# Adversarial Stress Test for Autonomous Vehicle via Series Reinforcement Learning Tasks with Reward Shaping

Xuan Cai , Xuesong Bai , Zhiyong Cui , Peng Hang , Haiyang Yu , and Yilong Ren , *Member, IEEE*

***Abstract*—Testing is a pivotal phase for uncovering potential vulnerabilities in autonomous vehicles (AVs) to develop a secure autonomy system. However, existing methods often lack consideration for efficiently exploring multiple vulnerability-revealing cases, particularly under adversarial game scenarios. We introduce an evolving series reinforcement learning (RL) framework for adversarial policy training, integrating Responsibility Sensitive Safety (RSS) and Dynamic Time Warping (DTW) theories to shape the reward function to steer the evolving direction of the subsequent series agents for exploring vulnerability-revealing attack scenarios uncharted in the refined buffered repository. Our method undertakes adversarial stress tests for both black-box and white-box AV systems under test in driving tasks that engage in games with traffic vehicles and pedestrians. The results indicate that our approach expedites the exploration of additional scenarios blamed for the AV, outperforming the baselines in the vulnerability-revealing accident and scenario diversity. Furthermore, the causality of the collisions is qualitatively analyzed to provide insights for AV system vulnerability repair. Code is available at https://github.com/caixxuan/AST-SRL.**

***Index Terms*—Autonomous vehicle, adversarial stress test, series reinforcement learning, reward shaping.**

## I. Introduction

AUTONOMOUS driving holds promising prospects for reducing traffic accidents and enhancing travel efficiency [1], primarily because AVs can drastically curtail driving risks associated with human distraction, inexperience, and operational errors, courtesy of their computer systems' superior real-time performance and hard coding [2][3]. However, the comprehensive driving ability of AVs at the current stage is generally deemed inferior to that of proficient human drivers. This deficit undermines public trust in AVs and hampers their widespread application, largely due to safety concerns[4][5][6].

Testing serves as an integral approach in the development of AVs, designed to reveal vulnerabilities and faults in both software and hardware systems, thereby guiding their repair or upgrades. This process aims to mitigate driving risks to the greatest extent possible before commercial applications [7][8]. At present, prevailing testing methods can be categorized into two main types:

*Data-Driven Test*. The Naturalistic Field Operational Test (N-FOT) [9] has emerged as a predominant data-driven solution within the industrial circles. Nevertheless, relying solely on the relatively limited sensor data garnered by N-FOT, such as that from nuScenses [10] and WAYMO [11], is insufficient in fully addressing safety assurance concerns, regardless of whether the data spans thousands of kilometers [12]. It can be attributed to two main reasons. Firstly, the constraints of the database sizes and delivery deadlines pose challenges. The presence of a long-tail effect means that it is difficult to extract sparse safety-critical corner cases from a vast quantity of natural driving data. Consequently, it results in a large number of repetitive non-vulnerability-revealing test fragments, which inefficiently evaluate and hinder the development process. Secondly, the lack of interactivity between the system under test (SUT) and the traffic participants limits the utility of N-FOT [13][14].

*Predefined Generative Test*. In order to automatically evolving more perilous scenarios, the generative test generates a potentially infinite number of traffic game scenarios for closed-loop testing, using seed datasets or prior knowledge as the foundation [15]. Within these scenarios, the traffic participants function as non-player characters (NPCs), which are managed by predefined driver models [16]. These models are designed to produce corner cases that pose serious threats to the safety of autonomous vehicles, thus significantly mitigating the limitations of the data-driven test.

However, some limitations still plague the generative test method:

1) Predefined rules governing NPCs frequently fall short in generating menacing attack scenarios. It inadequacy potentially masks underlying defects, particularly in the case of high-level AVs, where conventional threat models may fail to uncover significant vulnerabilities. For instance, a rule-based system might inadequately simulate the erratic behavior of an

This work was supported by National Key Research and Development Project of China under Grant 2022YFB4300400. *(Corresponding author: Zhiyong Cui and Yilong Ren.)*

Xuan Cai is with the Research Institute for Frontier Science, Beihang University, Beijing 100191, P.R.China, and also with the State Key Laboratory of Intelligent Transportation Systems, Beijing, 100191, P.R.China.

Xuesong Bai and Zhiyong Cui are with the State Key Laboratory of Intelligent Transportation Systems, School of Transportation Science and Technology, Beihang University, Beijing 100191, P.R.China. (e-mail: zhiyongc@buaa.edu.cn)

Peng Hang is with the Department of Traffic Engineering, Tongji University, Shanghai 201800, P.R.China.

Haiyang Yu and Yilong Ren are with the State Key Laboratory of Intelligent Transportation Systems, School of Transportation Science and Engineering, Beihang University, Beijing 100191, P.R.China, and Zhongguancun Laboratory, Beijing 100094, P.R.China. (e-mail: yilongren@buaa.edu.cn).

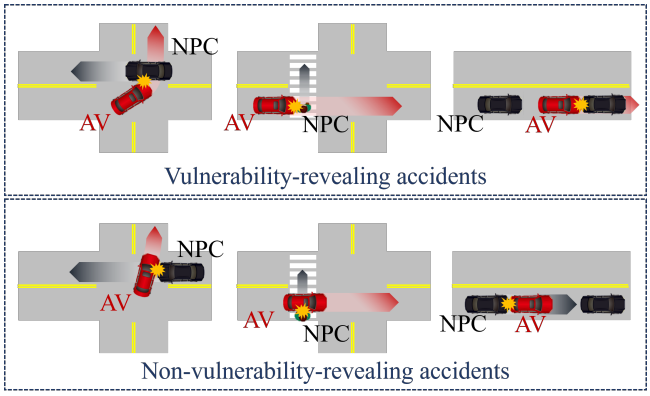



Fig. 1: Schematic diagram of accidents including vulnerability-revealing and non-vulnerability-revealing types.

attacking vehicle encroaching into the AV's lane unexpectedly, a scenario that requires more sophisticated modeling to accurately reflect real-world threats [17]. Such complexities suggest the need for advanced, dynamic control techniques that better capture the nuances of aggressive driving behaviors, further discussed as follows and supported by findings from Tuncali *et al*. [18].

2) Numerous test cases repetitively generated tend not to reveal vulnerabilities. While an abundance of corner cases can be extracted through generalization algorithms, the relevance of these scenarios for the rectification of AV systems is often overlooked. For instance, a scenario of an inevitable non-vulnerability-revealing accident involves a traffic NPC intentionally and rapidly colliding with the AV, where no viable solution was present within the allowed solution space, as depicted in the second row of Fig.1. To the best of our knowledge, this safety conundrum remains a formidable challenge for the existing AV technology to surmount. Consequently, such a test case may not bear significant relevance for revealing inherent vulnerabilities that could guide developers towards effective solutions.

3) The propensity to be entrapped in local optima often leads to the repetitive generation of analogous adversarial scenarios. Both optimization and learning-based methodologies may get ensnared in this prevalent issue, which consequently result in repeated exposure of similar vulnerabilities, thereby diminishing the efficiency of the test. From the developers' point of view, the higher the coverage of the test cases, the greater the possibility for enhancing the generability of AV systems.

Existing data-driven and predefined generative testing methods still face limitations in effectively exposing vulnerabilities across diverse scenarios and providing conclusive guidance for repairing AV systems. To address the aforementioned issues, we propose a series RL method as illustraed as Fig.2 that generates multiple adversarial testing scenarios successively. We consider the interaction between the AV and NPC as a Markov game problem, and subsequently train the NPC to actively launch intentional attacks on the AV while adhering to traffic rules, thereby generating adversarial scenarios that could potentially compromise the safety of the AV. We measure the appropriateness of AV operations and the dissimilarity of scenarios based on the RSS [19] and DTW [20] respectively, which helps to shape the reward function, to steer the evolving direction of the subsequent series agents for exploring vulnerability-revealing scenarios uncharted in the refined repository. The contributions of this paper can be encapsulated in three aspects as follows:

- We propose a continuously evolving and efficient series RL framework for conducting adversarial stress tests (ASTs). Our framework is specifically designed to generate a multitude of safety-critical corner cases rapidly where vulnerabilities with higher coverage are actively exposed.
- Our approach seamlessly integrates the principles of Responsibility Sensitive Safety and Dynamic Time Warping into the reward function. Additionally, we maintain a scenario repository as a prior knowledge of the adversarial agents in series to guide them towards unexplored areas, which serves to rapidly unearth a multitude of vulnerabilities inherent in AVs.
- Our proposed framework exhibits compatibility with both white-box and black-box AV systems, increasing practical applicability. The insights we have gleaned from the frame-by-frame qualitative analysis of discovered vulnerability-revealing collisions, shedding light on their root causes to inform targeted repair strategies for enhancing AV system robustness.

Overall, it expedites the testing process by rapidly uncovering a diverse set of vulnerabilities with higher coverage, ultimately improving the safety and robustness of autonomous driving systems. The rest of the paper is organized as follows. Section II provides a comprehensive review of autonomous driving testing methodologies currently prevalent in the academic sphere. Section III articulates the formulation of the AST via the application of the Markov Decision Process (MDP). Section IV details the adversarial policy of the NPC trained by the series RL. In Section V, the safety-critical game scenarios are simulated to validate the efficacy of the series RL method and further deduce the causality of vulnerability-revealing AV-blamed accidents. Section VI scrutinizes the potential threats to validity of the proposed testing methodology. Finally, we consolidate our conclusions and outline the scope for future research in Section VII.

## II. Related Works

### A. Data-Driven Test

The data-driven testing approach involves the direct replay of real-world driving scenarios, utilizing raw data gathered from various sources. Two fundamental methods employed in this process are data sampling and density estimation.

Data sampling evaluates the adaptability of AV through the replay of sensor data. Arief *et al.* [21] have expedited the AV assessment process by identifying safety-critical fragments within extensive collected data, thereby quantifying the risk intensities of the public road traffic system. To alleviate the testing burden of analogous scenarios, the unsupervised clustering method is utilized to categorize the test scenarios in [22] [23]. However, the sole reliance on simulation from collected data restricts the exploration of potential threat scenarios.

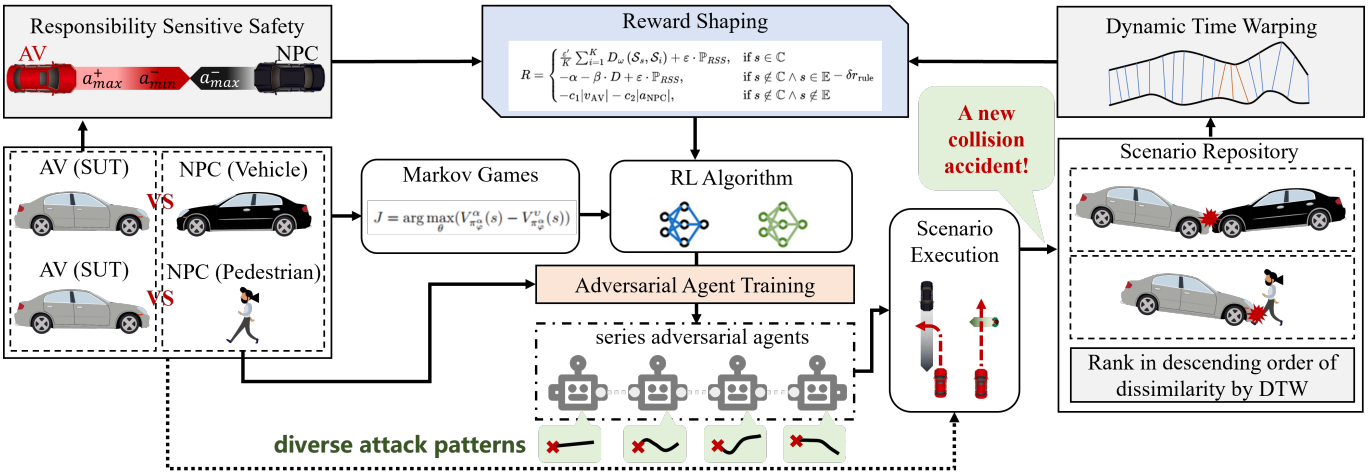



Fig. 2: Overview of the proposed method. Series RL integrating RSS and DTW is used to train the series adversarial agents' policy to generate multiple risk scenarios revealing vulnerabilities.

