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

Controllable Collision Scenario Generation via Collision Pattern Prediction

Pin-Lun Chen[1], Chi-Hsi Kung[2], Che-Han Chang[3], Wei-Chen Chiu[1], Yi-Ting Chen[1,†]

***Abstract*— Evaluating the safety of autonomous vehicles (AVs) requires diverse, safety-critical scenarios, with collisions being especially important yet rare and unsafe to collect in the real world. Therefore, the community has been focusing on generating safety-critical scenarios in simulation. However, controlling attributes such as collision type and time-to-accident (TTA) remains challenging. We introduce a new task called controllable collision scenario generation, where the goal is to produce trajectories that realize a user-specified collision type and TTA, to investigate the feasibility of automatically generating desired collision scenarios. To support this task, we present COLLIDE, a large-scale collision scenario dataset constructed by transforming real-world driving logs into diverse collisions, balanced across five representative collision types and different TTA intervals. We propose a framework that predicts *Collision Pattern*, a compact and interpretable representation that captures the spatial configuration of the ego and the adversarial vehicles at impact, before rolling out full adversarial trajectories. Experiments show that our approach outperforms strong baselines in both collision rate and controllability. Furthermore, generated scenarios consistently induce higher planner failure rates, revealing limitations of existing planners. We demonstrate that these scenarios fine-tune planners for robustness improvements, contributing to safer AV deployment in different collision scenarios. Additional generated scenarios are available at this project webpage.**

I. Introduction

Evaluating the reliability of autonomous vehicles (AVs) demands testing in diverse, safety-critical scenarios with varying attributes, including scenario categories (e.g., lane change, junction crossing), traffic participant types and states (e.g., time to accident), and road topologies [1]–[5]. Among these, collision scenarios stand out as especially critical, since they directly test an AV's ability to anticipate, react, and ensure safety under high-risk conditions. However, collision scenarios are statistically rare and inherently unsafe for testing in the real world. Therefore, the community has been focusing on generating safety-critical scenarios in simulation [1], [6] and controlling attributes for diverse scenario generation [1].

In this work, we propose a new task: controllable collision scenario generation. Given a target collision type and a desired time-to-accident (TTA), the goal is to generate trajectories that realize the specified collision. Prior work has primarily focused on adversarial trajectories that force a crash with the test vehicle. However, they lack explicit controllability over key aspects of the scenario, such as varying the granularity of collision types or adjusting the time-to-accident. For example, recent advances [7], [8] can generate accident-prone scenarios, but these are often limited to specific accident types, such as rear-end or head-on crashes (see Fig. 6 in [7]). More recently, conditional generation with diffusion models [9], [10] has opened new possibilities for controllable collision scenario generation, yet existing formulations have not addressed fine-grained controllability. These gaps motivate our investigation into what it would take to achieve controllable collision scenario generation.

To this end, we introduce COLLIDE, a large-scale collision scenario dataset specifically designed for controllable collision scenario generation. Existing collision scenario datasets [11]–[17] lack collision-type annotations and do not provide scenarios with varying time-to-accident (TTA). To address this, we propose an automatic pipeline that transforms non-collision trajectories from real-world driving logs into diverse collision scenarios. Specifically, we select target collision types based on common accident categories defined

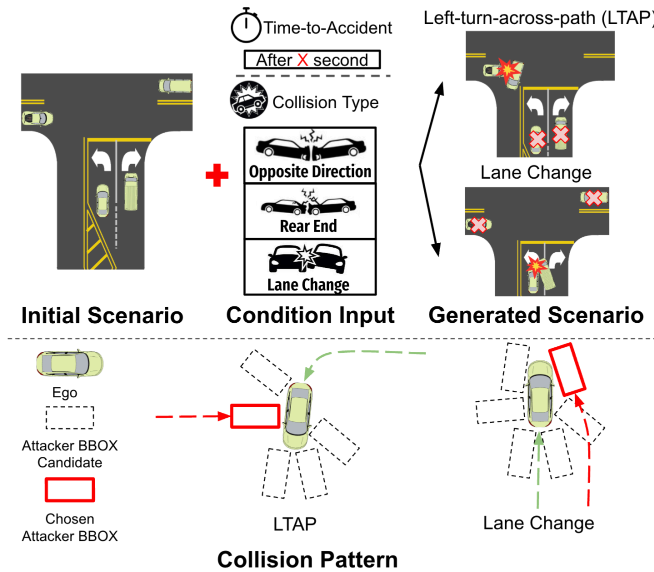

Fig. 1: Given an *initial scenario*, users specify condition inputs including collision type and time-to-accident (TTA). The framework predicts a *Collision Pattern*, defined as the relative configuration between the ego vehicle (green car) and the attacker (red bounding box) at the collision moment. This predicted pattern then guides the trajectory planner to generate a feasible attacker motion (red arrow), resulting in controllable *generated scenarios* such as lane change or LTAP.

