VILTA: A VLM-in-the-Loop Adversary for Enhancing Driving Policy Robustness

Qimao Chen[1*‡], Fang Li[2,3*], Shaoqing Xu[2,3*†], Zhiyi Lai[3], Zixun Xie[4‡], Yuechen Luo[1‡], Shengyin Jiang[3], Hanbing Li[3], Long Chen[3], Bing Wang[3], Yi Zhang[1§], Zhi-Xin Yang[2§]

[1]Tsinghua University
[2]University of Macau
[3]Xiaomi EV
[4]Peking University
cqm24@mails.tsinghua.edu.cn

Abstract

The safe deployment of autonomous driving (AD) systems is fundamentally hindered by the long-tail problem, where rare yet critical driving scenarios are severely underrepresented in real-world data. Existing solutions including safety-critical scenario generation and closed-loop learning often rely on rule-based heuristics, resampling methods and generative models learned from offline datasets, limiting their ability to produce diverse and novel challenges. While recent works leverage Vision Language Models (VLMs) to produce scene descriptions that guide a separate, downstream model in generating hazardous trajectories for agents, such two-stage framework constrains the generative potential of VLMs, as the diversity of the final trajectories is ultimately limited by the generalization ceiling of the downstream algorithm. To overcome these limitations, we introduce **VILTA** (**V**LM-**I**n-the-**L**oop **T**rajectory **A**dversary), a novel framework that integrates a VLM into the ***closed-loop training*** of AD agents. Unlike prior works, VILTA actively participates in the training loop by comprehending the dynamic driving environment and strategically generating challenging scenarios through direct, fine-grained editing of surrounding agents' future trajectories. This ***direct-editing*** approach fully leverages the VLM's powerful generalization capabilities to create a diverse curriculum of plausible yet challenging scenarios that extend beyond the scope of traditional methods. We demonstrate that our approach substantially enhances the safety and robustness of the resulting AD policy, particularly in its ability to navigate critical long-tail events.

1 Introduction

The rapid advancement of artificial intelligence has catalyzed a paradigm shift in the automotive industry, accelerating the development of autonomous driving (AD) systems. In recent years, significant technological breakthroughs have been achieved across the entire AD stack, from perception and prediction to motion planning and end-to-end learning models (Zhao, Shi, and Zhuo 2024; Huang et al. 2023; Teng et al. 2023; Chen et al. 2024). These innovations have steadily enhanced the safety and efficiency of autonomous vehicles, bringing the prospect of fully autonomous transportation closer to reality.

Despite this remarkable progress, the long-tail distribution of real-world driving data (Liu and Feng 2024; Song

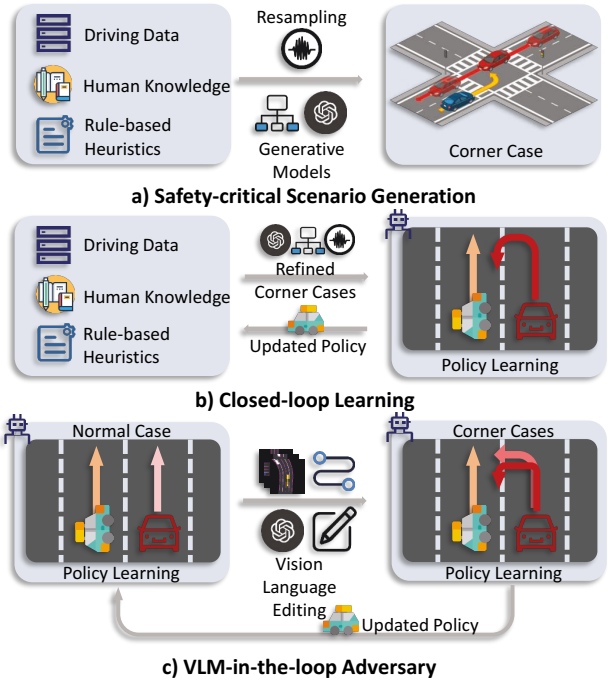

Figure 1: Comparison of different approaches to handling long-tail problem in autonomous driving. (a) Safety-critical scenario generation, which lacks use of the generated scenarios in training; (b) Closed-loop learning, which struggles to generate diverse scenarios; (c) Our proposed VLM-in-the-loop adversary, which is capable of generating scenarios that are both challenging and diverse.

*These authors contributed equally.
†Project Leader.
‡Work done during internships at Xiaomi EV.
§Corresponding authors.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

et al. 2023), a fundamental and persistent challenge of autonomous driving, impedes AD's widespread, reliable deployment. The vast majority of data collected from real-world driving consists of common, uneventful scenarios, such as highway cruising or following vehicles in moderate traffic. Conversely, safety-critical and complex situations called "corner cases" are inherently rare. Such data imbalance leads to a critical performance gap: while AD models excel in handling nominal driving conditions (Li et al. 2024), their ability to navigate rare and hazardous events remains underdeveloped and is a primary source of safety concerns.

To mitigate risks associated with the long-tail problem, researchers have created challenging offline datasets and online environments for testing and validation of pretrained driving models (Xu et al. 2025a,b; Huang et al. 2024c,a; Rowe et al. 2025; Lin et al. 2025; Wang et al. 2021), as depicted in Fig 1a. Pioneering studies (Zhang, Xu, and Li 2024; Mei et al. 2025; Sheng et al. 2025; Zhang et al. 2025) utilized vast knowledge embedded within Vision-Language Models (VLMs) to generate high-level textual descriptions of a scene to adjust the priors of a separate, downstream generation framework. Such indirect, two-stage approaches fail to harness the full potential of VLMs, particularly their powerful and fine-grained generative capabilities. Fig. 1b shows a different line of works where dynamically generated challenging scenarios are integrated into the policy training loop (Zhang et al. 2023; Sheng et al. 2025; Niu et al. 2024). However, these generated scenarios still relies on rule-based heuristics, resampling methods and generative models learned from existing datasets, which lacks the broad generality to create scenarios.