Waymo [24] has addressed this limitation by reconstructing fatal accident scenarios using counterfactual simulation and incorporating random disturbances to investigate similar scenarios.

Density estimation aims to mitigate the long-tail effect by learning the distribution of driving scenarios. The Gaussian process [25], a widely adopted non-parametric method, is utilized to model the scenario distribution. When combined with the Dirichlet process [26], which designates each observation to a specific scenario, this approach leverages non-parametric Bayesian learning to extract latent motion primitives for the simulation and reproduction of a multitude of traffic scenarios. Furthermore, Important sampling [27] represented by Bayesian networks, addresses the problem of repetitive testing by estimating high-risk driving scenarios through sampling.

While these method preserves the authenticity and credibility of scenarios, it necessitates a high degree of scenario diversity, unfortunately.

### B. Predefined Generative Test

The generative method, contingent upon the predefined and explicit knowledge of traffic and physical laws that govern the behavior of traffic participants like the Intelligent Driver Model (IDM) car-following model [28], overcomes the problem of extended generalization. It is achieved by deriving an infinite number of sub-scenarios from a limited number of parent-scenarios.

The formal simulation algorithm, utilized in conjunction with the formal specification of scenarios and safety attributes [29] or a priori formalized equations [30], generates random risk test cases. Kim *et al.* [31] employed a multi-objective optimization algorithm to mutate scenarios, thereby probing the vulnerability of AV sysems in simulation platform. Genetic algorithms [32][33] evaluate the risk level of scenarios based on specific risk quantification indicators, constantly evolving scenarios wherein the behavior of traffic participants is normal yet potentially jeopardizing to the safety of the ego vehicle. Nevertheless, these methods constrain the behavior of traffic participants within strict predefined rules, limiting their ability to solve long-tail problems. To further alleviate the stress of rule constraints, predefined knowledge only serves as a soft basis for learning-based neural networks in [34][35].

Nonetheless, the long-tail problem remains unresolved due to the prior constraints imposed on traffic participants. These constraints result in scenario singularity and an inherent issue of local convergence in optimization or learning methodologies.

### C. Adversarial Stress Test

AST is a promising way to generate safety-critical scenarios. It has the potential to construct numerous fatal accidents by controlling traffic participants to actively jeopardize the standard driving of AVs. Building upon the concept of the airborne adaptive stress test [36], a plethora of research findings on AST have been amassed [37][38]. Chen *et al.* [39] for instance, employed reinforcement learning methodologies to manage three agent vehicles, instigating them to deliberately collide with the AV during lane-changing tasks. However, they did not probe into the potential significance of post-clustering scenario remediation. Considering the human-like driving behavior of traffic flow [40], a proposal has been made for a reinforcement learning adversarial behavior generation method, which is based on the imitation of natural driving data. Sharif *et al.* [41] suggested a black-box testing framework, ReMAV, which models failure event probability-aware rewards to highlight areas where uncertain behavior might be present. Dense reinforcement learning [42] effectively and swiftly unveils potential vulnerabilities of AVs by eliminating non-safety critical states and reconnecting critical states through the editing of MDP.

Nevertheless, there is a dearth of studies that specifically target the identification of vulnerability-revealing scenarios in order to guide the development and repair of AVs. These studies encounter similar suboptimal issues as with optimization methods, which consequently restricts the breadth of scenarios covered. To address these two concerns and expedite the autonomous driving test process, this paper proposes a series RL framework predicated on AST.

## III. Test Formulation

In this section, AST is formulated as a MDP and a solution is proposed via the implementation of a deep RL methodology. Essentially, AST functions as a two-player Markov games, providing a descriptive framework for the MDP as well as viable application methods within the realm of autonomous driving testing.

### A. Deep Reinforcement Learning

Deep RL trains a deep network agent to solve sequential decision problems that can be described as a MDP which can be formalized as a quadruple-tuple $\langle S, A, P, R\rangle$. $S$ is the set of states that the agent can perceive; $A$ is the set of actions taken by the agent; $P : S \times A \to S$ is the transition function that describes the world model with the agent interacts, such as $P(s'|s, a)$ representing the probability of environment model transitioning to state $s' \in S$ after the agent takes action $a \in A$ at state $s \in S$. $R : S \times A \to \mathbb{R}$ defines the reward function $r = R(s, a)$, which is a real reward obtained after taking action $a$ at state $s$, to guide the agent to strive for a higher total return in each iterative episode.

- Policy: Policy, $\pi : S \to A$, refers to the strategy that an agent uses to decide an action $a$ at a given state. Therefore, a policy can be construed as a mapping from a state $s$ to an action $a$ which is determined by either a stochastic policy $a \sim \pi(a|s)$ or a deterministic one $a = \pi(s)$ once the agent perceives its environment.
- Reward Signal: The agent learns the optimal policy by obtaining immediate sense rewards through trail-and-error interactions with the environment, with the aim of maximizing the cumulative discounted return.
- Value function: Value function, $V_\pi(s)$, is the mathematical expectation of the cumulative return after performing a series of actions according to policy $\pi$ at state $s$. The optimal policy $\pi^*$ is derived by maximizing $V_\pi(s)$ through recursively solving the following equation according to the Bellman Equation:

$$V_\pi(s) = R\left(s, \pi\left(s\right)\right) + \gamma \sum_{s'} P\left(s' \mid s, \pi\left(s\right)\right) V_\pi(s') \quad (1)$$

where $\gamma \in (0, 1)$ is a discount factor determining the priority of short-term rewards, representing the degree to which an agent cares about future rewards. Identifying the optimal policy is tantamount to discovering the optimal value function [43]:

$$V_\pi(s) = \mathbb{E}_{\tau\sim\pi}[R(\tau_\pi)|s_0 = s] \quad (2)$$

where $\tau$ is the future discrete sequence $(\langle s_0, a_0, r_0\rangle, \langle s_1, a_1, r_1\rangle, ..., s_T)$ generated by the agent interacting with the environment subjecting to policy $\pi$. Eq.2 is to maximize $R(\tau_\pi) = \sum_{k=0}^{T} \gamma^k R_k(s_k, a_k)$ in MDP, where $T$ is the time horizon.

### B. Two-player Markov Games

Markov Games (MGs) model the decisions and strategies of agents as part of the MDP. In MGs, multiple agents perform a series of actions to maximize their collective or individual benefits. Particularly, two-player zero-sum MGs [44] involve a pair of agents with completely opposite interests. Although these two agents share an identical reward function, their objectives diverge: one (the victim) seeks to maximize the cumulative future rewards, whereas the other (the attacker) strives to minimize them [45][46].

In an effort to bolster generalizability, this study extends its scope to encompass the general scenario of non-zero-sum games [47]. The rationale behind this is that the paper does not focus on the training or design processes of the victim, and presumes its internal knowledge to be opaque, *i.e.*, the black-box assumption. The study's sole aim is to identify the victim's vulnerabilities through adversarial attack methodologies. Consequently, the training of adversarial strategies to maximize the expected reward $V_\pi^\alpha$ for the **attacker** $\alpha$ and to minimize the expected reward $V_\pi^\upsilon$ for the **victim** $\upsilon$ can efficaciously probe the victim's vulnerabilities, as follows:

$$J = \arg\max_\theta (V^\alpha_{\pi^\alpha_\varphi}(s) - V^\upsilon_{\pi^\alpha_\varphi}(s)) \quad (3)$$

where $\varphi$ is the network parameter of the adversarial strategy $\pi^\alpha_\varphi$ taken by the attacker. $V^\alpha_{\pi^\alpha_\varphi}$ and $V^\upsilon_{\pi^\alpha_\varphi}$ are the state value functions of the attacker and victim respectively. Note that under the black-box assumption and the fact that $\upsilon$ follows a fixed driving strategy, $\upsilon$ is considered as an integral part of the environment in the two-player MGs problem. Therefore, this approach addresses the absence of knowledge concerning $\upsilon$'s internal driving strategy, transforming it into a readily manageable single-agent MDP problem. Thus, the state transition function is solely dependent on $\alpha$. RL contemplates the interaction model between the agent and the environment, enabling the agent to adopt behaviors that maximize the revenue of Eq.3. Numerous RL algorithms have been devised to resolve optimal or sub-optimal solutions of MDP problems [43][48]. A recent empirical study [49] has demonstrated the widespread application of the Twin Delayed Deep Deterministic policy gradient (TD3) [50] across various industries. Consequently, this paper opts for TD3 as the preferred training method for adversarial strategies.

### C. Adversarial Stress Test for Mature Autonomous Vehicles

In this study, a mature AV, either well-trained for end-to-end or well-designed for modular systems, assumes the role of a victim. It aims to safely accomplish the pre-set driving task using a fixed strategy $\pi_\upsilon\left(s|a\right)$ typically referring to a specific learning or analytical driving strategy, in the face of the adversarial strategy $\pi_\alpha\left(s|a\right)$ deployed by the NPC $\alpha$, be it a vehicle or a pedestrian. In the MGs, the attacker optimizes the expected reward by learning $\pi^*_\alpha\left(s|a\right)$. The implication in this paper is that the NPC creates potentially hazardous scenarios via the optimal action sequence strategy, inducing the AV to collide.

Existing AV systems, regardless of being end-to-end or modular, commonly suffer from an overfitting problem [51] indicating their general ability to cover the test cases in the V-model development process [52] but their unreliability in dealing with intricate, unseen scenarios. As such, the verification

and validation of long-tail corner cases are crucial in testing the generalization capability of AV systems. AST is aimed at mature AV systems, generating a multitude of corner cases that are challenging to cover in the development process through extensive NPC attacks, thereby revealing their vulnerabilities.

## IV. Adversarial Policy Training

In this section, we introduce a series RL framework designed for the adversarial policy training of the NPC. The appropriateness of the AV's operations and the similarity of the generated scenarios are gauged using the RSS model and DTW algorithm, respectively, which are integrated into reward function. Lastly, we maintain a refined scenario repository, ranked by the DTW, to guide the directional exploration of the series adversarial agents.

### A. TD3 RL for Adversarial Training

The two-player MGs problem in this paper degenerates into an MDP, with its tuple $<S, A, P, R>$ described as follows:

- $S$: the fully observable state space that the NPC can obtain, including the state of the AV;
- $A$: $A_\alpha$, the action space of the NPC;
- $P$: $S \times A \to S$, the state transition probability matrix of the physical traffic world;
- $R$: $S \times A \to \mathbb{R}$, the immediate reward of the NPC.

