2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

CRASH: Context-aware Recognition of Agents for Simulation of High-risk Driving

Minhee Cho*, Hayeon Jo*, Dongbo Min†

***Abstract*— Evaluating the safety of autonomous vehicles requires simulation of safety-critical scenarios such as potential collisions, which are difficult to reproduce in real-world environments. Prior methods rely on future trajectory predictions and heuristically select adversarial agents based on spatial proximity to the ego vehicle, often producing unrealistic scenarios that misalign with real-world temporal dynamics and contextual risk. To address these issues, we propose CRASH, the first learning-based adversarial agent selection approach that operates solely on past and present observations. It comprises two key components: (1) a Motion-Aware Masking (MAM) module that filters out static agents unlikely to collide with the ego vehicle due to negligible movement, and (2) an Adversarial agent Selection Module (ASM) that models contextual interactions to probabilistically estimate each agent's likelihood of inducing a collision with the ego vehicle. Experiments on the nuScenes and Waymo datasets demonstrate that CRASH significantly improves the success rate of generating realistic collision scenarios under both replay and rule-based planners, validating the effectiveness of context-aware agent modeling without access to future information.**

I. INTRODUCTION

Ensuring the safety of autonomous vehicles necessitates rigorous assessment of their responses under safety-critical scenarios such as potential collisions. However, such hazardous events occur rarely in real-world driving and are difficult to reproduce due to ethical and safety constraints. Consequently, simulation-based frameworks have become crucial for systematically assessing the reliability and robustness of autonomous systems. Simulations enable controlled and repeatable generation of critical events, allowing for precise quantitative analysis of autonomous vehicle behavior, particularly collision avoidance performance.

Recently, several model-based approaches [1], [2], [3] have been proposed to automatically generate safety-critical scenarios for autonomous vehicles. As illustrated in Fig. 1 (a), these methods typically identify adversarial agents that are likely to induce a collision by forecasting the future trajectories of all non-ego agents and selecting those in close proximity to the ego vehicle that refers to the autonomous vehicle under evaluation. Although intuitive, such designs face two fundamental limitations.

First, relying on full future trajectories deviates from realistic driving conditions, where such long-horizon predictions are infeasible. In practice, drivers and autonomous systems make decisions based on short-term observations rather than long-horizon predictions. Accessing precise future information several seconds in advance is not only unrealistic but also unnecessary. Adversarial agent selection grounded in past and current motion and interaction is therefore more consistent with real-world assumptions and, as shown in our experiments, results in higher collision success rates and improved scenario realism. Furthermore, the future trajectories used in prior methods are typically predicted under normal, collision-free driving conditions that are not designed with adversarial intents. Attempting to induce collisions based on these trajectories often leads to implausible scenarios and reduces the effectiveness of realistic scenario generation.

Fig. 1: **Conventional distance-based adversarial agent selection [1], [2] and our context-aware selection method.** (a) Distance-based selection designates the adversarial agent based on future proximity, which often results in the selection of static agents unlikely to collide with the ego vehicle. (b) Ours identifies the adversarial agent by probabilistically estimating collision likelihoods of non-ego vehicles based on their past and present interactions with the ego vehicle, thereby better reflecting real-world driving constraints.

Second, spatial proximity is an inadequate proxy for identifying truly threatening agents as it fails to capture critical contextual cues such as relative velocity, heading, and interaction dynamics. Heuristic strategies that always select the closest agent to the ego vehicle may overlook agents posing higher risks. For instance, Rempe et al. [1], which heuristically chooses the nearest agent and optimizes its trajectory, restricts adversarial scenario generation to a predetermined agent, thereby missing genuinely threatening agents and leading to less realistic scenarios.

To address these limitations, we introduce CRASH (***C**ontext-aware **R**ecognition of **A**gents for **S**imulation of **H**igh-risk driving*), a novel framework that formulates adversarial agent selection as a learnable task based on past and current observations. CRASH jointly optimizes agent selection and scenario generation through a unified, task-coupled loss. It is composed of two key components. First, a Motion-Aware Masking (MAM) module filters out static, non-threatening agents (*e.g.* parked vehicles) by excluding those with negligible recent motion and low likelihood

∗: Contributed equally, †: Corresponding author, The authors are with the Division of AI and Software at Ewha Womans University, Korea; {aezjk12, johayeon, dbmin}@ewha.ac.kr. This work was supported by the NRF of Korea grant (RS-2025-24803204).

of colliding with the ego vehicle. Second, an Adversarial agent Selection Module (ASM) probabilistically estimates the threat levels of non-ego agents by modeling their past and present interactions with the ego vehicle using a lightweight attention mechanism over spatio-temporal features. As illustrated in Fig. 1(b), this context-aware selection enables CRASH to generate realistic, diverse, and high-risk scenarios aligned with real-world driving behaviors.

Experimental results on the nuScenes and Waymo datasets [4], [5] demonstrate that CRASH significantly improves the success rate of critical scenario generation compared to prior methods [2], [1], [3] under both replay and rule-based planners. This highlights the value of context-aware agent modeling based on past and present contextual information. Our learnable framework is the first to achieve competitive performance without access to future trajectories, better aligning with the temporal and contextual constraints of real-world driving.

