Transportation Research Part C 179 (2025) 105256

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

# Adaptive safety performance testing for autonomous vehicles with adaptive importance sampling

Jingxuan Yang [a], Zihang Wang [b], Daihan Wang [c], Yi Zhang [a,d], Qiujing Lu [a], Shuo Feng [a,*]

[a] Department of Automation, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, 100084, China
[b] Weiyang College, Tsinghua University, Beijing, 100084, China
[c] Guangzhou Automobile Group R&D Center, Guangzhou, 511434, China
[d] Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing, 210096, China

ARTICLE INFO

*Keywords:*
Adaptive testing
Autonomous vehicles
Adaptive importance sampling
Dense reinforcement learning

ABSTRACT

Efficient and accurate safety testing and evaluation are crucial for autonomous vehicles (AVs). Recent studies have utilized prior information, such as surrogate models, to enhance testing efficiency by deliberately generating safety-critical scenarios. However, discrepancies between this prior knowledge and actual AV performance can undermine their effectiveness. To address this challenge, adaptive testing methods dynamically adjust testing policies based on posterior information of AVs, such as testing results. Most existing approaches focus on adaptively optimizing testing policies during pre-tests, yet neglecting how to adapt the testing policies in the large-scale testing process that is required for unbiased safety performance evaluation. To fill this gap, we propose an adaptive testing framework that continuously optimizes testing policies throughout large-scale testing. Our approach iteratively learns AV dynamics through deep learning and optimizes testing policies based on the learned dynamics using reinforcement learning. To tackle the challenge posed by the rarity of safety-critical events, our method focuses exclusively on learning safety-critical states in both the dynamics learning and the policy optimization processes. Additionally, we enhance evaluation robustness by integrating multiple pre-trained testing policies and optimizing their combination coefficients. To accurately assess safety performance, we evaluate testing results obtained from various testing policies using adaptive importance sampling. Experimental validation in overtaking and unprotected left-turn scenarios demonstrates the significant evaluation efficiency of our method.

## 1. Introduction

Safety testing and evaluation are critical components in the development and deployment of AVs. One recommended method for assessing the safety performance (e.g., estimating crash rates) of AVs is to test them in naturalistic driving environment (NDE), observe their behaviors, and statistically compare it to that of human drivers. However, the rarity of safety-critical events (e.g., near-misses and crashes) in NDE necessitates large-scale testing to evaluate AV safety comprehensively, often requiring testing mileage to extend into the billions of miles (Kalra and Paddock, 2016). In recent years, substantial research advances have been made in

* Corresponding author.
*E-mail addresses:* yangjx20@mails.tsinghua.edu.cn (J. Yang), zihang-w21@mails.tsinghua.edu.cn (Z. Wang), wangdaihan@gacrnd.com (D. Wang), zhyi@mail.tsinghua.edu.cn (Y. Zhang), qiujinglu@mail.tsinghua.edu.cn (Q. Lu), fshuo@tsinghua.edu.cn (S. Feng).

https://doi.org/10.1016/j.trc.2025.105256
Received 19 October 2024; Received in revised form 23 May 2025; Accepted 25 June 2025

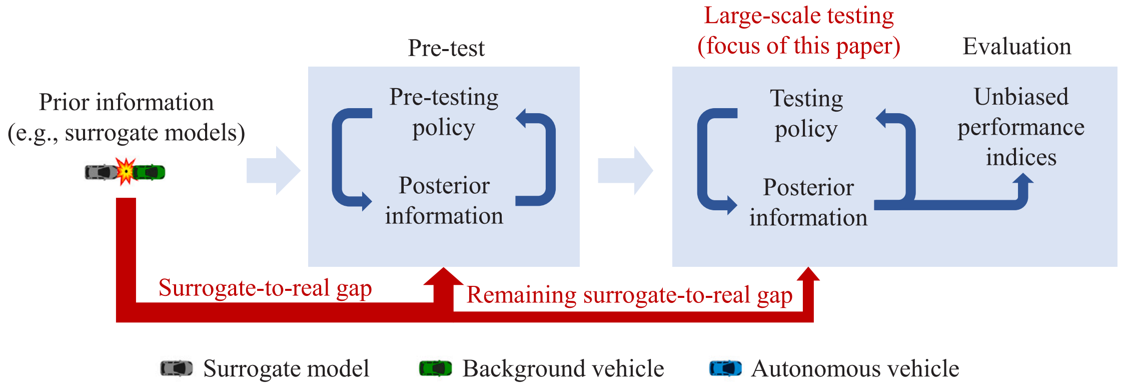


**Fig. 1.** Illustration of three adaptive testing paradigms, each targeting a different stage: (1) The pre-test stage aimed at adapting the testing policy toward the AV under test using minimal number of tests. Note that the testing policy is the driving policy of BVs interacting with the AV under test, modeled as a probabilistic distribution of actions given the current traffic state. (2) The large-scale testing stage involving extensive scenario generation and testing for statistically robust and unbiased evaluation of AV performance metrics. (3) The evaluation stage that aims to evaluate the performance metrics from the testing results. This paper focuses on the large-scale testing stage, introducing an adaptive testing framework that continuously optimizes testing policies.

improving the efficiency of AV testing and evaluation (Li et al., 2016, 2018, 2019; Menzel et al., 2018; Tian et al., 2018; Bagschik et al., 2018; Norden et al., 2019; Klischat and Althoff, 2019; Tuncali et al., 2019; Sinha et al., 2020; Nonnengart et al., 2020; Weng et al., 2020, 2021; Nalic et al., 2020; Li et al., 2021; Wang et al., 2021; Chen et al., 2021; Chelbi et al., 2022; Rempe et al., 2022). A key concept in these studies is leveraging prior knowledge of AVs, such as surrogate models (SMs), to generate testing scenarios (i.e., temporal sequences of traffic scenes (Ulbrich et al., 2015)) rich in safety-critical events. These SMs effectively capture the overall characteristics of AVs, thus significantly enhancing testing efficiency (Feng et al., 2020a,c, 2021; Li et al., 2024a,b). However, the high complexity and black-box nature of AVs lead to discrepancies between SMs and real AV performance. This surrogate-to-real gap can severely reduce the effectiveness of the generated testing scenarios in evaluating the safety performance of diverse AVs.

To address this challenge, the core concept of existing adaptive testing methods (Zhao et al., 2016; Mullins et al., 2017, 2018; Koren et al., 2018; O'Kelly et al., 2018; Corso et al., 2019; Feng et al., 2020b; Sun et al., 2021b; Wang et al., 2022; Yang et al., 2022, 2023, 2025; Gong et al., 2023; Zhou et al., 2023, 2024) is to dynamically adjust the testing policy based on posterior information of AVs, such as testing results. As more testing results are gathered, the posterior information of AVs is progressively enriched, enabling the optimization of testing policies that are better tailored to the specific AV under test. Most existing adaptive testing approaches focus on efficiently optimizing testing policies through a few pre-tests, and then unbiasedly estimate the safety performance of AVs through large-scale testing, utilizing techniques such as importance sampling (Zhao et al., 2016; Feng et al., 2020b; Gong et al., 2023; Yang et al., 2025). In importance sampling, the testing policies (i.e., importance functions) must meet certain criteria to ensure evaluation unbiasedness (Owen, 2013). With the large-scale testing results, the estimation efficiency of performance metrics can be further improved during the evaluation stage (Yang et al., 2022, 2023). The three adaptive testing stages mentioned are illustrated in Fig. 1.

Most existing adaptive testing methods have not addressed adaptive testing during the large-scale testing stage that is required for unbiased safety performance evaluation. The primary distinction between adaptive testing in the pre-test stage and the large-scale testing stage lies in how posterior information is collected. In the pre-test stage, posterior information is actively gathered through specially designed pre-testing policies aimed at minimizing the surrogate-to-real gap and optimizing testing policies with minimal tests. In contrast, during large-scale testing, posterior information is gathered reactively through testing policies that aim to efficiently and accurately evaluate the safety performance of AVs. Consequently, it is challenging to leverage such information to further optimize the testing policies during the large-scale testing stage, while maintaining the unbiasedness of the performance metrics. This optimization is particularly challenging due to the curse of rarity (CoR) (Liu and Feng, 2024), as the safety-critical scenarios (e.g., near-misses and crashes) sampled by the testing policies are usually rare in the high-dimensional scenario spaces involving numerous sequential decision variables of road participants.

To address this challenge, we propose a novel adaptive testing framework that continuously optimizes testing policies during large-scale testing, as shown in Fig. 2. While existing adaptive testing methods focus primarily on optimizing testing policies in the pre-test stage, they often fail to utilize the abundant posterior information generated during large-scale testing to further refine these policies. This oversight limits their ability to adaptively focus on emerging safety-critical scenarios, especially in high-dimensional scenarios where safety-critical events are rare. Our method explicitly addresses this gap by learning AV dynamics based on posterior information (i.e., testing results) and using the learned dynamics to generate simulation testing results for testing policy optimization via reinforcement learning (RL). Two main challenges arise. First, because testing policies must ensure unbiased evaluation, safety-critical dynamics data in testing scenarios are inherently rare. This rarity renders ordinary deep learning methods highly inefficient for learning AV dynamics, as they suffer from the CoR. Second, due to the limited safety-critical dynamics data, the learned AV dynamics may be more accurate in certain states but lack robustness across the entire state space. To overcome these issues, we first focus on learning AV dynamics from safety-critical dynamics data rather than from all data. We then optimize the testing policies

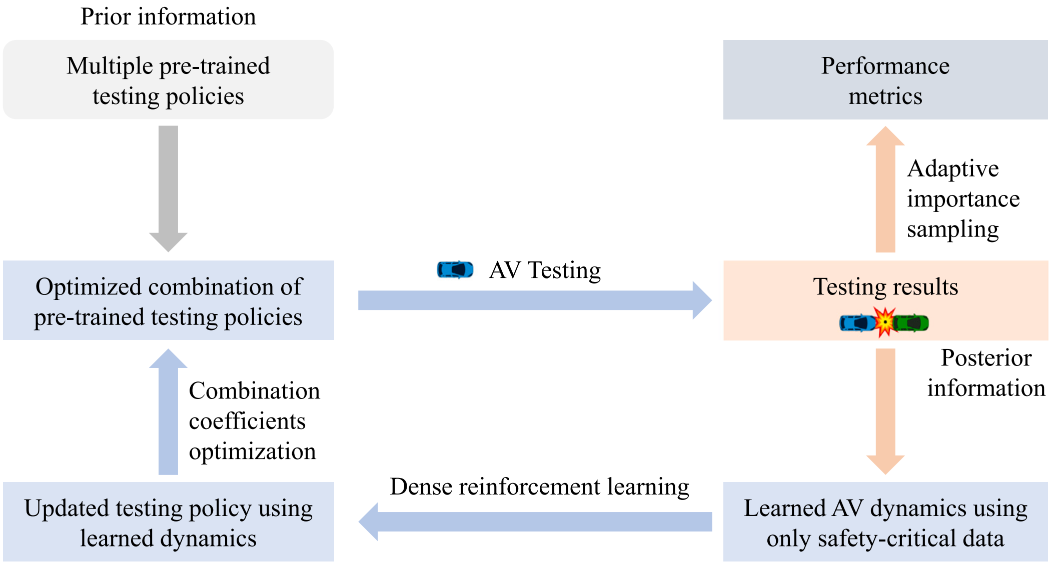

Prior information
Multiple pre-trained testing policies
Optimized combination of pre-trained testing policies
AV Testing
Testing results
Performance metrics
Adaptive importance sampling
Posterior information
Combination coefficients optimization
Updated testing policy using learned dynamics
Dense reinforcement learning
Learned AV dynamics using only safety-critical data

**Fig. 2.** Illustration of the adaptive testing framework. The AV dynamics is learned using only safety-critical states (i.e., traffic scenes where the AV may crash with background vehicles), and the testing policy is then optimized based on the learned dynamics through dense reinforcement learning. Note that traffic scenes are snapshots of traffic scenarios (Ulbrich et al., 2015). Evaluation robustness is improved by utilizing multiple pre-trained testing policies and optimizing their combination coefficients. The testing results from different testing policies are evaluated using adaptive importance sampling.

using dense reinforcement learning (Feng et al., 2023; Yang et al., 2025), which addresses the CoR by initializing from safety-critical states, employing an off-policy learning framework, editing the Markov chains by removing non-critical states and reconnecting safety-critical ones, and backpropagating rewards along the modified chains. To enhance robustness, we utilize multiple pre-trained policies and optimize their combination coefficients. Finally, to estimate performance metrics from the testing results of different testing policies, we employ adaptive importance sampling (Bugallo et al., 2017) to effectively aggregate these results. Validation in overtaking and unprotected left-turn scenarios demonstrates that our method significantly improves evaluation efficiency for various AVs.

The subsequent sections of this paper are structured as follows. In Section 2, we elaborate on the preliminary methods for testing AVs in NDE and formulate the adaptive testing problem in large-scale testing stage. Section 3 then addresses this problem by proposing an adaptive testing framework. Next, Section 4 provides a theoretical analysis of the convergence, consistency, and efficiency of our method. Finally, Section 5 validates the proposed method's effectiveness through testing various AVs in overtaking and unprotected left-turn scenarios.

## 2. Problem formulation

This section introduces the foundational concepts for testing AVs in NDE in Section 2.1. To improve the evaluation efficiency of NDE, the naturalistic and adversarial driving environment (Feng et al., 2021) is introduced in Section 2.2. The adaptive testing problem in large-scale testing stage is subsequently formulated in Section 2.3. Table 1 presents the list of abbreviations, while Table 2 summarizes the notation used.