In the field of RL, TD3 is an optimized version of the Deep Deterministic Policies Gradient (DDPG) [53] with an Actor-Critic structure that employs continuous control output. The agent can search using the gradient $\nabla_\varphi J(\varphi)$ provided by the value-function for the policy network $\pi_\varphi$ parameterized by $\varphi$ in DDPG. In the Actor-Critic framework, the policy function, also known as the Actor, updates the policy through a deterministic policy gradient algorithm:

$$\nabla_\varphi J(\varphi) = \mathbb{E}_{s\sim p_\pi}[\nabla_a Q_\pi(s,a)|_{a=\pi_\varphi(s)} \nabla_\varphi \pi_\varphi(s)] \quad (4)$$

where the action-value function $Q_\pi(s,a) = \mathbb{E}_{s\sim p_\pi, a\sim\pi}[R(\tau_\pi)|s,a]$ (Critic), represents the expected return obtained when following the policy $\pi$ to perform an action $a$ at the state $s$. In the Q-learning algorithm, $Q_\pi$ can be iteratively updated using the Bellman Equation, which relates the current state-action pair $(s,a)$ and the return at the next timestep $(s',a')$:

$$Q_\pi(s,a) = r + \gamma \mathbb{E}_{s',a'}\left[Q_\pi(s',a')\right], a' \sim \pi(s') \quad (5)$$

Critic is updated by minimizing the loss:

$$L(w) = \mathbb{E}_{s\sim\rho_\pi, a\sim\pi_\phi}[(Q_w(s,a) - Q_\pi(s,a))^2] \quad (6)$$

where $\rho$ symbolizes the state distribution. Experience replay buffer is used to assists the agent in adapting rapidly to complex and evolving task environments. Simultaneously, it effectively hinders overfitting and expedites the convergence process.

Expanding upon the foundation of DDPG, TD3 evolves into a structure comprising six networks. Within the Actor, online policy network, target policy network, and two Critics each encompassing the online Q network and target Q network are included. Moreover, TD3 has also incorporated the following enhancements:

**Algorithm 1** Single Adversarial TD3

1: **Initialize** Actor $(\pi_\varphi, \pi_{\varphi'})$, Critic $(Q_{\theta_1}, Q_{\theta_2}, Q_{\theta_1'}, Q_{\theta_2'})$ and Experience Replay Buffer $\mathcal{D}$
2: **for** episode = 1:$episode_{max}$ **do**:
3: **while** $s \notin \mathbb{C} \wedge s \notin \mathbb{E}$ **do**:
4: $(r, s') \leftarrow ENV(s, a)$
5: $\theta_i \leftarrow \text{argmin}_{\theta_i}\ N^{-1}\sum\left(y - Q_{\theta_i}(s,a)\right)^2$
6: **end while**
7: **if** episode % $N_{int}$==0 **do**:
8: $\nabla_\phi J(\phi) = N^{-1}\sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_\varphi(s)} \nabla_\varphi \pi_\varphi(s)$
9: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
10: **end if**
11: **end for**

*1) Clipped Double-Q learning:* In an effort to avoid overestimation of the Q-table, TD3 strategically employs the smaller value of the two target Q's during the computation:

$$y = r + \gamma \cdot \min_{i=1,2} Q_{\theta_i'}(s', \pi_\phi(s')) \quad (7)$$

where $\theta$ denotes the network parameters of the estimated function $Q(\cdot)$, and $\cdot'$ symbolizes the current network utilized for updating parameters.

*2) Delayed Policy Updates:* The policy network's updating frequency should be maintained at a less rate than the value network to ensure that the estimation error diminishes prior to the policy update. Alongside the reduction in update frequency, soft updates should also be employed for target networks:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \quad (8)$$

*3) Target Policy Smoothing Regulation:* In order to attain analogous Q values for similar actions, a regularization policy is implemented to streamline the target policy. This equation estimates the action value via a bootstrapping technique, which possesses the benefit of smoothing the estimation. In practical applications, we can introduce a minimal variance noise to the target policy and compute the average of the mini-batch updates for the expected action value:

$$\begin{aligned} y &= r + \gamma Q_{\theta'}(s', \pi_{\phi'}(s') + \epsilon) \\ \epsilon &\sim \text{clip}\left(\mathcal{N}(0,\sigma), -c, c\right) \end{aligned} \quad (9)$$

The pseudocode for the series adversarial TD3 RL is presented in Algorithm 1. The specific methodology of training has been extensively elaborated in this section as above. It is important note that $\mathbb{C}$ and $\mathbb{E}$ (line 3) which both serve as criteria for determining simulation, represent the state $s$ steps to collision moment and termination frame once time out or AV finish driving task respectively. $N_{int}$ in line 7 represents the interval for updating the policy network.

### B. Measurement of Operational Appropriateness and Scenario Dissimilarity

The RSS model and the DTW method are collaboratively employed to gauge the appropriateness of the AV operation

and the degree of dissimilarity between adversarial scenarios. It serves to facilitate the formulation of the reward function and the organization of the scenario repository.

*1) Responsibility Sensitive Safety-Critical Identification:* Formal verification offers a standardized validation of abstract mathematical models for a system. In essence, if the system's performance aligns with the designed mathematical model, it is deemed safe and reliable. The Mobileye RSS model [19] serves as a prime example of this approach. The RSS model establishes five fundamental principles that AVs must comply with. These principles are informed by prevalent traffic regulations and conventional driving wisdom. The model also formally stipulates appropriate responses for AVs in hazardous situations, asserting that adherence to the RSS model by all vehicles will result in an accident-free environment:

- Rule 1: Keep a safe longitudinal distance (*Safety Distance*);
- Rule 2: Keep a safe lateral distance (*Cutting in*);
- Rule 3: Don't fight for the right of way (*Right of Way*);
- Rule 4: Be aware of situations where your vision is obstructed (*Limited Visibility*);
- Rule 5: If a collision can be avoided, make every effort to do so, even if it means breaking Rules 1-4. However, do not cause a new collision (*Avoid Crashes*) [54].

It should be noted that Rules 1-4 have higher priority on account of the conservative nature of the estimations involved. But we neglect the Rule 4 due to the fully observable conditions. In this study, we employ the RSS model, inspired by common-sense rules, to assess the operational appropriateness of an AV actions at each instant, thereby determining if a system failure has occurred. This paper concentrates on assessing the safe distance (Rules 1 and 2) while the right of the way (Rule 3) contributes to the posterior evaluation of accident responsibility to eliminate the concern of the irresponsible entity regarding RSS-critical issues.

To facilitate analysis, the RSS model decouples the evaluation of safe responses into two dimensions - longitudinal and lateral. The safe distance, which ought to be predetermined, serves as the benchmark for appraising the adequacy of responses. In the longitudinal direction, if both the AV and NPC are moving in the same direction, the minimum safe distance for the AV can be deduced as follows:

$$d_{safe}^{lon} = \left[ v_{AV}\rho + \frac{1}{2}a_{max}^{lon,+}\rho^2 + \frac{\left(v_{AV} + \rho a_{max}^{lon,+}\right)^2}{2a_{min}^{lon,-}} - \frac{v_{NPC}^2}{2a_{max}^{lon,-}} \right]_+ \tag{10}$$

where $[\cdot]_+ = max\{\cdot, 0\}$, $v_{AV}$ and $v_{NPC}$ are the longitudinal velocity of the AV and NPC; $\rho$ is the response time at maximum braking of the AV; $a_{max}^{lon,+}$ and $a_{max}^{lon,-}$ are the maximum throttle and braking acceleration. $a_{min}^{lon,-}$ symbolizes the minimum braking deceleration that ensures complete stop without a collision after accelerating at $a_{max}^{lon,+}$ within the response time. $d$ represents the longitudinal distance between the AV and NPC. If both are moving in opposite directions, the velocity vector $\mathbf{v}_{NPC}$ of $c_{NPC}$ is negative. The minimum safe longitudinal distance is defined as:

$$\begin{aligned} d_{safe}^{lon} =& \frac{v_{AV} + v_{AV,\rho}}{2}\rho + \frac{v_{AV,\rho}^2}{2a_{min,AV}^{lon,-}} + \\ & \frac{|v_{NPC}| + v_{NPC,\rho}}{2}\rho + \frac{v_{NPC,\rho}^2}{2a_{min,NPC}^{lon,-}} \end{aligned} \tag{11}$$

where $v_{AV,\rho} = v_{AV} + \rho a_{max,AV}^{lon,+}$, $v_{NPC,\rho} = |v_{NPC}| + \rho a_{max,NPC}^{lon,+}$. If $d$ falls below the threshold $d_{safe}^{lon}$, it necessitates an evaluation of the AV's longitudinal response due to the present scenario is deemed longitudinally dangerous. The longitudinal proper response should satisfy the conditions proposed in the [19].

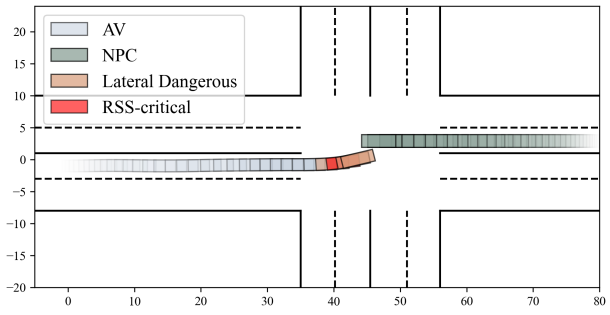
A similar methodology can be employed in the lateral direction, the minimum lateral safe distance can then be defined as:

$$d_{safe}^{lat} = \left[ \frac{v_{AV} + v_{AV,\rho}}{2}\rho + \frac{v_{AV,\rho}^2}{2a_{min}^{-}} - \frac{v_{NPC} + v_{NPC,\rho}}{2}\rho + \frac{v_{NPC,\rho}^2}{2a_{min}^{-}} \right]_+ \tag{12}$$

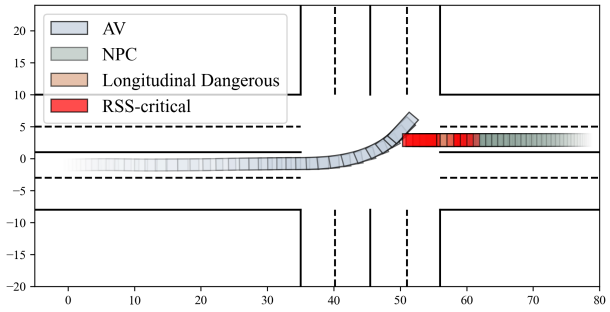
where all the speed and acceleration vectors are transformed into the lateral direction. In a similar vein, if $d$ falls below $d_{safe}^{lat}$, it becomes essential to assess the aptness of AV's lateral response action, as the current scenario is deemed to be laterally dangerous. The lateral proper response should also fulfill the conditions listed in [19].

Employing the RSS method enables us to discern the response of each discrete state within the AV driving trajectory, effectively differentiating between proper and improper reactions. An improper response highlights the shortcomings of the AV system when confronted with the present scenario. This insight is beneficial for subsequent reward shaping, facilitating the discovery of more vulnerability-exposing fault scenarios. In this article, the state when the vehicle enters the minimum safe distance is termed the dangerous state, and the improper response is referred to as the RSS-critical frame.

As illustrated in Fig.3, dangerous states (depicted in orange) and RSS-critical frames (depicted in red) are discerned following a collision accident that transpired while the AV was executing an unprotected left turn, supervised by a specific autonomous driving algorithm. In scenario (a), the AV detectes a lateral dangerous state in the final four frames preceding the collision, and responds appropriately with braking to decelerate (shown in orange), with the exception of the third to last frame which is marked in red. Nonetheless, the collision could not be averted. In the scenario (b), the AV was sidestruck by the NPC while in the midst of its left turn task. The NPC neglects to brake appropriately when it is identified by the RSS as being a dangerous longitudinal state in the final several frames, hence bearing primary responsibility for the accident. For the AV developers, accidents akin to scenario (b) are frequently difficult to circumvent as the responsibility is assigned to the NPC's erroneous operation, rendering effective feedback on AV vulnerability rectification inconspicuous. In contrast, in scenario (a), the NPC has an absolute right of way, and the accident is precipitated by a defect in the AV




(a) Vulnerability-revealing collision accident blamed for the AV.