Our contributions are summarized as follows: **(1)** We introduce CRASH, the first learnable framework that jointly optimizes adversarial agent selection and scenario generation based solely on past and present observations through a unified, task-coupled loss. **(2)** We propose the Motion-Aware Masking (MAM) to exclude static, non-threatening agents from consideration and the Adversarial agent Selection Module (ASM) that probabilistically estimates the threat level of each agent by modeling temporal interactions with the ego vehicle. **(3)** We demonstrate the effectiveness of our method on the nuScenes and Waymo datasets, demonstrating consistent improvements in critical scenario generation under both replay and rule-based planners.

II. Related Work

A. Traffic Simulation.

Traffic simulation is essential for evaluating autonomous driving systems, as it enables evaluation across diverse scenarios that are difficult to reproduce in the real world. Existing approaches can be broadly categorized into heuristic and learning-based methods. Heuristic models such as IDM [6] specify explicit behavioral rules, but often fail to capture the complexity of real-world interactions. Learning-based approaches, in contrast, leverage real trajectory data to better imitate realistic traffic behaviors. Supervised approaches such as MultiPath [7], MultiPath++[8], Wayformer[9], SceneTransformer [10], and DenseTNT [11] model multimodal trajectories, each employing different strategies for goal or anchor selection. Generative approaches instead capture uncertainty to synthesize diverse scenarios, with TrafficSim [12] adopting a VAE framework and conditional VAE methods [13], [14] improving realism through probabilistic modeling. Recently, diffusion models originally developed for image synthesis [15], [16], [17], [18] have been adapted for traffic simulation and planning [19], [20], [21], [22], [23]. However, most existing work emphasizes common driving scenarios, leaving safety-critical long-tail events underexplored. These long-tail events are crucial to revealing vulnerabilities in autonomous driving systems, emphasizing the necessity for targeted simulation techniques to generate these scenarios.

B. Safety-Critical Traffic Simulation.

Generating safety-critical traffic scenarios has emerged as an important direction for exposing vulnerabilities in autonomous driving planners. Gradient-based methods leverage differentiable or kinematic dynamics to generate adversarial scenarios [24], [25], while black-box approaches perturb actions or trajectories to induce prediction errors [26], [27], [28]. Recent generative approaches employ VAEs [1], normalizing flows, GANs, and diffusion models [2], [3] to synthesize safety-critical events, with STRIVE [1] optimizing the latent space of a conditional VAE and diffusion-based methods incorporating adversarial objectives into the denoising process. Despite these advances, reliably generating planner-sensitive safety-critical scenarios remains challenging. We address this by introducing a VAE-based framework that enables end-to-end learning of safety-critical scenarios. Our method directly optimizes the latent space conditioned on planner feedback, improving the success rate of generating collision-inducing scenarios while maintaining realism and feasibility.

III. Problem Formulation

We consider a simulation comprising of N interactive traffic agents to evaluate the safety of autonomous driving planners. CRASH aims to generate safety-critical traffic scenarios that are rarely observed and difficult to intentionally collect in real-world driving environments. These scenarios impose stress conditions, enabling an evaluation of the planner's robustness under critical situations. We denote the state sequence of all N agents within a scenario as $\mathbf{Y} = \{Y_i\}_{i=1}^{N}$. The state of agent i at timestep t is characterized by $Y_i^t = [x_i^t, y_i^t, hx_i^t, hy_i^t, v_i^t, \omega_i^t]$, which covers the 2D position $q_i^t = (x^t, y^t)$, the heading vector $h_i^t = (hx_i^t, hy_i^t)$, the speed v_i^t, and the yaw rate ω_i^t. The timestep is divided into a historical timestep range $T_p \in [1, t_p]$ and a future timestep range $T_f \in [t_p + 1, t_f]$.

We model the $N-1$ non-ego agents to exhibit realistic and responsive behaviors, emulating real-world traffic dynamics in interaction with the ego vehicle. The future trajectory of the ego vehicle is predicted by a black-box planner conditioned on the historical trajectories of surrounding agents. Since the internal mechanisms of the planner are assumed to be inaccessible, no gradients or internal states can be observed or utilized. Consequently, CRASH cannot directly influence the ego agent, and adversarial interactions are restricted to non-ego agents. This assumption is consistent with prior methods [1], [2], [3], [25], which similarly treat the ego as a black-box planner and manipulate only non-ego agents with meaningful motions. The non-ego agents are represented through a reactive model f, which is initialized using a sequence of real-world data. The initialization process [1] produces latent vectors z_{init}, which are used as regularization to maintain realistic driving behavior during scenario generation. The model f consists of a scene context encoder that processes the HD map M and historical trajectories $\mathbf{Y}^{T_p}$, followed by a state decoder d. The state decoder

