SafeMVDrive: Multi-view Safety-Critical Driving Video Synthesis in the Real World Domain

Jiawei Zhou[1], Linye Lyu[1], Zhuotao Tian[1], Cheng Zhuo[2], Yu Li[2*]

[1]Harbin Institute of Technology, Shenzhen
[2]Zhejiang University

Abstract

Safety-critical scenarios are rare yet pivotal for evaluating and enhancing the robustness of autonomous driving systems. While existing methods generate safety-critical driving trajectories, simulations, or single-view videos, they fall short of meeting the demands of advanced end-to-end autonomous systems (E2E AD), which require real-world, multi-view video data. To bridge this gap, we introduce SafeMVDrive, the first framework designed to generate high-quality, safety-critical, multi-view driving videos grounded in real-world domains. SafeMVDrive strategically integrates a safety-critical trajectory generator with an advanced multi-view video generator. To tackle the challenges inherent in this integration, we first enhance scene understanding ability of the trajectory generator by incorporating visual context – which is previously unavailable to such generator – and leveraging a GRPO-finetuned vision-language model to achieve more realistic and context-aware trajectory generation. Second, recognizing that existing multi-view video generators struggle to render realistic collision events, we introduce a two-stage, controllable trajectory generation mechanism that produces collision-evasion trajectories, ensuring both video quality and safety-critical fidelity. Finally, we employ a diffusion-based multi-view video generator to synthesize high-quality safety-critical driving videos from the generated trajectories. Experiments conducted on an E2E AD planner demonstrate a significant increase in collision rate when tested with our generated data, validating the effectiveness of SafeMVDrive in stress-testing planning modules. Our code, examples, and datasets are publicly available at: https://zhoujiawei3.github.io/SafeMVDrive/.

1 Introduction

Vision-based end-to-end (E2E) autonomous driving (AD) systems, which directly map visual inputs to driving decisions, have gained growing attention and are gradually deployed in real-world environments [34; 15; 18; 20; 16]. However, ensuring their safety in diverse scenarios remains a significant challenge and addressing this issue requires large-scale and diverse driving datasets, especially the safety-critical ones. Yet, collecting such data in the real world is both costly and inherently dangerous. As a promising alternative, synthetic generation of safety-critical driving scenarios offers a scalable, low-risk, and cost-effective solution. These synthetic datasets can significantly enhance the robustness and generalization of E2E AD systems, especially in rare and hazardous conditions.

Most existing methods for safety-critical data generation aim to generate realistic and controllable adversarial trajectories with diffusion models [32; 38; 37; 6], which can be used to evaluate and improve the performance of the AD planning modules. However, these methods produce non-visual trajectories, which are incompatible with E2E AD systems that require visual input. While some

*Corresponding author: Yu Li at yu.li.sallylee@gmail.com

Figure 1: Keyframes from diverse realistic, multi-view, safety-critical videos generated by **SafeMV-Drive**. Red boxes indicate safety-critical vehicles involved in events like cut-ins, rapid rear approaches, and sudden braking. Additional video examples are available via the link provided in the abstract.

approaches use simulators to generate safety-critical driving videos [33; 1], their effectiveness is limited by the domain gap between simulation and reality. Besides, deep video generative models like Open-Sora [36] are also explored to generate real-world driving accident videos, but the produced videos are typically low-quality and limited to single-view outputs [17].

With recent AD video models supporting realistic multi-view video generation and controllability via signals [29; 23; 9], a naive method to generate multi-view safety-critical videos in the real domain is to convert the safety-critical trajectories—produced by the trajectory generator—into control signals to guide the video generator. However, this naive approach faces several challenges: firstly, current trajectory generators need to select an adversarial vehicle from multi-vehicle traffic, typically relying on heuristic rules based on non-visual data such as annotated vehicle kinematics and map features. Due to the inherent limitations of heuristic methods and lack of critical visual cues, it is difficult for the selector to comprehensively understand the complex physical scene to choose the appropriate vehicle (as detailed in Section 3.2). Consequently, the safety-criticality and realism of the generated videos can be compromised. Furthermore, current safety-critical trajectory generators aim to generate collision events. However, existing multi-view video generators struggle to realistically simulate them due to a lack of multi-view collision training data. When the control signal corresponds to collision trajectories, the realism of the generated video degrades significantly.

To address the above issues, we present *SafeMVDrive* to generate high-quality, multi-view safety-critical videos in the real-world domain. Our key insight lies in enhancing adversarial vehicle selection by incorporating visual information and simulating evasion trajectories that align with the capacity of existing multi-view video models. For adversarial vehicle selection, we leverage the strong scene comprehension capabilities of Vision-Language Models (VLMs) [7] and design a VLM-based selector to selects the most critical adversarial vehicle based on visual data from the initial scene. To adapt the VLM to the selection task, we first construct an automatically annotated dataset that maps scenes to set of vehicles capable of colliding with the ego vehicle. We then fine-tune the VLM using the GRPO algorithm [21], enabling it to reason about complex traffic situations and significantly improve the success rate of adversarial vehicle selection. Furthermore, to address the limitations of video generators in rendering collisions, we introduce a two-stage trajectory generator. In the first stage, we simulate a valid collision trajectory. In the second stage, we refine this into a natural evasion trajectory that maintains safety-critical features while avoiding direct collisions. This ensures that the generated videos remain realistic and within the capability of current video models. Finally, by integrating a state-of-the-art multi-view video generator, we produce high-quality safety-critical driving videos in the real-world domain. As shown in Figure 1, our method successfully generates realistic traffic scenes suitable for testing and improving E2E AD systems.

The main contributions of our work are summarized as follows:

- In this paper, we introduce **SafeMVDrive**, the first framework capable of generating high-quality, safety-critical multi-view video in the real-world domain. The core insight underlying our framework is the strategic integration of a safety-critical trajectory simulator with a multi-view driving video generator, and addressing the main challenge of their integration using a VLM-based adversarial vehicle selector and a two-stage evasion trajectory generator.