(b) Passive impact accident blamed for the NPC.

Fig. 3: Collision accidents under the unprotected left turn driving task blamed for the opposite entities. The vehicle state is represented by the rectangles of different colors, the orange is refered to dangerous states and the red is refered to RSS-critical frame.

system itself. It mirrors the vulnerabilities of the AV and is significantly beneficial for its system's repair. In essence, augmenting the frequency of RSS-critical frames can compel the blamed entity to violate traffic laws. Consequently, scenario (a) is precisely the one we need to generate in large quantities, while striving to prevent the occurrence of scenario (b).

*2) Dynamic Time Warping Scenario Similarity Measurement:* To facilitate the generation of a broader spectrum of adversarial scenarios, it is crucial to undertake a comprehensive exploration of the NPC behavior. It will help identify potential vulnerabilities in the AV. Generally, scenarios are characterized based on the position and velocity trajectories of adversarial NPCs. However, it should be noted that the lengths of pairwise spatiotemporal trajectories may not always be identical. Consequently, the DTW [20] method is employed to manage the diversity measurement of multi-dimensional spatiotemporal trajectories of unequal lengths.

Let's assume that there exist two time series, $\mathcal{M} = q_1, q_2, ..., q_m$ and $\mathcal{N} = p_1, p_2, ..., p_n$ with lengths $m$ and $n$, respectively. The elements $q$ and $p$ share the same dimensions, which include $\langle x, y, v\rangle$ in this paper, where $x$, $y$ are the Cartesian coordinates, and $v$ denotes the velocity of the vehicle. By utilizing $\mathcal{M}$ and $\mathcal{N}$ as the horizontal and vertical coordinates respectively, we proceed to construct a warping matrix:

$$D = \begin{bmatrix} d(q_1,p_1) & d(q_1,p_2) & \cdots & d(q_1,p_n) \\ d(q_2,p_1) & d(q_2,p_2) & \cdots & d(q_2,p_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(q_m,p_1) & d(q_m,p_2) & \cdots & d(q_m,p_n) \end{bmatrix} \tag{13}$$

where the element $d(q_i, p_j)$ represents the dissimilarity measure between the discrete nodes $q_i$ and $p_j$, which can be directly measured using Euclidean distance:

$$d(q_i, p_j) = \|q_i - p_j\|_2 \tag{14}$$

The warping path is a sequence of continuous matrix elements within the distance matrix $D$ which exists between two distinct time series. This path illustrates the dissimilarity relationship between the time series: $W = \omega_1, \omega_2, ..., \omega_K$, and the $k$th element of $W$, represented as $\omega_k = (i, j)_k$, delineates the mapping between the sequences $\mathcal{M}$ and $\mathcal{N}$. The path $W$ must conform to certain constraints, including boundary conditions, continuity, and monotonicity. While numerous paths could potentially satisfy these stipulations, our interest lies in the path that minimizes the ensuing warping cost:

$$D_\omega(\mathcal{M}, \mathcal{N}) = min \frac{\sqrt{\sum_{k=1}^{K} \omega_k}}{K} \tag{15}$$

where the denominator $K$ is primarily serves to adjust for the regularization paths of varying lengths. DTW manipulates two time series by extending and shortening them to derive a warping path with the minimum distance, thereby maximizing the similarity between the two time series. Within this framework, our goal is to identify a path that ultimately minimizes the total distance. To ascertain the cumulative distance $\gamma(i, j)$, which acts as a metric for the similarity between the two sequences, DTW employs the principle of dynamic programming:

$$\gamma(i,j) = d(q_i, p_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{16}$$

where the cumulative distance $\gamma(i, j)$ is the distance at the current grid point $d(q_i, p_j)$, which is the Euclidean distance (or dissimilarity) between points $q_i$ and $p_j$ in addition to the cumulative distance from the smallest neighboring element that can reach this point.

### C. Reward Shaping

*1) Baseline Reward:* Two additional motivations for relaxing the zero-sum condition of MGs in this paper are the constraint of NPC behaviour and the enhancement of attack scenario diversity. A reward function designed strictly in accordance with the zero-sum condition may prompt the NPC to collide with the AV in the shortest possible path. It not only diminishes the diversity of collision scenarios, but the likelihood of such instances in the traffic environment is also relatively low, given that NPC behaviour is somewhat regulated by traffic rules. Most of the existing literature [37][38] merely identifies scenarios involving collisions into the reward function:

$$R = \begin{cases} 100, & \text{if } s \in \mathbb{C} \\ -\alpha - \beta \cdot D, & \text{if } s \notin \mathbb{C} \wedge s \in \mathbb{E} \\ -c_1|v_{\text{AV}}| - c_2|a_{\text{NPC}}|, & \text{if } s \notin \mathbb{C} \wedge s \notin \mathbb{E} \end{cases} - \delta r_{\text{rule}} \tag{17}$$

where the rewards for NPC are categorized into three stages. For the stage where $s \notin \mathbb{C} \wedge s \notin \mathbb{E}$: the attack task is ongoing, neither a collision occurring nor the termination frame being reached. $c_1$ and $c_2$ are hyperparameters. $c_1$ is designed to

encourage the NPC to interact with the AV to decelerate its speed, while $c_2$ penalizes NPC for abrupt acceleration, aiming to adhere to regular driving habits as closely as possible. In the stage where $s \notin \mathbb{C} \wedge s \in \mathbb{E}$: the last frame due to timeout or the completion of a task by the AV. In this case, $\alpha$ and $\beta$ are hyperparameters, $D$ is the distance between the AV and the NPC in the concluding frame. The aim is to coax the NPC into reducing the distance to the AV as much as possible, thereby provoking a collision. For the stage where $s \in \mathbb{C}$: the final frame due to a collision, indicating a successful attack launched by the NPC and resulting in a collision. As such, a reward of +100 is given to stimulate such behavior. $\delta$ determines the traffic compliance of the NPC.

*2) RSS Shaping:* However, collision accidents identified through the reward function of Eq.17 may not necessarily provide substantial feedback for rectifying the vulnerabilities of the AV, as depicted in Fig.3b. Therefore, designing a reward function predicated on the RSS model to induce a swifter generation of AV-blamed collisions revealing vulnerabilities is a potential solution. Assuming that the RSS judgment sequence of SUT is $S = \{x_1, x_2, ..., x_\tau\}$, where the RSS critical condition is met, *i.e.*:

$$P(x) = \{x \in (d^{lon} < d^{lon}_{safe} \vee d^{lat} < d^{lat}_{safe})] \tag{18}$$

So the set of elements is $S_p = \{x \in S | P(x)\}$. Let $\mathbb{P}_{RSS}$ denote as the proportion of RSS-critical frames in $\tau$:

$$\mathbb{P}_{RSS} = \frac{|S_P|}{|S|} \tag{19}$$

Compared to Eq.17, only two components require modification:

$$R = \begin{cases} 100 + \varepsilon \cdot \mathbb{P}_{RSS}, & \text{if } s \in \mathbb{C} \\ -\alpha - \beta \cdot D + \varepsilon \cdot \mathbb{P}_{RSS}, & \text{if } s \notin \mathbb{C} \wedge s \in \mathbb{E} \\ -c_1|v_{\text{AV}}| - c_2|a_{\text{NPC}}|, & \text{if } s \notin \mathbb{C} \wedge s \notin \mathbb{E} \end{cases} - \delta r_{\text{rule}} \tag{20}$$

where the hyperparameter $\varepsilon$-related term $\varepsilon \cdot \mathbb{P}_{RSS}$ is incorporated into the first and second terms to with the intent to foster the generation of accidents featuring a higher proportion of RSS-critical frames, as there is a robust correlation between RSS-critical and the revelation of vulnerabilities in AV-blamed collision accidents (see Fig.3).

*3) DTW Shaping:* Conversely, a plethora of attack scenarios can elucidate a broader range of vulnerabilities inherent in AV. Capitalizing on the DTW method, renowned for measuring the dissimilarity between two temporal sequences, we integrate it into the first term of the reward function as follows:

$$R = \begin{cases} \frac{\varepsilon'}{K} \sum_{i=1}^{K} D_\omega(\mathcal{S}_s, \mathcal{S}_i) + \varepsilon \cdot \mathbb{P}_{RSS}, & \text{if } s \in \mathbb{C} \\ -\alpha - \beta \cdot D + \varepsilon \cdot \mathbb{P}_{RSS}, & \text{if } s \notin \mathbb{C} \wedge s \in \mathbb{E} \\ -c_1|v_{\text{AV}}| - c_2|a_{\text{NPC}}|, & \text{if } s \notin \mathbb{C} \wedge s \notin \mathbb{E} \end{cases} - \delta r_{\text{rule}} \tag{21}$$

where the hyperparameter $\varepsilon'$ determines the significance of the first richness reward term, with $K$ representing the maximum allowable scenario number in the scenario repository. $\mathcal{S}_s$ and $\mathcal{S}_i$ represent the freshly generated scenario and the $i$th scenario in the repository, respectively. The inclusion of the term $\frac{\varepsilon'}{K} \sum_{i=1}^{K} D_\omega(\mathcal{S}_s, \mathcal{S}_i)$ serves to guide subsequent agents towards the generation of attack scenarios that are novel. This term implies a comparison of $\mathcal{S}_s$ with each scenario in the repository individually. A larger average distance signifies a more novel, recently discovered adversarial scenario $\mathcal{S}_s$, which in turn leads to a higher reward.

### D. Series Reinforcement Learning Framework

The series RL framework which trains a series agents to launch attacks in a multitude of styles to reveal the vulnerabilities of the AV, as depicted in Fig.4, is structured based on the preceding research content and can be summarized in the following steps:

- Deployment of well-trained/designed AV: Allocate vehicles equipped with advanced autonomous driving algorithms to execute driving assignments. Subsequently, ascertain the global route of the AV to establish the NPC's initial position. Finally, the AV independently accomplishes the predetermined driving task;
- Training of the subsequent TD3-based adversarial agent: This stage involves the training of a TD3-based adversarial agent in a simulator (Section IV.A) and integrates both the RSS and DTW (Section IV.B) into reward shaping (Section IV.C), thereby generating a rich set of vulnerability-revealing AV-blamed collisions;
- Incorporation of generated adversarial scenarios into the scenario repository: This procedure entails three steps - *Append*, *Sort*, and *Pop*. The *Append* step involves adding the adversarial scenarios attacked by the well-trained TD3 to the repository. The *Sort* step measures the dissimilarity of each scenario with all the others based on DTW, and arranges them in descending order of cumulative dissimilarity. The *Pop* step removes the scenario with the least dissimilarity if the maximum allowable value $K$ is exceeded to ensure the repository's richness and inform the next agent of the refined buffered scenarios;
- Iteration of the above steps until the maximum TD-3 agent count limit is reached.

## V. Experiments and Results

In this section, the ASTs using the high-fidelity Carla simulator [55] are conducted to validate the efficiency of the proposed series RL framework.