† Corresponding Author.
[1] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu City, Taiwan. yp201141413.en11, ychen@nycu.edu.tw
[2] School of Informatics, Computing, and Engineering, Indiana University, Bloomington, Indiana, USA. kung@iu.edu
[3] MediaTek Inc.

by the U.S. National Highway Traffic Safety Administration (NHTSA) [18]. For a given ego trajectory, we choose a timestamp and extract the corresponding ego bounding box as a reference. An adversarial vehicle is then placed at all feasible spatial locations. The configuration that satisfies the definition of the desired collision type (see the bottom of Fig. 1) is selected. Finally, given the spatial configuration of the ego and adversarial vehicles and the specified TTA, we use a quintic polynomial planner to generate a plausible trajectory. This pipeline enables scalable and controllable collision scenario generation, supporting systematic evaluation of the corresponding algorithms. This raises a question: how can we automatically generate collision scenarios given a collision type and TTA?

We argue that the key to generating collision scenarios lies in contextualizing *collision type*. For example, a *Lane Change* collision evokes a rough pattern, as illustrated in the bottom part of Fig. 1. A Left-Turn-Across-Path (LTAP) collision would be a different pattern. Once we have a rough pattern, we can generate the corresponding plausible trajectories. We term these patterns as *Collision Pattern*, which encodes the relative position, heading, and roles of the ego and adversarial vehicles at the moment of impact. Rather than modeling full trajectories, *Collision Patterns* focuses on the critical endpoint and offers a compact and interpretable representation for collision scenario generation.

Our framework is inspired by region proposal networks [19] in object detection, where a rough location is first estimated and then refined. Similarly, we first predict a coarse collision pattern as an anchor and refine it to determine the precise spatial configuration. Based on this configuration, we generate the full adversarial trajectory. This outcome-driven and coarse-to-fine strategy enables effective generation, avoiding error accumulation from stepwise prediction while improving controllability and flexibility.

We conduct experiments on the COLLIDE dataset and benchmark conditional trajectory prediction baselines, including SGAN [20], MID [10], and STRIVE [7]. Our method consistently generates collisions that match user-specified time-to-accident (TTA) and collision types, achieving higher collision rates and thus more plausible scenarios. We further evaluate the generated scenarios on three rule-based motion planners—IDM [21], a rule-based planner [7], and PDM [22]—and find that our method induces more planner failures than C-STRIVE, exposing critical blind spots in existing planning strategies. To further evaluate the impact of the generated scenarios, we use them to update AV planners and found that ours can offer improvement in terms of collision rate in diverse dangerous scenarios.

Our contributions are summarized as follows:

- We propose controllable collision scenario generation, a new task designed for comprehensive and systematic evaluation of autonomous vehicle safety.
- We introduce an automatic data collection pipeline that accounts for collision types and TTA, enabling benchmarking of controllable collision scenario generation algorithms.
- We introduce a compact and interpretable representation *Collision Pattern*, which encodes the relative position, heading, and roles of the ego and adversarial vehicles at the moment of impact.
- We demonstrate that our framework can effectively generate user-specified scenarios, outperforming strong baselines. Moreover, the generated scenarios reveal limitations in rule-based motion planners and can be used to improve their robustness.

II. Related Work

Scenario Generation. Existing approaches can be categorized as *data-driven*, *adversarial*, and *knowledge-based*, according to [1]. Data-driven methods [23]–[25] extract rare collision events from large-scale real-world datasets [26], but such events are sparse and highly imbalanced in terms of both collision type and urgency (TTA), limiting their value for training or evaluation. Adversarial methods [7], [8], [27] perturb agent trajectories within learned traffic dynamics models [28] to induce failures, but they often bias toward specific failure modes and lack explicit controllability over collision semantics. Knowledge-based approaches [17], [29]–[31] incorporate rule-based priors or pre-defined templates to guarantee coverage of certain collision types but are not scalable, as increasing intra-class variation requires substantial manual effort and does not generalize efficiently.

Our formulation does not fall into the existing three categories. Instead, it bridges data-driven realism by learning the distribution from COLLIDE and knowledge-based structure via collision pattern supervision. This formulation directly addresses the limitations of prior work: by training on synthesized collision trajectories derived from real-world logs , we mitigate the data scarcity faced by real-world–only methods; by conditioning on structured collision patterns and TTA, we enable intra-category diversity that knowledge-based methods, which rely on rule-based or heuristic-driven methods, often fail to provide; and by aligning each generated outcome with collision types, our design ensures controllability over collision types, which adversarial optimization approaches like STRIVE [7] and KING [8] cannot guarantee.

Trajectory Prediction and Traffic Simulation. Since our task involves generating future agent trajectories under collision constraints, it is most closely related to trajectory prediction, so we briefly review generative models for traffic simulation. Early generative models such as Social-GAN [20] learn from past trajectories of each agent to predict diverse but plausible future motions. TrafficSim [28] and BITS [32] utilize map information as additional input to generate traffic flows. However, these autoregressive predictors typically generate trajectories step by step, which often leads to error accumulation over longer horizons and makes it difficult to guarantee precise long-term outcomes such as specific collision types or TTA. STRIVE [7] builds upon TrafficSim to generate safety-critical scenarios by optimizing

TABLE I: **Comparison of existing collision-related datasets.** The table lists representative collision datasets with their primary task, the number of collision cases, and available attributes.

