vehicle (yellow bounding box) in Figure 14b as non-threatening, failed to anticipate a surpass from the black vehicle in yellow in Figure 14c, and did not foresee the U-turn maneuver of the black vehicle in magenta in the back-left view of Figure 14d—all resulting in failures to avoid collisions.

Misplanning. Misplanning refers to situations where the autonomous driving model fails to generate a safe driving plan despite having reasonably accurate forecasts of surrounding agents. This typically indicates that the planning module is not robust enough to handle complex or adversarial scenarios. For example, UniAD [8] generated unsafe plans that led to a collision with a front vehicle and road curb when surrounded by nearby vehicles (Figure 15a), a collision with a vehicle on the right while being overtaken from the left (Figure 15b), a collision with a vehicle on the left while being overtaken from the right (Figure 15c), and a frontal collision with an oncoming vehicle when a silver car obstructed its path (Figure 15d). Similarly, SparseDrive [10] planned unsafe maneuvers, such as turning left into the path of two vehicles in the left lane (Figure 16a), turning left while forecasting a black vehicle overtaking from the left (Figure 16b), turning right and hitting the curb with an oncoming vehicle approaching from the left (Figure 16c), and turning left into a vehicle that was clearly blocking the way (Figure 16d). These failures are particularly concerning as the planned actions are inconsistent with the system's own forecasts, highlighting critical deficiencies in decision-making under adversarial conditions.



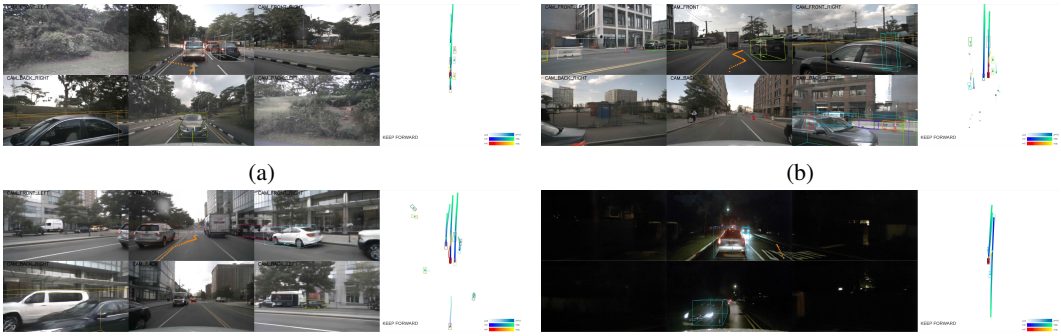
(a) (b) (c) (d)

Figure 13: Misforecasting failure cases of UniAD [8].



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

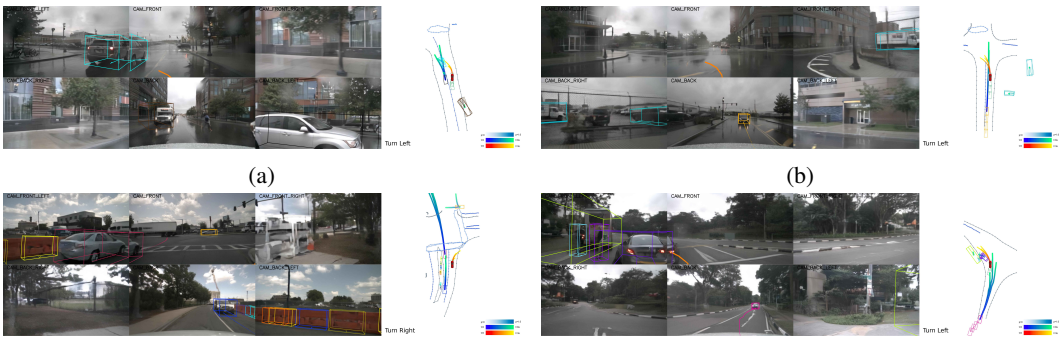
Figure 14: Misforecasting failure cases of SparseDrive [10].



(a) (b)

(c) (d)

Figure 15: Misplanning failure cases of UniAD [8].



(a) (b)

(c) (d)

Figure 16: Misplanning failure cases of SparseDrive [10].