### 2.1. Naturalistic driving environment testing

Let $\mathcal{E} := (\mathcal{S}, \mathcal{A}, \mathcal{A}^{\mathrm{AV}}, \phi, \pi, \rho)$ represent the NDE, where $\mathcal{S}$ is the state space comprising the positions and velocities of the AV and background vehicles (BVs), $\mathcal{A}$ and $\mathcal{A}^{\mathrm{AV}}$ are the action spaces (i.e., accelerations) for BVs and AV, respectively, $\phi$ and $\pi$ denote the driving policies of BVs and AV, respectively, and $\rho$ is the initial state distribution. The scenario can then be defined as

$$\boldsymbol{x} := (\boldsymbol{s}_0, \boldsymbol{a}_0, \dots, \boldsymbol{s}_{T-1}, \boldsymbol{a}_{T-1}, \boldsymbol{s}_T) \in \mathcal{X}, \tag{1}$$

where $\boldsymbol{s}_t \in \mathcal{S}$ represents the state at time $t$, $\boldsymbol{a}_t \in \mathcal{A}$ is the action of BVs at time $t$, $T$ is the time horizon, and $\mathcal{X}$ is the set of all feasible scenarios. In NDE, the naturalistic distribution $p$ of scenarios is

$$p(\boldsymbol{x}) := \rho(\boldsymbol{s}_0) \prod_{t=0}^{T-1} \phi(\boldsymbol{a}_t|\boldsymbol{s}_t) P_\pi(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t), \ \forall \boldsymbol{x} \in \mathcal{X}, \tag{2}$$

where $P_\pi$ is the state transition probability associated with the AV policy $\pi$.

**Table 1**
List of abbreviations.

| Abbreviation | Definition |
|---|---|
| AAR | average acceleration ratio |
| AV | autonomous vehicle |
| BV | background vehicle |
| CoD | curse of dimensionality |
| CoR | curse of rarity |
| FVDM | full velocity difference model |
| IDM | intelligent driver model |
| LV | leading vehicle |
| MSE | mean squared error |
| NADE | naturalistic and adversarial driving environment |
| NDE | naturalistic driving environment |
| NeuDyM | neural dynamics model |
| RHW | relative half-width |
| RL | reinforcement learning |
| SM | surrogate model |

**Table 2**
Summary of notation.

| Notation | Definition | Notation | Definition |
|---|---|---|---|
| $\boldsymbol{a}, \boldsymbol{a}_t$ | BVs' action, BVs' action at time $t$ | $R_1, R_2$ | range between LV and BV, BV and AV |
| $\boldsymbol{a}^{\mathrm{AV}}$ | action of AV | $\dot{R}_1, \dot{R}_2$ | range rate between LV and BV, BV and AV |
| $a_{\min}, a_{\max}$ | min and max accelerations | $\mathbb{R}$ | set of real numbers |
| $\boldsymbol{A}_t$ | action random variable of BVs | $\boldsymbol{s}, \mathcal{S}$ | state, state space |
| $\boldsymbol{A}^{\mathrm{AV}}$ | action random variable of AV | $\boldsymbol{s}_t, \boldsymbol{S}_t$ | state at time $t$, random variable of $\boldsymbol{s}_t$ |
| $\mathcal{A}$ | action space of BVs | $\mathcal{S}_{\mathrm{crash}}$ | set of crash states |
| $\mathcal{A}^{\mathrm{AV}}$ | action space of AV | $\mathcal{S}_c$ | set of safety-critical states |
| $B$ | batch size | $t, T$ | time step, time horizon |
| $\mathcal{D}^{(k)}$ | set of testing results at update step $k$ | $\mathcal{T}_c$ | set of critical time steps |
| $\mathcal{E}$ | naturalistic driving environment | $V, V^*$ | criticality, optimal criticality |
| $F$ | crash event | $w, W$ | importance policy weight, importance weight |
| $\mathcal{F}$ | $\sigma$-algebra | $W^{(k)}$ | importance weight at update step $k$ |
| $\mathbb{I}_F$ | indicator function of $F$ | $\mathcal{X}$ | scenario space |
| $J$ | number of pre-trained maneuver challenges | $\boldsymbol{x}, \boldsymbol{X}$ | scenario, scenario random variable |
| $k$ | update step in adaptive testing | $v_{\mathrm{LV}}, v_{\mathrm{BV}}, v_{\mathrm{AV}}$ | longitudinal velocities of LV, BV, AV |
| $\mathcal{L}$ | loss function | $x_{\mathrm{LV}}, x_{\mathrm{BV}}, x_{\mathrm{AV}}$ | longitudinal positions of LV, BV, AV |
| $n_k$ | number of tests in update step $k$ | $\alpha, \alpha_j$ | combination coefficients |
| $n^{(k)}$ | total number of tests up to update step $k$ | $\alpha^{(k)}$ | combination coefficients at update step $k$ |
| $\mathbb{N}$ | set of natural numbers | $\gamma$ | discount ratio |
| $p$ | naturalistic distribution | $\delta_t$ | temporal difference error at time $t$ |
| $\mathbb{P}_p$ | probability measure | $\mu$ | crash rate in NDE |
| $P_\pi$ | state transition probability | $\hat{\mu}_p$ | estimation of $\mu$ in NDE |
| $\mathcal{P}$ | power set | $\hat{\mu}_q$ | estimation of $\mu$ in NADE |
| $q$ | importance function | $\tilde{\mu}^{(k)}$ | estimation of $\mu$ by adaptive testing at update step $k$ |
| $q^*$ | optimal importance function | $\nu_t$ | learning rate at time $t$ |
| $q^{(k)}$ | importance function at update step $k$ | $\sigma_q^2$ | asymptotic variance of $\hat{\mu}_q$ |
| $\mathcal{Q}$ | function space of importance functions | $\phi$ | driving policy of BVs |
| $Q$ | maneuver challenge | $\psi$ | importance policy |
| $Q^*$ | optimal maneuver challenge | $\psi^*$ | optimal importance policy |
| $Q^{(k)}$ | maneuver challenge at update step $k$ | $\pi$ | driving policy of AV |
| $Q_j$ | pre-trained maneuver challenges | $\pi^{(k)}$ | NeuDyM policy at update step $k$ |
| $Q_\alpha$ | $\alpha$-combination of $Q_j$ | $\rho$ | initial state distribution |
| $r, R$ | reward, reward random variable | $\epsilon$ | defensive parameter |

Consider the probability space $(\mathcal{X}, \mathcal{F}, \mathbb{P}_p)$, where $\mathcal{F} := \mathcal{P}(\mathcal{X})$ is the $\sigma$-algebra, $\mathcal{P}(\mathcal{X}) := \{\mathcal{X}' : \mathcal{X}' \subseteq \mathcal{X}\}$ is the power set of $\mathcal{X}$, and $\mathbb{P}_p(\{\boldsymbol{x}\}) := p(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ is the probability measure. Denote the crash event between the AV and BVs as $F := \{\boldsymbol{x} \in \mathcal{X} : \boldsymbol{s}_T \in \mathcal{S}_{\mathrm{crash}}\} \in \mathcal{F}$, where $\mathcal{S}_{\mathrm{crash}}$ represents the set of crash states. The crash rate in NDE is then given by

$$\mu := \mathbb{P}_p(F) = \mathbb{E}_p[\mathbb{I}_F(\boldsymbol{X})], \tag{3}$$

where $\boldsymbol{X} : \boldsymbol{x} \mapsto \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathcal{X}$ is the scenario random variable, and $\mathbb{I}_F$ is the indicator function of $F$,

$$\mathbb{I}_F(\boldsymbol{X}) = \begin{cases} 1, & \text{if } \boldsymbol{X} \in F, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

**Algorithm 1:** Scheme of the adaptive testing process in pre-test stage

```
Input: Naturalistic distribution $p$, AV, acquisition function
Output: Optimized importance function
while not stop do
    Choose testing scenarios according to the acquisition function;
    Test the AV in testing scenarios;
    Update acquisition function and importance function using testing results;
end
return Optimized importance function;
```

According to Monte Carlo theory (Owen, 2013), the crash rate can be estimated in NDE as

$$\hat{\mu}_p := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_F(\boldsymbol{X}_i),\ \boldsymbol{X}_i \sim p, \tag{5}$$

where $n$ is the number of tests, and $\boldsymbol{X}_i$ are scenario random variables sampled independently and identically distributed (i.i.d.) from the naturalistic distribution $p$.

### 2.2. Naturalistic and adversarial driving environment testing

The evaluation efficiency of crash rate in NDE is severely hindered due to the CoR (Liu and Feng, 2024), as the rarity of crash events requires an impractically large number of testing miles-often reaching hundreds of millions or even billions (Kalra and Paddock, 2016; Zhang et al., 2018). Using importance sampling to replace the naturalistic distribution $p$ with an importance function $q$ can enhance the evaluation efficiency (Zhao et al., 2016, 2017; Feng et al., 2020a,c; Ren et al., 2025). However, the importance sampling approach cannot be directly applied in high-dimensional scenarios because of the curse of dimensionality (CoD) (Au and Beck, 2003), as the estimation variance of the crash rate using non-optimal importance functions increases exponentially with the number of scenario dimensions. To address this issue, the naturalistic and adversarial driving environment (NADE) has been proposed (Feng et al., 2021), which applies importance sampling only to critical variables at critical time steps, while retaining the naturalistic distribution for other variables. Specifically, the importance function is

$$q(\boldsymbol{x}) := \rho(\boldsymbol{s}_0)\prod_{t=0}^{T-1}\psi(\boldsymbol{a}_t|\boldsymbol{s}_t)P_\pi(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t),\ \forall \boldsymbol{x}\in\mathcal{X}, \tag{6}$$

where $\psi$ is the importance policy defined as

$$\psi(\boldsymbol{a}|\boldsymbol{s}) := \begin{cases}\phi(\boldsymbol{a}|\boldsymbol{s}), & \text{if } \boldsymbol{s}\notin\mathcal{S}_c,\\ \epsilon\phi(\boldsymbol{a}|\boldsymbol{s}) + (1-\epsilon)\dfrac{Q(\boldsymbol{s},\boldsymbol{a})\phi(\boldsymbol{a}|\boldsymbol{s})}{V(\boldsymbol{s})}, & \text{if } \boldsymbol{s}\in\mathcal{S}_c.\end{cases} \tag{7}$$

Here, $\mathcal{S}_c$ represents the set of safety-critical states, $\epsilon\in(0,1)$ is a defensive parameter, $Q(\boldsymbol{s},\boldsymbol{a})\in[0,1]$ is the maneuver challenge indicating the crash probability when BVs take action $\boldsymbol{a}$ in state $\boldsymbol{s}$, $V(\boldsymbol{s}) := \mathbb{E}_\phi[Q(\boldsymbol{s},\boldsymbol{A})]\in[0,1]$ is the criticality (i.e., the expected crash probability when BVs take actions according to the driving policy $\phi$ in state $\boldsymbol{s}$) (Sun et al., 2021a; Bai et al., 2024), and $\boldsymbol{A} : \boldsymbol{x}\mapsto\boldsymbol{a}$ for all $\boldsymbol{x}\in\mathcal{X}$ is the action random variable. The crash rate can then be estimated in NADE as

$$\hat{\mu}_q := \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbb{I}_F(\boldsymbol{X}_i)p(\boldsymbol{X}_i)}{q(\boldsymbol{X}_i)} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_F(\boldsymbol{X}_i)W(\boldsymbol{X}_i),\ \boldsymbol{X}_i\sim q, \tag{8}$$

where $W(\boldsymbol{x}) := p(\boldsymbol{x})/q(\boldsymbol{x}) = \prod_{t\in\mathcal{T}_c} w(\boldsymbol{a}_t|\boldsymbol{s}_t)$ for all $\boldsymbol{x}\in\mathcal{X}$ is the importance weight, $w(\boldsymbol{a}|\boldsymbol{s}) := \phi(\boldsymbol{a}|\boldsymbol{s})/\psi(\boldsymbol{a}|\boldsymbol{s})$ is the importance policy weight, and $\mathcal{T}_c := \{t\in\{0,\dots,T-1\} : \boldsymbol{s}_t\in\mathcal{S}_c\}$ denotes the set of critical time steps.

### 2.3. Adaptive testing in large-scale testing stage

While NADE have demonstrated great potential for efficient testing and evaluation of AVs using the importance function derived from a single SM (Feng et al., 2021, 2023), its evaluation efficiency is significantly affected by the surrogate-to-real gap. In NADE, this gap refers to the discrepancies between the designed importance function and the optimal importance functions for different AVs. The adaptive testing method aims to address this issue. Mathematically, the objective of adaptive testing (in both pre-test stage and large-scale testing stage) is to minimize the variance of the crash rate estimate in NADE over the function space $\mathcal{Q}$, which contains all probability distributions $q$ that satisfy

$$q(\boldsymbol{x}) > 0,\ \forall \boldsymbol{x}\in\{\boldsymbol{x}\in\mathcal{X} : \mathbb{I}_F(\boldsymbol{x})p(\boldsymbol{x}) > 0\}. \tag{9}$$

**Algorithm 2:** Scheme of the adaptive testing process in large-scale testing stage

```
Input: Naturalistic distribution $p$, AV, initial importance function
Output: Evaluation results of the AV
while not stop do
    Choose testing scenarios according to the importance function;
    Test the AV in testing scenarios;
    Update importance function using testing results;
    Evaluate performance metrics using testing results;
end
return Evaluation results;
```

This optimization problem can be formulated as

$$\min_{q\in\mathcal{Q}} \sigma_q^2 := \mathrm{Var}_q\left(\frac{\mathbb{I}_F(\boldsymbol{X})p(\boldsymbol{X})}{q(\boldsymbol{X})}\right). \tag{10}$$

By optimizing $q$ within the function space $\mathcal{Q}$, the importance function can be customized for specific AVs, thereby improving the evaluation efficiency.