Dataset	Task	#Collision	Attributes
YouTubeCrash [14]	collision prediction	122	x
Street Accident [11]	collision prediction	678	x
Collision [34]	collision detection	803	x
VIENA [35]	behavior prediction	1200	single behavior
CTA [15]	collision reasoning	1935	collision cause
NIDB [12], [13]	collision prediction	4595	topology
GTACrash [14]	collision prediction	7720	x
RiskBench [17]	risk identification	1873	single behavior
HazardVLM [16]	hazard description	3860	single behavior
COLLIDE	scenario generation	**8586**	**collision type TTA**

perturbations in the latent space. More recently, diffusion-based predictors [9], [10] have demonstrated a stronger ability to capture the diverse distribution of multi-agent behaviors, alleviating the mode-collapse problem often observed in previous trajectory predictors. CTG++ [33] introduces language-guided conditional generation via large language models (LLMs), where users provide prompts to influence future behavior. However, such natural-language conditions only guide agent behaviors at a coarse semantic level (e.g., intent or style), rather than enforcing precise spatial relations or collision timing.

Our method instead adopts a back-to-front, coarse-to-fine generation process: rather than incrementally predicting each next step, which risks compounding errors and missing the desired outcome, we first predict the final collision pattern that encodes the collision type and TTA. The attacker's full trajectory is then generated to realize this outcome while maintaining physical plausibility.

Collision Dataset. Existing real-world datasets such as YouTubeCrash [14], StreetAccidents [11], and VIENA [35] have been widely used for training or evaluating models on accident anticipation or classification tasks. However, these datasets passively collect crash incidents from traffic camera footage or dashcams, offering biased coverage and lacking detailed semantic annotations such as collision types and time-to-accident (TTA). Some simulation-based datasets, such as RiskBench [17], SafeBench [31], and Target [36], offer synthetic crash scenes for wider coverage of rare events, but they lack semantic labels for collision types, limited to per-vehicle behaviors rather than structured collision patterns. To bridge this gap, we propose COLLIDE, a structured dataset derived from real-world logs via an automatic generation pipeline. It ensures balanced coverage across five NHTSA-defined collision types and TTA intervals, providing fine-grained control over scenario attributes. As shown in Table I, COLLIDE contains 8,586 task-specific collision cases, enabling scalable training and evaluation compared to prior datasets. Moreover, COLLIDE is the only dataset among existing works that supports scenario generation with explicit collision type control.

III. Data Collection

This section introduces the definition of collision types, the data collection methodology, and scenario augmentation. We construct a dedicated dataset called **COLLIDE**, providing large-scale, balanced, and semantically grounded collision scenarios that are ideal for controllable scenario generation.

A. Collision Types

We adopt five representative collision categories based on NHTSA [18] statistics, which reflect societal impact across crash frequency, injury severity, and economic burden. These include: (1) Junction Crossing (JC), (2) Lane Change, (3) Opposite Direction, (4) Rear-End, and (5) Left-Turn Across Path (LTAP). We use the relative heading angle at the collision point to assign categories: Rear-End ($\sim 0°$), Lane Change ($\sim 20°$), Opposite Direction ($\sim 180°$), Junction Crossing ($\sim 90°$), and LTAP ($\sim 90°$). A $10°$ tolerance is applied to account for noise. Since both JC and LTAP involve a $90°$ collision angle, we distinguish them by the relative heading when entering the intersection: JC maintains orthogonal paths throughout, while LTAP involves a left turn with vehicles approaching at a $180°$ difference.

B. Data Collection

Given the scarcity of labeled collision scenarios in existing real-world datasets, we develop an automatic data collection pipeline that transforms **non-collision** scenes collected in the nuScenes dataset into synthetic **collision** scenarios. Each modified scene incorporates our self-defined collision pattern and preserves physical feasibility.

The original nuScenes scenarios span 20 seconds at 2Hz. We segment these into 4.5s to 9s clips, and at every 0.5-second interval, define a target collision point. The ego vehicle's bounding box serves as a reference to compute a collision target for an attacker. For example, in a rear-end scenario, the attacker must reach a predefined position behind the ego vehicle at the moment of collision.

A quintic polynomial planner is used to generate smooth attacker trajectories toward the target location. Candidate attackers are selected from other vehicles in the same scene that exhibit continuous motion. If a generated trajectory leads to infeasibility, such as collisions with other agents or off-road paths, the scenario is discarded. This ensures that the resulting dataset exhibits high control over both the collision type and TTA.

C. Scenario Coverage

To ensure comprehensive coverage of safety-critical cases, our dataset incorporates multiple sources of variation.

Collision geometry augmentation. To enrich collision diversity, we sample collision angles uniformly within the defined margins to generate varied but semantically consistent scenarios.

TTA augmentation. The time-to-accident (TTA) is defined as the temporal gap between the current time and the moment when the ego reaches the collision point. By uniformly sampling such collision points, we obtain TTAs distributed in the