### A. Experiment Setups

We employ the Python API interface of Carla to compile all programs relevant to the series RL. We select two game scenarios for the research, namely *unprotected left turn* (#ULT) and *pedestrian cross walk* (#PCW) are illustrated in Fig.5. These tasks respectively address situations where the AV at the exit faces a green signal conflicting with straight ahead traffic and a pedestrian crossing signal. #ULT and #PCW represent considerable challenges for the existing AV system due to their long-standing high accident rates. This is largely attributed to the heterogeneous traffic environment at intersections and the intricate process of the game. The AV will be governed either by an end-to-end system or a modular system, which will be

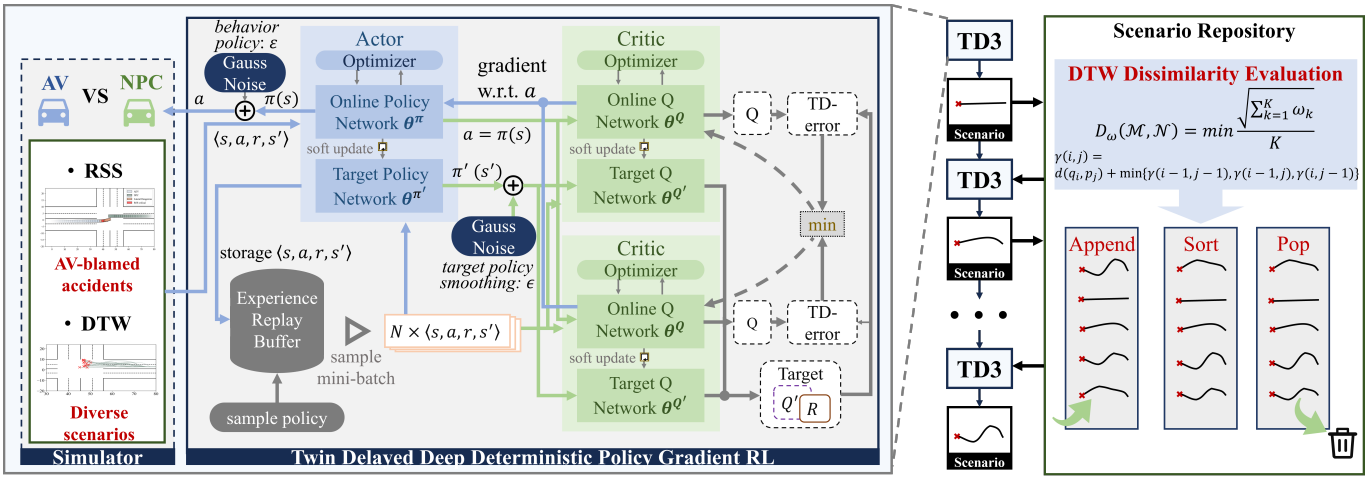



Fig. 4: Overall framework of the series RL for adversarial policy training. The NPC is trained by the TD3 integrating the RSS and DTW to attack the AV to reveal its multiple vulnerabilities. The scenario repository is perpetually curated and ranked in descending order of dissimilarity by the DTW, ensuring that the subsequent series of agents are well-informed about the existing scenarios while simultaneously incentivizing them to venture into uncharted territories by attempting novel attack scenarios that have not yet been encountered in previous simulations.

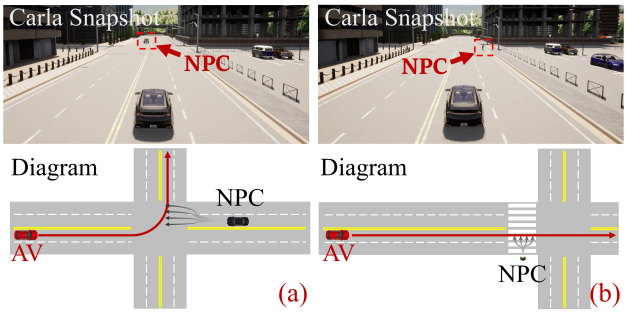



Fig. 5: Snapshots of Carla simulation and driving task diagrams including (a) unprotected left turn (#ULT) and (b) pedestrian cross walk (#PCW).

TABLE I: Settings of the hyperparameters of TD3.

| Parameter | Value |
|---|---|
| discount factor $\gamma$ | 0.99 |
| actor learning rate | 0.008 |
| critic learning rate | 0.01 |
| soft update rate | 0.01 |
| batch size | 128 |
| buffer size | 100000 |
| exploration noise std | 0.1 |
| policy noise std | 0.4 |
| delay update frequency | 2 |
| **(Reward function related)** | |
| $\alpha$ | 100 |
| $\beta$ | 1 |
| $\varepsilon$ | 100 |
| $\varepsilon\prime$ | 0.1 |
| $c_1$ | 0.01 |
| $c_2$ | 0.01 |
| $\delta$ | 10 |

elaborated upon in the following. Meanwhile, a TD3-based adversarial agent controls the NPC.

The state space of the MDP is an 8-dimensional vector space: $[x_{AV}, y_{AV}, yaw_{AV}, v_{AV}, x_{NPC}, y_{NPC}, yaw_{NPC}, v_{NPC}]$, where $x, y, yaw$ and $v$ respectively represent the horizontal coordinate, vertical coordinate, yaw angle, and absolute speed of the AV or NPC as specified by the subscript. This paper presumes that the state information is fully observable, thanks to vehicle-to-vehicle information sharing technology [56]. The action space of the NPC is a 2-dimension vector space, but there are discrepancies in the control methods employed by vehicles and pedestrians in Carla. For vehicles, the control action is denoted by $[\phi_{NPC}, \delta_{NPC}]$, where $\phi_{NPC} \in [-1, 1]$ symbolizes the degree of the vehicle's throttle pedal. If it is greater or less than zero, it represents throttle or braking respectively, and the absolute value signifies the degree of the pedal. For pedestrians, the control action is described as $[v_{NPC}, yaw_{NPC}]$.

The training hyperparameters for TD3 are established as illustrated in Tab.I with reference to [37][57]. Within TD3, the Actor is a fully connected neural network comprising 3 layers, with the number of hidden units sequentially arranged as $[128, 64, 2]$. The activation function for the first two layers is ReLU, and for the last two layers is Tanh, and outputs $[a_1, a_2]^T \in [-1, 1]$ which will then be mapped to a predefined control range. A linear mapping method is applied: for vehicles, it will be a replicative linear mapping, $[\phi_{NPC}, \delta_{NPC}]^T \leftarrow [a_1, a_2]^T$; for pedestrians, $[v_{NPC}, yaw_{NPC}]^T \leftarrow [v_{max}/2, yaw_{max}/2]^T \circ [a_1, a_2]^T + [v_{max}/2, yaw_{max}/2]^T$, where the symbol $\circ$ represents Hadamard product [58], and the maximum velocity is $v_{max} = 2.5\ mps$, the maximum yaw angle is $yaw_{max} = 360°$. The Critic in TD3 is a 4-layer fully connected neural network, with the number of hidden layers being [128, 64, 32, 1] respectively. The series RL trains 100 agents in sequence, with the scenario repository capable of storing a maximum of $K = 15$ scenarios.

In order to enhance the credibility of the system's appropriateness judgement, the parameters of the RSS should be

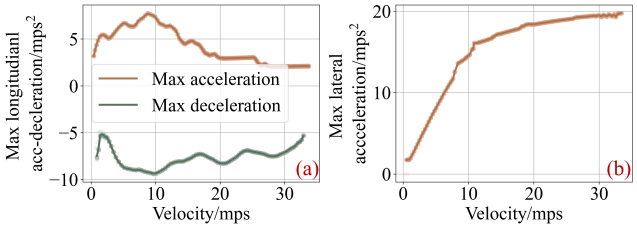



Fig. 6: Calibrations of (a) the max longitudinal acc-deceleration and (b) the max lateral acceleration for Carla Lincoln MKZ-2017.

TABLE II: Settings of the RSS ($g = 9.8m/s^2$, response time $\rho = 0.1s$).

| | $a^{+}_{max}$ | $a^{-}_{max}$ | $a^{-}_{min}$ |
|---|---|---|---|
| Longitudinal | Interpolate. | Interpolate. | $0.4g$ |
| Lateral | Interpolate. | / | $0.08g$ |

distinctly defined. Certain studies model the RSS of AV using estimated constants [57], which may not necessarily correspond with the actual performance dynamics of the vehicle. Hence, this study employs dynamic parameters to establish the RSS. Fig.6 depicts a one-dimensional calibration table of the maximum longitudinal acceleration $a^{lon,+}_{max}$, deceleration $a^{lon,-}_{max}$ and maximum lateral acceleration $a^{lat,+}_{max}$ of the Carla Lincoln MKZ-2017 (SUT) relative to the vehicle speed, calibrated with maximum throttle, brake opening, and steering wheel angle respectively. The acceleration performance of the vehicle is subject to various factors such as the maximum torque curve of the vehicle's power system, the friction force of the transmission shaft, and the ground adhesion conditions, but developed in Carla simulator, with its superior dynamic modeling capabilities, is highly credible. The values of $a^{lon,+}_{max}$, $a^{lon,-}_{max}$, $a^{lat,+}_{max}$ are obtained through interpolation, while $a^{lon,-}_{min}$ and $a^{lat,-}_{min}$ are set as conservative parameters referring to [19], as shown in Tab.II.

### B. System under Tests

The AST method proposed in this study is universally applicable to any test subject, irrespective of its level of algorithm transparency, thereby eliminating the need for prior knowledge of its system development details. Consequently, Carla BehaviorAgent [55] and Interfuser [59], two exemplary autonomous driving systems, representing white-box and black-box systems respectively, are selected as the SUT:

- Carla BehaviorAgent: Carla BehaviorAgent is a typical interpretable white-box autonomous driving model that uses a modular method, which is Carla's built-in robust control method. Note that BehaviorAgent skips the development of the perception system and directly obtains accurate information about the surrounding traffic environment, so the vulnerabilities found during the testing process point entirely to the planning and control system.
- Interfuser[1]: Interfuser (Interpretable Sensor Fusion Transformer) is a black-box end-to-end strategy that uses Transformers for multi-sensor fusion and uses interpretable features to increase the safety of autonomous driving. Interfuser uses four cameras and a LIDAR sensor data as inputs, and outputs a series of waypoints with velocity information as the reference planning trajectory. Finally, a PID (Proportional-Integral-Derivation) controller is used to control the throttle and brake pedals to track the waypoints.

[1]As of the submission date, Interfuser is ranked 2nd in the Carla Leaderboard.

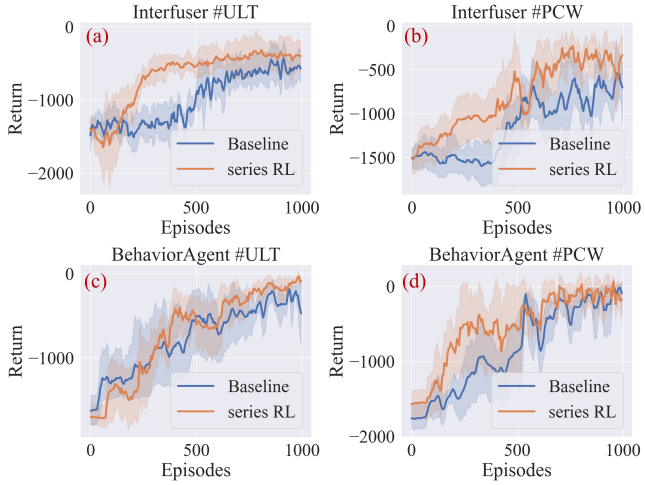



Fig. 7: Episodic return of RL for the Interfuser under (a) #ULT and (b) #PCW and the BehaviorAgent under (c) #ULT and (d) #PCW.