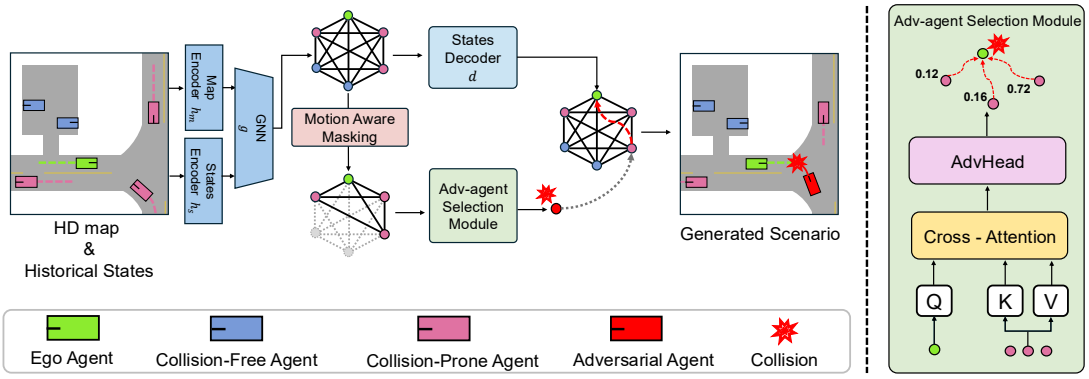

Fig. 2: **Overview of the CRASH framework.**The input scene comprises historical states and an HD map, which are encoded separately by h_s and h_m. The extracted features are processed by a GNN g to produce latent representations for each agent. A motion-aware masking strategy filters out static agents, and the Adversarial agent Selection Module (ASM) learns to identify agents with high collision potential. The selected adversarial agents and the remaining non-adversarial agents are then passed to a trajectory generation module, which jointly learns to produce collision-inducing and plausible collision-free trajectories, respectively.

d, following [1], operates auto-regressively. At each timestep t, it performs one round of message passing to model inter-agent interactions and then predicts the next position for each agent.

IV. Adaptive Adversarial Agent Selection.

A. Motivation and Overview

Recent methods [1], [2], [3] generate safety-critical scenarios by selecting one or a few non-ego agents as adversarial and optimizing their future trajectories to induce collisions with the planner-controlled ego vehicle. To maintain the framework planner-agnostic, these approaches restrict adversarial control to non-ego agents only, and typically rely on future trajectory prediction and apply heuristic rules, such as choosing the agents with a minimum distance to the future trajectory of the ego agent. However, this strategy presents two key limitations. First, it assumes access to full future trajectories of all agents, which is unrealistic in real-world settings, and long-horizon trajectory prediction is inherently uncertain and unreliable [29]. Second, spatial proximity offers a limited view of collision risk, as it ignores cues such as relative speed, heading, and interaction dynamics. The heuristic selection strategy that designates the nearest agent as adversarial can easily overlook agents with higher threat potential.

Our approach, CRASH, addresses the limitations of prior works by introducing a two-step scenario generation process. In Step 1, Motion-Aware Masking (MAM) filters collision-prone non-ego agents using only historical trajectories (Sec. IV-C). A learning-based Adversarial Selection Module (ASM) then adaptively selects adversarial agents from this masked set, moving beyond static, heuristic distance thresholds (Sec. IV-D). In Step 2, the trajectories of the selected adversarial agents are optimized to induce collisions with the ego, while the remaining agents follow plausible, collision-free behaviors (Sec. IV-E). To the best of our knowledge, CRASH is the first learnable framework that explicitly focuses on adversarial agent selection, while jointly optimizing both selection and scenario generation through a unified, task-coupled loss. This joint design tightly couples which non-ego agents are chosen with how their trajectories are generated, enabling realistic, safety-critical scenarios that more effectively reveal planner weaknesses. The overall architecture of CRASH is shown in Fig. 2.

B. Scene Context Encoding

Given the vectorized historical states $\mathbf{Y}^{T_p}$ of agents and the cropped map M, we first encode each modality separately to preserve modality-specific structure. Since these initial representations are often sparse and suboptimal for fusion, we employ MLP-Mixer networks [30] to obtain richer, information-dense embeddings. Specifically, the map M is processed via an MLP-Mixer encoder $\mathrm{h_m}$ to extract dense map features, while the historical agent states $\mathbf{Y}^{T_p}$ are encoded using another MLP-Mixer $\mathrm{h_s}$. The resulting embeddings are then passed into a Graph Neural Network (GNN) g_θ [1] to model intra-scene interactions. This encoding process is formally written as:

$$F_\mathrm{m} = \mathrm{h_m}(M), \quad F_\mathrm{s} = \mathrm{h_s}(\mathbf{Y}^{T_p}), \quad Z = g_\theta(F_\mathrm{s}, F_\mathrm{m}), \quad (1)$$

where F_m and F_s indicate the features extracted from the map and historical trajectories, respectively. The latent embeddings $Z \in \mathbb{R}^{N\times D}$ consists of $Z_{ego} \in \mathbb{R}^{1\times D}$ of a single ego agent and $Z_{agent} \in \mathbb{R}^{(N-1)\times D}$ of $N-1$ non-ego agents, where D is the feature dimension.