- We incorporate visual information into the selection of safety-critical vehicles by adapting a vision-language model (VLM) to this task. To facilitate this, we propose an automated annotation method to generate pairs of driving scenes and the corresponding safety-critical vehicles. Using this dataset, we fine-tune the VLM with the GRPO algorithm to improve its ability to understand multi-view driving scenes, allowing it to accurately identify adversarial vehicles capable of inducing safety-critical scenarios.
- We propose a two-stage trajectory generator to produce collision evasion trajectories that remain within the generative capacity of existing multi-view video generators. In the first stage, a collision trajectory is generated. In the second stage, we design a method to transform the first-stage collision trajectories to natural evasion trajectories, preserving the safety-critical characteristics while ensuring compatibility with video generation models.
- Using our framework, we construct the first high-quality, multi-view, safety-critical driving video dataset in the real domain. The dataset contains 41 diverse 9-second scenes and can serves as a valuable benchmark to evaluate and enhance the robustness of end-to-end autonomous driving planners in terms of their ability to avoid collisions.

Our dataset exhibits strong safety-critical traits. Tested with the classic end-to-end autonomous planner UniAD [15], our generated videos show a 30% increase in collision rate compared to the original NuScenes videos [5]. Moreover, our videos maintain high visual realism. In the user study evaluating video quality, our safety-critical videos achieve 87% of the realism score compared to those generated by the state-of-the-art video generator using original, non-safety-critical trajectories.

2 Related work

Safety-critical data generation is essential to enhance end-to-end AD systems' robustness in the real world. The existing work can be categorized into trajectory-based and video-based approaches in terms of their output formats. Trajectory-based approaches generate non-visual adversarial trajectories, while video-based approaches produce safety-critical driving videos.

Given the initial traffic context, trajectory approaches first select the adversarial vehicle and then optimize trajectories that lead to safety-critical situations. Recent works often leverage diffusion models for controllable traffic generation. For example, Controllable Traffic Generation (CTG) trains diffusion models on large-scale driving data to generate realistic trajectories [38]. Safe-Sim [6] and CTG++ [37] further select adversarial vehicles via heuristics and apply adversarial losses to guide generation. While these methods show progress, they are incompatible with E2E AD systems, which require visual input. Besides, they use simple heuristic methods like the nearest vehicle to select the adversarial vehicle, which can fail to create safety-critical cases. For instance, as shown in Figure 3, a nearby bus may be chosen despite being blocked by obstacles, making collision impossible.

Another research direction is to directly provide safety-critical visual data. Some works [33; 1; 32] use simulators like Carla [10] to generate adversarial driving videos but suffer from the domain gap between simulation and reality. To generate realistic visuals, ADV2 [17] employs generative models like open-sora [36], finetuned on real traffic accident data with text captions, to produce adversarial videos from user prompts. However, the videos are low-quality and single-view, limiting their use in E2E AD systems that require high-quality, multi-view inputs.

In contrast to the current works, our proposed framework strategically combines a novel vehicle selector, an evasion trajectory generator, and a high-performance driving video generator to produce realistic, high-quality, multi-view adversarial driving video data compatible with E2E AD systems.

3 Methods

3.1 Overview

Figure 2 shows our framework for generating safety-critical multi-view videos, comprising three parts: (1) a VLM-based adversarial vehicle selector; (2) a two-stage evasion trajectory generator; and (3) a trajectory-to-video generator. The input is single-frame holistic information of an initial scene, combining visual data (multi-view camera images) and non-visual data (camera parameters, vehicle states, and road maps)—all available in datasets like NuScenes [5], Waymo [24], and Argoverse2

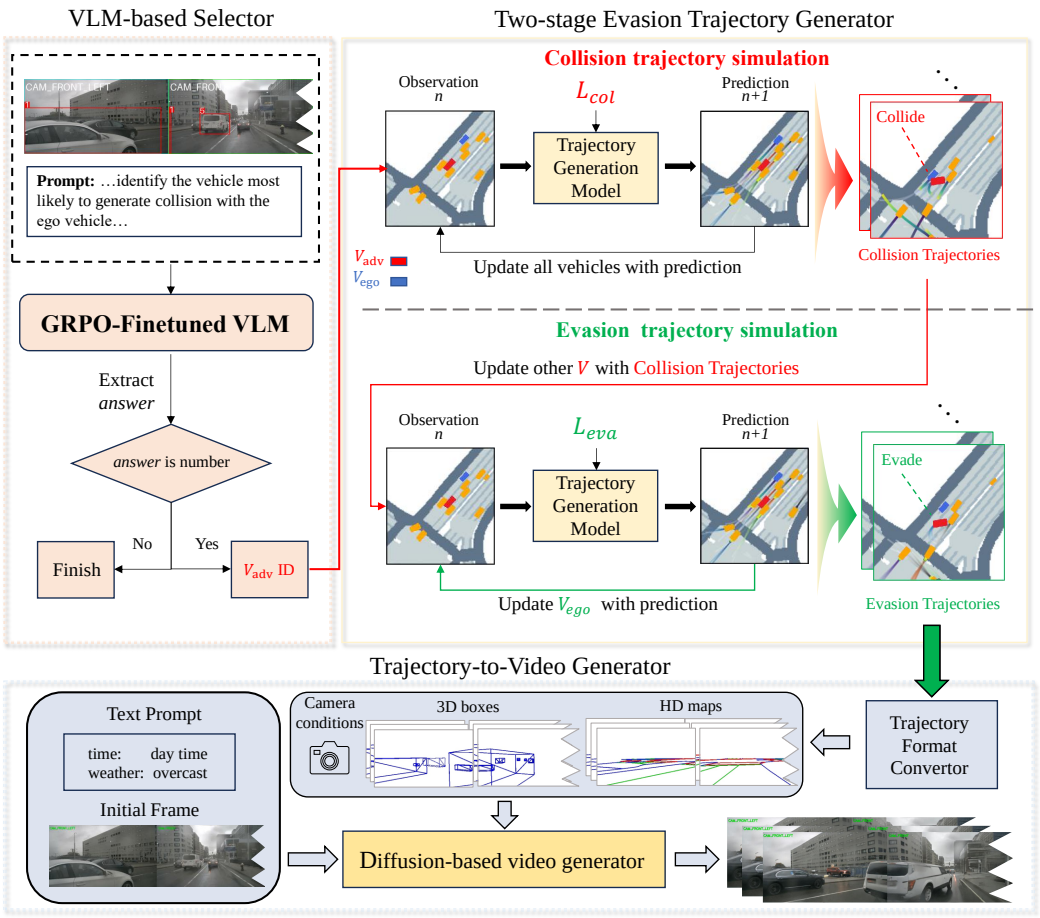

Figure 2: The SafeMVDrive framework for generating realism, multi-view safety-critical videos.