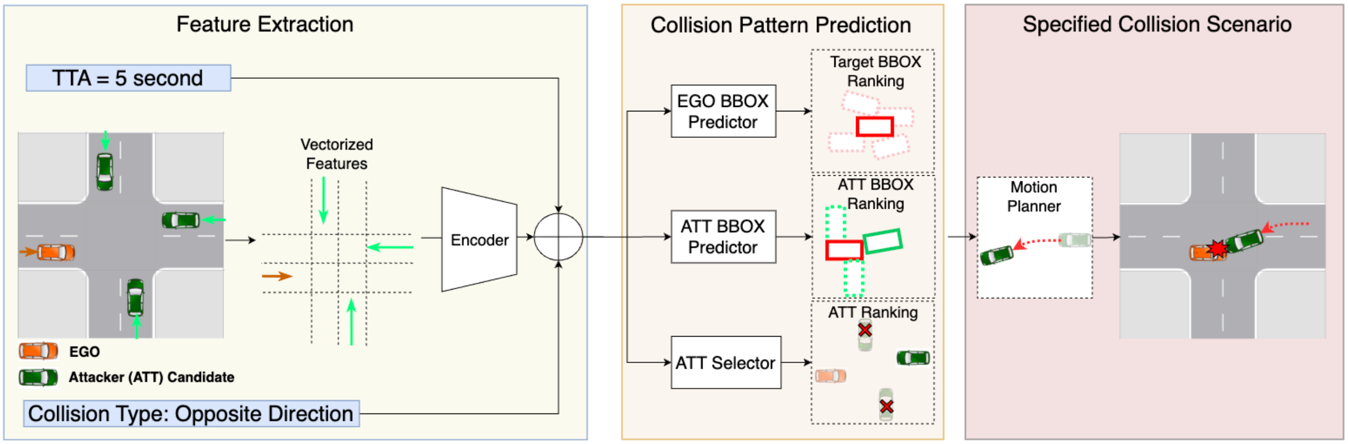

Fig. 2: **The proposed architecture for controllable collision scenario generation**. Given structured scene context and user-specified conditions (collision type and TTA), our model predicts the "collision pattern", representing the relative configuration of ego and attacker vehicles (ATT) at the moment of collision. The predicted pattern then guides a motion planner to produce a feasible attacker trajectory.

range of 4.5s to 9.0s, which balances between short-horizon, high-risk interactions and longer-horizon, low-urgency cases. **Map diversity.** Our dataset covers all four cities in nuScenes, which provides diverse road structures and driving norms (e.g., left-hand vs. right-hand driving). This geographical diversity naturally enriches the collision scenarios with varied map topologies, ensuring broader coverage of intersection layouts, lane geometries, and driving conventions.

IV. Method

A. Problem Formulation

We aim to generate controllable collision scenarios conditioned on user-specified attributes, namely the desired collision type and time-to-accident (TTA). The input to the task consists of past trajectories of all traffic participants and the map topology. Each agent trajectory is represented as a sequence of positions and headings: $s^i_{1:T_{\text{hist}}} = \{(x^t_i, y^t_i, \theta^t_i)\}_{t=1}^{T_{\text{hist}}}$, where (x^t_i, y^t_i) denotes the 2D location of agent i at time t, and θ^t_i is its heading angle. Let $\mathcal{M}$ denote the map topology. Given the historical trajectories of N agents $\{s^i_{1:T_{\text{hist}}}\}_{i=1}^N$ and map $\mathcal{M}$, the goal is to generate an attacker trajectory $s^a_{T_{\text{hist}}:t_{\text{TTA}}}$ that collides with the ego vehicle exactly at $t = t_{\text{TTA}}$. The generated trajectory is consistent with the user-specified collision type $\mathbf{C}_{\text{type}}$ and the urgency level determined by TTA. This formulation emphasizes two key challenges: (1) localizing where and how the collision occurs, and (2) realizing a kinematically feasible attacker trajectory that satisfies the specified conditions.

B. Framework Overview

As illustrated in Fig. 2, rather than directly predicting the full attacker trajectory, we first predict a *collision pattern*, a compact and interpretable representation of the final spatial configuration of the ego and attacker vehicles at the collision moment. Specifically, we define the collision pattern as a 4D vector $\mathbf{p} = (\Delta x, \Delta y, \Delta\theta, \hat{a})$ where $(\Delta x, \Delta y)$ denotes the relative position of the attacker with respect to the ego vehicle, $\Delta\theta$ denotes their relative heading at impact, and $\hat{a}$ indicates the selected attacker. The predicted pattern is then used as a target for trajectory realization by a quintic polynomial planner. Three trainable modules, Ego Position Prediction, Attacker Offset Prediction, and Attacker Selection, operate on the encoded scene and condition features to predict the collision pattern.

C. Input Representation

We adopt a VectorNet-based encoder [37] to extract vectorized features from map topology and agent trajectories, forming the scene feature $\mathbf{F}_{\text{scene}}$. The user-specified condition, including collision type (one-hot vector $\mathbf{C}_{\text{type}}$) and normalized TTA value, is concatenated with $\mathbf{F}_{\text{scene}}$ to produce the final feature: $\mathbf{F}_{\text{final}} = [\mathbf{F}_{\text{scene}}, \mathbf{C}_{\text{type}}, \text{TTA}]$. This final feature serves as the input to all subsequent modules.

D. Collision Pattern Prediction

a) Ego Position Prediction.: We localize the anticipated collision point of the ego vehicle by framing it as a region proposal task. Candidate anchors $\{T_i\}$ are sampled along lane centerlines and classified as positive or negative depending on their proximity to the ground-truth collision point $x_{\text{ego, GT}}$ within radius r:

$$\text{label}(T_i) = \begin{cases} \text{positive}, & \|T_i - x_{\text{ego,GT}}\| \le r, \\ \text{negative}, & \text{otherwise.} \end{cases}$$

Each candidate anchor T_i is represented by a feature vector that includes its own spatial attributes (x, y, θ) together with concatenated local map around T_i. For the top-ranked candidate, heading and offset regression produce the final ego bounding box $\mathbf{B}_{\text{ego}}(t_{\text{TTA}})$. The local feature design allows the model to exploit richer local geometric information for more reliable ranking.