C. Motion-Aware Masking (MAM)

In real-world traffic, agents that remain stationary over sustained durations, *e.g.* parked vehicles, pose minimal collision risk to the ego vehicle. Including such agents during training introduces noise and dilutes the effectiveness of adversarial scenario generation. To encourage the model to prioritize high-risk interactions during training, we propose a Motion-Aware Masking (MAM) strategy that explicitly filters out non-ego agents with negligible motion. This design

follows from the black-box assumption of the ego planner, which restricts adversarial interactions to non-ego agents. For each non-ego agent i, we define a motion-aware mask M_i by computing an average motion d_i over the past T_p timesteps as:

$$d_i = \frac{1}{T_p} \sum_{t=1}^{T_p} \left\| q_i^t - q_i^{t-1} \right\|_2 , \quad M_i = \mathbb{I}(d_i > \delta) , \tag{2}$$

where $q_i^t \in \mathbb{R}^2$ denotes 2D position of agent i at timestep t, δ is a predefined threshold for motion distance, and $\mathbb{I}(\cdot)$ is an indicator function. The binary motion mask $M \in \mathbb{R}^{(N-1)\times 1}$ is then applied to filter out collision-free agents, generating the masked latent feature Z_{MAM}:

$$Z_{\text{MAM}} = M \odot Z_{agent} , \tag{3}$$

where $\odot$ denotes element-wise multiplication. This masked feature Z_{MAM} represents only non-ego agents exhibiting meaningful motion and is subsequently passed to the ASM. By excluding collision-free agents early in the pipeline, MAM ensures that the model prioritizes interactions with agents that pose a genuine threat, leading to more targeted and effective critical scenario generation.

D. Adversarial-agent Selection Module (ASM)

The ASM is designed to identify the most contextually threatening agent to the ego vehicle, based solely on past and present observations. It consists of two key components: (1) quantifying the relative interaction impact of each candidate agent on the ego vehicle, and (2) estimating the probability that each candidate agent induces a collision. To effectively capture interactions between the ego agent and the set of motion-filtered non-ego agents, we employ a cross-attention mechanism [31] between the two embeddings $Z_{\text{ego}} \in \mathbb{R}^{1\times D}$ and $Z_{\text{MAM}} \in \mathbb{R}^{(N-1)\times D}$. Specifically, the ego agent embeddings Z_{ego} serve as the query, attending over the keys and values from the non-ego agent embeddings Z_{MAM}. The influence of collision-prone agents on the ego is formulated using attention scores, conditioned on historical context:

$$\alpha = \text{Softmax}\left(\frac{Z_{\text{ego}} \cdot Z_{\text{MAM}}^{\top}}{\sqrt{d}} \right) Z_{\text{MAM}} , \tag{4}$$

where $\alpha \in \mathbb{R}^{(N-1)\times D}$ encodes interaction-aware representations of non-ego agents, modulated by their collision relevance to the ego vehicle. The attention representation α is passed through a lightweight decoder, denoted as AdvHead. It produces a probability distribution $P \in \mathbb{R}^{N-1}$ over the non-ego agents, which represents the likelihood of non-ego agents being selected as an adversarial agent to collide with the ego agent:

$$P = \text{AdvHead}(\alpha), \quad P \in \mathbb{R}^{N-1} . \tag{5}$$

The non-ego agent with the highest probability is selected as the adversarial agent at the current step in the scenario. This probabilistic approach enables the model to flexibly reason about contextual threat, in contrast to conventional deterministic approaches [1], [2], [25] based solely on spatial proximity. Moreover, ASM is fully differentiable, allowing end-to-end training of adversarial agent selection and subsequential trajectory generation within a unified framework. By leveraging attention-based interaction modeling and learnable threat inference, ASM facilitates the generation of more realistic and high-fidelity critical scenarios.

E. Collide with Planner

CRASH is designed to achieve two primary goals: to adaptively identify the most threatening agent using past and present context, and to generate its future trajectory to induce a collision with the ego agent. To jointly optimize these objectives, we introduce a unified loss that couples the adversarial agent selection with trajectory generation. The task-coupled loss encourages the selected adversarial agent, identified with high collision probability, to exhibit collision-inducing behaviors, while constraining non-adversarial agents to maintain realistic and safe trajectories. Based on the adversarial agent selection probability P, we define the adversarial collision loss L_{adv} as:

$$L_{\text{adv}} = \sum_{i=1}^{N-1} p_i \sum_{t=t_p+1}^{t_f} \left\| Y_i^t - Y_{\text{ego}}^t \right\|^2 + \sum_{t=t_p+1}^{t_f} \left\| Y_{\text{adv}}^t - Y_{\text{ego}}^t \right\|^2 ,$$
$$\text{adv} = \arg\max_i p_i . \tag{6}$$

The first term is a soft objective, where the proximity of each non-ego agent to the ego agent is weighted by its adversarial agent selection probability p_i for $i = 1, ..., N-1$, probabilistically encouraging adversarial behaviors. The second term is a hard objective that explicitly enforces collision between the ego agent and the most probable adversarial agent. Additionally, to ensure that non-ego agents not selected as adversaries continue to exhibit plausible driving behaviors, we introduce a regularization loss L_{reg}, following [1], combined with our proposed weight p_i:

$$L_{\text{reg}} = -\frac{1}{N-1} \sum_{i=1}^{N-1} (1 - p_i) \cdot \log g_\theta(z_i \mid \mathbf{Y}_{T_p}, M)$$
$$+ \frac{1}{N-1} \sum_{i=1}^{N-1} (1 - p_i) \left\| z_i - z_i^{\text{init}} \right\|^2 , \tag{7}$$

where z_i^{init} is the latent embedding from the initialization phase. The first term encourages the latent distribution to stay consistent with realistic behavior, while the second term penalizes deviations from the initial latent z_i^{init}. The total objective is given by:

$$L = L_{adv} + L_{reg} . \tag{8}$$

This formulation enables joint optimization of both adversarial agent selection and trajectory generation. Our key modification lies in the use of adversarial agent selection probability P in the loss functions. The prior methods [1], [2] using distance-based weights in (6) and (7) introduce a critical limitation: when non-adversarial agents are spatially close to the ego agent, the distance-based weighting encourages them to unintentionally stay closer to the ego agent, deviating from realistic driving behaviors. In contrast, by using the adversarial agent selection probability P, the proposed method ensures that non-adversarial agents with low p_i are strongly regularized to maintain realistic behaviors, while adversarial agents with high p_i effectively induce collision behaviors. Qualitative results are presented in Sec. V-E.

TABLE I: **EVALUATION OF GENERATED SAFETY-CRITICAL SCENARIOS.** WE COMPARE STRIVE [1], DIFFSCENE [3], SAFE-SIM [2], AND OUR METHOD.

Planner	Method	Collision (%) ↑	Other Offroad (%) ↓	Adv Offroad (%) ↓	Kinematic Value (m/s)
Replay	STRIVE	43.7	2.8	10.8	4.47
	Ours	65.4 (+21.7%)	1.5	4.3	6.46
Rule-based	STRIVE	36.4	2.2	11.4	5.52
	DiffScene	18.2	11.4	9.0	16.4
	Safe-Sim	43.2	1.8	11.4	-0.12
	Ours	64.8 (+21.6%)	1.4	10.5	8.01

TABLE II: **EVALUATION OF SCENARIO DIFFICULTY AND FEASIBILITY.** WE COMPARE STRIVE AND OUR METHOD UNDER RULE-BASED PLANNERS BY MEASURING SOLUTION RATES AND PLANNER TRAJECTORY STATISTICS.

Planner	Method	Solution Rate (%) ↑	Accel (%) ↑	Coll Vel (%) ↑
Rule-based	STRIVE	86.8	17.18	13.80
	Ours	87.9 (+1.1%)	44.79	65.09

V. EXPERIMENTS

A. Experimental Setup

Dataset. We conducted experiments using nuScenes [4], a large-scale real-world driving dataset containing 5.5 hours of data collected across two cities. The dataset consists of $20s$ traffic clips annotated at 2 Hz. The model was trained on the official training split and evaluated on the validation split, following the official data configuration of the nuScenes prediction challenge.

Baselines and Planners. We compared our approach against three baselines: STRIVE [1], DiffScene [3] and Safe-Sim [2]. These methods are recognized for their ability to generate safety-critical scenarios using generative models. Scenario generation was evaluated under two different planners used in prior work [1]: the *Replay* planner, which replays the ground-truth ego trajectory from nuScenes in an open-loop setting without re-planning, and the *Rule-based* planner, which enables a closed-loop setting by reacting to surrounding agents and re-planning at 5 Hz.

Evaluation Metrics. Our evaluation is based on several key metrics: the success rate of collision scenario generation between the ego and adversarial agents ("Collision"), the off-road rate of other agents ("Other Offroad"), the off-road rate of the adversarial agent ("Adv Offroad"), and the relative collision speed between the ego and adversarial agent ("Kinematic Value").

Implementation Details. Adversarial scenario optimization was implemented in PyTorch [32] using the ADAM [33] optimizer with a learning rate of 0.05. Following the setup of [1], we optimized for 300 iterations under the replay planner and 200 iterations under the rule-based planner, respectively. A threshold value of $\delta = 10^{-4}$ was used to mask static agents with negligible motion in (2). All experiments were conducted on a single NVIDIA RTX 3090 GPU with a batch size of 1.

B. Evaluation on Safety-Critical Scenario Generation

Our primary goal is to generate realistic, safety-critical scenarios that involve meaningful collisions, while preserving naturalistic behavior of non-ego agents. As shown in Tab. I, our method achieves the highest performance in terms of generating safety-critical scenarios compared to the baselines. Additionally, CRASH demonstrates that non-ego agents remain on the road, maintaining realistic driving behavior. This indicates that CRASH not only excels in creating challenging scenarios but also preserves realism, which is crucial for practical safety evaluations. The primary purpose of generating safety-critical scenarios is to provide non-trivial yet solvable situations, offering meaningful evaluation scenarios for planner assessment. CRASH effectively generates realistic collision scenarios while maintaining plausible non-ego agent behavior, demonstrating its robustness and applicability for safety validation in autonomous driving.