To address these limitations, we introduce a novel framework, ***VILTA***, which directly integrates a multimodal large model into the closed-loop training of autonomous driving agents for enhancing driving policy robustness, as shown in Fig. 1c. In our proposed method, the VLM actively participates in the training loop by first comprehending the fine-grained representation of dynamic driving environment and then strategically generating challenging scenarios. This is achieved by having the VLM edit the future trajectories of surrounding agents to create scenarios that are both plausible and difficult for the ego-vehicle to handle.

By doing so, our method makes two key contributions:

- Our framework fully leverages VLM's ability to produce a diverse and challenging curriculum of safety-critical scenarios integrated with policy learning.
- We introduce a highly efficient generation mechanism, where direct trajectory editing by the VLM provides a targeted way to craft adversarial interactions. Experiments demonstrate that this approach improves the safety and robustness of the resulting autonomous driving policy, especially in its ability to navigate long tail events. Empirical analysis confirms that trajectories generated via VLM editing are more challenging in nature compared to those generated directly.

2 Related Work

2.1 Safety-Critical Scenario Generation

Generating diverse and complex safety-critical scenarios in simulation is essential for evaluating autonomous driving (AD) systems, bypassing the high costs of real-world testing (Feng et al. 2023). Existing generation approaches include several main categories (Ding et al. 2023). Data-driven methods (Wheeler, Kochenderfer, and Robbel 2015; Wu et al. 2020) suffer from the imbalanced nature of real-world data. Knowledge-based methods (Xu et al. 2025a) use predefined rules, which limits scenario diversity and generalizability. Adversarial generation (Kuutti, Fallah, and Bowden 2020) can produce diverse scenarios but fails to leverage real-world driving data.

More recently, Foundation Models (FMs) have been applied for their powerful generalization capabilities (Bommasani et al. 2021; Gao et al. 2025b,a). For instance, Chatscene (Zhang, Xu, and Li 2024) translates language instructions into scenarios, CrashAgent (Li et al. 2025) converts crash reports into simulations. However, these FM-based methods typically adopt a two-stage approach: an FM first describes a scenario, and a downstream module then performs the generation. This indirect process fails to fully exploit the FM's generalization power and can be inefficient.

2.2 Closed-loop Learning for Autonomous Driving

Closed-loop learning is increasingly applied to AD to enhance model robustness by training on progressively challenging situations (Lowd and Meek 2005). Generating evolving curricula through environmental design can enhance the robustness of reinforcement learning (RL) agents (Parker-Holder et al. 2022). Notable works include CAT (Zhang et al. 2023), which uses a closed-loop adversarial framework with real-world data, and SDM (Niu et al. 2023), which models vehicle interactions as a Stackelberg game. CurricuVLM (Sheng et al. 2025) further integrates Vision-Language Models (VLMs) to analyze an AV's weaknesses and create personalized training curricula. However, it still operates within a two-stage framework, using VLM's comprehension ability to guide the resampling of scenarios from an existing dataset. In contrast, our work breaks this two-stage paradigm, enabling VLM to utilize both its understanding and generative capabilities to directly craft complex scenarios, significantly improving the diversity and novelty of the generated challenges.

3 Preliminaries

3.1 Partially Observable Markov Decision Process

A Partially Observable Markov Decision Process (POMDP) (Cassandra 1998) extends the Markov Decision Process (MDP) (Puterman 1990) to formalize sequential decision-making under state uncertainty. A POMDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$. After an agent takes an action $a \in \mathcal{A}$ in a state $s \in \mathcal{S}$, the environment transitions via $\mathcal{T}(s'|s, a)$ and yields a reward $\mathcal{R}(s, a, s')$.