[30]. First, we mark vehicles within distance D from the ego vehicle with ID-labeled 2D boxes in the multi-view images. The images are then fed into the VLM-based selector to identify the adversarial vehicle V_{adv}. With V_{adv}'s ID, the two-stage evasion trajectory generator can produce safety-critical trajectories. In the first stage, we generate a collision trajectory where V_{adv} collides with the ego vehicle V_{ego}; in the second stage, we convert this collision trajectory into a realistic evasion trajectory using our proposed method. The generated trajectories are then converted to control signals that guide a diffusion-based video generator to synthesize realistic safety-critical multi-view videos.

3.2 VLM-based Adversarial Vehicle Selector

The first step in generating safety-critical data is selecting the adversarial vehicle V_{adv} from the initial scene. Prior methods rely on simple heuristics using non-visual data like vehicle kinematics and maps, such as choosing the closest vehicle, applying fixed distance-velocity rules [37], or randomly picking nearby lane vehicles [6]. However, these heuristics lack crucial visual cues and fail to capture complex driving scenarios, often resulting in inappropriate selections. Figure 3 illustrates this: from the BEV-view, non-visual data misses an obstacle separating the chosen vehicle from the ego vehicle, causing all heuristic methods to select V_{adv} incorrectly (red boxes).

To address the aforementioned problems, we propose incorporating visual information into adversarial vehicle selection by leveraging the scene understanding capabilities of Vision-Language Models (VLMs) [7]. Specifically, we introduce a VLM-based selector that selects the critical adversarial vehicle using the multi-view images from the initial scene. Our first attempt is to guide the VLM with task-specific prompts. To aid comprehension and accurate vehicle ID output, we annotate safety-critical vehicle candidates with ID-labeled 2D bounding boxes (Figure 3, left) and exclude distant vehicles (beyond distance D from the ego vehicle). We also design a tailored prompt using these annotations (see Appendix C). However, due to VLMs' limited exposure to multi-view data during training, prompting alone proves insufficient for effective multi-view understanding.

Figure 3: Comparison between the real-world scene (left) and the BEV-rendered non-visual data (right). Obstacles that physically prevent a collision between Vehicle 1 and the ego vehicle are visible in the real-world view but missing in the non-visual data, potentially misleading heuristic methods.

To address the above problem, we fine-tune the VLM to better adapt it to our task, which requires constructing a suitable fine-tuning dataset. A key challenge is determining the correct VLM output for each multi-view image—specifically, identifying which vehicles could realistically collide with the ego vehicle via natural trajectories. Manual labeling is costly and error-prone. To solve this, we propose an automated method using a controllable diffusion-based traffic simulator [38] that generates naturalistic trajectories and supports test-time guidance via loss functions. For each safety-critical vehicle candidate V_{adv}, we apply a loss encouraging collision with the ego vehicle based on their distance (see Section 3.3). We select vehicles that successfully collide and filter out unrealistic cases, such as those entering non-drivable areas or colliding with other vehicles first. This yields annotated data defining the set of effective safety-critical vehicles S_{coll} for each scene.

After obtaining the fine-tuning dataset, we apply the GRPO algorithm [21], a recent RL method proven to effectively enhance reasoning of LLMs [13] and VLMs [22]. GRPO enhances the model's reasoning capabilities through a self-improving RL process, which makes it well-suited for helping the model better understand complex multi-view physical scenarios [28; 14]. Following [22], we augment the prompt with: 'Output the thinking process in <think> </think> and final answer (number) in <answer> </answer> tags.' and use the following format reward:

$$R_{form} = \begin{cases} 1 & \text{if } O \sim \texttt{<think>...</think> <answer>...</answer>} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where O refers to the output from the VLM. The format reward is designed to enforce the model to place its reasoning process between '`<think>`', and outputs the final answer within the '`<answer>`' tags. To promote accurate outputs, we add the following accuracy reward,

$$R_{Acc} = \begin{cases} similarity(extract\_answer(O), \text{"no vehicle is appropriate"}) & \text{if } S_{\text{coll}} = \emptyset \\ 1 & \text{if } S_{\text{coll}} \neq \emptyset \wedge extract\_answer(O) \in S_{\text{coll}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $extracted\_answer()$ refers to extracting the content between the tags `<answer>...</answer>` from the VLM output. S_{coll} denotes the set of vehicles that can collide with the ego vehicle in our automated annotation process. If $S_{\text{coll}} = \emptyset$, it indicates that no vehicle in the scene can collide with the ego vehicle, in which case we expect the model to output "no vehicle is appropriate". If $S_{\text{coll}} \neq \emptyset$, at least one adversarial vehicle exists, and the model should output an ID belonging to this set.

3.3 Two-stage Evasion Trajectory Generator

Several safety-critical trajectory simulators have been introduced [32; 37; 6], primarily focusing on generating collision events. However, current multi-view video generators struggle to realistically generate such events, degrading visual quality when control signals cause collisions. To address this, we propose a two-stage evasion trajectory generator. It produces safety-critical yet non-colliding evasion trajectories compatible with current video generators while retaining safety-critical features.

Our generator builds upon a popular controllable trajectory generation framework [38], which uses a diffusion-based model trained on real-world driving data for realistic trajectories. It enables test-time control via loss functions. Since our framework takes a single-frame initial scene as input, we retrain the trajectory generation model to align with this setup. We adopt the closed-loop simulation strategy

described in [38]. At each step n, the model predicts future trajectories from the current scene, applying only the first few to update the scene. This iterates to form the full trajectory sequence.

Once the VLM-based selector identifies the adversarial vehicle V_{adv}, our trajectory generator performs a two-stage simulation. In the first stage, V_{adv} is guided to collide with V_{ego}. If the collision occurs before V_{adv} entering non-drivable areas or hitting others, the collision trajectory is considered valid. In the second stage, we introduce a trajectory update mechanism with an evasion-targeted loss, which guides V_{ego} in evading V_{adv}. Finally, a collision-evasion trajectory sequence is generated.