TABLE II: **Results of controllable collision scenario generation on COLLIDE.** Our method achieves the highest collision rate and similarity across five collision types, outperforming conditional baselines.

Method	Metric	Lane Change	Opposite Direction	Rear End	Junction Crossing	LTAP	Average
C-SGAN [20]	**Collision Rate**	20%	20%	18%	9%	0%	14%
C-STRIVE [7]		32%	21%	39%	16%	15%	25%
C-MID [10]		25%	10%	9%	5%	10%	11%
Ours		**73%**	**77%**	**84%**	**81%**	**90%**	**81%**
C-SGAN	**Similarity**	34%	33%	68%	0%	0%	39%
C-STRIVE		52%	75%	**93%**	52%	18%	68%
C-MID		42%	68%	74%	33%	22%	49%
Ours		**62%**	**90%**	87%	**84%**	**76%**	**81%**

b) Attacker Offset Prediction.: Given $\mathbf{B}_{\text{ego}}$ and $\mathbf{F}_{\text{final}}$, the model predicts a relative offset to determine the adversarial bounding box:

$$\mathbf{B}_{\text{attacker}} = \mathbf{B}_{\text{ego}} + (\Delta x, \Delta y, \Delta\theta).$$

This ensures controllable geometric relations consistent with the target collision type.

c) Attacker Selection.: For each candidate agent i, its current state $\mathbf{P}_i$ is concatenated with $\mathbf{F}_{\text{final}}$ and scored via an MLP with softmax. The selected attacker is

$$\hat{a} = \arg\max_i \text{MLP}([\mathbf{P}_i, \mathbf{F}_{\text{final}}]).$$

E. Trajectory Realization

Given the predicted collision pattern, which specifies the relative spatial configuration of the ego and attacker at $t = T_{\text{TTA}}$, we employ a quintic polynomial planner to realize a smooth and kinematically feasible attacker trajectory. The collision pattern serves as the boundary condition for trajectory generation: the attacker must start from its observed history $s^a_{1:T_{\text{hist}}}$ and reach the configuration $\mathbf{B}_{\text{attacker}}$ exactly at the designated TTA. Formally, the trajectory is obtained as

$$s^{\hat{a}}_{T_{\text{hist}}:T_{\text{TTA}}} = \mathcal{P}\left(s^a_{T_{\text{hist}}}, \mathbf{B}_{\text{attacker}}, T_{\text{TTA}}\right),$$

where $\mathcal{P}$ denotes the polynomial planner. This construction guarantees physical feasibility while ensuring that the trajectory faithfully realizes the user-specified collision type and timing.

V. Experimental Results

A. Baselines and Evaluation Setup

We compare our method with three conditional scenario generation baselines, C-SGAN, C-MID, and C-STRIVE, implemented using Social-GAN [20], MID [10], and the traffic dynamic model of STRIVE [7], respectively. For all baselines, a one-hot vector of collision type and the normalized time-to-accident (TTA) are concatenated as conditional inputs to the decoder. To ensure a fair comparison, baselines are extended to the same prediction horizon by applying open-loop autoregression: each iteration predicts 4 frames and appends them to the input sequence for the next step. Our method uses the same history input to directly predict a compact collision pattern, the terminal spatial configuration of the ego and attacker. A quintic polynomial planner then generates the full attacker trajectory to meet the predicted pattern at the desired TTA. This difference in generation strategy (pattern prediction + planner vs. autoregressive rollout) is explicitly accounted for in the evaluation by enforcing the same input horizon and final prediction horizon for all methods.

We do not compare with script-based or adversarial-based scenario generation methods, as they rely on pre-defined attacker scripts or agent-level adversarial optimization within simulation environments. These approaches are designed to explore possible crashes but are not directly applicable to offline driving logs such as COLLIDE, and they do not provide explicit controllability over collision type or TTA. Instead, we adapt conditional trajectory prediction models as baselines, since they naturally operate on logged data and can be extended to support generation conditioned on collision attributes.

B. Evaluation Metrics

To evaluate controllable scenario generation, we define two key metrics:

- **Collision Rate**: the percentage of modified attacker trajectories that result in a collision with the ego vehicle.
- **Scenario Similarity**: the percentage of collisions in which the relative angle between ego and attacker matches the desired collision type, within a 10-degree tolerance.

C. Implementation and Training Details

Three modules, Ego Position Prediction, Attacker Offset Prediction, and Attacker Selection, are trained jointly in an end-to-end manner using supervision from ground-truth collision patterns extracted from COLLIDE. The full training loss is defined as a weighted sum of the ego position prediction loss, attacker offset regression loss, and attacker selection classification loss. We train the network for 1000 epochs using the Adam optimizer with a learning rate of 5×10^{-5} and a batch size of 128. The overall inference latency is approximately 0.37 ms per scene on a single GPU, including both collision pattern prediction and polynomial trajectory realization.