In the pre-test stage, one solution to the adaptive testing problem (10) is to employ the Bayesian optimization framework (Snoek et al., 2012), where acquisition functions can be designed to identify the next most informative scenarios, as shown in Algorithm 1. The primary goal of these acquisition functions is to minimize the difference between $q$ and the optimal importance function $q^*$ with as few tests as possible. For example, the acquisition function can be designed to sample the next testing scenario $\boldsymbol{x}'$ such that the absolute gap between $q(\boldsymbol{x}')$ and $q^*(\boldsymbol{x}')$ is maximized across the scenario space. In contrast, during large-scale testing, to accurately and efficiently evaluate performance metrics, the testing scenarios are sampled from importance functions (either initially designed or optimized through adaptive testing) that meet the condition in Eq. (9), as shown in Algorithm 2. As a result, the testing scenarios cannot be deliberately selected during large-scale testing, and we must rely on the testing scenarios and results generated by the importance functions to optimize them. This optimization is highly challenging due to the CoR, as the informative scenarios (e.g., crashes) sampled by the importance functions are so rare that an impractically large number of tests is needed to optimize $q$ effectively.

From Eq. (6), it is clear that optimizing $q$ is equivalent to optimizing the importance policy $\psi$. According to importance sampling theory (Owen, 2013), the optimal importance policy is given by $\psi^*(\boldsymbol{a}|\boldsymbol{s}) := Q^*(\boldsymbol{s},\boldsymbol{a})\phi(\boldsymbol{a}|\boldsymbol{s})/V^*(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$, where $Q^* := \mathbb{P}_p(F|\boldsymbol{S},\boldsymbol{A})$ is the optimal maneuver challenge, and $V^* := \mathbb{P}_p(F|\boldsymbol{S})$ is the optimal criticality. Thus, the key to adaptive testing is optimizing $Q$ toward $Q^*$ based on the currently available testing results. Let $\mathcal{D}^{(k)}$ represent the testing scenarios and results accumulated up to update step $k$, and $Q^{(k)}$ denote the optimized maneuver challenge at update step $k$, for $k = 1, 2, \ldots$. The optimization of $Q^{(k)}$ based on $\mathcal{D}^{(k-1)}$ can be formulated as a reinforcement learning problem. In our experiments, we designate the BV with the highest criticality—i.e., the one most likely to be involved in a crash with the AV—as the RL agent (Feng et al., 2021, 2023). This vehicle is referred to as the principal other vehicle (POV), resulting in a single-agent RL setup, while other BVs and the AV serve as the environment in the RL framework. We note that extending this approach to multiple BV agents is straightforward using multi-agent RL algorithms. Let $\mathcal{M} := (\mathcal{S}, \mathcal{A}, R, P_\pi, \rho, \gamma)$ denote the Markov decision process, where $\mathcal{A}$ is the action space consisting of the POV's acceleration, $R$ is the reward function defined as $R(\boldsymbol{s}) := \mathbb{I}_{\mathcal{S}_{\text{crash}}}(\boldsymbol{s})$ for all $\boldsymbol{s} \in \mathcal{S}$, and $\gamma \in (0, 1]$ is the discount factor. The optimal maneuver challenge $Q^*$ can then be expressed as the state–action value function, i.e.,

$$Q^* = \mathbb{E}_{(\phi,\pi)}\left[R_{t:T}|\boldsymbol{S}_t, \boldsymbol{A}_t\right], \tag{11}$$

where $t$ is any time step, and $R_{t:T} := \sum_{\tau=t+1}^{T} \gamma^{\tau-t-1} R_\tau$ is the discounted sum of future rewards, with $R_\tau := R$ for all $\tau = t+1, \ldots, T$.

The maneuver challenges $Q^{(k)}$ can then be trained using reinforcement learning based on $\mathcal{D}^{(k-1)}$. However, this process encounters the CoR, as safety-critical states and actions are rare, and rewards (i.e., crash events) are highly sparse. As the rarity of the crash events increases, the amount of training data for reinforcement learning to effectively learn $Q^{(k)}$ increases significantly (Liu and Feng, 2024). To mitigate this issue, our previous work introduced the dense reinforcement learning method (Feng et al., 2023; Yang et al., 2025), which learns exclusively the safety-critical states during the temporal difference learning process. However, during testing, the number of safety-critical states in $\mathcal{D}^{(k-1)}$ is far less than what is required for $Q^{(k)}$ to converge to $Q^*$. We address this challenge in the following section.

## 3. Methods

This section presents our adaptive testing method. The core idea is to learn AV dynamics based on testing scenarios and results $\mathcal{D}^{(k-1)}$, and then optimize maneuver challenges $Q^{(k)}$ based on the learned dynamics. Two primary issues emerge. First, the rarity of safety-critical dynamics data in testing scenarios renders ordinary deep learning for learning AV dynamics highly inefficient. Second, due to the limited availability of safety-critical dynamics data, the learned AV dynamics may not generalize well across the entire state space. To address the first issue, Section 3.1 introduces an adaptive dense reinforcement learning method to effectively learn maneuver challenges. To address the second, Section 3.2 formulates a quadratic programming for optimizing the combination coefficients of multiple maneuver challenges pre-trained with different SMs. In Section 3.3, we evaluate the crash rate using testing results obtained from continuously updated importance functions based on adaptive importance sampling. Finally, Section 3.4 summarizes the complete adaptive testing algorithm.

### 3.1. Adaptive dense reinforcement learning

To effectively learn maneuver challenges $Q^{(k)}$, we propose an adaptive dense reinforcement learning method, which iteratively learns both the environment dynamics—specifically, the dynamics of the AV under test—through policy functions $\pi^{(k)}$ and state–action value functions (i.e., maneuver challenges) $Q^{(k)}$. Here, $Q^{(k)} := \mathbb{E}_{(\phi,\pi^{(k)})}[\boldsymbol{R}_{t:T}|\boldsymbol{S}_t, \boldsymbol{A}_t]$ represents the optimal maneuver challenges under the AV policies $\pi^{(k)}$. To capture the dynamics $\pi^{(k)}$, we train a deep neural network using the dynamics data collected during testing, referred to as the neural dynamics model (NeuDyM). The training of NeuDyM faces the CoR, as safety-critical dynamics data is so rare that ordinary training methods (i.e., using all dynamics data) fail to capture meaningful information about safety-critical maneuvers. To resolve this challenge, we propose to use exclusively the safety-critical dynamics data to train NeuDyM. Specifically, the gradient of the loss function $\mathcal{L}$ with respect to the NeuDyM parameters $\boldsymbol{\theta}$ is

$$\tilde{g} := \frac{1}{B}\sum_{i=1}^{B}\frac{\partial\mathcal{L}(\boldsymbol{S}_i, \boldsymbol{A}_i^{\mathrm{AV}})}{\partial\boldsymbol{\theta}}\mathbb{I}_{\mathcal{S}_c}(\boldsymbol{S}_i), \tag{12}$$

where $B \in \mathbb{N}_{>0}$ is the batch size, and $\boldsymbol{A}_i^{\mathrm{AV}} : \boldsymbol{x} \mapsto \boldsymbol{a}^{\mathrm{AV}} \in \mathcal{A}^{\mathrm{AV}}$ is the action random variable of AV. By focusing solely on safety-critical dynamics data, the variance in gradient estimation for the loss function is significantly reduced, enabling NeuDyM to effectively learn AV's safety-critical maneuvers. We assume that the safety-critical dynamics data collected during the large-scale testing process provide sufficient coverage of the safety-critical states for learning the NeuDyM policy. Although these data may be sparse initially, it accumulates progressively as testing continues, enabling the NeuDyM policy to be effectively trained—albeit not to perfection—on safety-critical states.

We then utilize the NeuDyM policies $\pi^{(k)}$ to learn the maneuver challenges $Q^{(k)}$ through dense reinforcement learning (Feng et al., 2023; Yang et al., 2025). The process begins by initializing $\hat{Q}(\boldsymbol{s}, \boldsymbol{a}) = 0$ for all $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{a} \in \mathcal{A}$. In each training iteration, the initial state is uniformly sampled from $\mathcal{S}_c$, after which the BVs follow the uniform behavior policy while the AV follows the NeuDyM policy $\pi^{(k)}$. In dense reinforcement learning, the update rule for $\hat{Q}$ is

$$\hat{Q}(\boldsymbol{S}_t, \boldsymbol{A}_t) \leftarrow \hat{Q}(\boldsymbol{S}_t, \boldsymbol{A}_t) + \nu_t\delta_t\mathbb{I}_{\mathcal{S}_c}(\boldsymbol{S}_t), \tag{13}$$

where $\nu_t$ is the learning rate, and $\delta_t$ is the temporal difference error defined as

$$\delta_t := \boldsymbol{R}_{t+1} + \gamma\mathbb{E}_{\phi}\left[\hat{Q}(\boldsymbol{S}_{t+1}, \boldsymbol{A}_{t+1})|\boldsymbol{S}_{t+1}\right] - \hat{Q}(\boldsymbol{S}_t, \boldsymbol{A}_t). \tag{14}$$

Note that training the NeuDyM policy exclusively on safety-critical data may introduce bias for non-safety-critical states. However, our method does not rely on accurate modeling of the AV's behavior across the entire state space. Instead, the NeuDyM policy is used solely to simulate the AV's actions when generating training data for maneuver challenge learning via dense reinforcement learning. In this process, initial states are sampled from safety-critical states, BVs' actions are drawn from a uniform distribution, and AV actions are derived from the NeuDyM policy. Therefore, since training is confined to safety-critical states, any bias in non-safety-critical states do not impact the effectiveness of maneuver challenge training.

Under mild assumptions, dense reinforcement learning is guaranteed to converge almost surely to $Q^{(k)}$ (see Theorem 1 in Yang et al., 2025). However, the learned $Q^{(k)}$ may not generalize well to out-of-distribution state–action pairs. Specifically, the $Q^{(k)}$ values are typically more accurate within a subspace of the entire state–action space—primarily the space covered by the safety-critical dynamics data in $\mathcal{D}^{(k-1)}$—while values in other subspaces may degrade, potentially becoming less accurate than before. As a result, directly applying the importance function derived from $Q^{(k)}$ for AV testing and evaluation can be risky. We address this issue in the following subsection.

### 3.2. Combination coefficient optimization

To improve the evaluation robustness of $Q^{(k)}$, we propose using a convex combination of multiple maneuver challenges $Q_j$ for $j = 1, \ldots, J$, pre-trained with diverse SMs, to create a constrained optimization space. The combination coefficients are then optimized by solving the following regression problem:

$$\begin{aligned}&\min_{\boldsymbol{\alpha}\in\mathbb{R}^J}\ \frac{1}{2}\sum_{(\boldsymbol{s},\boldsymbol{a})\in\mathcal{S}_c\times\mathcal{A}}\left[Q^{(k)}(\boldsymbol{s},\boldsymbol{a}) - Q_{\boldsymbol{\alpha}}(\boldsymbol{s},\boldsymbol{a})\right]^2\\ &\ \text{s.t.}\ \ \mathbf{1}^\top\boldsymbol{\alpha} = 1,\ \boldsymbol{\alpha} \geqslant \mathbf{0},\end{aligned} \tag{15}$$

where $Q_{\boldsymbol{\alpha}} := \sum_{j=1}^{J}\alpha_j Q_j$ is the $\boldsymbol{\alpha}$-combination of $Q_j$, and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_J]^\top$ is the vector of combination coefficients. This regression problem is a quadratic programming and can be solved using standard convex optimization tools such as CVXOPT (Andersen et al., 2004). Let $\boldsymbol{\alpha}^{(k)}$ represent the solution to Eq. (15). The importance functions $q^{(k)}$ can then be derived from Eqs. (6) and (7) using $Q_{\boldsymbol{\alpha}^{(k)}}$ in place of $Q$. By iteratively updating the importance functions based on testing results and sampling new testing scenarios from the updated importance functions to gather more results, we establish the adaptive testing process, except for crash rate estimation, which is elaborated in the next subsection.