During the collision-stage trajectory simulation, we employ three loss functions for test-time guidance: an adversarial loss, a no-collision loss, and an on-road loss. The adversarial loss is necessary to encourage V_{adv} to collide with V_{ego}, typically based on their distance [37; 6]. However, this often causes V_{adv} to remain stuck to V_{ego} after collision, resulting in unnatural dynamics—shifting of V_{adv} from aggressive (e.g., rapid acceleration) to passive, ego-like behaviors (e.g., slow driving). To solve this, we propose the following adversarial loss formulation:

$$L_{adv} = \begin{cases} \sum_{t=1}^{T} w_t \cdot d_t \cdot \mathbb{I}(d_t > d_{penalty}) & \text{Before } V_{adv} \text{ collides with } V_{ego} \\ 0 & \text{After } V_{adv} \text{ collides with } V_{ego} \end{cases} \tag{3}$$

where T denotes predicted future steps, d_t is the distance between V_{adv} and V_{ego} at time step t, and d_{penalty} is the non-collision distance threshold. Gradients are detached with respect to V_{ego} to ensure only V_{adv} has the adversarial behavior. A time-decay weight $w_t = \frac{\lambda^t}{\sum_{k=0}^{T-1} \lambda^k}$, controlled by a decay factor λ, emphasizes earlier trajectory predictions and is shared across all losses. Moreover, to avoid unnatural sticking post-collision, we explicitly set the adversarial loss L_{adv} to zero once the collision between V_{adv} and V_{ego} has occurred in the updated trajectories during closed-loop simulation. This leads to more natural post-collision behavior of the adversarial vehicle.

To prevent undesired collisions (excluding that between the ego and adversarial vehicles), we utilize a no-collision loss L_{no_coll}, which penalizes inter-vehicle collisions in denoised trajectories, excluding the ego–adversarial pair. To keep vehicles on drivable areas, an on-road loss L_{on_road} penalizes trajectories entering non-drivable zones and guides them back. Full definitions of L_{no_coll} and L_{on_road} can be found in Appendix B.

Overall, the loss function of the collision stage trajectory simulation can be summarized as

$$L_{coll} = \alpha L_{adv} + \beta L_{no\_coll} + \gamma L_{on\_road} \tag{4}$$

where α, β, γ control each loss's contribution. We obtain trajectory sequences through closed-loop simulation and we filter out trajectories that either do not collide with the ego vehicle or that collide with other vehicles or go off-road beforehand to ensure safety-criticality and physically validity. Subsequently, the collision trajectories are fed into the second stage for evasion trajectory simulation.

During the evasion stage trajectory simulation, we only use $L_{\text{no\_coll}}$ and $L_{\text{on\_road}}$ as follow,

$$L_{eva} = \beta L_{no\_coll} + \gamma L_{on\_road} \tag{5}$$

where the $L_{\text{no\_coll}}$ is applied to all vehicles in the scene to guide V_{ego} in evading V_{adv}. The evasion-stage simulation starts from the same initial scene as the collision stage. During closed-loop rollout, only the ego vehicle's trajectory is updated via the diffusion model; other vehicles retain their collision-stage trajectories to preserve adversarial behavior. This converts a collision scenario to a safety-critical evasion one, staying within the video generator's capability. Finally, we select successful evasive trajectories for video generation.

3.4 Trajectory-to-Video Generator

To convert the simulated collision evasion trajectories into multi-view driving videos, we leverage diffusion-based video generation models tailored for autonomous driving scenarios. To ensure that the generated videos accurately reflect our trajectory-based traffic scenarios, we require a video generation model that explicitly encodes both the ego vehicle's motion and the surrounding vehicles' motion. Moreover, we require a model capable of producing sufficiently long video sequences as safety-critical scenarios typically spans a relatively long duration. Accordingly, we choose UniMLVG [9] as our backbone. UniMLVG delivers state-of-the-art video quality and supports motion control

Table 1: Comparison of baseline effectiveness by evaluating video realism and planner's collision rate (CR) on the videos. Sample-level (CR) measures the average collision probability per valid sample, while scene-level (CR) counts the average number of colliding valid samples per scene.

Methods	Sample-level $CR\uparrow$				Scene-level $CR\uparrow$				FID $\downarrow$	Natural score $\uparrow$
	1s	2s	3s	Avg.	1s	2s	3s	Avg.		
Origin	0.001	0.003	0.007	0.004	0.024	0.049	0.122	0.065	16.254	0.633 ± 0.133
Naive	0.082	0.274	0.445	0.267	0.556	1.861	3.028	1.815	23.346	0.207 ± 0.183
SafeMVDrive	0.097	0.207	0.303	0.202	1.659	3.561	5.220	3.378	20.626	0.560 ± 0.122

Figure 4: Comparison of videos generated by different methods, only showing front view. Origin is ordinary, Naive loses realism near the end, while only ours exhibits both realism and safety-criticality.

signals like 3D bounding boxes, HD maps, and camera conditions. Its multi-task training scheme also reduces autoregressive errors, enabling high-quality long-video synthesis.

We first convert generated trajectories into frame-level control inputs—3D boxes, HD maps, and camera conditions—combined with multi-view initial frames and time-weather text to guide the video generation. Due to the extended duration of the collision evasion scenarios, we use an autoregressive roll-out to generate videos: the final frame of each roll-out serves as the initial frame for the next, and the corresponding control signals for the new time window are used to guide the subsequent generation. Through this iterative process, the complete collision evasion trajectories are ultimately transformed to a multi-view video in the real domain.

4 Experiments

4.1 Experimental Settings

Datasets: We use the large-scale real-world NuScenes dataset [5], featuring diverse driving scenarios. To train the VLM for our adversarial vehicle selector, we randomly select 1,500 samples from the training split and generate the safety-critical annotations within each scene with the automated annotation method (proposed in Section 3.2). The trajectory generation diffusion model is trained on the full training split. For evaluation, 250 samples are randomly selected from the validation split.

Baseline: As the first to generate multi-view, realistic, safety-critical videos, we design two intuitive baselines for comparison. **Naive**, generates safety-critical videos by converting collision trajectories into control signals for the video generator. These trajectories are produced using the vehicle selection method and loss function from [37], combined with our retrained conditional diffusion-based model. **Origin**, uses original NuScenes trajectories to benchmark video quality under natural conditions. All baselines generate videos via UniMLVG with identical settings, as detailed in Section 3.4.