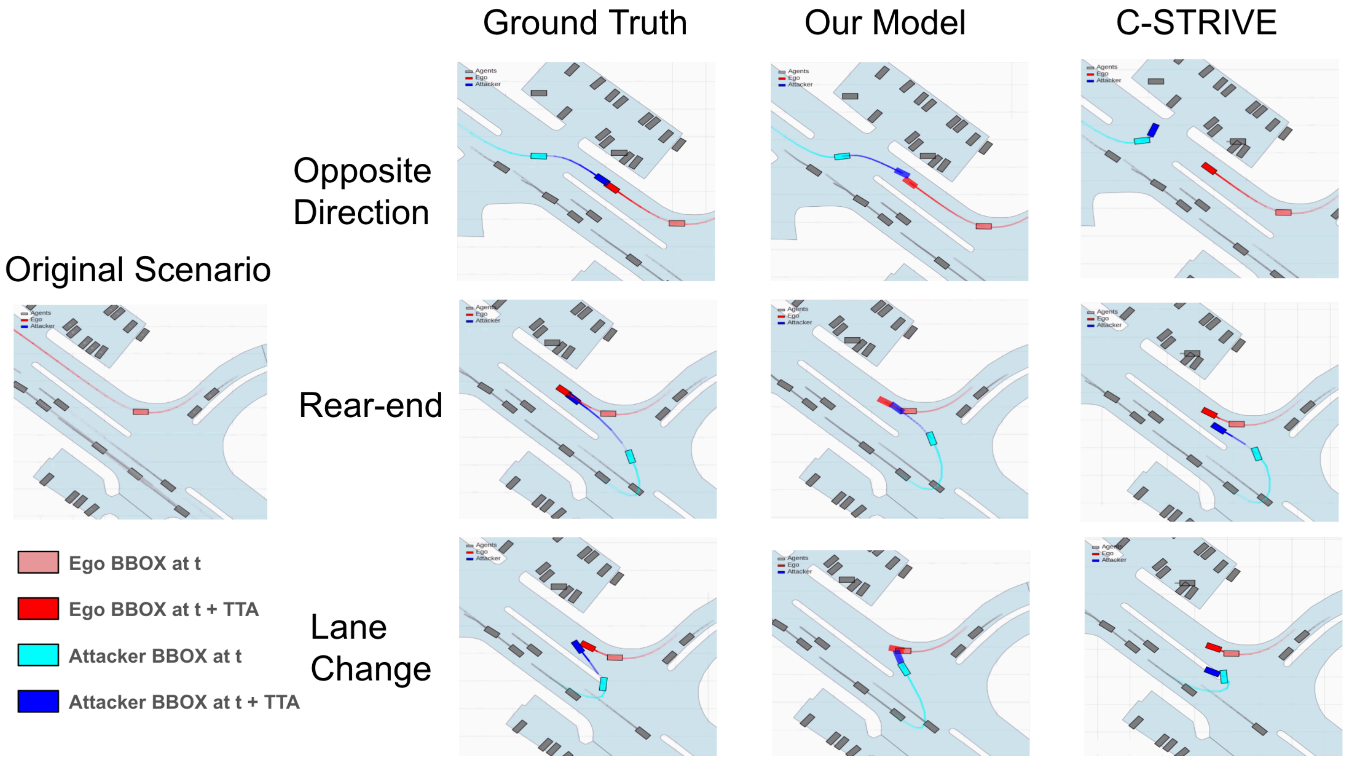

Fig. 3: **Qualitative comparison of generated scenarios across different collision types.** We compare our dataset and model against C-STRIVE [7], the strongest baseline. Our models demonstrate strong controllability in generating user-specified collision types.

TABLE III: **Ablation study on ego position prediction module.** We evaluate different approaches for predicting the ego vehicle's position at the collision moment, comparing regression-based prediction, target point ranking, and region proposal.

Setting	Displacement Error (m)	Angle Distance (°)	Collision Rate	Similarity
Regression-based	3.95	16	69%	67%
Target point ranking	**2.15**	11.4	**80%**	74%
Region proposal	3.1	**10.18**	77%	**78%**

D. Results of Conditional Collision Scenario Generation

Tab. II shows that our method consistently achieves higher collision rates and scenario similarity across all five collision types compared to conditional baselines. The generated scenarios not only cause more collisions but also better match the intended collision patterns specified by time-to-accident (TTA) and collision type. Fig. 3 highlights common failure cases of the baselines. For instance, in the Opposite Direction example, C-STRIVE [7] predicts U-turn behaviors rather than intrusions into the ego lane, resulting in high deviation from the ground-truth trajectory. Similarly, in Rear-End scenarios, C-STRIVE [7] fails to capture overtaking attackers that decelerate sharply in front of the ego and instead defaults to benign forward-driving behaviors. In contrast, C-STRIVE [7] performs well in the common scenario where both vehicles are driving in the same direction. This limitation arises because C-STRIVE lacks fine-grained collision type guidance and therefore tends to overfit to the dominant prevalent, and safer behaviors in the dataset. By introducing collision patterns as explicit intermediate targets, our method overcomes this bias and reliably generates hazardous yet underrepresented events consistent with the specified collision type and TTA.

E. Ablation Study on Ego Bounding Box Prediction

We investigate three strategies for localizing the ego collision endpoint, a key step for shaping the collision pattern. The regression-based approach directly predicts offsets in (x, y, θ), but often produces imprecise goals. Target point ranking [38] improves positional accuracy by selecting among candidate anchors along centerlines, yet may still misalign with the lane topology. Our region proposal design further attaches local subgraph features to anchors, emphasizing lane-conforming orientations.