C. Evaluation of Scenario Difficulty and Solvability

To effectively evaluate the safety of autonomous vehicles, it is crucial that safety-critical scenarios are both challenging and solvable for planners. Scenarios that merely induce collisions without providing a solution lack practical utility in simulation. To quantitatively evaluate this aspect, we conducted experiments using the nuScenes dataset with Rule-based planners. To evaluate how challenging the scenarios are for planners, we measured two key metrics: acceleration and collision velocity, both quantifying the degree to which the planner changes its behavior to avoid collisions. Acceleration indicates how abruptly the planner must maneuver, with higher values representing more challenging scenarios for the planner. Collision velocity measures the relative speed between the planner and adversarial agent at the point of collision, where higher values indicate more severe and challenging collision scenarios. As shown in Tab. II, CRASH generates scenarios challenging enough to induce substantial behavioral changes in planners, achieving increases of 27.61% and 51.29% in acceleration and collision velocity, respectively, compared to the STRIVE baseline [1]. To assess the feasibility of the scenarios, we measure the solution rate. The solution rate represents the proportion of collision scenarios in which the planner successfully finds a solution, with higher values indicating more useful scenarios. Tab. II demonstrates that CRASH outperforms STRIVE in terms of solution rate, achieving 1.1% higher rates. Additionally, we qualitatively compared the two models in Fig. 3. The two figures on the left show a scenario generated by STRIVE, where unrealistic driving behavior causes the planner to fail to find a solution. In contrast, the two figures on the right present the scenario generated by CRASH, demonstrating a difficult yet feasible situation where the planner successfully finds a solution.

D. Ablation Study on CRASH Components

We conducted an ablation study to evaluate the impact of each module in our framework: Motion-Aware Masking (MAM) and Adversarial Agent Selection Module (ASM). Additionally, we compared our framework with the Distance-based Selection (DS) method [1] to demonstrate its effectiveness. All experiments were conducted using the closed-

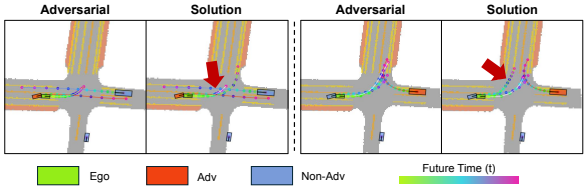

Fig. 3: **Qualitative comparison of adversarial scenarios using a rule-based planner.** The left side shows adversarial scenarios from STRIVE and their planner outcomes, while the right side shows those from our method. STRIVE fails due to unrealistic collision-driving cases, whereas our method generates solvable scenarios, indicated by red arrows.

TABLE III: **ABLATION STUDY ON THE CRASH COMPONENTS.** WE EVALUATE THE INDIVIDUAL EFFECT OF DISTANCE-BASED SELECTION (DS), MAM, AND ASM ON COLLISION METRICS AND NON-ADV STABILITY.

DS	MAM	ASM	Collision Metrics		Non-Adv Stability
			Collision (%) ↑	Adv Offroad (%) ↓	Other Offroad (%) ↓
×	✓	✓	64.8	10.5	1.4
×	×	✓	57.2	10.8	1.6
✓	✓	×	50.6	11.2	2.1
✓	×	×	36.4	11.4	2.2

loop rule-based planner. As shown in Tab. III, incorporating MAM into the conventional distance-based method [1] effectively filters out static agents with movements below a predefined threshold, leading to a reduced offroad rate for adversarial agents (row 3). This suggests that excluding collision-free agents mitigates unrealistic collision attempts and suppresses unnatural adversarial behavior. Additionally, MAM preserves stable offroad behavior for non-adversarial agents, contributing to the overall simulation stability. Although MAM combined with DS [1] enhances trajectory stability (as observed in rows 3 and 4), it yields limited effectiveness in generating collision-inducing behaviors. In contrast, ASM exploits interactions between the ego and non-ego agents to identify the most contextually threatening ones, enabling the successful generation of safety-critical scenarios (cf. rows 2 and 3). The integration of MAM and ASM (row 1) produces a synergistic effect, improving both trajectory stability and collision success rates. MAM filters out non-threatening agents, stabilizing the trajectory, while ASM focuses on selecting the most threatening agents based on contextual cues. Together, they enable more realistic and effective scenario generation.

E. Qualitative Analysis of Safety-Critical Scenarios

Fig. 4 provides a qualitative comparison between STRIVE [1] and our proposed method. For each case, we visualize the ego vehicle, the selected adversarial agent, and surrounding non-adversarial agents, along with their predicted trajectories.

In Fig. 4 (a), both methods select the same adversarial agent in nearly identical positions. However, only our method successfully induces a collision. This difference can be attributed to how the decoder of each method is trained to generate non-ego agent trajectories. STRIVE is trained with an emphasis on future proximity between the

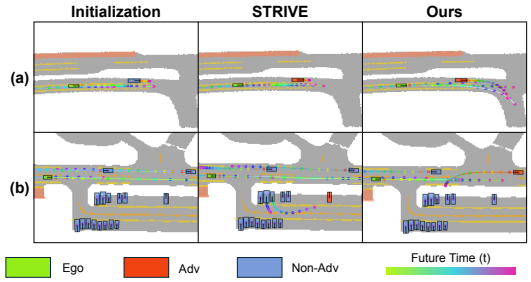

Fig. 4: **Qualitative comparison between STRIVE and our method using a rule-based planner.** (a) Scenario where both methods select the same adversarial agent. (b) Scenario where the two methods select different adversarial agents.

adversarial agent and the ego agent, which may encourage the agent to approach the ego without necessarily producing a collision. In contrast, our method is trained based on past and current interaction context between the ego and non-ego agents, enabling the decoder to generate trajectories where the selected agent is more likely to intersect with the ego's path or follow a more threatening route. As a result, the same adversarial agent exhibits qualitatively different behaviors under each method, leading to divergent collision outcomes.