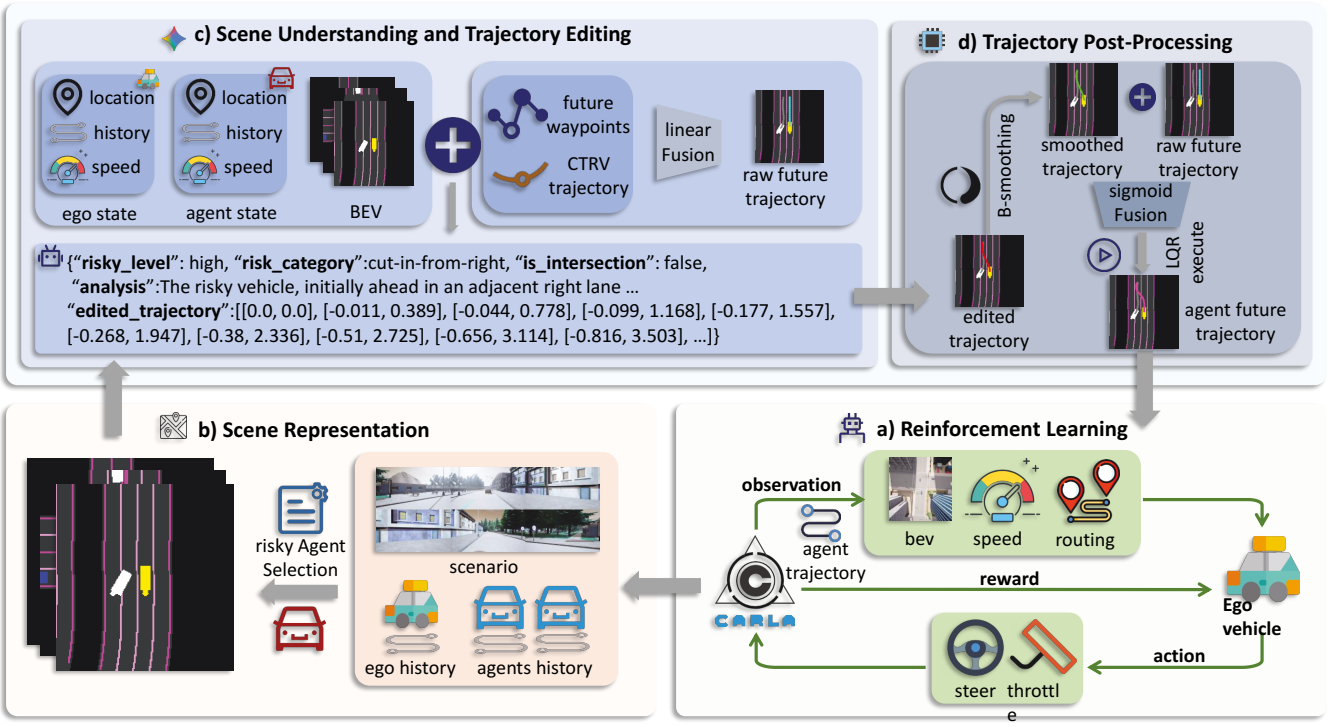

Figure 2: Overview of VILTA framework. (a) Reinforcement learning environment executes the edited trajectory of the risky agent, facilitates the training of the ego agent, and provides the initial scene representation; (b) The raw scene data is processed into a specific representation formatted for input to VLM; (c) Gemini performs both scene understanding and trajectory editing in a single pass, outputting a raw edited trajectory; (d) The post-processing stage ensures that the final trajectory is both smooth and kinematically feasible.

The agent receives an observation $o \in \Omega$ based on the probability $\mathcal{O}(o|s', a)$ instead of the true state. The objective is to find an optimal policy $\pi(a|s)$ that maximizes the expected discounted return $E\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1})|\pi\right]$, where γ is the discount factor.

3.2 Problem Formulation

Our objective is to train the autonomous vehicle by alternately exposing it to both complex and nominal scenarios, denoted as $\mathcal{S}_C, \mathcal{S}_N$, respectively. Based on this training approach, the objective is formulated in Eq. 1 as follows:

$$\pi^* = \arg\max_{\pi} E_{\substack{a_t \sim \pi(a|s) \\ \mathcal{S} \in \{\mathcal{S}_N, \mathcal{S}_C, \dots\} \\ s_{t+1} \sim \mathcal{T}(s'|s,a,\mathcal{S})}} \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s, a, s')|\pi \right] \tag{1}$$

, where $\mathcal{T}(s_{t+1}|s_t, a_t, \mathcal{S})$ denotes state transition in scenario $\mathcal{S}$. The notation $\mathcal{S} \in \{\mathcal{S}_N, \mathcal{S}_C, \mathcal{S}_N, \mathcal{S}_C, \dots\}$ indicates that the scenario $\mathcal{S}$ is drawn alternately from $\mathcal{S}_N$ and $\mathcal{S}_C$.

4 Methodology

The overall framework of our proposed method is illustrated in Fig. 2. We choose to use gemini-2.5-flash (Comanici et al. 2025) as our VLM, which is a publicly accessible API at no cost. This section begins by detailing the overall training framework for VILTA. Subsequently, we introduce the representation for scenarios, and then describe each of the core components in sequence: scene understanding, trajectory editing, and trajectory post-processing.

4.1 VLM In-the-loop Trajectory Adversary

As depicted in Fig. 2, VILTA operates as a closed-loop training framework. During the training process, the RL environment provides state information about the current scene, which is then consolidated into a scene representation that is fed as input to the VLM. The VLM simultaneously performs scene understanding and trajectory editing. Finally, the resulting edited trajectory undergoes post-processing and is used to control the behavior of the "risky agent" within the RL environment, thus closing the loop.

4.2 Scene Representation

To comprehensively capture scene information, we first structure the scene into a Bird's-Eye-View (BEV) representation as shown in Fig. 2 (b). Next, the vehicle closest to the ego vehicle within a predefined circle hazardous zone is selected as the agent responsible for generating the critical scenario. We then identify this agent's driving mode relative to the ego vehicle and assign a corresponding hazardous maneuver based on the selection rule as detailed in Tab. 1.

Driving Direction	Driving Lane	Longitudinal Postion	Horizontal Position	Hazardous Maneuver
Same	Same	Front	–	Sudden-brake
		Rear		Overtake
	Different	–	Left	Cut-in-left
			Right	Cut-in-right
Opposite	–	–	–	Lane-encoachment
				U-turn

Table 1: Agent's hazardous maneuver selection rule.

4.3 Scene Understanding

Given the scene's BEV representation, the agent's driving mode relative to the ego vehicle, and the specified hazardous maneuver, the VLM outputs a structured understanding and analysis of the scene, as depicted in Fig. 2c. VLM leverages its scene understanding capabilities to assess the current risk level and evaluate the appropriateness of the intended hazardous maneuver, using this analysis to produce the final edited trajectory. In addition, VLM performs a check to determine if the ego vehicle is located at an intersection. If so, greater flexibility is permitted in the subsequent trajectory editing.

4.4 Trajectory Editing

To ensure usability and generalizability, we leverage the pre-trained VLM without any fine-tuning to generate trajectories. A direct consequence, however, is that future trajectories generated directly via the standard Vision-Language-Action (VLA) (Brohan et al. 2023) paradigm would lack the challenging nature, as shown in Sec. 5.4.