Metrics: To demonstrate that the videos generated by our framework present significant challenges to end-to-end planners and are likely to induce collisions, we evaluate the classical end-to-end planner, UniAD [15] on our generated data. Following the methodology proposed by Li et al. [19], we compute the collision rate for each video sample. Specifically, the collision rate is defined as:

$$cr(t) = \left(\sum_{i=0}^{N} \mathbb{I}_i \right) > 0, \quad N = \frac{t}{0.5}. \tag{6}$$

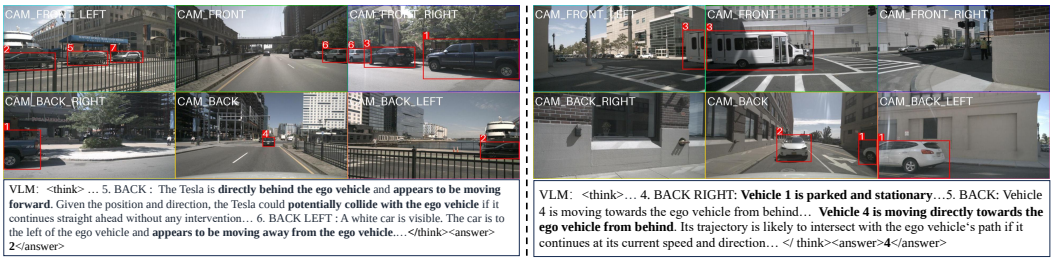

Figure 5: Adversarial vehicle selection examples using the GRPO-finetuned VLM. The VLM accurately analyzes spatial relationships between vehicles and makes reasonable selections.

where N denotes the number of the trajectory points in the planning before t seconds, and $\mathbb{I}_i$ indicates whether the ego vehicle collides at step i. The final collision rate $CR(t)$ is averaged over all samples. We follow this method with one adaptation: if a collision already occurs at the initial step ($\mathbb{I}_0 = 1$), the collision ratio cr(t) for this sample is always 1, making it unsuitable for evaluating planner performance. Therefore, we consider such samples as invalid initializations and exclude them from our evaluation. We report both sample-level and scene-level averaged $CR(t)$: the former represents the averaged $cr(t)$ before time t for each valid sample; the latter reflects the average number of valid samples in which the planner collides before time t within each scene.

Regarding the realism of generated videos, existing automatic metrics such as FVD [26] are widely recognized as insufficient for accurately reflecting perceptual quality and real-world dynamics [11; 2; 4; 12; 31]. Consequently, we rely on human evaluation to obtain a more reliable and authentic assessment. In line with recent studies [11; 2; 4; 3; 8; 27], we employ the Two-Alternative Forced Choice (2AFC) protocol to evaluate the videos (see Appendix A for details) and refer to the resulting preference rate as the realism score in our experiments. In addition, we also compute FID as an auxiliary quantitative metric to facilitate comparison with prior works.

Implement Details: We choose Qwen2.5VL-7B-Instruct [25] as our base VLM due to its strong vision-language understanding capabilities (fine-tuning details in Appendix D). For safety-critical candidate selection, we set the distance threshold $D = 25\,\text{m}$. We retrain the trajectory generation model to align with our single-frame input setup (details in Appendix D). In the two-stage simulation process, we set $\alpha = 1$, $\beta = 50$, $\gamma = 1$ in the collision stage, $\beta = 1$, $\gamma = 1$ in the evasion stage, and use $\lambda = 0.9$ for all loss terms. Ablation studies on these hyperparameters can be found in Appendix F and G. Our video generator produces 19 frames per iteration, using the last frame of the previous roll-out as the reference frame for the next. This results in a final video of 9 seconds at 12 Hz.

4.2 Evaluation of SafeMVDrive

This section compares the realism of videos generated by different baselines and the collision rate (CR) of the UniAD planner on these videos. Each method generates videos from 250 samples randomly selected from the NuScenes validation split (see Appendix E for details). Table 1 shows SafeMVDrive videos significantly increase the planner's CR while maintaining realism comparable to Origin. Specifically, average sample-level CR rises by nearly 0.2, and scene-level CR by 3.3, challenging the planner due to aggressive adversarial trajectories and ego evasion causing speed and acceleration variations. Origin achieves high realism but lacks safety-critical events, thus failing to challenge the planner. Naive's scene-level CR is much lower (1.8 vs. 3.4) and its realism is only about half of SafeMVDrive's, as collision events exceed the video generator's capacity. Its slightly higher sample-level CR stems from vehicles getting stuck post-collision, increasing invalid samples and average CR. All methods show similar FID scores, indicating comparable image quality. Figure 4 compares the videos, highlighting SafeMVDrive as the only method combining high realism with strong safety-critical features.

4.3 Evaluation of Our VLM-based Adversarial Vehicle Selector

In this section, we evaluate the effectiveness of our VLM-based adversarial vehicle selector. On 250 validation scenes, we use automated annotation to identify all vehicles that can collide with the ego vehicle. We compare precision, recall, and F1-score of our VLM-based selector against three heuristic methods: Closest Vehicle, Rule-based Selector [37], and Random Adjacent [6]. As shown in Table 2,

Table 2: Comparison of different methods for adversarial vehicle selection.

Methods	Precision	Recall	F1-score
Closest vehicle	0.528	0.861	0.654
Rule-based selector	0.758	0.497	0.600
Random Adjacent	0.606	0.437	0.507
VLM-based selector	0.750	0.675	**0.710**

Table 3: Comparison of the performance of different models on adversarial vehicle selection.

Model	Precision	Recall	F1-score
Base 3B	0.360	0.596	0.449
Base 7B	0.455	0.530	0.489
Base 72B	0.433	0.602	0.504
SFT-finetuned Model 7B	0.582	0.748	0.655
GRPO-finetuned Model 7B	0.750	0.675	**0.710**

Table 4: Evaluation of the effectiveness of the two-stage simulation.

Methods	Sample-level $CR\uparrow$				Scene-level $CR\uparrow$				FID $\downarrow$	Natural score $\uparrow$
	1	2	3	Avg.	1	2	3	Avg.		
Origin	0.001	0.003	0.007	0.004	0.024	0.049	0.122	0.065	16.254	0.601 ± 0.134
Collision stage only	0.058	0.167	0.228	0.151	0.707	2.049	2.805	1.854	22.204	0.330 ± 0.128
Two-stage simulation	0.097	0.207	0.303	0.202	1.659	3.561	5.220	3.378	20.626	0.569 ± 0.075

our method achieves the highest F1-score, demonstrating its effectiveness in accurately identifying safety-critical vehicles. Figure 5 shows examples of adversarial vehicle selections, where our VLM correctly analyzes positional relationships and driving directions to make appropriate selections.