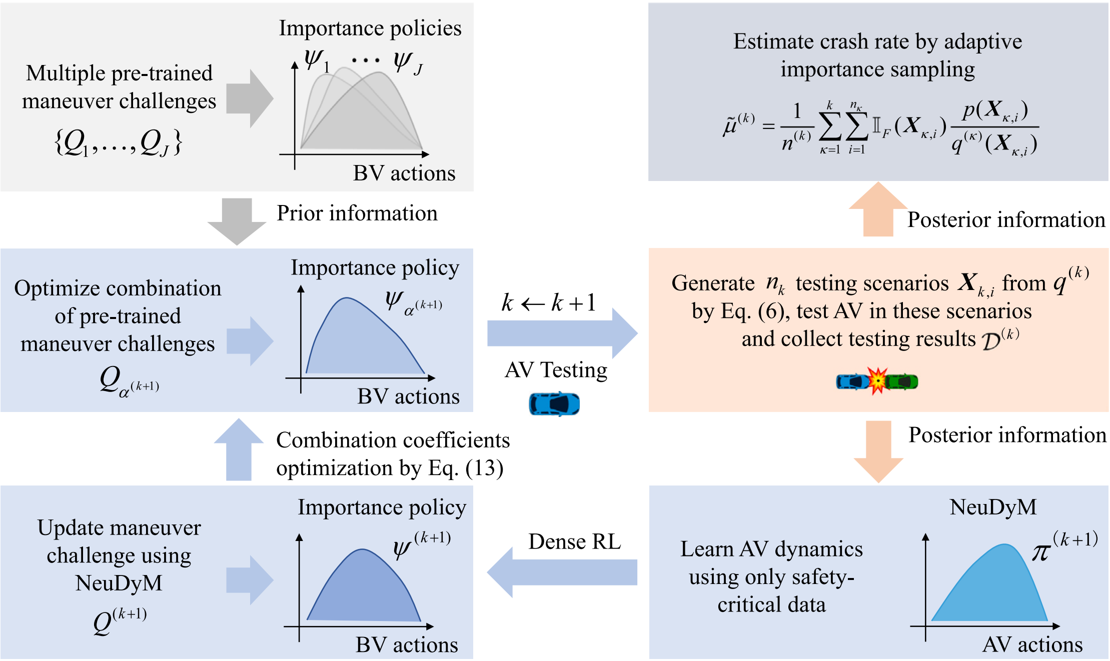

Multiple pre-trained maneuver challenges
$\{Q_1,\dots,Q_J\}$
Importance policies
$\psi_1 \cdots \psi_J$
BV actions
Prior information
Estimate crash rate by adaptive importance sampling
$\tilde{\mu}^{(k)} = \frac{1}{n^{(k)}}\sum_{\kappa=1}^{k}\sum_{i=1}^{n_\kappa}\mathbb{I}_F(X_{\kappa,i})\frac{p(X_{\kappa,i})}{q^{(\kappa)}(X_{\kappa,i})}$
Posterior information
Optimize combination of pre-trained maneuver challenges
$Q_{\alpha^{(k+1)}}$
Importance policy
$\psi_{\alpha^{(k+1)}}$
BV actions
$k \leftarrow k+1$
AV Testing
Generate $n_k$ testing scenarios $X_{k,i}$ from $q^{(k)}$ by Eq. (6), test AV in these scenarios and collect testing results $\mathcal{D}^{(k)}$
Combination coefficients optimization by Eq. (13)
Posterior information
Update maneuver challenge using NeuDyM
$Q^{(k+1)}$
Importance policy
$\psi^{(k+1)}$
BV actions
Dense RL
Learn AV dynamics using only safety-critical data
NeuDyM
$\pi^{(k+1)}$
AV actions

**Fig. 3.** Illustration of the workflow of the adaptive testing framework.

### 3.3. Adaptive importance sampling

To estimate the crash rate based on the testing results obtained from different importance functions, we propose applying adaptive importance sampling techniques (Bugallo et al., 2017) to aggregate these results effectively. The crash rate can be estimated using adaptive importance sampling as

$$\tilde{\mu}^{(k)} = \frac{1}{n^{(k)}}\sum_{\kappa=1}^{k}\sum_{i=1}^{n_\kappa}\mathbb{I}_F(\boldsymbol{X}_{\kappa,i})W^{(\kappa)}(\boldsymbol{X}_{\kappa,i}),\ \boldsymbol{X}_{\kappa,i}\sim q^{(\kappa)}, \tag{16}$$

where $n^{(k)} := \sum_{\kappa=1}^{k} n_\kappa$ is the total number of tests conducted, $n_\kappa$ is the number of testing scenarios sampled from $q^{(\kappa)}$, and $W^{(\kappa)} := p/q^{(\kappa)}$, for $\kappa = 1,\dots,k$ and $k = 1,2,\dots$. It is worth noting that various adaptive importance sampling methods are available (see Bugallo et al., 2017 and references therein). We use Eq. (16) due to its simplicity, effectiveness, and because it provides theoretical guarantees of consistency and approximate normality—properties often lacking in most adaptive importance sampling approaches (Bugallo et al., 2017). In Section 4, we will show that our method ensures both consistency and approximate normality under mild assumptions.

### 3.4. Adaptive testing workflow and algorithm

The adapting testing workflow is shown in Fig. 3. The pipeline for update step $k$ (where $k \geqslant 1$) is

$$\mathcal{D}^{(k)} \xrightarrow{(12)} \pi^{(k+1)} \xrightarrow{(13)} Q^{(k+1)} \xrightarrow{(15)} Q_{\boldsymbol{\alpha}^{(k+1)}} \xrightarrow{(6)\text{ and }(7)} q^{(k+1)} \xrightarrow{(16)} (\mathcal{D}^{(k+1)}, \tilde{\mu}^{(k+1)}). \tag{17}$$

The core idea is to train the NeuDyM policies $\pi^{(k+1)}$ using only safety-critical dynamics data and apply dense reinforcement learning to learn the maneuver challenges $Q^{(k+1)}$. Next, the combination coefficients $\boldsymbol{\alpha}^{(k+1)}$ are optimized through solving a quadratic programming problem. Following this, the importance functions $q^{(k+1)}$ are updated, and testing scenarios are sampled from $q^{(k+1)}$. Then the crash rate is estimated using adaptive importance sampling. This iterative process continues until the termination criteria are met. We use the relative half-width (RHW) (Zhao et al., 2016) as the stopping criterion, with the threshold set at 0.3. In summary, the adaptive testing process is given in Algorithm 3.

## 4. Theoretical analysis

This section presents a theoretical analysis of the proposed adaptive testing method, covering the convergence analysis of adaptive dense reinforcement learning in Section 4.1, the consistency analysis of adaptive testing in Section 4.2, and the efficiency analysis of adaptive testing in Section 4.3.

**Algorithm 3:** Adaptive testing with adaptive importance sampling

```
Input: naturalistic distribution $p$, surrogate maneuver challenges $Q_j$, $j = 1, \dots, J$, number of testing scenarios $n_k$ at each
    update step $k$
Output: crash rate estimate
Initialize $Q_{\alpha^{(1)}} = (1/J)\sum_{j=1}^{J} Q_j$, $\ell = 1$, $\ell_{\text{th}} = 0.3$, $k = 1$;
while $\ell > \ell_{th}$ do
    Compute $q^{(k)}$ based on Eqs. (6) and (7) with $Q_{\alpha^{(k)}}$ for $Q$;
    Sample $n_k$ testing scenarios from $q^{(k)}$;
    Estimate crash rate $\tilde{\mu}^{(k)}$ using adaptive importance sampling via Eq. (16);
    Set $\ell \leftarrow$ relative half-width of $\tilde{\mu}^{(k)}$;
    Set $\mathcal{D}^{(k)} \leftarrow$ testing scenarios and results up to now;
    Set $k \leftarrow k + 1$;
    Train NeuDyM policy $\pi^{(k)}$ based on Eq. (12);
    Apply dense reinforcement learning to learn $Q^{(k)}$ using policy $\pi^{(k)}$;
    Solve the quadratic programming in Eq. (15) to update $\alpha^{(k)}$ (e.g., via CVXOPT (Andersen et al., 2004));
end
Return $\tilde{\mu}^{(k)}$;
```

### 4.1. Convergence analysis

First, we prove the convergence of adaptive dense reinforcement learning, i.e., under certain assumptions, the maneuver challenge $Q^{(k)}$ learned by our adaptive testing method converges almost surely to the optimal maneuver challenge $Q^*$.

**Assumption 1.** The NeuDyM policies $\pi^{(k)}$ converge to $\pi^\dagger \in \Pi$ almost surely, i.e., $\mathbb{P}_p\left(\lim_{k\to\infty} \pi^{(k)} = \pi^\dagger\right) = 1$, denoted as $\pi^{(k)} \xrightarrow{\text{a.s.}} \pi^\dagger$.

**Assumption 2.** The assumptions of Theorem 1 in Yang et al. (2025) hold for all $\pi^{(k)}$, $k = 1, 2, \dots$.

**Theorem 1.** *Suppose that Assumptions 1 and 2 hold, then $Q^{(k)} \xrightarrow{a.s.} Q^\dagger$, where $Q^\dagger := \mathbb{E}_{(\phi,\pi^\dagger)}[R_{t:T}|\boldsymbol{S}_t, \boldsymbol{A}_t]$ and $t$ is any time step.*

**Proof.** From Assumption 1 we have $\pi^{(k)} \xrightarrow{\text{a.s.}} \pi^\dagger$. Under Assumption 2, the dense reinforcement learning algorithm will converge to $Q^{(k)}$ for any NeuDyM policy $\pi^{(k)}$. Since $\pi^{(k)}(\boldsymbol{a}|\boldsymbol{s}) \leqslant 1$ for all $\boldsymbol{s} \in \mathcal{S}$, $\boldsymbol{a} \in \mathcal{A}$, and $k \geqslant 1$, each $\pi^{(k)}$ is dominated by the constant function 1. Therefore, by the Lebesgue's dominated convergence theorem (Rudin, 1987), it follows that $Q^{(k)} \xrightarrow{\text{a.s.}} Q^\dagger$. □

**Corollary 1.** *If $\pi^{(k)} \xrightarrow{a.s.} \pi$ and Assumption 2 holds, then $Q^{(k)} \xrightarrow{a.s.} Q^*$.*

**Proof.** This follows directly from Theorem 1. □

**Remark 1.** As shown in Theorem 1, if NeuDyM policies $\pi^{(k)}$ can converge to some policy $\pi^\dagger$ (not necessarily optimal) almost surely and Assumption 2 holds, then $Q^{(k)}$ will converge to $Q^\dagger$ almost surely. Assumption 2 is necessary to ensure the convergence of dense reinforcement learning. Since NeuDyM is a deep neural network, it can approximate AV's driving policy $\pi$ well given infinite training data, though convergence to $\pi$ cannot be guaranteed. Nonetheless, Corollary 1 shows that if $\pi^{(k)}$ converges to $\pi$ almost surely and Assumption 2 holds, then $Q^{(k)}$ will converge to $Q^*$ almost surely. In practice, due to limited data and the approximation capacity of neural networks, the convergence to the exact driving policy $\pi$ may not be fully realized. However, our empirical results show that even approximate convergence can lead to substantial gains in optimizing maneuver challenges.

### 4.2. Consistency analysis

Next, we prove the consistency of the proposed adaptive testing method, i.e., $\tilde{\mu}^{(k)} \xrightarrow{\text{a.s.}} \mu$ as $k \to \infty$. The proof is based on Lemma 1 (Oh and Berger, 1992). To elaborate on this lemma, we define some notations used in Oh and Berger (1992). Suppose the goal is to estimate $\eta = \mathbb{E}_f[\varphi]$, where $f$ is a probability distribution. Denote the parametric family of importance functions as $\mathcal{G} := \{g_\lambda : \lambda \in \Lambda\}$, where $\Lambda$ is the parameter space. At each update step $k = 1, 2, \dots$, denote the parameter as $\lambda^{(k)}$ and the corresponding importance function as $g^{(k)} := g_{\lambda^{(k)}}$. The estimate for $\eta$ is given by

$$\hat{\eta}^{(k)} = \frac{1}{n^{(k)}} \sum_{\kappa=1}^{k} \sum_{i=1}^{n_\kappa} \frac{\varphi(\boldsymbol{X}_{\kappa,i}) f(\boldsymbol{X}_{\kappa,i})}{g^{(\kappa)}(\boldsymbol{X}_{\kappa,i})}, \quad \boldsymbol{X}_{\kappa,i} \sim g^{(\kappa)}. \tag{18}$$

**Assumption 3.** The importance functions $g^{(k)}$, $k = 1, 2, \dots$, have the same support as $f$.

**Assumption 4.** The importance weights $f/g^{(k)}$ are bounded for all $k = 1, 2, \ldots$ .

**Assumption 5.** The expectation $\eta = \mathbb{E}_f[\varphi]$ exists.

**Lemma 1.** *Suppose that Assumptions 3, 4 and 5 hold and $\varphi$ has finite second moment, then $\hat{\eta}^{(k)} \xrightarrow{a.s.} \eta$.*

**Proof.** This is the Theorem 3.1 in Oh and Berger (1992). □

**Theorem 2.** *Let $\tilde{\mu}^{(k)}$ be given by Eq. (16). Then we have $\tilde{\mu}^{(k)} \xrightarrow{a.s.} \mu$.*

**Proof.** The correspondence to Lemma 1 is established by associating $\varphi$ with $\mathbb{I}_F$, $f$ with $p$, and $g$ with $q$. We then verify Assumptions 3, 4 and 5 accordingly.

(1) Verification of Assumption 3. From Eq. (7), we know that $\psi^{(k)}(\boldsymbol{a}|\boldsymbol{s}) > 0$ whenever $\phi(\boldsymbol{a}|\boldsymbol{s}) > 0$ for all $k = 1, 2, \ldots$ . Therefore, $q^{(k)}(\boldsymbol{x}) > 0$ whenever $p(\boldsymbol{x}) > 0$ for all $k = 1, 2, \ldots$ , meaning that all $q^{(k)}$ share the same support as $p$.
(2) Verification of Assumption 4. The importance policy weight for any critical state $\boldsymbol{s} \in \mathcal{S}_c$ is

$$\begin{aligned} w(\boldsymbol{a}|\boldsymbol{s}) &= \frac{\phi(\boldsymbol{a}|\boldsymbol{s})}{\epsilon\phi(\boldsymbol{a}|\boldsymbol{s}) + (1-\epsilon)\dfrac{Q(\boldsymbol{s},\boldsymbol{a})\phi(\boldsymbol{a}|\boldsymbol{s})}{V(\boldsymbol{s})}} \\ &\leqslant \frac{\phi(\boldsymbol{a}|\boldsymbol{s})}{\epsilon\phi(\boldsymbol{a}|\boldsymbol{s})} \\ &= \frac{1}{\epsilon},\ \forall \boldsymbol{a} \in \mathcal{A}. \end{aligned} \tag{19}$$

Thus, the importance weight for all $\boldsymbol{x} \in \mathcal{X}$ is given by

$$W(\boldsymbol{x}) = \prod_{t\in\mathcal{T}_c} w(\boldsymbol{a}_t|\boldsymbol{s}_t) \leqslant \frac{1}{\epsilon^L}, \tag{20}$$

where $L$ is the maximum number of critical time steps. Therefore, the importance weights $W^{(k)}$ are bounded by $1/\epsilon^L$ for all $k = 1, 2, \ldots$ .
(3) Verification of Assumption 5. Clearly, $\eta = \mathbb{E}_f[\varphi] = \mathbb{E}_p[\mathbb{I}_F] = \mu$ exists.