To address this challenge, we instead employ a Vision-Language-Editing (VLE) paradigm. The inspiration for this approach is drawn from editing methodologies in the field of image manipulation, where a model can alter an image to adopt a specific style while preserving its underlying structure (Meng et al. 2021; Zhang, Rao, and Agrawala 2025). We hypothesize that a similar phenomenon exists in the generation processes of large models, where editing raw information can preserve certain underlying structures. Applying this premise to the problem of challenging trajectory generation, our goal is to edit an initial, normal trajectory to render it more difficult, while simultaneously retaining its original overall motion trend. In this approach, the VLM does not generate the trajectory from scratch; rather, it edits an initial trajectory for the agent which is produced by a rule-based method. More specifically, we employ a weighted fusion method that combines the trajectory output from a Constant Turn Rate and Velocity (CTRV) model with predefined waypoints on the map. Details can be found in Alg. 1. The generated trajectory is predominantly influenced by the CTRV model during its initial phase and by the map waypoints towards its conclusion. This baseline trajectory provides a stable foundation that effectively grounds the VLM's generation process, guiding it to produce feasible results and focusing its power on the editing task. By editing this fused trajectory (denoted as T_{base}), we enhance the realism and reliability of the VLM's final output.

Algorithm 1: Trajectory Fusion via Linear Weighting

Input:
T_{model}: Trajectory from CTRV model.
T_{map}: Map's waypoints.
N: Output trajectory length.
Output: T_{base}: The final fused trajectory.

$T_{base} \leftarrow [\,]$ `// Initialize an empty list for the fused trajectory`
for $i = 1$ to N **do**
  $p_{fused} \leftarrow (1 - \frac{i}{N}) \cdot T_{model}[i] + \frac{i}{N} \cdot T_{map}[i]$
  Append p_{fused} to T_{base}
end for
return T_{base}

4.5 Trajectory Post-processing

While the design of the editing module enhances the reliability of the VLM's output trajectory, it does not in itself guarantee that the trajectory is kinematically feasible. To address this, we introduce a post-processing module comprising three components. We denote the output trajectory from the editing module as T_{edit}. First, B-spline smoothing (He and Shi 1998) is applied to the raw edited trajectory T_{edit}, yielding the smoothed trajectory T_B. The second component is sigmoid fusion, where the smoothed trajectory T_B is combined with the rule-based fused trajectory T_{base} using a sigmoid weighting scheme:

$$
\begin{aligned}
w_i &= \frac{1}{1 + \exp\left(\frac{M(2i-N)}{N}\right)},\ i = 1, 2, \ldots, N \\
T_{curve}[i] &= w_i \cdot T_{base}[i] + (1 - w_i) \cdot T_B[i]
\end{aligned}
\tag{2}
$$

where N is the trajectory length and M is the weighting factor. The weight w_i is initially high for T_{base} to ensure behavioral continuity and prevent abrupt turns, and rapidly decreases to shift the trajectory's reliance towards the smoothed path. Finally, the third component uses a Linear-Quadratic Regulator (LQR) controller (Khatoon, Gupta, and Das 2014), taking the agent's kinematics to execute T_{curve} and generate the final, kinematically plausible result T_{final}.

5 Experiments

5.1 Experimental Setups

Environments We conduct our experiments in the CARLA simulator (Dosovitskiy et al. 2017). All models are trained on the Town02 map and evaluated on Town01, Town02, and Town03. For a detailed description of the map layout, please refer to the supplementary materials. Each map is populated with 20 autopilot vehicles, one of which is designated as a "risky agent" per the methodology in Sec. 4.2. For each episode, the ego vehicle navigates between a randomly sampled start and destination, following the shortest path computed by the A* algorithm.

RL Setting Our state representation follows VLM-RL (Huang et al. 2024b), concatenating an ego-centric BEV

Town	Envrionment	Model	Route Completion ↑	Total Distance ↑	Crash Rate ↓	Collision Per Kilometer ↓	Collision Speed ↓	Average Speed ↑
01	Challenging	VLM-RL	0.80±0.03	4341.78±104.41	0.37±0.06	1.66±0.44	1.95±0.25	22.85±0.99
		CAT	0.70±0.06	3715.67±294.55	0.50±0.10	2.49±0.79	3.65±1.30	23.84±0.15
		VILTA	0.93±0.09	5197.11±639.33	0.13±0.15	0.67±0.63	0.29±0.17	22.94±1.14
	Normal	VLM-RL	0.88±0.10	4758.34±690.13	0.20±0.17	0.82±0.98	1.00±3.25	23.13±0.65
		CAT	0.85±0.05	4912.2±134.19	0.33±0.15	1.45±0.78	1.99±0.25	23.03±0.28
		VILTA	0.92±0.04	5156.23±208.23	0.13±0.06	0.96±0.57	2.17±1.57	22.73±0.56
02	Challenging	VLM-RL	0.68±0.03	1441.66±121.46	0.57±0.06	18.86±3.39	4.87±0.95	18.96±2.11
		CAT	0.58±0.01	1076.81±17.70	0.73±0.06	22.44±0.44	6.61±0.79	21.87±0.12
		VILTA	0.77±0.05	1588.91±157.71	0.50±0.10	18.13±7.36	3.63±3.13	20.13±2.74
	Normal	VLM-RL	0.86±0.06	1746.99±163.41	0.23±0.06	4.83±1.21	1.27±0.97	21.91±0.75
		CAT	0.89±0.06	1861.00±178.35	0.30±0.10	2.47±0.94	1.44±0.91	21.69±1.25
		VILTA	0.91±0.08	1884.12±228.64	0.27±0.15	2.47±2.47	2.83±3.14	20.25±0.60
03	Challenging	VLM-RL	0.62±0.08	2532.29±462.51	0.50±0.10	31.22±6.83	8.36±2.11	21.83±0.31
		CAT	0.62±0.02	2569.61±149.02	0.47±0.06	42.57±2.01	7.41±0.08	20.72±0.04
		VILTA	0.68±0.06	2862.55±270.66	0.40±0.10	29.82±7.01	7.89±2.92	21.91±1.06
	Normal	VLM-RL	0.80±0.03	3461.05±183.96	0.33±0.06	12.36±0.66	4.08±2.49	23.61±0.27
		CAT	0.90±0.01	3545.09±44.65	0.23±0.06	0.75±0	0.33±0.14	22.26±0.84
		VILTA	0.87±0.04	3787.55±171.41	0.23±0.06	11.00±0.75	2.17±0.25	22.15±0.55
Total	Challenging	VLM-RL	2.10	8305.73	1.44	51.74	15.18	63.64
		CAT	1.90	7362.09	1.70	67.5	17.67	**66.43**
		VILTA	**2.38**	**9648.57**	**1.03**	**48.62**	**11.81**	64.98
	Normal	VLM-RL	2.54	9966.38	0.76	18.01	6.35	**68.65**
		CAT	2.64	10318.29	0.86	**4.67**	**3.76**	66.98
		VILTA	**2.70**	**10827.90**	**0.63**	14.43	7.17	65.13