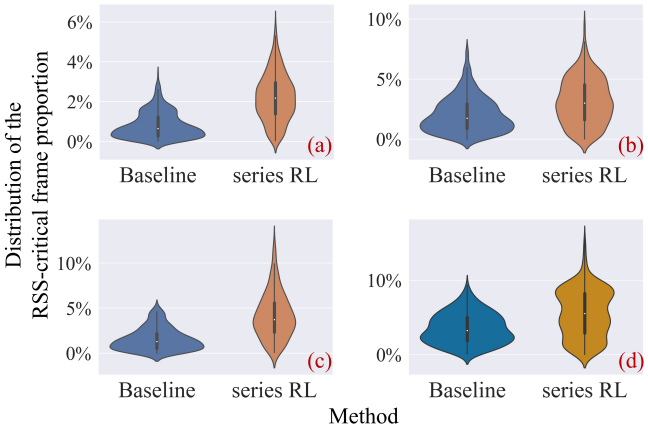



Fig. 8: Distribution of RSS-critical frame proportion for Interfuser under (a) #ULT and (b) #PCW and BehaviorAgent under (c) #ULT and (d) #PCW among all scenarios.

In the #ULT scenario, the SUT needs to complete a left turn task. In the #PCW scenario, the SUT needs to complete a straight driving task. The SUT is considered successful only if it completes the driving task within the specified time without any collisions or traffic violations.

### C. Adversarial Stress Test Result

In this paper, AdvSim [60], Learning-to-Collide (LC) [61] and the TD3 employing the reward function denoted as Eq.17 (Baseline) are selected as the baseline strategies for the generation of an equivalent number of the proposed series TD3-based agents. Subsequently, the top $K$ scenarios exhibiting maximum dissimilarity are retained to form the control group.

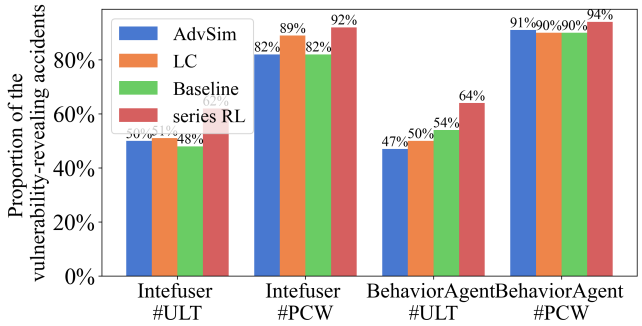



Fig. 9: Proportion of the vulnerability-revealing accidents among all scenarios.

Fig.7 demonstrates that TD3 is able to achieve convergence under both #ULT and #PCW tasks, regardless of the types of SUT employed during the training process.

As illustrated in Fig.8, it is apparent that the proportion of RSS-critical frames in the entire SUT discrete trajectory undergoes a significant increase upon consideration of the RSS reward (Eq.20). The average proportion of RSS-critical frame in the baseline method is 1.79%, while that of series RL is 3.55%, resulting in an increase of 98.32% in the average RSS-critical frame.

The reward inducing the occurrence of RSS-critical frames promotes the larger proportion of accidents revealing vulnerabilities. A comparison was conducted between the efficiency of the baselines and the consideration of RSS rewards in the exploration of AV-blamed accidents. As depicted in Fig.9, the ratio of scenarios revealing vulnerabilities among all scenarios in thoroughly well-trained agents is presented. It is evident that the series RL significantly enhances the generation of scenarios revealing vulnerabilities compared to the baselines devoid of RSS average of up to 16.33%. The baselines only exhibit a vulnerability-revealing accident rate approximating 50% under the #ULT. The principal reason for the notable increase in the proportion of this event to nearly 90% under the #PCW is the difficulty faced by slow-moving pedestrians in posing threats to fast-moving vehicles, leading to a low incidence of pedestrian-vehicle collisions. Conversely, vehicles are more likely to collide with pedestrians.

The combined operation of the reward shaping, Eq.21, and scenario repository maintenance based on DTW can facilitate the accelerated exploration of multiple adversarial scenarios. As depicted in Fig.10, the final top 15 heterogeneous adversarial scenarios retained in the repository post-utilization of the baselines and the series RL are compared across two distinct SUTs and tasks. In the four test cases, the NPC's position trajectory is represented by the gradient solid lines, while the speed trajectory is depicted by the color of the position trajectory corresponding to the right colorbar. It should be noted that AdvSim directly manipulates existing trajectories to perturb the driving paths of NPCs, posing dangers to the SUT. Therefore, trajectories often conform to traffic rules but are homogeneous. LC is a black-box algorithm that optimizes the initial poses of a NPC to attack the SUT. The orientation angle and manipulation of NPCs are set constant in this article, resulting in multiple collision points but tending towards consistent behavior. Evidently, adversarial scenarios utilizing series RL are more diverse, thereby enabling AVs to expose a wider array of vulnerabilities. Conversely, the baselines exhibit a relatively homogeneous type of position and speed trajectories, often opting to strike the NPC at the repetitive behavior to accomplish the adversarial task, thereby overlooking the diversity of scenarios. This occurs even though the method can encounter various local optima.

In order to quantify the coverage of the evaluation scenario, the quantitative indicator results are presented in Tab.III and Tab.IV. In conjunction with Tab.III, it is observed that the average $D_\omega$ of all scenarios in the repository has undergone an increase ranging from a minimum of 29.46% to a maximum of 355.56% with an average metric of 210.41%. The LC method exhibits inferior performance relative to other approaches due to its failure in optimizing process behavior. It is interesting that in the #PCW series RL outperforms than other baselines much better owing to the greater freedom in pedestrian motion control, which is not limited by the dynamic constraints typically imposed on vehicles. To facilitate a more granular evaluation of trajectories, information entropy [62] is employed as a metric to assess the multi-dimensional coverage provided by the trajectories, as illustrated in Tab.IV.

$$H = -\sum_{i=1}^{n} P_i \log(P_i) \tag{22}$$

where $n$ is the total number of the grid, $P_i$ is the relative frequency of trajectory points in the $i$-th grid. Series RL outperforms the baseline models notably in terms of the $x$, $y$, and $v$ dimensions, with its superior performance being especially pronounced in the #PCW scenario. Conversely, in the Interfuser #ULT, AdvSim exhibits marginally greater velocity entropy compared to series RL. This phenomenon is attributed to the frequent acceleration behaviors of vehicles within the scenario, leading to a tendency towards uniformity in speed patterns. Different collisions may be attributed to different module defects or malfunctions in AVs. The series RL generates a series of agents with heterogeneous adversarial strategies, and using them together can help discover the safety level of AVs in the face of different game behaviors, thereby comprehensively evaluating the intelligent performance of AVs.

### *D. Ablation studies*

In order to authenticate each module's contribution, we executed ablation studies—the findings of which are presented in Tab.V. The reward functions of TD3, calculated by employing the Eq.17, 20, and 21, are identified respectively as TD3, TD3-RSS, and TD3-RSS-DTW, as three variants of series RL. The TD3-RSS model notably augmented the number of vulnerability-revealing collisions (NVC), whereas both TD3-RSS-DTW and series RL exhibited a modest reduction in NVC due to a balanced reward structure that resulted in some performance trade-offs, yet still preserved a high standard of function. TD3-RSS-DTW moderately enhanced the diversity of the scenarios, and with the integration of a scenario repository *(i.e.*, series RL), the diversity experienced a more

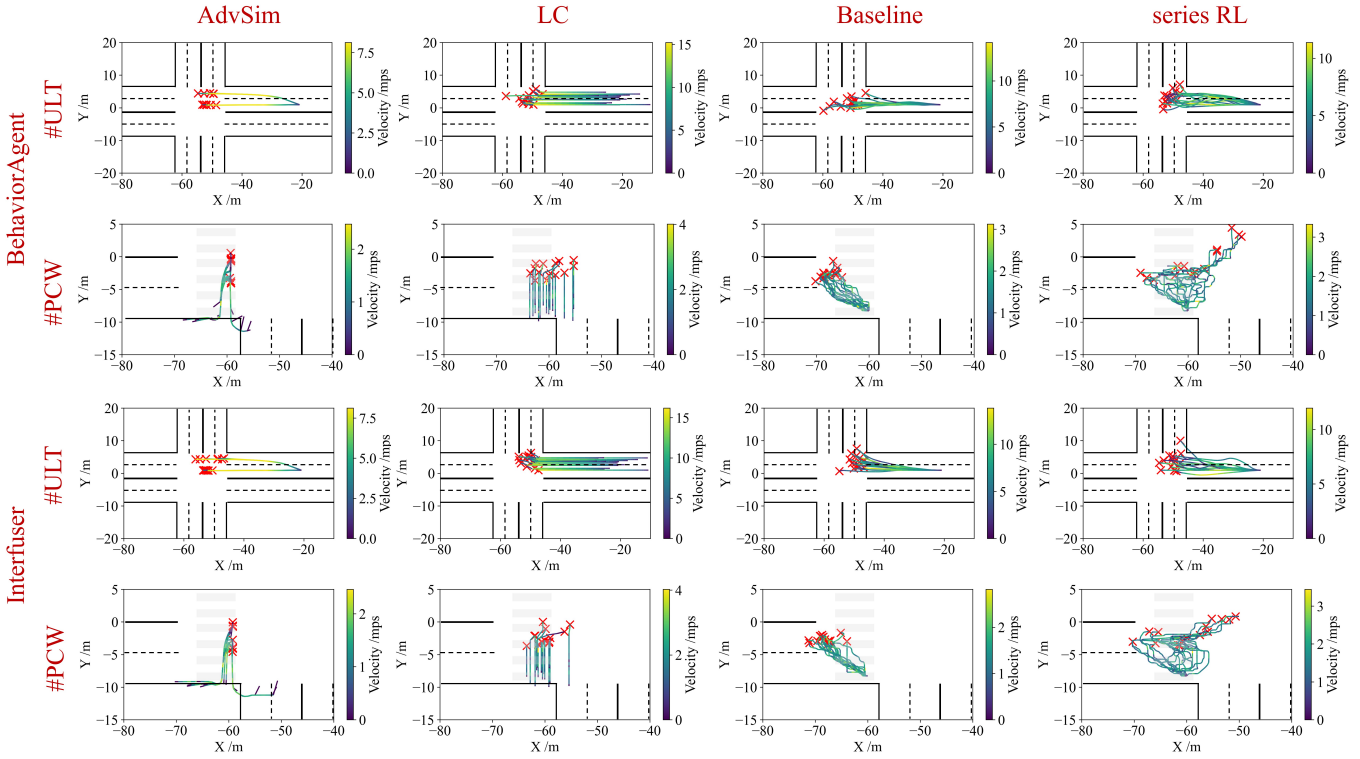



Fig. 10: Scenario repository of the trajectories for the four adversarial policies.