We further evaluate the effectiveness of our GRPO fine-tuning. As shown in Table 3, the GRPO-finetuned model significantly outperforms the untuned baseline, achieving an F1-score improvement of 0.21 over the strongest 72B base model. We also evaluate supervised fine-tuning (SFT) for comparison (see Appendix D for configurations), but it performs worse than GRPO, with an F1-score reduction of more than 0.05. These findings highlight both the necessity and effectiveness of adopting GRPO for our adversarial vehicle selection task.

4.4 Evaluation of the Effectiveness of the Two-stage Simulation

We propose a two-stage trajectory simulator to generate collision-evasion scenarios that are both safety-critical and within the capability of current multi-view video generators. In this section, we assess the necessity of the two-stage simulation by comparing videos from collision-stage-only trajectories with those from the full two-stage process. Additionally, to assess the naturalness of the generated videos, we include the Origin baseline for comparison. Each method generates videos based on 250 samples randomly selected from the NuScenes validation split. Detailed generation procedures can be found in Appendix E. As shown in Table 4, our two-stage simulation leads to both higher collision rates for the planner and significantly improved realism.

5 Conclusion

We present *SafeMVDrive*, the first framework for generating multi-view safety-critical driving videos in the real-world domain. By strategically combining a safety-critical trajectory simulator with a realistic multi-view video generator, we build a bridge from safety-critical trajectory simulation to multi-view video generation. To address the integration challenge, we introduce a VLM-based adversarial vehicle selector and a two-stage collision-evasion trajectory generation strategy. Experiments demonstrate the effectiveness of our approach in producing realistic and safety-critical multi-view videos, which lead to a high collision rate for end-to-end planners. The generated video data can serve as valuable resources for evaluating and enhancing autonomous driving systems.

Limitations. Since this is the first work to generate multi-view safety-critical driving videos in the real world, we have several limitations. One is the reliance on the complete initial scene configuration, which restricts its ability to generate scenarios directly from raw multi-view camera inputs. Additionally, although our framework uses guidance signals to generate annotations, it lacks a mechanism to discard outdated or irrelevant ones—e.g., vehicles that have exited the ego's view. Future research could address these challenges by reducing dependency on dense annotations and incorporating dynamic filtering strategies to maintain temporal relevance in the guidance signals.

Societal Impact. This work aims to improve the safety and robustness of autonomous driving systems by generating realistic, safety-critical driving scenarios for testing and training. The potential for misuse is limited, as the primary application—autonomous driving—rarely involves malicious intent.

References

[1] Yasasa Abeysirigoonawardena, Florian Shkurti, and Gregory Dudek. 2019. Generating adversarial driving scenarios in high-fidelity simulators. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8271–8277. IEEE.

[2] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. 2024. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11.

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

[4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575.

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

[6] Wei-Jer Chang, Francesco Pittaluga, Masayoshi Tomizuka, Wei Zhan, and Manmohan Chandraker. 2024. Safe-sim: Safety-critical closed-loop traffic simulation with diffusion-controllable adversaries. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXI*, pages 242–258, Berlin, Heidelberg. Springer-Verlag.

[7] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465.

[8] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. 2023. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*.

[9] Rui Chen, Zehuan Wu, Yichen Liu, Yuxin Guo, Jingcheng Ni, Haifeng Xia, and Siyu Xia. 2024. Unimlvg: Unified framework for multi-view long video generation with comprehensive control capabilities for autonomous driving.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

[11] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. 2024. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[12] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. 2024. Factorizing text-to-video generation by explicit image conditioning. In *European Conference on Computer Vision*, pages 205–224. Springer.

[13] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

[14] Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

[15] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[16] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. 2024. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*.

[17] Cheng Li, Keyuan Zhou, Tong Liu, Yu Wang, Mingqiao Zhuang, Huan-ang Gao, Bu Jin, and Hao Zhao. 2025. Avd2: Accident video diffusion for accident video description. *arXiv preprint arXiv:2502.14801*.

[18] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. 2024. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*.

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

[20] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. 2024. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*.

[21] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

[22] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

[23] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. 2024. Drivelm: Driving with graph visual question answering. In *European Conference on Computer Vision*, pages 256–274. Springer.

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454.

[25] Qwen Team. 2025. Qwen2.5-vl.

[26] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*.

[27] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. 2025. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, 133(5):3059–3078.

[28] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

[30] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. 2023. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*.

[31] Jay Zhangjie Wu, Guian Fang, Haoning Wu, Xintao Wang, Yixiao Ge, Xiaodong Cun, David Junhao Zhang, Jia-Wei Liu, Yuchao Gu, Rui Zhao, et al. 2024. Towards a better metric for text-to-video generation. *arXiv preprint arXiv:2401.07781*.

[32] Chejian Xu, Aleksandr Petiushko, Ding Zhao, and Bo Li. 2025. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8797–8805.

[33] Jiawei Zhang, Chejian Xu, and Bo Li. 2024. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15459–15469.

[34] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. 2024. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer.

[35] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

[36] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. 2024. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*.

[37] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. 2023. Language-guided traffic simulation via scene-level diffusion. In *7th Annual Conference on Robot Learning*.

[38] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. 2023. Guided conditional diffusion for controllable traffic simulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3560–3566.

Figure 6: The questionnaire used to evaluate the realism of videos generated by different baselines in the user study.

A User Study Setting

In our experiments, participants are presented with two videos displayed side-by-side and are asked to choose the one they perceive to be of higher visual quality. In addition to choosing one of the two videos, an 'uncertain' option is also provided. A selected video receives 1 point; in the case of an "uncertain" response, both videos receive 0.5 point each. The final realism score is computed as the total number of points received divided by the total number of comparisons. The questionnaire we used is shown in Figure 6. The user studies in Section 4.2 and Section 4.4 are conducted separately. In the experiment of Section 4.2, we randomly select ten initial scenes that are present across all three video sets—Origin, Naive, and SafeMVDrive. For each selected scene, we retrieve the corresponding video from each set, forming ten matched triplets for pairwise comparison. Similarly, in Section 4.4, we randomly select ten initial scenes that exist in all three sets—Origin, Collision Stage Only, and Two-Stage Simulation—and obtain the corresponding video per method for each scene, again resulting in ten matched triplets for evaluation. For each user study, we collect 660 answers from 22 participants.