Table 2: Performance comparison with several baselines. Averaged over 3 random seeds. Best results are marked in **bold**.

semantic map, the ego vehicle's state (steering, throttle, speed), and navigation information from 15 upcoming waypoints. The action space is a 2D continuous vector representing the steering angle in [-1, 1] and a combined throttle/brake value, where positive and negative values control throttle and brake, respectively. For detailed information on the RL models and parameters used, please refer to the supplementary materials.

To steer the agent towards safe and efficient driving behaviors, we designed a comprehensive reward function, R_{total}, which is a weighted sum of three key components: a driving style reward (R_{style}), a vehicle-following reward (R_{follow}), and a safety penalty (P_{safety}). The total reward for a given state is calculated as:

$$R_{\text{total}} = \alpha R_{\text{style}} + \beta R_{\text{follow}} - \gamma P_{\text{safety}} \tag{3}$$

where α, β, and γ are hyperparameters that balance the trade-offs between these objectives.

- **Driving Style Reward (R_{style})** encourages smooth, stable, and centered driving. Following VLM-RL (Huang et al. 2024b), it is a product of four normalized factors, ensuring the agent must perform well across all aspects to achieve a high reward:

$$R_{\text{style}} = r_{\text{speed}} \cdot r_{\text{center}} \cdot r_{\text{angle}} \cdot r_{\text{stable}} \tag{4}$$

  Here, r_{speed} rewards maintaining a velocity within a predefined target range. r_{center} is a function of the vehicle's lateral distance from the lane centerline. r_{angle} penalizes angular deviation between the vehicle's heading and the lane's direction. Finally, r_{stable} promotes stability by penalizing high variance in the vehicle's lateral position over a recent time window.

- **Following Reward (R_{follow})** incentivizes car-following behavior by encouraging ego vehicle to maintain a safe, speed-dependent safe distance from the front vehicle:

$$R_{\text{follow}}(d, v) = \begin{cases} \text{clip}(\frac{d - d_{\text{danger}}}{d_{\text{opt}}(v) - d_{\text{danger}}}, 0, 1) & \text{if } d \leq d_{\text{opt}}(v) \\ \frac{d_{\text{opt}}(v)}{d} & \text{if } d > d_{\text{opt}}(v) \end{cases} \tag{5}$$

  where d is the distance to front vehicle, d_{danger} is the dangerous distance threshold, and $d_{\text{opt}}(v)$ is the optimal following distance considering ego speed v.

- **Safety Penalty (P_{safety})** serves as a general collision avoidance mechanism. It is a penalty term that increases sharply as the distance to the nearest surrounding vehicle (in any direction) falls below a critical safety threshold.

Evaluation Metrics To provide a comprehensive assessment of the autonomous vehicle's performance, we evaluate both its safety and efficiency using a suite of metrics. Driving progression is measured by the average speed (**AS**, km/h), route completion rate (**RC**), and total distance (**TD**, m) traveled per episode. Safety is assessed based on the overall collision rate (**CR**), collision speed (**CS**, km/h), and collisions per kilometer (**CPM**).

Baselines Our method is compared with two baselines:

- **VLM-RL (Huang et al. 2024b):** A unified framework that combines VLMs with RL for safe autonomous driving by generating reward signals from image observations and contrasting language features from CLIP (Radford et al. 2021).
- **CAT (Zhang et al. 2023):** A framework that continuously improves the safety of autonomous driving agents by training them on safety-critical scenarios dynamically generated from pretrained motion prediction model.

5.2 Performance Comparison

Our experiments are designed to answer the following two key questions:

- Can VILTA agent maintain safe and efficient driving performance when confronted with challenging scenarios?
- Does VILTA suffer from "catastrophic forgetting", where its performance on normal scenarios degrades after training on challenging ones?

We present a detailed evaluation of proposed VILTA against several baselines as shown in Tab. 2. The behavior of the surrounding vehicles differs by scenario type: in "normal" scenarios, they operate under the control of a standard autopilot. In "challenging" scenarios, they are dynamically controlled by the VILTA editing module to execute hazardous maneuvers targeted at the ego vehicle. The ego vehicle is tasked with following a predefined navigation route. All models are trained in Town 02 and evaluated in Town 01-03 in both challenging and normal scenarios.