Note that $(\mathbb{I}_F)^2 = \mathbb{I}_F$, which implies that $\mathbb{I}_F$ has finite second moment. Thus, the theorem follows. □

**Remark 2.** As shown in Theorem 2, our adaptive testing method converges to the true crash rate almost surely. That is, the probability that the crash rate estimated by our method converges to the true value approaches one as the number of samples tends to infinity. This is achieved primarily by employing the defensive importance sampling paradigm in Eq. (7), where we incorporate the naturalistic policy $\phi$ with probability $\epsilon$, which ensures that Assumptions 3 and 4 are satisfied. Furthermore, since $\eta = \mathbb{E}_f[\varphi] = \mathbb{E}_p[\mathbb{I}_F] = \mu$ represents the crash rate we aim to estimate, Assumption 5 is clearly met. Practically, the experimental results shown in Fig. 8(a)–(c) in Section 5.2 and Fig. 11(a)–(c) in Section 5.3 will demonstrate that the crash rate estimates from our adaptive testing method closely align with the ground truth values obtained via NDE. It is worth noting that in real-world applications, convergence may require a substantial number of samples, and computational limitations may affect the practical implementation of theoretical guarantees. Nonetheless, our method remains effective in producing accurate performance evaluations with significantly improved efficiency.

### *4.3. Efficiency analysis*

Finally, we analyze the efficiency of the adaptive testing method. In addition to Assumptions 3, 4 and 5, we also require the following two additional assumptions.

**Assumption 6.** The parameters $\lambda^{(k)}$ converge to $\lambda^{\dagger} \in \Lambda$ almost surely, i.e., $\lambda^{(k)} \xrightarrow{\text{a.s.}} \lambda^{\dagger}$.

**Assumption 7.** Given Assumption 6, the importance functions $g^{(k)}$ converge to $g_{\lambda^{\dagger}}$ almost surely, i.e., $g^{(k)} \xrightarrow{\text{a.s.}} g_{\lambda^{\dagger}}$.

**Lemma 2.** *Suppose that Assumptions 3, 4, 5, 6 and 7 hold, $\varphi$ has finite fourth moment, and $\sum_{\kappa=1}^{k} n_\kappa^2 / \left(n^{(k)}\right)^2 \to 0$ as $k \to \infty$. Then*

$$\sqrt{n^{(k)}}\left(\hat{\eta}^{(k)} - \eta\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{g_{\lambda^{\dagger}}}\right), \tag{21}$$

*where $\xrightarrow{d}$ denotes convergence in distribution, and $\sigma^2_{g_{\lambda^{\dagger}}} := \mathrm{Var}_{g_{\lambda^{\dagger}}}(\varphi f/g_{\lambda^{\dagger}})$.*

**Proof.** This is the Theorem 3.2 in Oh and Berger (1992). □

**Theorem 3.** *Let $\tilde{\mu}^{(k)}$ be given by Eq. (16). Suppose that Assumptions 1 and 2 hold, and $\sum_{\kappa=1}^{k} n_\kappa^2 / \left(n^{(k)}\right)^2 \to 0$ as $k \to \infty$. Then*

$$\sqrt{n^{(k)}}\left(\tilde{\mu}^{(k)} - \mu\right) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2_{q^\dagger}\right), \tag{22}$$

*where $\sigma^2_{q^\dagger} := \mathrm{Var}_{q^\dagger}(\mathbb{I}_F p / q^\dagger)$ and $q^\dagger$ is given by Eqs. (6) and (7) with $Q^\dagger$ in place of $Q$.*

**Proof.** The correspondence to Lemma 2 follows from associating $\varphi$ with $\mathbb{I}_F$, $f$ with $p$, $g$ with $q$, and $\lambda$ with $Q$. In the proof of Theorem 2, Assumptions 3, 4 and 5 have already been verified. Therefore, we only need to verify the remaining Assumptions 6 and 7.

(1) Verification of Assumption 6. Given Assumptions 1 and 2, Theorem 1 establishes that $Q^{(k)} \xrightarrow{\text{a.s.}} Q^\dagger$. Hence, Assumption 6 holds.
(2) Verification of Assumption 7. Since $Q^{(k)} \xrightarrow{\text{a.s.}} Q^\dagger$, by the Lebesgue's dominated convergence theorem (Rudin, 1987), we have $q^{(k)} \xrightarrow{\text{a.s.}} q^\dagger$.

Since $(\mathbb{I}_F)^4 = \mathbb{I}_F$, it is clear that $\mathbb{I}_F$ has finite fourth moment, which concludes the proof of this theorem. □

**Remark 3.** Note that if $n_k \leqslant N$ for some $N > 0$ and for all $k = 1, 2, \ldots$, then we have

$$\sum_{\kappa=1}^{k}\left(\frac{n_\kappa}{n^{(k)}}\right)^2 \leqslant k\left(\frac{N}{k}\right)^2 = \frac{N^2}{k} \to 0,\ k \to \infty. \tag{23}$$

This can be easily achieved if $n_k$ are the same for all $k = 1, 2, \ldots$. Theorem 3 shows that if $q^{(k)}$ converge almost surely to an importance function $q^\dagger$ (not necessarily optimal), then the distributions of $\tilde{\mu}^{(k)}$ converge to $\mathcal{N}\left(\mu, \sigma^2_{q^\dagger}/n^{(k)}\right)$. This implies that as the importance functions are iteratively optimized, the asymptotic estimation variances of crash rates decrease and eventually converge to $\sigma^2_{q^\dagger}$. The closer the importance functions are optimized toward $q^*$, the more the estimation variance is reduced, and the greater the improvement in estimation efficiency. In practice, the convergence to the optimal importance function $q^*$ may not be feasible due to limited knowledge of the AV's behavior model or high-dimensional scenario spaces. However, even partial convergence to a reasonably well-optimized $q^\dagger$ still yields substantial variance reduction, as validated by our experimental results.

## 5. Results

This section first details the generation processes of NDE and NADE in Section 5.1. The testing and evaluation results under NDE, NADE, and adaptive testing are then presented for overtaking scenarios in Section 5.2, and for unprotected left-turn scenarios in Section 5.3.

### 5.1. Generation of NDE and NADE

The generation of NDE and NADE adheres to the methodology outlined in Feng et al. (2021).[1] In NDE, the driving policies of BVs are aligned with naturalistic policies extracted from naturalistic driving data, which effectively replicates realistic traffic conditions (Yan et al., 2023; He et al., 2024). As a result, the crash rates estimated in NDE can be regarded as ground truth. In NADE, the driving policies of BVs are adjusted at critical moments using importance policies, while preserving naturalistic policies at other moments. Critical moments are identified by evaluating the criticality of each BV at every time step. Time steps where the criticality exceeds a specified threshold (e.g., zero) are marked as critical moments. The BV with the highest criticality is selected as the POV, and its driving policy is replaced with the importance policy.

To show the generalizability of our method, we select three distinct AVs:

(1) AV-I: the intelligent driver model (IDM) (Treiber et al., 2000);
(2) AV-II: a variation of IDM with different calibrated parameters (Sangster et al., 2013), used to test the robustness of our method to IDM parameter changes;
(3) AV-III: a RL agent trained with proximal policy optimization (PPO) (Schulman et al., 2017), reflecting the common practice of using RL to train AV policies.

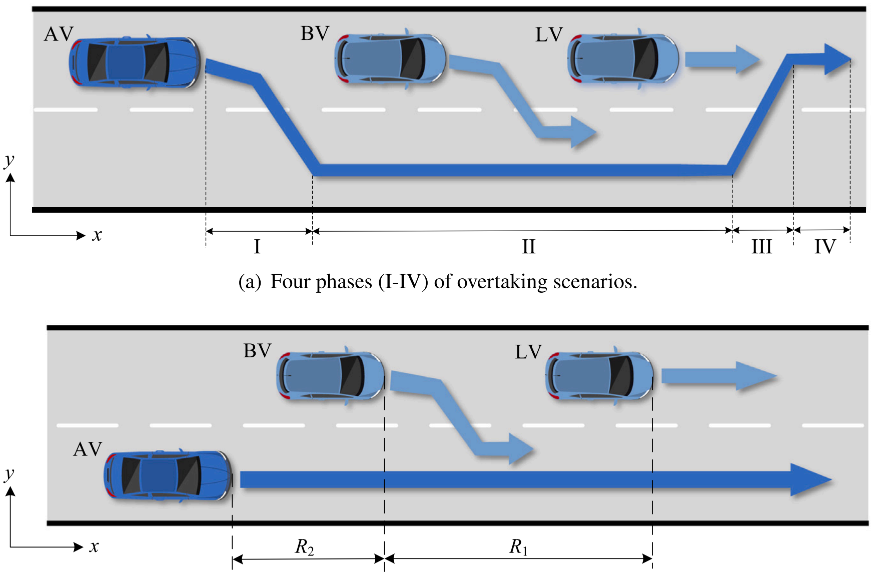
The IDM is a widely used car-following model that calculates the acceleration $a$ of the ego vehicle as

$$a = c_1\left[1 - \left(\frac{v}{c_2}\right)^{c_3} - \left(\frac{s(v, \Delta v)}{\Delta x - c_4}\right)^2\right], \tag{24}$$

where

$$s(v, \Delta v) = c_5 + c_6 v + \frac{v \Delta v}{2\sqrt{c_1 c_7}}. \tag{25}$$

[1] Link to source code: https://github.com/michigan-traffic-lab/Naturalistic-and-Adversarial-Driving-Environment.



(a) Four phases (I-IV) of overtaking scenarios.

(b) Passing phase of overtaking scenarios (focus of this paper).

**Fig. 4.** Illustrations of (a) the four phases of overtaking scenarios and (b) the passing phase (Phase II). In overtaking scenarios, the AV overtakes both BV and LV. While AV is engaged in passing, BV may overtake LV.

Here, $c_1, \ldots, c_7$ are constant parameters, $v$ is the ego vehicle's velocity, $\Delta v$ is the velocity difference with the leading vehicle, and $\Delta x$ is the distance to the leading vehicle. For AV-I, the constants are set to $c_1 = 2.5$, $c_2 = 18$, $c_3 = 4$, $c_4 = 4$, $c_5 = 2$, $c_6 = 1$, and $c_7 = 3$ (Ro et al., 2017). For AV-II, the constants are $c_1 = 5.948$, $c_2 = 28.31$, $c_3 = 16.79$, $c_4 = 4.5$, $c_5 = 1.42$, $c_6 = 1.72$, and $c_7 = 5.961$ (Sangster et al., 2013).

We use three archetypal SMs that capture a spectrum of driving behaviors:

(1) SM-I: the IDM (identical to AV-I), representing a neutral driving style;
(2) SM-II: the full velocity difference model (FVDM) (Jiang et al., 2001) with $a_{\min} = -1\ \text{m/s}^2$, representing an aggressive driving style;
(3) SM-III: the FVDM with $a_{\min} = -6\ \text{m/s}^2$, representing a conservative driving style.

The FVDM computes the acceleration $a$ of the ego vehicle as

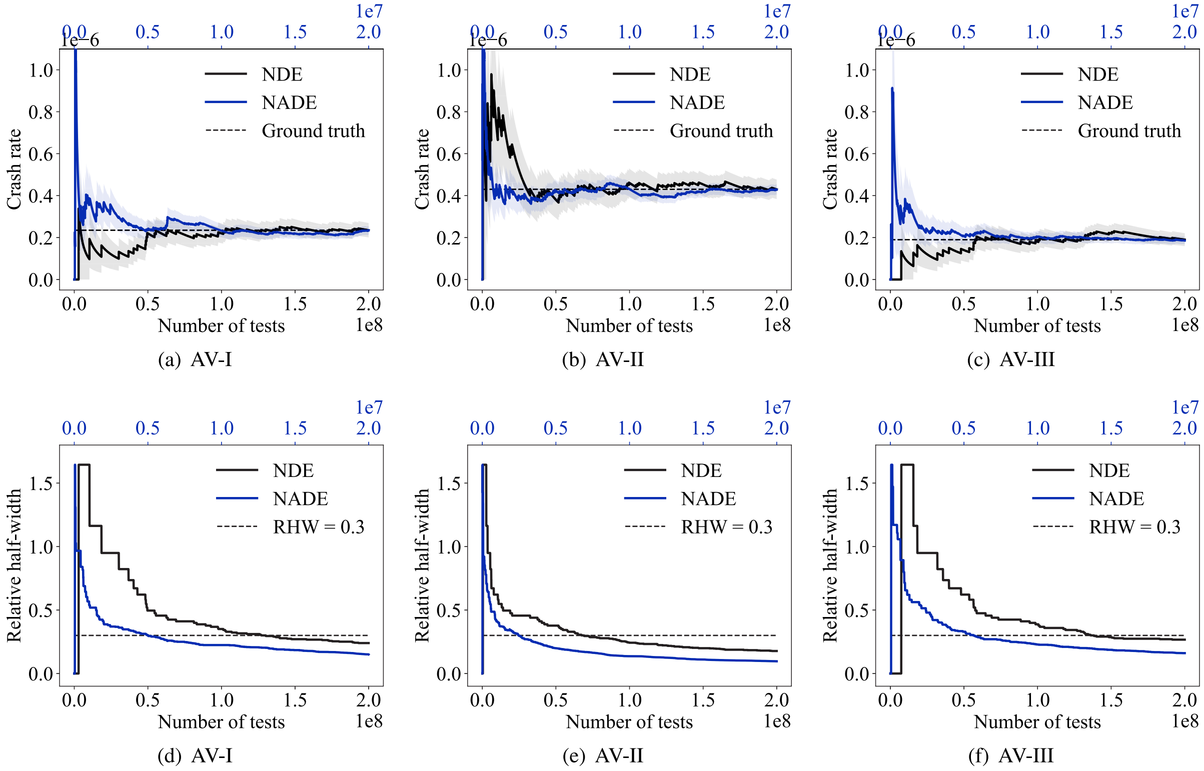
$$a = k_1[k_2 + k_3 \tanh(k_4(\Delta x - k_5) - k_6) - v]. \tag{26}$$

where $k_1 = 0.85$, $k_2 = 6.75$, $k_3 = 7.91$, $k_4 = 0.13$, $k_5 = 5$ and $k_6 = 1.57$ are constant parameters (Ro et al., 2017). The motivation for selecting these SMs is that most AV driving behaviors can be approximated as a weighted combination of neutral, aggressive, and conservative styles. By optimizing the combination coefficients of these SMs, we can construct a more efficient testing policy. All experiments are conducted using 50 threads on a computer equipped with an Intel® Xeon® Gold 5218R CPU, an NVIDIA® GeForce RTX™ 3090 GPU, and 256 GB RAM.