In Fig. 4 (b), the two methods select different adversarial agents. STRIVE chooses a stationary vehicle based solely on its future proximity to the ego, which fails to produce a collision due to its lack of motion. In contrast, our method selects a dynamic agent with stronger interaction with the ego. This agent follows a trajectory that naturally intersects with the ego's path, resulting in a successful and plausible collision. These results highlight the limitations of proximity-based selection and the benefits of interaction-aware modeling in adversarial agent selection. Additionally, the prior methods that assign regularization and adversarial loss weights based solely on spatial proximity can inadvertently cause non-adversarial agents to move toward the ego. This limitation is evident in STRIVE's result, where the parked vehicle, although not selected as an adversarial agent, moves unnaturally, deviating from typical driving behavior. In contrast, our method accurately identifies an adversarial agent, maintaining realistic behavior while effectively inducing collision scenarios. This illustrates the effectiveness of the proposed ASM and probability-based loss weighting in producing plausible, safety-critical interactions.

F. Simulation Video

Simulation videos are available at the Project Page.

VI. CONCLUSION

We introduced CRASH, a context-aware framework for safety-critical scenario generation that selects adversarial agents based on past and present observations. By combining motion-aware masking and probabilistic selection, CRASH overcomes limitations of distance-based methods and avoids reliance on future trajectories. Experiments on nuScenes confirm consistent improvements under replay and rule-based planners, demonstrating its effectiveness in generating realistic safety-critical scenarios.

REFERENCES

[1] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 305–17 315.

[2] W.-J. Chang, F. Pittaluga, M. Tomizuka, W. Zhan, and M. Chandraker, "Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries," in *European Conference on Computer Vision*. Springer, 2024, pp. 242–258.

[3] C. Xu, A. Petiushko, D. Zhao, and B. Li, "Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 8, 2025, pp. 8797–8805.

[4] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[5] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9710–9719.

[6] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[7] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[8] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.

[9] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2980–2987.

[10] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified multi-task model for behavior prediction and planning," *arXiv preprint arXiv:2106.08417*, vol. 2, no. 7, 2021.

[11] J. Gu, Q. Sun, and H. Zhao, "Densetnt: Waymo open dataset motion prediction challenge 1st place solution," *arXiv preprint arXiv:2106.14160*, 2021.

[12] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "Trafficsim: Learning to simulate realistic multi-agent behaviors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 400–10 409.

[13] Y. J. Ma, J. P. Inala, D. Jayaraman, and O. Bastani, "Diverse sampling for normalizing flow based trajectory forecasting," *arXiv preprint arXiv:2011.15084*, vol. 7, no. 8, 2020.

[14] C. Schöller and A. Knoll, "Flomo: Tractable motion prediction with normalizing flows," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7977–7984.

[15] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[16] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8633–8646, 2022.

[17] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[19] D. Rempe, Z. Luo, X. Bin Peng, Y. Yuan, K. Kitani, K. Kreis, S. Fidler, and O. Litany, "Trace and pace: Controllable pedestrian animation via guided trajectory diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 756–13 766.

[20] Z. Zhong, D. Rempe, Y. Chen, B. Ivanovic, Y. Cao, D. Xu, M. Pavone, and B. Ray, "Language-guided traffic simulation via scene-level diffusion," in *Conference on Robot Learning*. PMLR, 2023, pp. 144–177.

[21] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone, "Guided conditional diffusion for controllable traffic simulation," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 3560–3566.

[22] C. Jiang, A. Cornman, C. Park, B. Sapp, Y. Zhou, D. Anguelov *et al.*, "Motiondiffuser: Controllable multi-agent motion prediction using diffusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9644–9653.

[23] M. Jiang, Y. Bai, A. Cornman, C. Davis, X. Huang, H. Jeon, S. Kulshrestha, J. Lambert, S. Li, X. Zhou *et al.*, "Scenediffuser: Efficient and controllable driving simulation initialization and rollout," *Advances in Neural Information Processing Systems*, vol. 37, pp. 55 729–55 760, 2024.

[24] Y. Cao, C. Xiao, A. Anandkumar, D. Xu, and M. Pavone, "Advdo: Realistic adversarial attacks for trajectory prediction," in *European Conference on Computer Vision*. Springer, 2022, pp. 36–52.

[25] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger, "King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients," in *European Conference on Computer Vision*. Springer, 2022, pp. 335–352.

[26] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "Advsim: Generating safety-critical scenarios for self-driving vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9909–9918.

[27] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 159–15 168.

[28] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8271–8277.

[29] D. Park, H. Ryu, Y. Yang, J. Cho, J. Kim, and K.-J. Yoon, "Leveraging future relationship reasoning for vehicle trajectory prediction," *arXiv preprint arXiv:2305.14715*, 2023.

[30] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *Advances in neural information processing systems*, vol. 34, pp. 24 261–24 272, 2021.

[35] 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.

[36] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE intelligent vehicles symposium (IV)*. IEEE, 2015, pp. 1094–1099.

[37] 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)*. IEEE, 2017, pp. 812–818.

[38] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[39] 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*. PMLR, 2023, pp. 2357–2372.

Supplementary Material

VII. Implementation Details

The CRASH model, introduced in Sec B and C of the main paper, is further elaborated here with a focus on its main architectural components.