Beginning with the challenging scenarios, VILTA exhibits superior performance across all tested Towns. In terms of safety, it achieved the highest route completion rate, exceeding VLM-RL and CAT by 13.3% and 25.3% in total, respectively. Simultaneously, VILTA records the lowest crash rate, representing a reduction of 28.5% and 39.4% compared to VLM-RL and CAT. Also, VILTA exhibits lowest average impact velocity, clearly indicating its enhanced safety profile. Furthermore, from an efficiency perspective, its average speed remained comparable to the leading baseline, CAT, showing only a marginal decrease of 2.2%. This demonstrates that the significant safety gains were not achieved at the cost of efficiency.

Turning to the normal scenarios, the statistics reveal that VILTA performs similarly to the SOTA baselines. Specifically, VILTA achieved the best results for route completion rate, total distance traveled, and crash rate. In terms of efficiency, its performance was only marginally lower (by 5.1%) than the top-performing baseline, VLM-RL. These findings provide strong evidence that our training method, which alternates between challenging and normal scenarios, effectively mitigates catastrophic forgetting, allowing VILTA to preserve a strong balance of both safety and efficiency in standard driving conditions.

5.3 Ablation Study

To validate the effectiveness of each component in our proposed method, we conducted the following ablation studies:

Env.	Model	RC↑	CR↓	CPM↓
C	w/o PP	0.73±0.03	0.60±0	18.75±2.09
	w/o FR	0.74±0.19	0.60±0.26	19.28±8.34
	w/o VLE	0.75±0.04	0.53±0.06	18.50±1.04
	x2	0.77±0.05	0.50±0.10	18.13±7.36
	x4	0.85±0.03	**0.27±0.06**	11.21±14.89
	x8	**0.87±0.01**	0.30±0.10	**5.33±2.27**
	x16	0.85±0.08	0.30±0.10	5.56±3.12
N	w/o PP	0.89±0.05	0.20±0.10	2.24±0.50
	w/o FR	0.80±0.05	0.37±0.06	4.88±0.22
	w/o VLE	0.91±0.07	0.31±0.06	2.60±0.88
	x2	0.91±0.08	0.27±0.15	2.47±2.47
	x4	**0.94±0.05**	0.20±0.10	**1.30±0.85**
	x8	0.92±0.05	0.23±0.15	2.30±1.62
	x16	0.91±0.07	**0.17±0.06**	1.60±0.88

Table 3: Ablation study results. Averaged over 3 random seeds. "C" and "N" are brief denotations of "Challenging" and "Normal".

- W/o Post-Processing (w/o PP): The risky agent is controlled directly by the raw trajectory generated from VLM without any subsequent refinement.
- W/o Following Reward (w/o FR): The model is trained without the reward component R_{follow} designed to encourage route adherence in Eq. 3.
- W/o Vision-Language-Editing (w/o VLE): The risky agent is controlled by trajectories generated from VLM, without using VLE paradigm.
- Different Alternation Frequency between Normal and Challenging Scenarios: This ablation study investigates the effect of varying the frequency of challenging scenarios during RL training across three conditions: one challenging scenario is presented per every 2 (origin), 4, 8 or 16 normal scenarios, briefly denoted as x2, x4, x8 and x16 respectively.

As shown in Tab. 3, the inclusion of post-processing improves the route completion rate in challenging scenarios by 5.5% while reducing CPM by 3.3%. Adding the "Following Reward" component boosts the route completion rate by 4.1% and decreases CPM by 6.0%. The ablation of "w/o VLE" likewise leads to a decline in performance on all evaluated metrics. This is because the trajectories generated by our VLE paradigm and post-processing module are kinematically feasible, ensuring that alternating between "hazardous" and "normal" scenarios during training does not disrupt the physical consistency of the simulation environment.

Regarding the scenario alternation frequency, the results indicate that performance degrades when challenging scenarios are introduced either too infrequently (a 1:16 challenging-to-normal ratio) or too frequently (a 1:2 ratio). This suggests that finding an optimal balance is a critical empirical consideration for the training process. In our experiments, a frequency of one challenging scenario per eight normal ones (an 8:1 ratio) yielded the best trade-off and is thus considered the most suitable configuration.