B No-collision Loss and On-road Loss

To prevent collisions between the vehicles in the scene (except between the ego vehicle and adversarial vehicles), we use the following no-collision loss,

$$L_{no\_coll} = \sum_{t=1}^{T} \sum_{i,j \in \mathcal{A}} w_t \cdot \left(1 - \frac{d_t^{i,j}}{d_{penalty}^{i,j}}\right) \cdot M_{i,j} \cdot \mathbb{I}((d_t^{i,j} < d_{penalty}^{i,j}) \wedge (v_i > v_{th})) \tag{7}$$

where $\mathcal{A}$ is the set of all vehicles in the scene, and $d_t^{i,j}$ and $d_{penalty}^{i,j}$ represent the distance at time step t and minimum non-collision threshold distance while detach the gradient of V_j. v_i is the velocity of the vehicle V_i, and v_{th} is a very small velocity threshold. When the vehicle's velocity exceeds v_{th}, it indicates that the vehicle is not in a completely stationary state. This condition ensures that when a moving vehicle is about to collide with a stationary vehicle, the moving vehicle will adjust its trajectory, rather than causing the stationary vehicle to evade the collision, which is a more natural way to prevent collisions. $M_{i,j}$ is a mask indicating which pairs of agents should evade collisions. In the collision-stage simulation, we configure that all agents, except for the ego and adversarial vehicles, are required to evade collisions. In the evasion-stage simulation, this mask includes all agenets.

To ensure that the vehicles stay within the driving area, we utlilize the following on-road loss,

$$L_{on\_road} = \sum_{t=1}^{T} \sum_{p \in P_{offroad}} w_t \cdot \left(1 - \frac{\min_{q \in P_{onroad}} dist(p,q)}{l_{diag}}\right) \cdot \mathbb{I}(v_i > v_{th}) \tag{8}$$

where $P_{offroad}$ and P_{onroad} are the set of sampled points located off-road and on-road, respectively, within the agent vehicle's bounding box, $dist(p,q)$ is the Euclidean distance between points p and q while detach the gradients of q, and l_{diag} is the diagonal length of the agent vehicle's bounding box. The gradient of this loss pulls the ego vehicle's off-road points toward the nearest on-road points, thereby encouraging the denoised trajectory to remain within drivable areas.

C Prompt

In this section, we present the prompt used in our VLM-based selection of adversarial vehicles in Figure 7. In the prompt, v is substituted with the ego vehicle's velocity in the given scene. For the non-finetuned VLM, we append "Output final answer (number) in <answer> </answer> tags." at the end to ensure it outputs the vehicle ID for evaluation. For the GRPO-finetuned VLM, we append "Output the thinking process in <think> </think> and final answer (number) in <answer> </answer> tags." at the end. For the SFT-finetuned VLM, we use the original prompt without any modifications.

D Finetuning Setting

D.1 VLM finetuning

GRPO-finetuning details: We set the learning rate to 0.00002 with a cosine scheduler, enable DeepSpeed Zero3, set the number of generations in GRPO to 6, do not freeze any modules, and follow other settings from the LoRA fine-tuning configuration in [22]. We fine-tune Qwen-VL 2.5 Instruct [25] using the GRPO algorithm within the framework of [22] for 2600 steps on 4 A800 GPUs.

SFR-finetuing details: We set the learning rate to 0.00002 with a cosine scheduler, use a gradient accumulation step size of 2, do not freeze any modules, and follow other settings from the LoRA fine-tuning configuration of Qwen-VL 2.5 Instruct in [35]. We fine-tune Qwen-VL 2.5 Instruct [25] using the SFT algorithm within the framework of [35] for 2600 steps on a single A800 GPU.

D.2 Trajectory diffusion model finetuning

Originally, the context length of the trajectory generation model [38] is set to 6. Since our framework takes a single-frame initial scene as input, we retrain the model to align with this setup. Following

Prompt: You are a collision scenario analysis expert. Based on the traffic scenario described in the input images, your task is to identify the vehicle most likely to generate collision with the ego vehicle. The scene consists of six camera views surrounding the ego vehicle, arranged as follows: The first row includes three images: FRONT LEFT, FRONT, and FRONT RIGHT. The second row includes three images: BACK RIGHT, BACK, and BACK LEFT. Potential Dangerous Vehicles are highlighted with red boxes, and each vehicle's ID is labeled in the top-left corner of the respective box. Select the one most likely to have its future trajectory modified (through manual intervention) to produce the collision with the ego vehicle. The speed of any car other than ego vehicle can be adjusted, as long as it is in accordance with the laws of physics, so there is no need to analyze the speed of other cars. If no vehicle is suitable for this task, please respond that 'no vehicle is appropriate'. In the current scenario, the initial speed of the ego vehicle is v m/s.

Figure 7: Original Prompt used in our VLM-based selection.

the configuration of [38], we introduce two key modifications: (1) the context length is set to 1, and (2) the motion restriction mask for static vehicles is removed to allow more vehicles to collide with the ego vehicle. The trajectory generation model is trained for 80,000 steps.

E Generated Videos Used for Evaluation.

The videos used in our evaluation are generated under a fixed set of 250 samples , randomly selected from the val split, hereafter referred to as the base dataset. The following sections provide the video generated process in each experiment.

Videos used in Section 4.2: In this section, we compare the generated videos under three baselines: Origin, Naive, and our proposed SafeMVDrive. For the SafeMVDrive set, we apply our full framework to the base dataset and ultimately obtain 41 collision-evasion videos. For the Origin set, we start from the same 41 initial scenarios used in SafeMVDrive set and convert their original NuScenes trajectories into videos. For the Naive set, we apply the naive baseline to all 250 initial scenarios in the base dataset and obtain 72 valid collision trajectories, which are then converted into videos. We evaluate the collision rates of the planner using videos on these three sets. Since FID scores are empirically affected by the number of images used in evaluation—more images generally lead to lower FID values—for fairness, we randomly sample 41 videos from the Naive set to compute FID. The videos used in the user study are described in Section A.

Videos used in Section 4.4: In this section, we compare the generated videos under three methods: Origin, Collision Stage Only, and Two-Stage Simulation. For the Two-Stage Simulation set, we apply our full framework to the base dataset and ultimately obtain 41 collision-evasion videos. For the Origin set, we start from the same 41 initial scenarios used in Two-Stage Simulation set and convert their original NuScenes trajectories into videos. For the Collision Stage Only set, we start from the same 41 initial scenarios used in Two-Stage Simulation set and skip the second simulation to generate videos and eventually get 41 collision videos. We evaluate the collision rates of the planner using videos and fid on these three sets. The videos used in the user study are described in Section A.