Encoder Network. We employ separate MLP-Mixer networks [34] as encoders to embed the 2D HD map M and the 2D spatial trajectories of historical agent states $\mathbf{Y}^{T_p}$ into information-dense representations. In the main paper, the encoding processes are briefly represented as $\mathrm{h_m}(M)$ and $\mathrm{h_s}(\mathbf{Y}^{T_p})$. These encoders iteratively process inputs through two separate MLP-Mixer networks, each formulated as:

$$\begin{aligned} \mathrm{h_m}(M): \quad & F_m = M + \mathrm{MLP}_m(M^\top)^\top, \\ & F_m \leftarrow F_m + \mathrm{MLP}_m(F_m), \end{aligned} \tag{9}$$

$$\begin{aligned} \mathrm{h_s}(\mathbf{Y}^{T_p}): \quad & F_s = \mathbf{Y}^{T_p} + \mathrm{MLP}_s((\mathbf{Y}^{T_p})^\top)^\top, \\ & F_s \leftarrow F_s + \mathrm{MLP}_s(F_s), \end{aligned} \tag{10}$$

After applying multiple MLP mixing layers, we perform average pooling along the vector dimension to obtain the final embedding representations $F_m, F_s \in \mathbb{R}^{N\times D}$, following the implementation used in [35].

Graph Neural Network. The encoder takes as input a fully-connected scene graph, where each node represents the context feature of an agent. The context feature is constructed by fusing the past trajectory F_s and local map information F_m. The encoder is built upon a Graph Neural Network (GNN) architecture, modeling node-to-node interactions via a single round of message passing, following [1]. After computing all features, they are aggregated into a single interaction feature $e_i \in \mathbb{R}^{128}$ using max-pooling. Finally, each node's context feature and the aggregated interaction feature e_i are processed by a 4-layer MLP-based update network to produce the agent's latent vector z_i.

States Decoder. The state decoder d operates in an autoregressive manner, predicting future states one step at a time, as proposed in prior work [1]. At each timestep, the input for each agent is constructed by concatenating the latent representation Z from the GNN, the previous state $\mathbf{Y}^{t-1}$, and the map feature F_{m}^{t-1}. At rollout step t, this concatenated input is processed by a 3-layer MLP with a hidden size of 128, resulting in a 64-dimensional feature vector for each node. Based on this representation, the decoder predicts the next state $\mathbf{Y}^t$ using a kinematic bicycle model [36], [37]. The predicted state is then fed into the next decoding step as input. This process is repeated over the future horizon, enabling the model to sequentially update agent states in accordance with the scenario progression.

AdvHead in Adversarial-agent Selection Module (ASM). Within the ASM, the AdvHead estimates each agent's threat level from attention scores. These are processed through an MLP with non-linear activations and a softmax, yielding a probability distribution over collision likelihoods. The probabilities guide adversarial agent selection and weight the adversarial loss in (6), enabling generation of physically plausible collision scenarios.

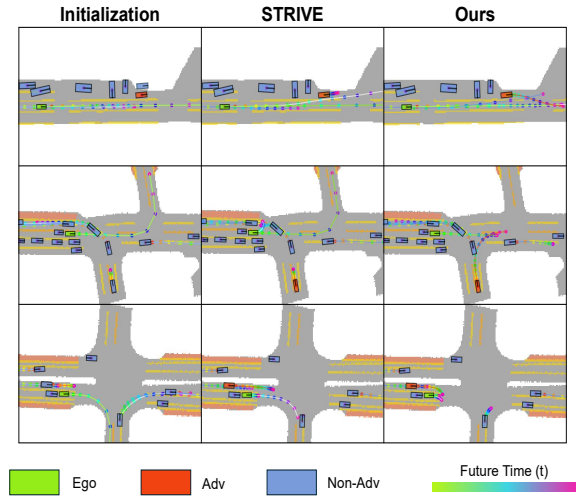

Fig. 5: **Additional qualitative comparison between STRIVE and our method using a rule-based planner.** While both approaches target the same adversarial agent, STRIVE fails to induce a collision. In contrast, our method generates a more plausible adversarial behavior that successfully leads to a collision, while maintaining realism.

TABLE IV: **Extended evaluation on the Waymo dataset.** We compare STRIVE, CAT, and CRASH under Replay, IDM, and RL-based planners.

Method	Replay	IDM [6]	RL-based planner [38]
STRIVE [1]	85.0%	85.0%	66.0%
CAT [39]	91.0%	86.0%	69.0%
CRASH	**92.4%**	**87.7%**	**75.5%**

VIII. Extended Evaluation

We extended our evaluation to the Waymo Open Motion Dataset [5] and compared CRASH with STRIVE [1] and CAT [39] under Replay, IDM [6], and RL-based planners [38]. As shown in Tab. IV, CRASH consistently achieves the highest success rates across all settings, confirming its robustness and scalability under diverse datasets, planners, and baselines.

IX. Discussion of ASM Design

In existing methods, the adversarial agent is selected by directly relying on distances computed from predicted future trajectories, making future trajectory distance an explicit and essential input to the selection process. In contrast, our ASM outputs a probability distribution over candidate agents based solely on past and present observations and does not access any predicted future trajectories during the selection process.