### *5.2. Testing and evaluation results in overtaking scenarios*

In this subsection, we analyze the testing and evaluation results of NDE, NADE and adaptive testing in overtaking scenarios. As illustrated in Fig. 4, we focus on the passing phase of overtaking scenarios, where a slower-moving lead vehicle (LV) travels ahead of the background vehicle (BV), and the AV is attempting to overtake both BV and LV. During this process, BV may also attempt to overtake LV, which could lead to a rear-end crash between AV and BV. The state of the overtaking scenario is defined as $\boldsymbol{s} := [v_{\text{BV}}, R_1, \dot{R}_1, R_2, \dot{R}_2]^\top$, where $R_1 := x_{\text{LV}} - x_{\text{BV}}$, $\dot{R}_1 := v_{\text{LV}} - v_{\text{BV}}$, $R_2 := x_{\text{BV}} - x_{\text{AV}}$, and $\dot{R}_2 := v_{\text{BV}} - v_{\text{AV}}$. The action is defined as the accelerations of LV and BV, $\boldsymbol{a} := [a_{\text{LV}}, a_{\text{BV}}]^\top$. Here, $x$, $v$ and $a$ refer to the longitudinal position, velocity, and acceleration, respectively, with the subscripts corresponding to each specific vehicle. The simulation runs for a maximum of 20 s with a time resolution of 0.1 s. Typically, overtaking scenarios involve more than 1,400 dimensions (201 time steps, each with 5 state variables and 2 action variables), presenting the high-dimensionality challenge.

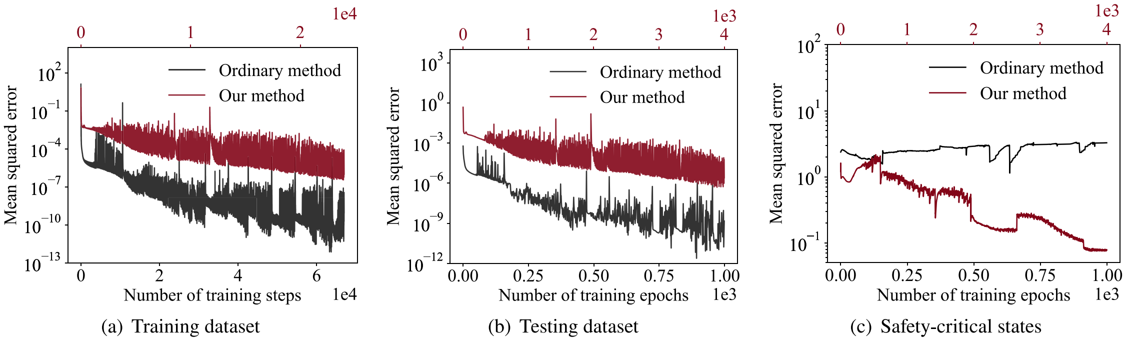
To enhance the robustness of crash rate estimation in NADE, we use three maneuver challenges, $Q_1$, $Q_2$, and $Q_3$, pre-trained with SM-I, SM-II, and SM-III, respectively, with equal combination coefficients to establish the importance function. Specifically, the importance function is given by Eqs. (6) and (7) with $(Q_1 + Q_2 + Q_3)/3$ in place of $Q$. Fig. 5 displays the crash rate estimations and



(a) AV-I (b) AV-II (c) AV-III

(d) AV-I (e) AV-II (f) AV-III

**Fig. 5.** The crash rate estimations for (a) AV-I, (b) AV-II, and (c) AV-III in NDE and NADE, and corresponding RHWs for (d) AV-I, (e) AV-II, and (f) AV-III in overtaking scenarios.

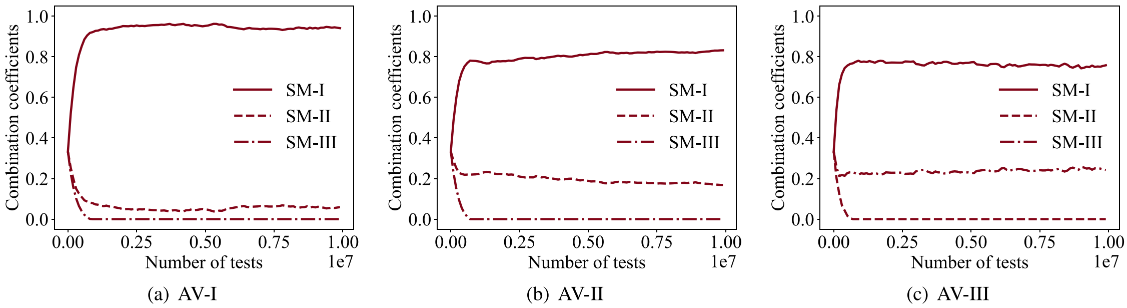


(a) Training dataset (b) Testing dataset (c) Safety-critical states

**Fig. 6.** The (a) MSE on the training dataset, (b) MSE on the testing dataset and (c) MSE on safety-critical states throughout the NeuDyM training process in overtaking scenarios. The black curves represent the MSEs from the ordinary learning method, which uses all dynamics data, while the red curves correspond to the MSEs from our method that learns with only safety-critical dynamics data.

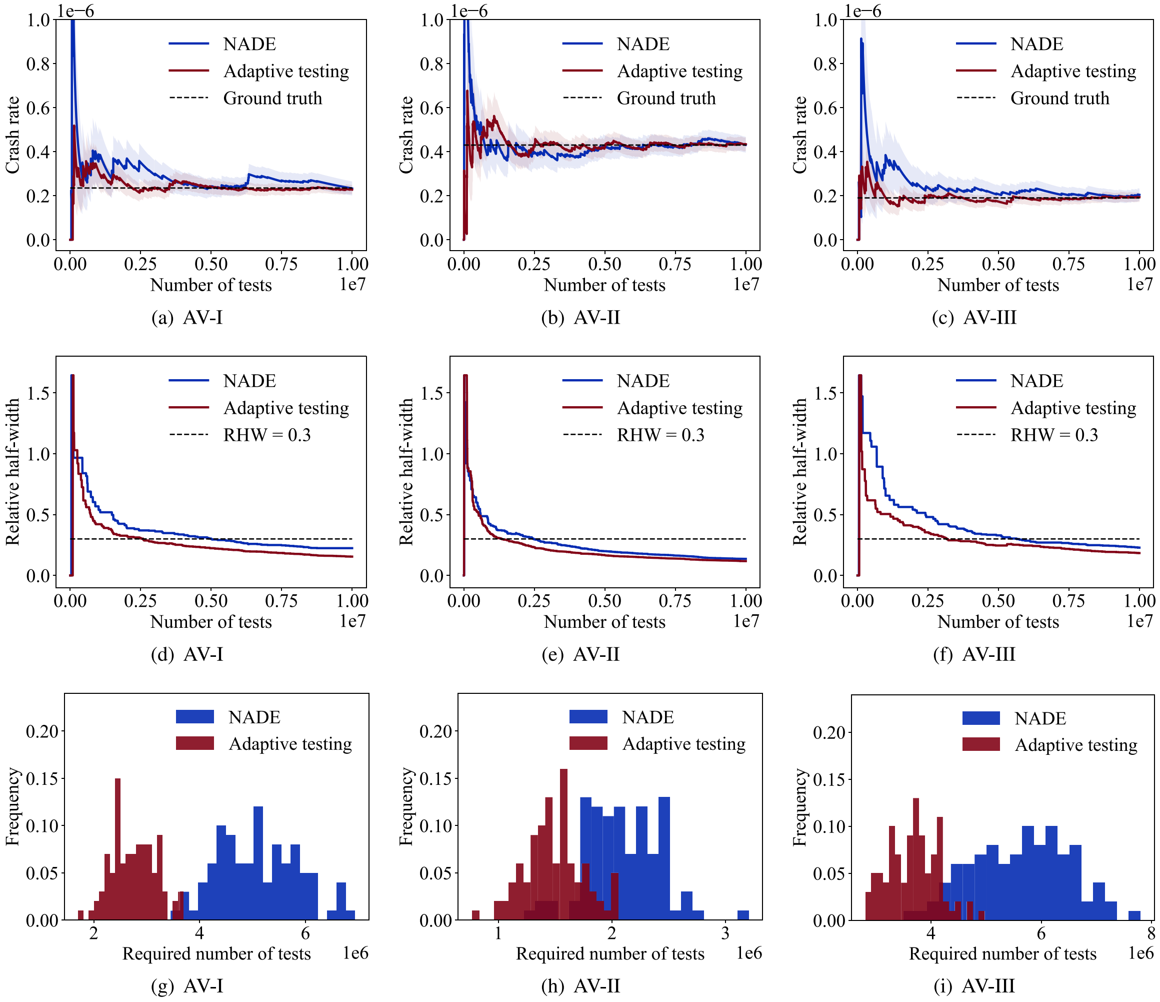
the corresponding RHWs for AV-I, AV-II, and AV-III in NDE and NADE. It can be seen that, across all three AVs, NADE converges to the same crash rate estimate as NDE, while requiring far fewer tests to reach the 0.3 RHW threshold. Although using multiple pre-trained maneuver challenges with average combination coefficients can enhance the evaluation robustness of NADE, it may reduce evaluation efficiency since such a configuration is not tailored for any specific AV under test.

To tackle this challenge, we continuously optimize the importance functions through the adaptive testing process. The number of tests is set to $n^{(k)} = 10^5$ for each update step $k = 1, 2, \dots$. The NeuDyM is implemented as a multilayer perceptron (MLP) with three hidden layers, each containing 256 neurons. We employ the Adam optimizer and the mean squared error (MSE) loss function, using the default hyperparameters provided in PyTorch 1.13.1 (Paszke et al., 2019), which are widely adopted in practice. Fig. 6 illustrates the MSEs in the NeuDyM training process. While the MSEs for both the training and testing datasets show a downward trend, only our learning method manages to reduce the MSE in safety-critical states, whereas the ordinary learning method fails to do so, with its MSE actually increasing. Based on the learned NeuDyM policies, the maneuver challenges are then learned using dense reinforcement learning, and the combination coefficients are optimized accordingly. Fig. 7(a)–(c) reveal that the combination coefficients are



(a) AV-I (b) AV-II (c) AV-III

**Fig. 7.** The combination coefficients optimized during adaptive testing for (a) AV-I, (b) AV-II, and (c) AV-III in overtaking scenarios.



(a) AV-I (b) AV-II (c) AV-III

(d) AV-I (e) AV-II (f) AV-III

(g) AV-I (h) AV-II (i) AV-III

**Fig. 8.** The crash rate estimations for (a) AV-I, (b) AV-II and (c) AV-III of NADE and adaptive testing, RHW of crash rate estimations for (d) AV-I, (e) AV-II and (f) AV-III, and frequency distributions of bootstrapped required number of tests for (g) AV-I, (h) AV-II and (i) AV-III in overtaking scenarios.

effectively optimized. Notably, at $10^7$ tests, the optimized coefficients for AV-I, AV-II, and AV-III are $\boldsymbol{\alpha}_{\text{AV-I}} = [0.94, 0.03, 0.03]^\top$ (with the ground truth $\boldsymbol{\alpha}^*_{\text{AV-I}} = [1, 0, 0]^\top$), $\boldsymbol{\alpha}_{\text{AV-II}} = [0.82, 0.17, 0.01]^\top$, and $\boldsymbol{\alpha}_{\text{AV-III}} = [0.64, 0.03, 0.33]^\top$, respectively. As the combination coefficients are optimized, the importance functions are updated, and the testing results obtained from these importance functions are aggregated through adaptive importance sampling.

To evaluate the performance of the adaptive testing method, we compare its results with those of NADE, as illustrated in Fig. 8. Fig. 8(a)–(c) show that adaptive testing produces the same crash rate estimates as NADE for all three AVs. However, as seen in Fig.