Tab. III shows that regression yields the largest errors, ranking reduces displacement but lacks angular precision, while region proposals achieve the best similarity to ground-truth patterns. Since our task prioritizes semantic consistency of collision patterns over raw collision frequency, we adopt the region proposal variant in the final framework.

While the target ranking strategy better predicts whether a collision will occur, the region proposal method more faithfully captures how the collision happens. Since our goal is to generate controllable scenarios that match specific types,

TABLE IV: **Evaluation of generated scenarios on autonomous vehicle planners.** We report collision rates of C-STRIVE and our method across five collision types, tested on different planners. Higher collision rates indicate stronger effectiveness in exposing planner failure cases.

Planner	Algorithms	lane change	opposite direction	rear end	junction crossing	LTAP	Average
IDM [21]	C-STRIVE	**17%**	11%	58%	9%	4%	21%
	Ours	16%	**30%**	**65%**	**20%**	**6%**	**29%**
Rule-based [7]	C-STRIVE	23%	24%	**40%**	13%	**23%**	25%
	Ours	**31%**	**52%**	18%	**28%**	22%	**30%**
PDM [22]	C-STRIVE	12%	12%	67%	11%	11%	24%
	Ours	**19%**	**37%**	**76%**	**18%**	**39%**	**39%**

we prioritize semantic consistency and pattern fidelity over mere collision frequency. Therefore, we adopt the region proposal variant in our final model.

F. Planner Evaluation under Safety-Critical Scenarios

To assess the practical impact of our generated scenarios, we evaluate three planners: IDM [21], the STRIVE [7] rule-based planner, and the PDM [22] planner from the nuPlan 2023 planning challenge, under scenarios generated by our method and C-STRIVE. As shown in Tab. IV, our scenarios consistently lead to higher collision rates across all scenario types and planners.

Both the IDM and STRIVE rule-based planners rely on the original ego trajectory as a reference path. The IDM planner treats the attacker as a lead vehicle and follows a conservative car-following policy, often resulting in early braking and thus lower collision rates. In contrast, the STRIVE rule-based planner applies a simple brake–accelerate rule to slightly adjust the replayed speed, producing behavior more similar to the original collision scenario. The PDM planner uses the lane centerline as a reference path, generates candidate maneuvers in both lateral and longitudinal directions, and evaluates them using an internal scoring function. To assess potential collisions, it simulates the future motion of surrounding vehicles under the assumption that each continues with its current velocity rather than relying on learned trajectory predictors. This uncertainty in reasoning about the front agent leads to a higher collision rate.

The results confirm that our method produces more adversarial scenarios, capable of consistently triggering planner failures even under motion planning setups.

G. Improving Planner with Generated Scenarios

Planner optimization process. To demonstrate the utility of our generated collision scenarios, we conduct parameter optimization of the PDM planner [22] via grid search under both C-STRIVE-generated scenarios and those from our method (Tab. V). The planner constructs candidate trajectories by laterally shifting the centerline reference path of the ego vehicle to simulate evasive maneuvers and combining each shifted path with different velocity scales to simulate varying levels of deceleration. Each candidate trajectory is evaluated with the IDM policy, and the best one is selected according to a scoring function. We tune both IDM parameters (e.g., minimum desired distance, maximum acceleration) and the scoring weights (e.g., progress and timing weights).

TABLE V: **Parameter tuning results on planners.** We tune the policy parameters of PDM to evaluate collision rate using scenarios generated by different methods.

Parameter	LC	OD	RE	JC	LTAP
Default	**8.1** %	43.4 %	72.6%	14.9%	27.8%
C-STRIVE's	8.7 %	**39.4** %	**70.3%**	12.9%	20.9%
Ours	8.7 %	40.0 %	70.4%	**12.5%**	**20.8%**

Results. C-STRIVE tends to yield overly aggressive attacker behaviors, often leading to collisions earlier than the specified TTA. This drives the PDM planner to favor more frequent lane changes, which is advantageous in scenarios such as *Opposite Direction* (OD) and *Rear-End (RE)*, where collision avoidance is only feasible through lateral maneuvers. These results highlight that different scenario generation methods expose different planner failure modes, underscoring the need for a broader and more controllable set of testing scenarios to comprehensively assess planner robustness and generalization. Qualitative visualizations of planner behavior after fine-tuning with the generated scenarios are provided in the project webpage.

VI. Conclusion and Future Works

We present controllable collision scenario generation as a new task for systematic AV safety evaluation. To this end, we build COLLIDE, the first dataset with balanced coverage of collision types and time-to-accident (TTA). We further introduce the concept of Collision Pattern and design a region-proposal-inspired framework that enables fine-grained control over collision outcomes. Experiments show that our method outperforms strong baselines in controllability and plausibility, while also revealing planner limitations and improving their robustness.

Acknowledgement

This work is sponsored in part by the National Science and Technology Council under grants 113-2628-E-A49-022- and 114-2628-E-A49-007-, MobileDrive, the Higher Education Sprout Project of National Yang Ming Chiao Tung

University, the Ministry of Education, and the Yushan Fellow Program Administrative Support Grant.

REFERENCES

[1] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation - a methodological perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 6971–6988, 2023.