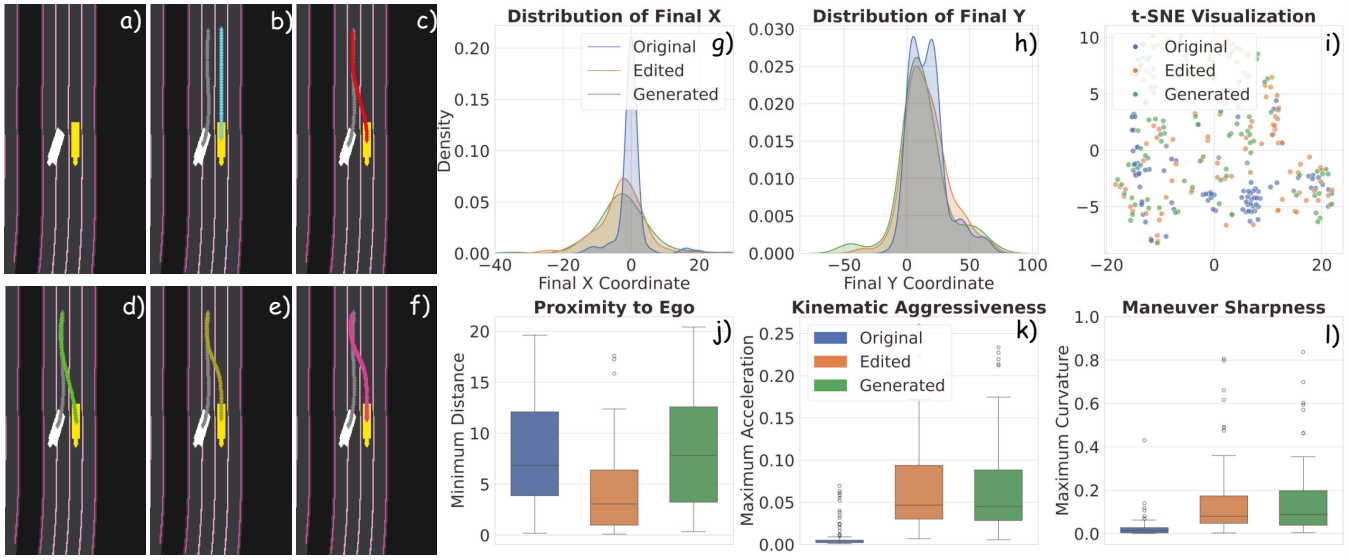

Figure 3: Trajectory visualization and empirical analysis. Panels (a-f) provide trajectory visualizations; panels (g-i) visualize the diversity of the trajectory distributions; and panels (j-l) present box plots of key trajectory features.

5.4 Empirical Analysis

In this section, we present both qualitative and quantitative analyses of the trajectories before and after the editing process. Fig. 3 a-f illustrate the detailed pipeline of our editing process, with the yellow and white vehicles representing the "risky agent" and the ego vehicle, respectively. Initially, the scene representation includes the road structure and past trajectories (a). A raw future trajectory is then generated by linearly fusing CTRV model's prediction with map waypoints and get T_{base} (b, cyan line). This raw trajectory is subsequently edited by the VLM (c, T_{edit}, red line) and then smoothed using a B-spline (d, T_B, green line). A sigmoid curve fusion stage then blends the smoothed trajectory from (d) with the raw future trajectory from (b) to produce an intermediate path (e, T_{curve}, yellow line). Finally, an LQR controller executes this path to produce the final, kinematically plausible trajectory (f, T_{final}, pink line).

Fig. 3 g-l provide a quantitative analysis of 100 randomly sampled cases to verify the increased challenge and diversity of the edited trajectories (**Edited**) compared with the original agent future trajectories (T_{base}, **Original**) and trajectories directly generated from VLM without using our VLE paradigm (T_{gen}, **Generated**). Fig. 3g and 3h show that the endpoint distribution of the final trajectories (T_{final}) is more dispersed than the initial ones (T_{base}). In Fig. 3i, we use t-SNE (Maaten and Hinton 2008) to visualize a set of trajectory features (length, average speed, max curvature, endpoint), where the resulting 2D distribution for T_{final} is significantly broader. The directly generated trajectories T_{gen} exhibit a similar level of distributional diversity as the edited trajectories.

Fig. 3 j-l plot the distributions for minimum distance to the ego vehicle, acceleration, and steering angle. In Fig. 3j, the minimum distance to the ego vehicle for the directly generated trajectories is comparable to that of the original ones, which indicates that the direct generation method merely enhances diversity without increasing the level of challenge to the ego vehicle. Meanwhile, the edited trajectories T_{final} feature smaller minimum distances as well as more extreme accelerations (Fig. 3k) and steering angles (Fig. 3l), all with wider distributions. Taken together, these results statistically confirm that our method enhances both the diversity and the challenging nature of the generated trajectories.

6 Conclusions and Limitations

This paper presented VILTA, a novel VLM-in-the-loop training framework for autonomous driving. By strategically integrating VLM into the training loop to perform fine-grained trajectory editing, VILTA generates a diverse curriculum of challenging scenarios. Experiments show that VILTA significantly enhances policy robustness in hazardous situations, increasing route completion and reducing collisions without degrading performance in normal conditions. The results validate that direct VLM integration is a highly effective strategy for creating more resilient autonomous driving agents. Despite its success, our work has several limitations. First, validation is confined to simulation, necessitating future real-world testing to bridge the "sim-to-real" gap. Second, the framework's performance is tied to the capabilities of the underlying VLM, warranting investigation into model-specific biases. Third, our current implementation focuses on single-agent adversarial scenarios, while real-world critical events can involve multiple actors. Finally, the initial identification of threats relies on predefined rules, suggesting an opportunity for more dynamic, learning-based threat discovery in future work.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant no.62473224 and 62461160260), in part by National Key Research and Development Program of China under grant 2023YFE0205800.

References

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Cassandra, A. R. 1998. A survey of POMDP applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, volume 1724.

Chen, L.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; and Li, H. 2024. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Ding, W.; Xu, C.; Arief, M.; Lin, H.; Li, B.; and Zhao, D. 2023. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 24(7): 6971–6988.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16.

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

Gao, Y.; Piccinini, M.; Moller, K.; Alanwar, A.; and Betz, J. 2025a. From Words to Collisions: LLM-Guided Evaluation and Adversarial Generation of Safety-Critical Driving Scenarios. *arXiv preprint arXiv:2502.02145*.

Gao, Y.; Piccinini, M.; Zhang, Y.; Wang, D.; Moller, K.; Brusnicki, R.; Zarrouki, B.; Gambi, A.; Totz, J. F.; Storms, K.; et al. 2025b. Foundation Models in Autonomous Driving: A Survey on Scenario Generation and Scenario Analysis. *arXiv preprint arXiv:2506.11526*.

He, X.; and Shi, P. 1998. Monotone B-spline smoothing. *Journal of the American statistical Association*, 93(442): 643–650.

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

Huang, R.; Zhuo, G.; Xiong, L.; Lu, S.; and Tian, W. 2023. A Review of Deep Learning-Based Vehicle Motion Prediction for Autonomous Driving. *Sustainability (2071-1050)*, 15(20).