**Table 3**
Average required number of tests and average acceleration ratios for AV-I, AV-II and AV-III in overtaking scenarios.

| Methods | AV-I (AAR) | AV-II (AAR) | AV-III (AAR) |
|---|---|---|---|
| NDE | $1.32 \times 10^8$ | $7.20 \times 10^7$ | $1.59 \times 10^8$ |
| NADE | $5.11 \times 10^6$ (26) | $2.09 \times 10^6$ (34) | $5.63 \times 10^6$ (28) |
| Adaptive testing | $2.75 \times 10^6$ (48) | $1.49 \times 10^6$ (48) | $3.68 \times 10^6$ (43) |

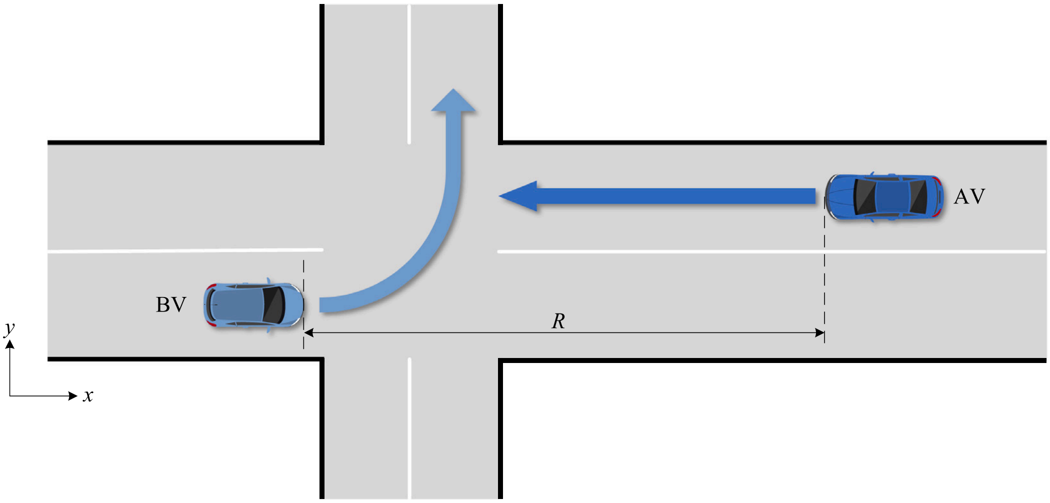


**Fig. 9.** Illustration of the unprotected left-turn scenarios.

8(d)–(f), adaptive testing requires fewer tests than NADE to reach the 0.3 RHW threshold. To mitigate experimental stochasticity, we bootstrap the testing results by shuffling them 100 times. The frequency distributions of the required number of tests are shown in Fig. 8(g)–(i), respectively. Table 3 presents the average required number of tests and the average acceleration ratios (AARs) for NDE, NADE, and adaptive testing across the three AVs, where AARs (given in parentheses) represent the ratio of the average number of tests required by NADE and adaptive testing compared to NDE. To evaluate the computational overhead of the adaptive testing method, we measure the average wall-clock time (AWT) needed by our approach to reach the required number of tests. This measurement excludes the time spent on the actual testing process, focusing solely on the overhead introduced by the adaptive mechanism. The resulting AWTs for AV-I, AV-II, and AV-III are 585.60 s, 429.88 s, and 1216.31 s, respectively. Compared with NADE, adaptive testing reduces the required number of tests by 46.17%, 29.01%, and 34.67% for AV-I, AV-II, and AV-III, respectively, demonstrating its significant improvement in evaluation efficiency while maintaining robustness for diverse AVs.

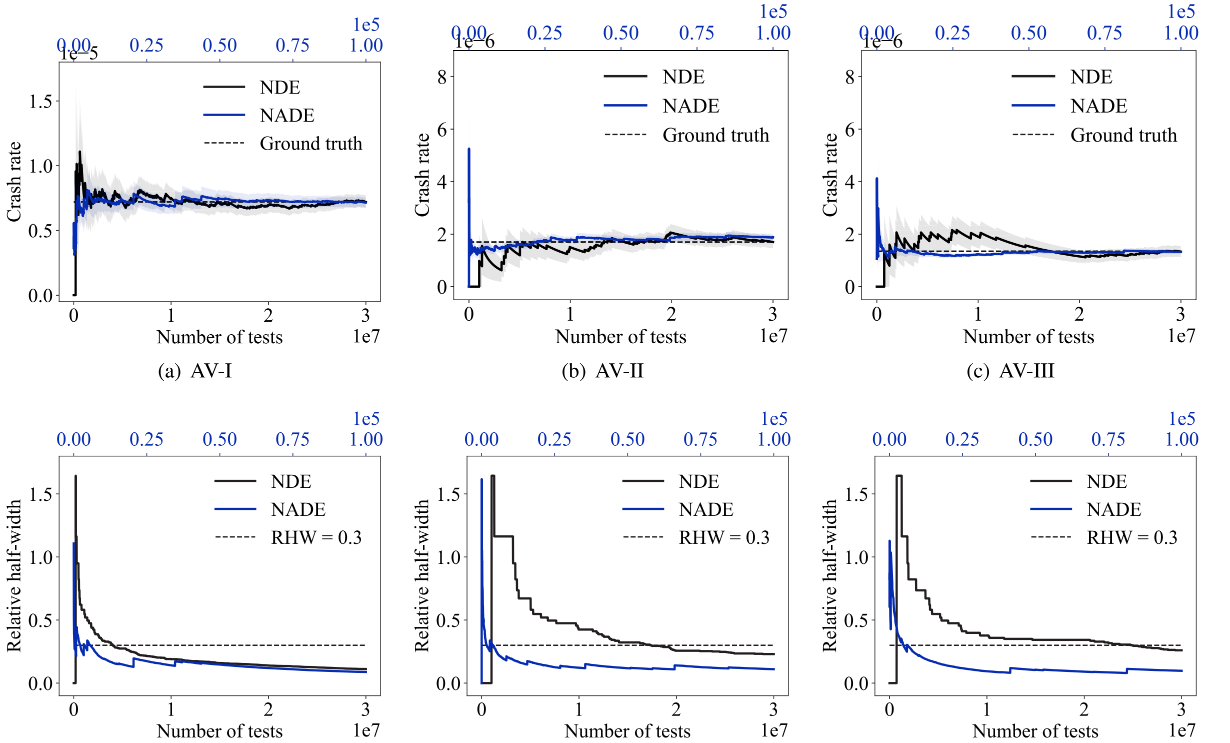
### 5.3. Testing and evaluation results in unprotected left-turn scenarios

In this subsection, we present the testing and evaluation results of NDE, NADE, and adaptive testing in unprotected left-turn scenarios. As illustrated in Fig. 9, these scenarios involve the BV attempting a left turn while an oncoming AV approaches from the opposite direction, potentially leading to a crash within the intersection area. The state of the unprotected left-turn scenario is defined as $s := [v_{\mathrm{BV}}, R, \dot{R}]^{\top}$, where $R := x_{\mathrm{AV}} - x_{\mathrm{BV}}$ represents the relative position between the AV and BV, and $\dot{R} := v_{\mathrm{AV}} - v_{\mathrm{BV}}$ denotes their relative speed. The action space consists of two discrete maneuvers available to the BV: wait ($a = 0$) or turn left ($a = 1$), i.e., $a \in \{0, 1\}$. To model the BV's decision-making process, we adopt the logit gap-acceptance model proposed by Madanat et al. (1994). Specifically, the probability of accepting a time gap $\Delta t := R/\dot{R}$ is given by

$$\mathbb{P}(\text{Accept } \Delta t) = \frac{1}{1 + \exp(c_1 - c_2 \Delta t)}, \tag{27}$$

where $c_1 = 5.212$ and $c_2 = 0.89934$ are constant parameters calibrated in Madanat et al. (1994). Each simulation runs for a maximum of 10 s with a time resolution of 0.1 s. Due to the temporal resolution and the state–action representation, unprotected left-turn scenarios typically involve more than 400 dimensions (101 time steps with 3 state variables and 1 action per step), posing a significant high-dimensionality challenge.

To assess the effectiveness of the adaptive testing method in unprotected left-turn scenarios, we compare its performance with that of NDE and NADE. Fig. 10 presents the crash rate estimations and the corresponding RHWs for AV-I, AV-II, and AV-III under both NDE and NADE. For all three AVs, NADE achieves the same crash rate estimates as NDE but with significantly fewer tests needed to reach the 0.3 RHW threshold. Furthermore, as illustrated in Fig. 11(a)–(c), the adaptive testing yields crash rate estimates identical



(a) AV-I (b) AV-II (c) AV-III

(d) AV-I (e) AV-II (f) AV-III

**Fig. 10.** The crash rate estimations for (a) AV-I, (b) AV-II, and (c) AV-III in NDE and NADE, and corresponding RHWs for (d) AV-I, (e) AV-II, and (f) AV-III in unprotected left-turn scenarios.

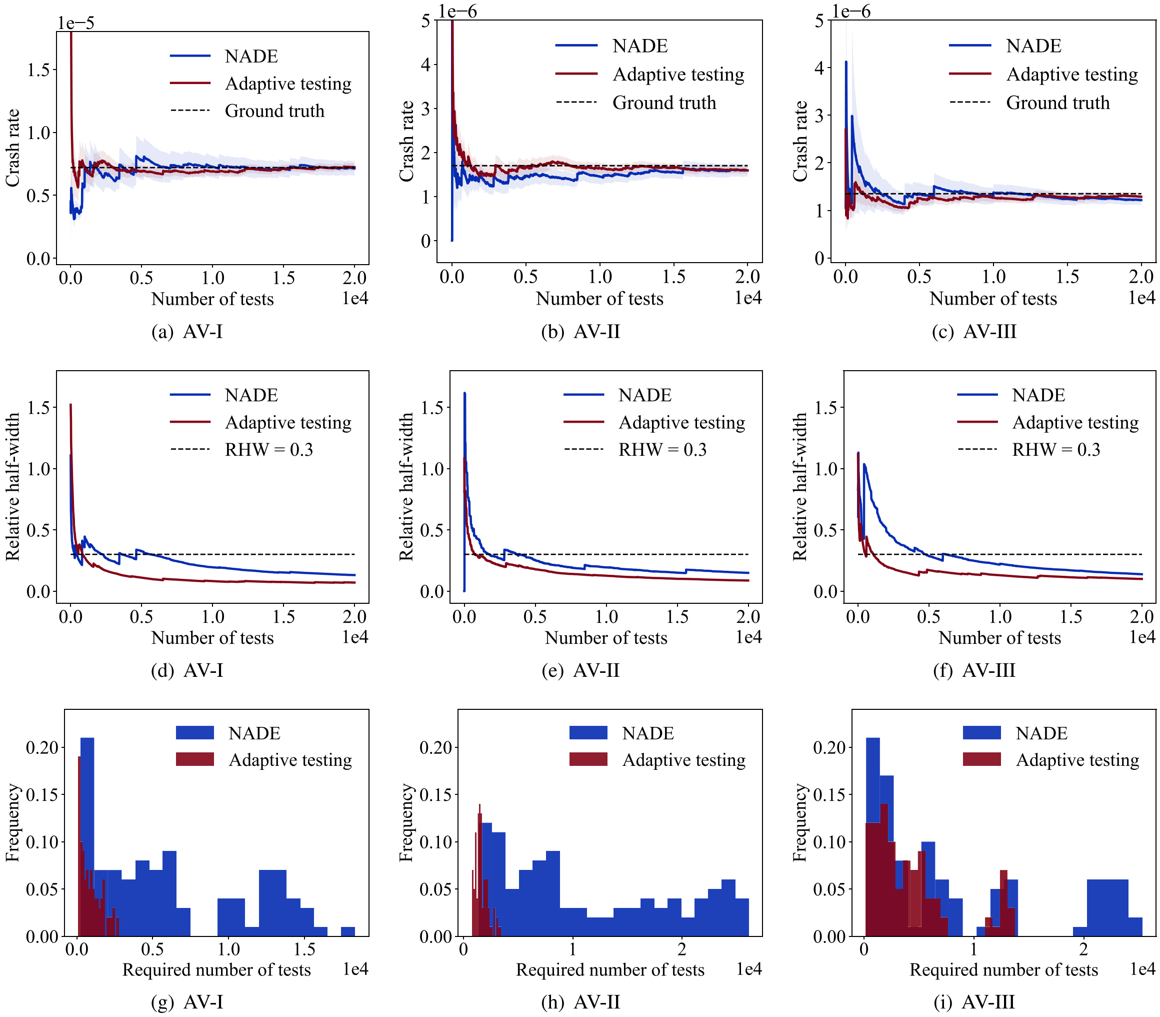
**Table 4**
Average required number of tests and average acceleration ratios for AV-I, AV-II and AV-III in unprotected left-turn scenarios.

| Methods | AV-I (AAR) | AV-II (AAR) | AV-III (AAR) |
|---|---|---|---|
| NDE | $4.41 \times 10^6$ | $1.82 \times 10^6$ | $2.27 \times 10^6$ |
| NADE | $6.22 \times 10^3$ (709) | $2.14 \times 10^3$ (8479) | $8.36 \times 10^3$ (2712) |
| Adaptive testing | $9.20 \times 10^2$ (4792) | $8.73 \times 10^2$ (20821) | $4.22 \times 10^3$ (5367) |

to those of NADE across all three AVs. However, Fig. 11(d)–(f) indicate that adaptive testing achieves the 0.3 RHW threshold with fewer test cases than NADE. The distributions of the required number of tests are illustrated in Fig. 11(g)–(i). Table 4 summarizes the average required number of tests and the average acceleration ratios for NDE, NADE, and adaptive testing across the three AVs. The AWTs needed to reach the average required number of tests—excluding the testing process itself—for AV-I, AV-II, and AV-III are 37.18 s, 42.23 s, and 166.24 s, respectively. Compared to NADE, adaptive testing reduces the number of required tests by 85.21%, 59.28%, and 49.48% for AV-I, AV-II, and AV-III, respectively, demonstrating its substantial gains in evaluation efficiency while preserving robustness across different AVs.