TABLE III: Average diversity of the top 15 most dissimilar scenarios.

| **Average** $D_\omega$ ↑ | Interfuser #ULT | Interfuser #PCW | BehaviorAgent #ULT | BehaviorAgent #PCW |
|---|---|---|---|---|
| AdvSim | 100.94 | 52.76 | 71.96 | 33.94 |
| LC | 48.16 | 43.51 | 82.58 | 49.08 |
| Baseline | 149.19 | 55.30 | 130.69 | 54.98 |
| series RL | **219.40** | **180.93** | **169.19** | **183.68** |

TABLE IV: Entropy of the location and velocity trajectory of the top 15 most dissimilar scenarios.

| **Entropy** (***x*** / ***y*** / ***v***) ↑ | Interfuser #ULT | Interfuser #PCW | BehaviorAgent #ULT | BehaviorAgent #PCW |
|---|---|---|---|---|
| AdvSim | 6.04 / 7.07 / **7.16** | 5.93 / 5.99 / 5.60 | 6.89 /6.93 / 6.95 | 6.63 /6.80 / 6.20 |
| LC | 6.55 / 7.05 / 5.17 | 4.13 / 5.91 / 3.04 | 6.70 / 6.26 /5.60 | 4.21 / 5.22 / 3.30 |
| Baseline | 5.05 / 7.57 / 6.32 | 6.92 / 6.65 / 6.87 | 4.83 / 7.35 / 6.13 | 6.65 / 6.46 / 6.67 |
| series RL | **6.60 / 8.26 /** 7.10 | **8.36 / 8.01 / 8.19** | **7.20 / 8.21 / 7.01** | **7.70 / 7.87 / 7.82** |

TABLE V: Ablation studies of the number of vulnerability-revealing collisions (NVC) and average diversity ($D_\omega$).

| **Method** | **NVC** ↑ | | **Average** $D_\omega$ ↑ | |
|---|---|---|---|---|
| | Interfuser #ULT / #PCW | BehaviorAgent #ULT / #PCW | Interfuser #ULT / #PCW | BehaviorAgent #ULT / #PCW |
| TD3 | 50 / 82 | 47 / 91 | 149.19 / 55.30 | 130.69 / 54.98 |
| TD3-RSS | 64 / 93 | 68 / 95 | 146.26 / 52.42 | 109.20 / 41.48 |
| TD3-RSS-DTW | 61 / 90 | 65 / 94 | 159.04 / 70.30 | 149.97 / 63.76 |
| series RL | 62 / 92 | 64 / 94 | 219.40 / 180.93 | 169.19 / 183.68 |

substantial increase. Collectively, the RSS reward function and the scenario repository modules stand out as the most powerful tools for augmenting NVC and expanding coverage.

### *E. Discussion and Limitations*

To further explore the causes of SUT vulnerability under accidents and provide insights for repairs, qualitative analyses of typical case involving the Interfuser and BehaviorAgent under the #ULT task are studied, as depicted in Fig.11. Regardless of whether it is the Interfuser or BehaviorAgent, the NPC has already initiated evasive action by swerving to the right (Fig.11.(a)(c)) or wandering on the zebra crossing (Fig.11.(b)(d)), thus ensuring the absolute right-of-way before the AV merges into the target lane. However, the AV does not exhibit the intended deceleration (*e.g.* Fig.11.(d)) or it decelerates excessively slowly (*e.g.* Fig.11.(b)), resulting in RSS critical frames before the collision. Interestingly, it was discovered that the Interfuser was able to decelerate several frames before the collision, indicating that the Interfuser possesses a delayed collision-free capability. This feature, to a certain extent, is superior to the BehaviorAgent, which completely lacks collision-avoidance capability. This finding suggests that end-to-end autonomous driving technology continues to hold

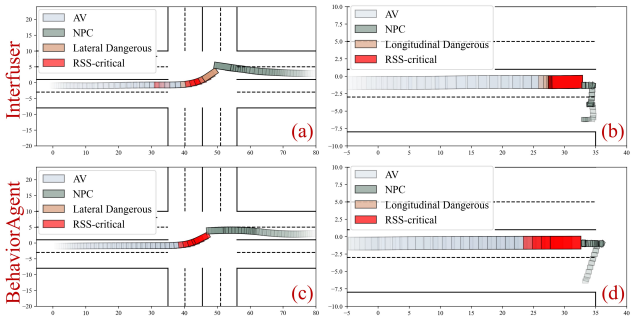



Fig. 11: Vulnerability-revealing accidents for Interfuser under (a) #ULT and (b) #PCW and BehaviorAgent under (c) #ULT and (d) #PCW.

vast potential for future development.

For the Interfuser, Fig.12 provides the four camera video streams and the Bird Eye View (BEV)'s consecutive frames preceding the collision. The AV and NPC are illustrated as the yellow and white boxes in the right BEV snapshot. And the brownish red NPC vehicle shown the in the left RGB picture which is captured by the front camera of the AV. At the time T1, when the NPC is at a significant distance from the AV and before entering the intersection, the AV opts to accelerate at full throttle to attain the target speed, which is safe and in accordance with the RSS. However, at time T2, subsequent to entering the intersection, the NPC appears in the BEV and initiates an evident evasion action, but the AV still chooses to accelerate at full throttle, disregarding the impending collision conflict, which is deemed RSS-critical. It is only at time T3, when the NPC occupies the AV's planned route as depicted in the figure, that the AV decides to apply full brake. Unfortunately, the decision comes too late and due to the high speed (8.72 m/s), the collision is inevitable. Hence, it is evident that the Interfuser lacks the ability to anticipate the behavior of surrounding traffic vehicles and the imminent conflicts, which may necessitate specialized training for this adversarial scenario to enhance the AV system's response capabilities [63]. Nonetheless, owing to the black box nature of neural networks [58], their interpretability continues to require further investigation [64].

The BehaviorAgent is essentially a rudimentary system that takes into account only global routing and vehicle control, completely disregarding the perception components [55]. The waypoints of BehaviorAgent are predetermined prior to departure, and it lacks traffic prediction and local behavior planning capabilities. Consequently, the AV only decides to accelerate or decelerate when the forthcoming waypoints are occupied by the NPC. It, however, is clearly unexpected for the NPC that suddenly enter waypoints, such as the intersection depicted in Fig.11.(c), and other scenarios like lane changing, ramp merging, etc. Therefore, the primary cause of the collision accident in the #ULT case is the BehaviorAgent's deficiency in predicting surrounding traffic.

While the series RL demonstrates noteworthy capabilities in generating a variety of adversarial scenarios pivotal to testing the SUT, it is imperative to acknowledge certain limitations:

- Lack of generalization: Current methodology necessitates distinct training for agents assigned to each scenario. The absence of a unified adversarial paradigm to generate risk scenarios is a drawback. We envision addressing this by investigating agent parameter tuning approaches in the future, mitigating the need for repeated single-agent training.
- Difficulty in engaging with complex traffic environments: Scenarios involving multiple NPCs present unique challenges. Given the stringent constraints required for NPCs, training becomes difficult, leading to a significant reduction in convergence efficiency. As a countermeasure, the employment of multi-agent reinforcement learning [65] in future research could facilitate better behavioral control over multiple NPCs.
- Sim-to-Real Gap [66]: Challenges in implementing RL strategies in real-world environments persist, due to physical constraints like collisions, time costs, and the sheer complexity of real world training. Barriers to deployment in actual vehicles remain steep. Nevertheless, our future research purports to deploy our methods in practical settings to generate high-risk scenarios, thereby enhancing SUT algorithmic performance.

## VI. Threats to Validity

We discussed threats to validity from three aspects: external validity, internal validity, and construct validity, as suggested by the literature [67].

External Validity: External validity pertains to the extent to which the study's results can be generalized beyond its scope, even when a causal relationship has been identified. The proposed series RL does not necessitate internal knowledge of the tested AV system for black-box attacks, except for information related to its driving task. As such, our attack methodology can learn threatening edge adversarial scenarios for any modular or end-to-end AV systems. Furthermore, aside from the #ULT and #PCW tasks, it can be extended to various gaming scenarios, such as lane changes, ramp merging, and following vehicles.

Internal Validity: Internal validity is concerned with the accuracy with which a research design can establish causal relationships. The AV system inherently possesses uncertainties, particularly the end-to-end algorithm. However, for an AV with a specific driving task, the NPC behavior is the sole external variable, while all other variables are strictly regulated. As such, the NPC can be determined as the only variable that exposes the AV's vulnerabilities. Our experiments demonstrate that as training advances, the NPC significantly learns adversarial behaviors that are more likely to cause collisions. Hence, the adversarial attack behavior of the TD3 adversarial agent can swiftly and accurately assess the AV's performance.

Construct Validity: We selected Interfuser and BehaviorAgent, which were developed based on the high-fidelity simulation platform, as our test subjects. Two hazardous driving tasks, #ULT and #PCW, ensuring that the simulation scenario can realistically portray the potential threats of SUTs if deployed on an actual vehicle. Additionally, RSS-critical

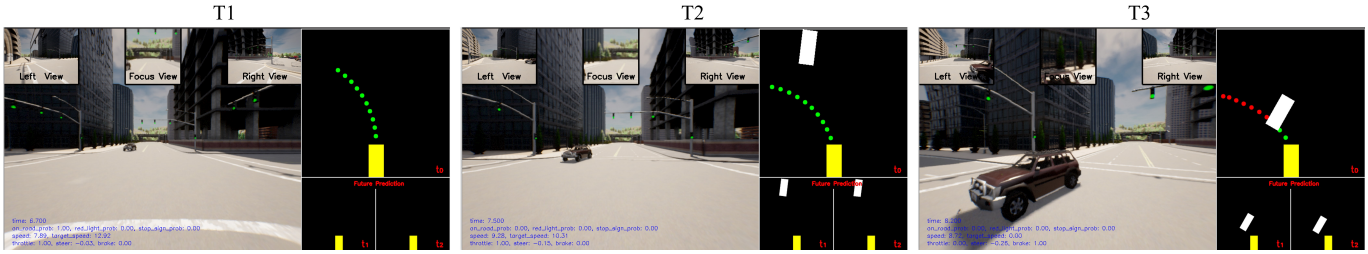



Fig. 12: Snapshots of the video stream for the Intefuser under #ULT. The AV (the yellow box in the right BEV snapshot) is at full throttle at the T1, T2 and full braking at the T3 attacked by the NPC (the white box in the right BEV snapshot and the brownish red vehicle in the left RGB picture).

identification and DTW measurement are employed as the rulers for the vulnerability exposure rate and diversity of attack scenarios, which are accelerated by the series RL.

## VII. Conclusion and Future Works

In this study, we propose a TD3-based RL framework to train series adversarial agents to attack the white-box and black-blox AV systems that generate safety-critical corner cases. The RSS and DTW are integrated into the reward function to gauge the appropriateness of AV operations and the dissimilarity of generated scenarios. We maintain a scenario repository to encourage subsequent adversarial agents to explore attack scenarios uncharted in the refined repository with high richness, thereby exposing multiple AV vulnerability types. AST simulation experiments demonstrate that, compared to the baseline, the series RL significantly enhances the number of vulnerability-revealing and richness scenarios. Through a qualitative analysis of collision accidents caused by SUTs and the provision of improvement suggestions, we guide AV vulnerability remediation, contributing to the development of safer autonomous systems. Consequently, our method can be considered a promising approach for accelerating the validation and verification of AV systems.

Future research will concentrate on extending our series RL to other game scenarios such as lane changing, ramp merging, and car following. Another area of endeavor involves minimizing the sim-to-real gap, thereby facilitating the application of the proposed adversarial stress testing framework in actual AV testing environments. Furthermore, enhancing the safety level of AVs through adversarial attacks launched by traffic participants is another focus of our future investigations.

## References

[1] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li *et al.*, "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1046–1056, 2022.

[2] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 142–10 162, 2021.

[3] X. Bai, P. Dong, Y. Huang, S. Kumari, H. Yu, and Y. Ren, "An ar-based meta vehicle road cooperation testing systems: Framework, components modeling and an implementation example," *IEEE Internet of Things Journal*, 2024.

[4] A. A. A. Aldakkhelallah, M. Todorovic, and M. Simic, "An investigation in autonomous vehicles acceptance," in *International KES Conference on Human Centred Intelligent Systems*. Springer, 2023, pp. 78–87.