Table 5: Ablation Study on Loss Functions in the Two-Stage Evasion Trajectory Generator

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
WHOLE LOSSES	0.750	0.402	0.042	0.002	0.312	5.37
$-L_{adv}$ IN COLLISION STAGE	0.471	0.703	0.034	0.000	0.308	9.11
$-L_{no\_coll}$ IN COLLISION STAGE	0.735	0.410	0.141	0.004	0.312	6.07
$-L_{on\_road}$ IN COLLISION STAGE	0.765	0.490	0.053	0.065	0.314	5.37
$-L_{no\_coll}$ IN EVASION STAGE	0.770	0.127	0.024	0.000	0.313	6.32
$-L_{on\_road}$ IN EVASION STAGE	0.770	0.304	0.057	0.007	0.310	5.23

F Ablation Study on Loss Functions in the Two-Stage Evasion Trajectory Generator

We conduct ablation studies on the loss functions used in our two-stage evasion trajectory simulator. On 250 validation scenes, we first use the VLM selector to identify safety-critical candidates. Then, we remove one specific loss from the two-stage simulation while keeping the remaining loss terms unchanged to simulate.

We report the following metrics:

- **Collision Success Rate (CSR)**: the proportion of adversarial vehicles that successfully collide with the ego vehicle during collision simulation. A higher value is better.
- **Evasion Success Rate (ESR)**: the proportion of adversarial vehicles that successfully evade during evasion simulation. A higher value is better.
- **Collision Rate**: in the final trajectories, the proportion of adversarial vehicles that collide with any vehicle. Since these trajectories are later used for multi-view video simulation and collision cases cannot be rendered, a lower value is preferred. This metric follows the implementation in CTG [38].
- **Off-Road Rate**: in the final trajectories, the proportion of adversarial vehicles that enter non-drivable areas. A lower value is better. This metric follows the implementation in CTG [38].
- **Realism**: in the final trajectories, the degree to which the trajectories resemble real-world behavior. In accordance with [38], We compare the statistical distribution between simulated trajectories and real-world trajectories. A lower value indicates better realism. This metric follows the implementation in CTG [38].
- **Closest Distance**: in the final trajectories, the minimum distance between the adversarial vehicle and the ego vehicle, measured by the distance between their center points, which reflects the potential danger level. A lower value is better.

The experimental results are shown in Table 5. The results demonstrate that each of our loss terms plays a crucial role. Removing L_{adv} during the collision stage leads to a higher Closest Distance, indicating a lower safety criticality of the scenes. Removing $L_{no\_collision}$ results in a higher Collision Rate in the final trajectories. Removing L_{on_road} increases the Off-Road Rate in the final trajectories. During the evasion stage, removing $L_{no\_collision}$ decreases the Evasion Success Rate (ESR), resulting in fewer generated scenarios, while removing L_{on_road} similarly increases the Off-Road Rate in the final trajectories. These results verify the rationality and necessity of our loss design.

G Hyperparameters study of the losses used in the Two-stage Evasion Trajectory Generator

In this section, we investigate the hyperparameters that control the contributions of different loss terms in the two-stage simulation.The positions of these hyperparameters can be found in Equations (4) and (5) in the main text. In addition to these, we also conduct hyperparameter studies on the weight decay rate factor λ. Similar to the previous section, we first use the VLM selector to identify safety-critical candidates on the 250 validation scenes. After that, we vary the hyperparameter corresponding to a specific loss term while keeping the other parameters fixed and then perform the two-stage simulation. We adopt the same evaluation metrics as in the previous section.

Table 6: Ablation Study on α in Collision Stage

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
$\alpha = 0$	0.471	0.703	0.034	0.000	0.308	9.11
$\alpha = 1$ (DEFAULT)	0.750	0.402	0.042	0.002	0.312	5.37
$\alpha = 50$	0.765	0.404	0.054	0.003	0.311	5.33

Table 7: Ablation Study on β in Collision Stage

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
$\beta = 0$	0.735	0.410	0.141	0.004	0.312	6.07
$\beta = 1$	0.735	0.440	0.083	0.005	0.311	5.63
$\beta = 50$ (DEFAULT)	0.750	0.402	0.042	0.002	0.312	5.37

The experimental results are shown in Table 6, 8, 7, 9, 10, and 11. We vary the hyperparameters controlling the loss contributions with values {0, 1, 50}. Overall, setting the value to 0 generally leads to worse performance across various metrics, indicating the necessity of each individual loss term. On the other hand, when the value is within the range of 1 to 50, the differences among the metrics are relatively small, suggesting that our framework is not highly sensitive to hyperparameter selection.

For the weight decay factor λ, we evaluate settings of 0, 0.9, and 1. A value of 0 means that the loss is computed using only the prediction at timestamp 1, while a value of 1 averages the loss across all timestamps (i.e., no decay is applied). We observe that the best performance across all metrics is achieved when $\lambda = 0.9$, which demonstrates the importance of applying a temporal weight decay in our loss design.

Table 8: Ablation Study on γ in Collision Stage

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
$\gamma = 0$	0.765	0.490	0.053	0.065	0.314	5.37
$\gamma = 1$ (DEFAULT)	0.750	0.402	0.042	0.002	0.312	5.37
$\gamma = 50$	0.779	0.396	0.059	0.003	0.311	5.60

Table 9: Ablation Study on β in Evasion Stage

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
$\beta = 0$	0.750	0.127	0.024	0.000	0.313	6.32
$\beta = 1$ (DEFAULT)	0.750	0.402	0.042	0.002	0.312	5.37
$\beta = 50$	0.750	0.422	0.041	0.003	0.311	4.92

Table 10: Ablation Study on γ in Evasion Stage

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
$\gamma = 0$	0.750	0.304	0.057	0.007	0.310	5.23
$\gamma = 1$ (DEFAULT)	0.750	0.402	0.042	0.002	0.312	5.37
$\gamma = 50$	0.750	0.363	0.002	0.048	0.310	5.59

Table 11: Ablation Study on λ

CONFIGURATION	CSR ↑	ESR ↑	COLLISION RATE ↓	OFF-ROAD RATE ↓	REALISM ↓	CLOSEST DISTANCE ↓
$\lambda = 0$	0.640	0.011	0.182	0.091	0.336	8.61
$\lambda = 0.9$ (DEFAULT)	0.750	0.402	0.042	0.002	0.312	5.37
$\lambda = 1$	0.765	0.433	0.060	0.004	0.317	5.85