Huang, Z.; Sheng, Z.; Qu, Y.; You, J.; and Chen, S. 2024b. VLM-RL: A Unified Vision Language Models and Reinforcement Learning Framework for Safe Autonomous Driving. *arXiv preprint arXiv:2412.15544*.

Huang, Z.; Zhang, Z.; Vaidya, A.; Chen, Y.; Lv, C.; and Fisac, J. F. 2024c. Versatile behavior diffusion for generalized traffic agent simulation. *arXiv preprint arXiv:2404.02524*.

Khatoon, S.; Gupta, D.; and Das, L. 2014. PID & LQR control for a quadrotor: Modeling and simulation. In *2014 international conference on advances in computing, communications and informatics (ICACCI)*, 796–802. IEEE.

Kuutti, S.; Fallah, S.; and Bowden, R. 2020. Training adversarial agents to exploit weaknesses in deep control policies. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 108–114. IEEE.

Li, M.; Ding, W.; Lin, H.; Lyu, Y.; Yao, Y.; Zhang, Y.; and Zhao, D. 2025. CrashAgent: Crash Scenario Generation via Multi-modal Reasoning. *arXiv preprint arXiv:2505.18341*.

Li, Z.; Yu, Z.; Lan, S.; Li, J.; Kautz, J.; Lu, T.; and Alvarez, J. M. 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*, 14864–14873.

Lin, H.; Huang, X.; Phan, T.; Hayden, D.; Zhang, H.; Zhao, D.; Srinivasa, S.; Wolff, E.; and Chen, H. 2025. Causal composition diffusion model for closed-loop traffic generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 27542–27552.

Liu, H. X.; and Feng, S. 2024. Curse of rarity for autonomous vehicles. *nature communications*, 15(1): 4808.

Lowd, D.; and Meek, C. 2005. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 641–647.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Mei, Y.; Nie, T.; Sun, J.; and Tian, Y. 2025. Seeking to collide: Online safety-critical scenario generation for autonomous driving with retrieval augmented large language models. *arXiv preprint arXiv:2505.00972*.

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Niu, H.; Chen, Q.; Li, Y.; Zhang, Y.; and Hu, J. 2023. Stackelberg driver model for continual policy improvement in scenario-based closed-loop autonomous driving. *arXiv preprint arXiv:2309.14235*.

Niu, H.; Xu, Y.; Jiang, X.; and Hu, J. 2024. Continual Driving Policy Optimization with Closed-Loop Individualized Curricula. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 6850–6857. IEEE.

Parker-Holder, J.; Jiang, M.; Dennis, M.; Samvelyan, M.; Foerster, J.; Grefenstette, E.; and Rocktäschel, T. 2022. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, 17473–17498. PMLR.

Puterman, M. L. 1990. Markov decision processes. *Handbooks in operations research and management science*, 2: 331–434.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Rowe, L.; Girgis, R.; Gosselin, A.; Paull, L.; Pal, C.; and Heide, F. 2025. Scenario dreamer: Vectorized latent diffusion for generating driving simulation environments. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 17207–17218.

Sheng, Z.; Huang, Z.; Qu, Y.; Leng, Y.; Bhavanam, S.; and Chen, S. 2025. Curricuvlm: Towards safe autonomous driving via personalized safety-critical curriculum learning with vision-language models. *arXiv preprint arXiv:2502.15119*.

Song, Z.; He, Z.; Li, X.; Ma, Q.; Ming, R.; Mao, Z.; Pei, H.; Peng, L.; Hu, J.; Yao, D.; et al. 2023. Synthetic datasets for autonomous driving: A survey. *IEEE Transactions on Intelligent Vehicles*, 9(1): 1847–1864.

Teng, S.; Hu, X.; Deng, P.; Li, B.; Li, Y.; Ai, Y.; Yang, D.; Li, L.; Xuanyuan, Z.; Zhu, F.; et al. 2023. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6): 3692–3711.

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

Wheeler, T. A.; Kochenderfer, M. J.; and Robbel, P. 2015. Initial scene configurations for highway traffic propagation. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 279–284. IEEE.

Wu, C.; Chen, L.; Wang, G.; Chai, S.; Jiang, H.; Peng, J.; and Hong, Z. 2020. Spatiotemporal scenario generation of traffic flow based on lstm-gan. *IEEE Access*, 8: 186191–186198.

Xu, C.; Petiushko, A.; Zhao, D.; and Li, B. 2025a. Diffscene: Diffusion-based safety-critical scenario generation for autonomous vehicles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 8797–8805.

Xu, Z.; Li, B.; Gao, H.-a.; Gao, M.; Chen, Y.; Liu, M.; Yan, C.; Zhao, H.; Feng, S.; and Zhao, H. 2025b. Challenger: Affordable adversarial driving video generation. *arXiv preprint arXiv:2505.15880*.

Zhang, J.; Xu, C.; and Li, B. 2024. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15459–15469.

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

Zhang, L.; Rao, A.; and Agrawala, M. 2025. Scaling in-the-wild training for diffusion-based illumination harmonization and editing by imposing consistent light transport. In *The Thirteenth International Conference on Learning Representations*.

Zhang, S.; Tian, J.; Zhu, Z.; Huang, S.; Yang, J.; and Zhang, W. 2025. Drivegen: Towards infinite diverse traffic scenarios with large models. *arXiv preprint arXiv:2503.05808*.

Zhao, J.; Shi, J.; and Zhuo, L. 2024. BEV perception for autonomous driving: State of the art and future perspectives. *Expert Systems with Applications*, 258: 125103.