[5] Z. Xu, N. Zheng, Y. Lv, Y. Fang, and H. L. Vu, "Analyzing scenario criticality and rider's intervention behavior during high-level autonomous driving: A vr-enabled approach and empirical insights," *Transportation research part C: emerging technologies*, vol. 158, p. 104451, 2024.

[6] Z. Xu, C. Wang, T. Jiang, and N. Zheng, "Impediments to environmental awareness in autonomous driving systems and its effect on user adoption," *IEEE Transactions on Intelligent Vehicles*, 2024.

[7] T. Zhao, E. Yurtsever, J. A. Paulson, and G. Rizzoni, "Formal certification methods for automated vehicle safety assessment," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 232–249, 2022.

[8] X. Zhang, J. Huang, Y. Huang, K. Huang, L. Yang, Y. Han, L. Wang, H. Liu, J. Luo, and J. Li, "Intelligent amphibious ground-aerial vehicles: State of the art technology for future transportation," *IEEE Transactions on Intelligent Vehicles*, 2022.

[9] FESTA-Consortium *et al.*, "Festa handbook version 2 deliverable t6. 4 of the field operational test support action," *Brussels: European Commission*, 2008.

[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[11] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[12] J. Zhang, Y. Lou, J. Wang, K. Wu, K. Lu, and X. Jia, "Evaluating adversarial attacks on driving safety in vision-based autonomous vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3443–3456, 2021.

[13] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[14] J. Tang, Q. Pan, Z. Chen, G. Liu, G. Yang, F. Zhu, and S. Lao, "An improved artificial electric field algorithm for robot path planning," *IEEE Transactions on Aerospace and Electronic Systems*, 2024.

[15] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical scenario generation for autonomous driving-a methodological perspective," *arXiv preprint arXiv:2202.02215*, 2022.

[16] H. Meng, J. Chen, T. Feng, B. Wang, L. Xiong, Z. Yu, and H. Chen, "An interactive car-following model (icfm) for the harmony-with-traffic evaluation of autonomous vehicles," SAE Technical Paper, Tech. Rep., 2023.

[17] A. Kusari, P. Li, H. Yang, N. Punshi, M. Rasulis, S. Bogard, and D. J. LeBlanc, "Enhancing sumo simulator for simulation based testing and validation of autonomous vehicles," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 829–835.

[18] C. E. Tuncali, G. Fainekos, H. Ito, and J. Kapinski, "Simulation-based adversarial test generation for autonomous vehicles with machine learning components," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1555–1562.

[19] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[20] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[21] M. Arief, P. Glynn, and D. Zhao, "An accelerated approach to safely and efficiently test pre-production autonomous vehicles on public streets,"

in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2006–2011.

[22] F. Kruber, J. Wurst, and M. Botsch, "An unsupervised random forest clustering technique for automatic traffic scenario categorization," in *2018 21st International conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2811–2818.

[23] F. Kruber, J. Wurst, E. S. Morales, S. Chakraborty, and M. Botsch, "Unsupervised and supervised learning with the random forest algorithm for traffic scenario clustering and classification," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2463–2470.

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

[25] Z. Huang, M. Arief, H. Lam, and D. Zhao, "Synthesis of different autonomous vehicles test approaches," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2000–2005.

[26] Y. Guo, V. V. Kalidindi, M. Arief, W. Wang, J. Zhu, H. Peng, and D. Zhao, "Modeling multi-vehicle interaction scenarios using gaussian random field," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3974–3980.

[27] B. Wulfe, S. Chintakindi, S.-C. T. Choi, R. Hartong-Redden, A. Kodali, and M. J. Kochenderfer, "Real-time prediction of intermediate-horizon automotive collision risk," *arXiv preprint arXiv:1802.01532*, 2018.

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

[29] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[30] A. Rana and A. Malhi, "Building safer autonomous agents by leveraging risky driving behavior knowledge," in *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*. IEEE, 2021, pp. 1–6.

[31] S. Kim, M. Liu, J. J. Rhee, Y. Jeon, Y. Kwon, and C. H. Kim, "Drivefuzz: Discovering autonomous driving bugs through driving quality-guided fuzzing," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 1753–1767.

[32] Z. Zhong, G. Kaiser, and B. Ray, "Neural network guided evolutionary fuzzing for finding traffic violations of autonomous vehicles," *IEEE Transactions on Software Engineering*, 2022.

[33] Y. Huai, Y. Chen, S. Almanee, T. Ngo, X. Liao, Z. Wan, Q. A. Chen, and J. Garcia, "Doppelgänger test generation for revealing bugs in autonomous driving software," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 2591–2603.

[34] S. Shiroshita, S. Maruyama, D. Nishiyama, M. Y. Castro, K. Hamzaoui, G. Rosman, J. DeCastro, K.-H. Lee, and A. Gaidon, "Behaviorally diverse traffic simulation via reinforcement learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2103–2110.

[35] W. Ding, H. Lin, B. Li, K. J. Eun, and D. Zhao, "Semantically adversarial driving scenario generation with explicit knowledge integration," *arXiv preprint arXiv:2106.04066*, 2021.

[36] R. Lee, M. J. Kochenderfer, O. J. Mengshoel, G. P. Brat, and M. P. Owen, "Adaptive stress testing of airborne collision avoidance systems," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 2015, pp. 6C2–1.

[37] M. Koren, A. Nassar, and M. J. Kochenderfer, "Finding failures in high-fidelity simulation using adaptive stress testing and the backward algorithm," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5944–5949.

[38] A.-K. Reuel, M. Koren, A. Corso, and M. J. Kochenderfer, "Using adaptive stress testing to identify paths to ethical dilemmas in autonomous systems." in *SafeAI@ AAAI*, 2022.

[39] B. Chen, X. Chen, Q. Wu, and L. Li, "Adversarial evaluation of autonomous vehicles in lane-change scenarios," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 8, pp. 10 333–10 342, 2021.

[40] Y. Ma, W. Jiang, L. Zhang, J. Chen, H. Wang, C. Lv, X. Wang, and L. Xiong, "Evolving testing scenario generation method and intelligence evaluation framework for automated vehicles," *arXiv preprint arXiv:2306.07142*, 2023.

[41] A. Sharif and D. Marijan, "Remav: Reward modeling of autonomous vehicles for finding likely failure events," *arXiv preprint arXiv:2308.14550*, 2023.

[42] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.

[43] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[44] T. Bansal, J. Pachocki, S. Sidor, I. Sutskever, and I. Mordatch, "Emergent complexity via multi-agent competition," *arXiv preprint arXiv:1710.03748*, 2017.

[45] P. Lv, Z. Xu, Y. Ji, S. Li, and X. Yin, "Optimal supervisory control of discrete event systems for cyclic tasks," *Automatica*, vol. 164, p. 111634, 2024.

[46] P. Lv, S. Li, and X. Yin, "Optimal deceptive strategy synthesis for autonomous systems under asymmetric information," *IEEE Transactions on Intelligent Vehicles*, 2024.

[47] P. Sun, X. Sun, L. Han, J. Xiong, Q. Wang, B. Li, Y. Zheng, J. Liu, Y. Liu, H. Liu *et al.*, "Tstarbots: Defeating the cheating level builtin ai in starcraft ii in the full game," *arXiv preprint arXiv:1809.07193*, 2018.

[48] Y. Hao, M. Chen, H. Gharavi, Y. Zhang, and K. Hwang, "Deep reinforcement learning for edge service placement in softwarized industrial cyber-physical system," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5552–5561, 2020.

[49] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski *et al.*, "What matters for on-policy deep actor-critic methods? a large-scale study," in *International conference on learning representations*, 2020.

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

[51] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1364–1384, 2020.

[52] B. Liu, H. Zhang, and S. Zhu, "An incremental v-model process for automotive development," in *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2016, pp. 225–232.

[53] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[54] D. Nistér, H.-L. Lee, J. Ng, and Y. Wang, "The safety force field," *NVIDIA White Paper*, 2019.

[55] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[56] K. Xiong, S. Leng, C. Huang, C. Yuen, and Y. L. Guan, "Intelligent task offloading for heterogeneous v2x communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2226–2238, 2020.

[57] A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, "Adaptive stress testing with reward augmentation for autonomous vehicle validatio," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 163–168.

[58] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, USA, 2015, vol. 25.

[59] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*. PMLR, 2023, pp. 726–737.

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

[61] W. Ding, B. Chen, M. Xu, and D. Zhao, "Learning to collide: An adaptive safety-critical scenarios generating method. in 2020 ieee," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2243–2250.

[62] L. Zhang, Y. Ma, X. Xing, L. Xiong, and J. Chen, "Research on the complexity quantification method of driving scenarios based on information entropy," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3476–3481.

[63] V. Behzadan and A. Munir, "Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 2, pp. 236–241, 2019.

[64] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5068–5078, 2021.
[65] J. Guo, Y. Chen, Y. Hao, Z. Yin, Y. Yu, and S. Li, "Towards comprehensive testing on the robustness of cooperative multi-agent reinforcement learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 115–122.
[66] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.
[67] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

**Xuan Cai** received the B.S. degree from Hunan University, Changsha, China, in 2016. He is currently pursuing the Ph.D. degree in traffic information engineering and control from the School of Frontier Science and Technology Innovation Research Institute, Beihang University. His research interests include autonomous vehicle test, control theory, and reinforcement learning.

**Xuesong Bai** received the B.S. degree in vehicle engineering from Beijing Forestry University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in vehicle engineering with Beihang University, Beijing. His research interests include intelligent vehicle intelligent testing, advanced optimization algorithm, metaverse system, microscopic traffic modeling, and its application in automatic driving virtual tests.

**Zhiyong Cui** received the B.S. degree in software engineering from Beihang University, Beijing, China, in 2012, the M.S. degree in software engineering and microelectronics from Peking University, Beijing, in 2015, and the Ph.D. degree in civil engineering from the University of Washington, Seattle, WA, USA, in 2021. He is currently a Professor in the School of Transportation Science and Engineering at Beihang University. His primary research interests include urban computing, traffic forecasting, and connected vehicles. He was the recipient of the IEEE ITSS Best Dissertation Award in 2021 and Best Paper Award at the 2020 IEEE International Smart Cities Conference.

**Peng Hang** received the Ph.D. degree with the School of Automotive Studies, Tongji University, Shanghai, China, in 2019. He is currently a Research Professor with the Department of Traffic Engineering, Tongji University, Shanghai, China. In 2018, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. From 2020 to 2022, he was a Research Fellow with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. His research interests include vehicle dynamics and control, decision making, motion planning, and motion control for autonomous vehicles. He is an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, Journal of Field Robotics, IET Smart Cities, and SAE International Journal of Vehicle Dynamics, Stability, and NVH.

**Haiyang Yu** Haiyang Yu received the Ph.D. degree in traffic environment and safety technology from Jilin University, China, in 2012. He is currently a University Professor with the School of Transportation Science and Engineering, Beihang University, China. His research interests include traffic big data, traffic control, intelligent vehicle infrastructure cooperative systems, and intelligent vehicle intelligent testing.

**Yilong Ren** (Member, IEEE) received the B.S. and Ph.D. degrees from Beihang University in 2010 and 2017, respectively. He is currently an Associate Professor with the Research Institute for Frontier Science, Beihang University. His research interests include vehicular communications, vehicular crowd sensing, traffic big data, intelligent vehicle infrastructure cooperative systems, and intelligent vehicle intelligent testing.