[2] B. Schütt, J. Ransiek, T. Braun, and E. Sax, "1001 ways of scenario generation for testing of self-driving cars: A survey," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–8.

[3] Y. Gao, M. Piccinini, Y. Zhang, D. Wang, K. Moller, R. Brusnicki, B. Zarrouki, A. Gambi, J. F. Totz, K. Storms *et al.*, "Foundation models in autonomous driving: A survey on scenario generation and scenario analysis," *arXiv preprint arXiv:2506.11526*, 2025.

[4] Z. Wang, J. Ma, and E. M. Lai, "A survey of scenario generation for automated vehicle testing and validation," *Future Internet*, vol. 16, no. 12, p. 480, 2024.

[5] C.-H. Kung, S.-W. Lu, Y.-H. Tsai, and Y.-T. Chen, "Action-slot: Visual action-centric representations for multi-label atomic activity recognition in traffic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 18 451–18 461.

[6] H. X. Liu and S. Feng, "Curse of rarity for autonomous vehicles," *nature communications*, vol. 15, no. 1, p. 4808, 2024.

[7] 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.

[8] 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.

[9] 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.

[10] T. Gu, G. Chen, J. Li, C. Lin, Y. Rao, J. Zhou, and J. Lu, "Stochastic trajectory prediction via motion indeterminacy diffusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17 113–17 122.

[11] F.-H. Chan, Y.-T. Chen, Y. Xiang, and M. Sun, "Anticipating accidents in dashcam videos," in *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part IV 13*. Springer, 2017, pp. 136–153.

[12] H. Kataoka, T. Suzuki, S. Oikawa, Y. Matsui, and Y. Satoh, "Drive video analysis for the detection of traffic near-miss incidents," in *2018 IEEE International Conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3421–3428.

[13] T. Suzuki, H. Kataoka, Y. Aoki, and Y. Satoh, "Anticipating traffic accidents with adaptive loss and large-scale incident db," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3521–3529.

[14] H. Kim, K. Lee, G. Hwang, and C. Suh, "Crash to not crash: Learn to identify dangerous vehicles using a simulator," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 978–985.

[15] T. You and B. Han, "Traffic accident benchmark for causality recognition," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*. Springer, 2020, pp. 540–556.

[16] D. Xiao, M. Dianati, P. Jennings, and R. Woodman, "Hazardvlm: A video language model for real-time hazard description in automated driving systems," *IEEE Transactions on Intelligent Vehicles*, 2024.

[17] C.-H. Kung, C.-C. Yang, P.-Y. Pao, S.-W. Lu, P.-L. Chen, H.-C. Lu, and Y.-T. Chen, "Riskbench: A scenario-based benchmark for risk identification," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 800–14 807.

[18] W. G. Najm, R. Ranganathan, G. Srinivasan, J. D. Smith, S. Toma, E. D. Swanson, A. Burgett *et al.*, "Description of light-vehicle pre-crash scenarios for safety applications based on vehicle-to-vehicle communications," United States. Department of Transportation. National Highway Traffic Safety..., Tech. Rep., 2013.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[20] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.

[21] 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.

[22] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1268–1281.

[23] C. Knies and F. Diermeyer, "Data-driven test scenario generation for cooperative maneuver planning on highways," *Applied Sciences*, vol. 10, no. 22, p. 8154, 2020.

[24] J. M. Scanlon, K. D. Kusano, T. Daniel, C. Alderson, A. Ogle, and T. Victor, "Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain," *Accident Analysis & Prevention*, vol. 163, p. 106454, 2021.

[25] W. Ding, W. Wang, and D. Zhao, "A new multi-vehicle trajectory generator to simulate vehicle-to-vehicle encounters," *arXiv preprint arXiv:1809.05680*, 2018.

[26] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2118–2125.

[27] 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.

[28] 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.

[29] W. Ding, H. Lin, B. Li, and D. Zhao, "Causalaf: Causal autoregressive flow for safety-critical driving scenario generation," in *Conference on robot learning*. PMLR, 2023, pp. 812–823.

[30] P. Huang, W. Ding, B. Stoler, J. Francis, B. Chen, and D. Zhao, "Cadre: Controllable and diverse generation of safety-critical driving scenarios using real-world trajectories," *arXiv preprint arXiv:2403.13208*, 2024.

[31] C. Xu, W. Ding, W. Lyu, Z. Liu, S. Wang, Y. He, H. Hu, D. Zhao, and B. Li, "Safebench: A benchmarking platform for safety evaluation of autonomous vehicles," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 667–25 682, 2022.

[32] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone, "Bits: Bi-level imitation for traffic simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2929–2936.

[33] 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.

[34] R. Herzig, E. Levi, H. Xu, H. Gao, E. Brosh, X. Wang, A. Globerson, and T. Darrell, "Spatio-temporal action graph networks," in *Proceedings of the IEEE/CVF international conference on computer vision workshops*, 2019, pp. 0–0.

[35] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Viena: A driving anticipation dataset," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 449–466.

[36] Y. Deng, J. Yao, Z. Tu, X. Zheng, M. Zhang, and T. Zhang, "Target: Automated scenario generation from traffic rules for testing autonomous vehicles," *arXiv preprint arXiv:2305.06018*, 2023. [Online]. Available: https://arxiv.org/abs/2305.06018

[37] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 525–11 533.

[38] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.