## 6. Conclusion

This paper proposes an adaptive testing framework that continuously optimizes importance functions during the large-scale testing process—a stage largely overlooked by prior work. Our method centers on learning NeuDyM policies from exclusively safety-critical dynamics data and then applying dense reinforcement learning to optimize maneuver challenges. To further enhance robustness and generalizability, we combine multiple pre-trained maneuver challenges and optimize their combination coefficients. We also employ adaptive importance sampling techniques to accurately aggregate testing results across varying importance functions. This work makes three primary contributions: (i) it addresses a critical gap in existing adaptive testing methods by focusing on large-scale testing, (ii) it introduces a novel integration of NeuDyM learning and dense reinforcement learning to overcome the curse of rarity, and (iii) it improves evaluation efficiency and accuracy through adaptive importance sampling and testing policy mixture optimization. Extensive experiments in overtaking and unprotected left-turn scenarios demonstrate that our method significantly outperforms baseline methods, including NDE and NADE, in both evaluation accuracy and sample efficiency.



(a) AV-I (b) AV-II (c) AV-III

(d) AV-I (e) AV-II (f) AV-III

(g) AV-I (h) AV-II (i) AV-III

**Fig. 11.** The crash rate estimations for (a) AV-I, (b) AV-II and (c) AV-III of NADE and adaptive testing, RHW of crash rate estimations for (d) AV-I, (e) AV-II and (f) AV-III, and frequency distributions of bootstrapped required number of tests for (g) AV-I, (h) AV-II and (i) AV-III in unprotected left-turn scenarios.

Future work will explore extending the framework to driving environments with continuous state and action spaces and integrating adaptive testing across all three stages (pre-testing, large-scale testing, and performance evaluation). Moreover, as all experiments in this study are conducted in simulation for practicality, we plan to apply our method to real-world AV testing in future research.

## CRediT authorship contribution statement

**Jingxuan Yang:** Writing – review & editing, Validation, Investigation, Visualization, Methodology, Conceptualization, Writing – original draft, Software, Formal analysis. **Zihang Wang:** Visualization, Software, Validation. **Daihan Wang:** Validation, Funding acquisition, Supervision. **Yi Zhang:** Supervision, Validation, Funding acquisition, Project administration. **Qiujing Lu:** Supervision, Validation, Writing – review & editing, Methodology. **Shuo Feng:** Validation, Methodology, Conceptualization, Writing – original draft, Project administration, Formal analysis, Writing – review & editing, Supervision, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is supported by National Natural Science Foundation of China No. 62473224.

## References

Andersen, M., Dahl, J., Vandenberghe, L., 2004. CVXOPT: Python software for convex optimization, version 1.3.2. Available At https://cvxopt.org.

Au, S.K., Beck, J., 2003. Important sampling in high dimensions. Struct. Saf. 25 (2), 139–163.

Bagschik, G., Menzel, T., Maurer, M., 2018. Ontology based scene creation for the development of automated vehicles. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1813–1820.

Bai, R., Yang, J., Gong, W., Zhang, Y., Lu, Q., Feng, S., 2024. Accurately predicting probabilities of safety-critical rare events for intelligent systems. In: 2024 IEEE 20th International Conference on Automation Science and Engineering. CASE, IEEE, pp. 3243–3249.

Bugallo, M.F., Elvira, V., Martino, L., Luengo, D., Miguez, J., Djuric, P.M., 2017. Adaptive importance sampling: The past, the present, and the future. IEEE Signal Process. Mag. 34 (4), 60–79.

Chelbi, N.E., Gingras, D., Sauvageau, C., 2022. Worst-case scenarios identification approach for the evaluation of advanced driver assistance systems in intelligent/autonomous vehicles under multiple conditions. J. Intell. Transp. Syst. 26 (3), 284–310.

Chen, B., Chen, X., Wu, Q., Li, L., 2021. Adversarial evaluation of autonomous vehicles in lane-change scenarios. IEEE Trans. Intell. Transp. Syst. 23 (8), 10333–10342.

Corso, A., Du, P., Driggs-Campbell, K., Kochenderfer, M.J., 2019. Adaptive stress testing with reward augmentation for autonomous vehicle validatio. In: 2019 IEEE Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 163–168.

Feng, S., Feng, Y., Sun, H., Bao, S., Zhang, Y., Liu, H.X., 2020a. Testing scenario library generation for connected and automated vehicles, part II: Case studies. IEEE Trans. Intell. Transp. Syst. 22 (9), 5635–5647.

Feng, S., Feng, Y., Sun, H., Zhang, Y., Liu, H.X., 2020b. Testing scenario library generation for connected and automated vehicles: An adaptive framework. IEEE Trans. Intell. Transp. Syst. 23 (2), 1213–1222.

Feng, S., Feng, Y., Yu, C., Zhang, Y., Liu, H.X., 2020c. Testing scenario library generation for connected and automated vehicles, part I: Methodology. IEEE Trans. Intell. Transp. Syst. 22 (3), 1573–1582.

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

Feng, S., Yan, X., Sun, H., Feng, Y., Liu, H.X., 2021. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. Nat. Commun. 12 (1), 1–14.

Gong, X., Feng, S., Pan, Y., 2023. An adaptive multi-fidelity sampling framework for safety analysis of connected and automated vehicles. IEEE Trans. Intell. Transp. Syst. 24 (12), 14393–14405.

He, H., Li, S., Yang, J., He, L., Lu, Q., Zhang, Y., Feng, S., 2024. KnowMoformer: Knowledge-conditioned motion transformer for controllable traffic scenario simulation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Data-Driven Autonomous Driving Simulation. IEEE, pp. 1–8.

Jiang, R., Wu, Q., Zhu, Z., 2001. Full velocity difference model for a car-following theory. Phys. Rev. E 64 (1), 017101.

Kalra, N., Paddock, S.M., 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? Transp. Res. Part A Policy Pr. 94, 182–193.

Klischat, M., Althoff, M., 2019. Generating critical test scenarios for automated vehicles with evolutionary algorithms. In: 2019 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 2352–2358.

Koren, M., Alsaif, S., Lee, R., Kochenderfer, M.J., 2018. Adaptive stress testing for autonomous vehicles. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1–7.

Li, A., Chen, S., Sun, L., Zheng, N., Tomizuka, M., Zhan, W., 2021. SceGene: Bio-inspired traffic scenario generation for autonomous driving testing. IEEE Trans. Intell. Transp. Syst. 23 (9), 14859–14874.

Li, S., He, H., Yang, J., Hu, J., Zhang, Y., Feng, S., 2024a. Few-shot testing of autonomous vehicles with scenario similarity learning. arXiv preprint arXiv:2409.14369.

Li, L., Huang, W.L., Liu, Y., Zheng, N.N., Wang, F.Y., 2016. Intelligence testing for autonomous vehicles: A new approach. IEEE Trans. Intell. Veh. 1 (2), 158–166.

Li, L., Lin, Y.L., Zheng, N.N., Wang, F.Y., Liu, Y., Cao, D., Wang, K., Huang, W.L., 2018. Artificial intelligence test: A case study of intelligent vehicles. Artif. Intell. Rev. 50 (3), 441–465.

Li, L., Wang, X., Wang, K., Lin, Y., Xin, J., Chen, L., Xu, L., Tian, B., Ai, Y., Wang, J., Cao, D., Liu, Y., Wang, C., Zheng, N., Wang, F.Y., 2019. Parallel testing of vehicle intelligence via virtual-real interaction. Sci. Robot. 4 (28), eaaw4106.

Li, S., Yang, J., He, H., Zhang, Y., Hu, J., Feng, S., 2024b. Few-shot scenario testing for autonomous vehicles based on neighborhood coverage and similarity. In: 2024 IEEE 35th Intelligent Vehicles Symposium. IV, IEEE, pp. 620–626.

Liu, H.X., Feng, S., 2024. Curse of rarity for autonomous vehicles. Nat. Commun. 15 (1), 4808.

Madanat, S.M., Cassidy, M.J., Wang, M.H., 1994. Probabilistic delay model at stop-controlled intersection. J. Transp. Eng. 120 (1), 21–36.

Menzel, T., Bagschik, G., Maurer, M., 2018. Scenarios for development, test and validation of automated vehicles. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1821–1827.

Mullins, G.E., Stankiewicz, P.G., Gupta, S.K., 2017. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles. In: 2017 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 1443–1450.

Mullins, G.E., Stankiewicz, P.G., Hawthorne, R.C., Gupta, S.K., 2018. Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles. J. Syst. Softw. 137, 197–215.

Nalic, D., Mihalj, T., Bäumler, M., Lehmann, M., Eichberger, A., Bernsteiner, S., 2020. Scenario based testing of automated driving systems: A literature survey. In: FISITA Web Congress. pp. 1–10.

Nonnengart, A., Klusch, M., Müller, C., 2020. CriSGen: Constraint-based generation of critical scenarios for autonomous vehicles. In: Formal Methods. FM 2019 International Workshops: Porto, Portugal, October 7–11, 2019, Revised Selected Papers, Part I 3. Springer, pp. 233–248.

Norden, J., O'Kelly, M., Sinha, A., 2019. Efficient black-box assessment of autonomous vehicle safety. arXiv preprint arXiv:1912.03618.

Oh, M.S., Berger, J.O., 1992. Adaptive importance sampling in Monte Carlo integration. J. Stat. Comput. Simul. 41 (3–4), 143–168.

O'Kelly, M., Sinha, A., Namkoong, H., Tedrake, R., Duchi, J.C., 2018. Scalable end-to-end autonomous vehicle testing via rare-event simulation. Adv. Neural Inf. Process. Syst. 31.

Owen, A.B., 2013. Monte Carlo Theory, Methods and Examples. Stanford.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 8024–8035.

Rempe, D., Philion, J., Guibas, L.J., Fidler, S., Litany, O., 2022. Generating useful accident-prone driving scenarios via a learned traffic prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17305–17315.

Ren, K., Yang, J., Lu, Q., Zhang, Y., Hu, J., Feng, S., 2025. Intelligent testing environment generation for autonomous vehicles with implicit distributions of traffic behaviors. Transp. Res. Part C Emerg. Technol. 174, 105106.

Ro, J.W., Roop, P.S., Malik, A., Ranjitkar, P., 2017. A formal approach for modeling and simulation of human car-following behavior. IEEE Trans. Intell. Transp. Syst. 19 (2), 639–648.

Rudin, W., 1987. Real and Complex Analysis. McGraw Hill.

Sangster, J., Rakha, H., Du, J., 2013. Application of naturalistic driving data to modeling of driver car-following behavior. Transp. Res. Rec. 2390 (1), 20–33.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Sinha, A., O'Kelly, M., Tedrake, R., Duchi, J.C., 2020. Neural bridge sampling for evaluating safety-critical autonomous systems. Adv. Neural Inf. Process. Syst. 33, 6402–6416.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical Bayesian optimization of machine learning algorithms. In: Neural Information Processing Systems. pp. 2951–2959.

Sun, H., Feng, S., Yan, X., Liu, H.X., 2021a. Corner case generation and analysis for safety assessment of autonomous vehicles. Transp. Res. Rec. 2675 (11), 587–600.

Sun, J., Zhou, H., Xi, H., Zhang, H., Tian, Y., 2021b. Adaptive design of experiments for safety evaluation of automated vehicles. IEEE Trans. Intell. Transp. Syst. 23 (9), 14497–14508.

Tian, Y., Pei, K., Jana, S., Ray, B., 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th International Conference on Software Engineering. pp. 303–314.

Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E 62 (2), 1805.

Tuncali, C.E., Fainekos, G., Prokhorov, D., Ito, H., Kapinski, J., 2019. Requirements-driven test generation for autonomous vehicles with machine learning components. IEEE Trans. Intell. Veh. 5 (2), 265–280.

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., Maurer, M., 2015. Defining and substantiating the terms scene, situation, and scenario for automated driving. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. IEEE, pp. 982–988.

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

Wang, X., Zhang, S., Peng, H., 2022. Comprehensive safety evaluation of highly automated vehicles at the roundabout scenario. IEEE Trans. Intell. Transp. Syst. 23 (11), 20873–20888.

Weng, B., Capito, L., Ozguner, U., Redmill, K., 2021. Towards guaranteed safety assurance of automated driving systems with scenario sampling: An invariant set perspective. IEEE Trans. Intell. Veh. 7 (3), 638–651.

Weng, B., Rao, S.J., Deosthale, E., Schnelle, S., Barickman, F., 2020. Model predictive instantaneous safety metric for evaluation of automated driving systems. In: 2020 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1899–1906.

Yan, X., Zou, Z., Feng, S., Zhu, H., Sun, H., Liu, H.X., 2023. Learning naturalistic driving environment with statistical realism. Nat. Commun. 14 (1), 2037.

Yang, J., Bai, R., Ji, H., Zhang, Y., Hu, J., Feng, S., 2025. Adaptive testing environment generation for connected and automated vehicles with dense reinforcement learning. IEEE Trans. Intell. Transp. Syst. 26 (4), 5135–5145.

Yang, J., He, H., Zhang, Y., Feng, S., Liu, H.X., 2022. Adaptive testing for connected and automated vehicles with sparse control variates in overtaking scenarios. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2791–2797.

Yang, J., Sun, H., He, H., Zhang, Y., Liu, H.X., Feng, S., 2023. Adaptive safety evaluation for connected and automated vehicles with sparse control variates. IEEE Trans. Intell. Transp. Syst. 25 (2), 1761–1773.

Zhang, S., Peng, H., Zhao, D., Tseng, H.E., 2018. Accelerated evaluation of autonomous vehicles in the lane change scenario based on subset simulation technique. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 3935–3940.

Zhao, D., Huang, X., Peng, H., Lam, H., LeBlanc, D.J., 2017. Accelerated evaluation of automated vehicles in car-following maneuvers. IEEE Trans. Intell. Transp. Syst. 19 (3), 733–744.

Zhao, D., Lam, H., Peng, H., Bao, S., LeBlanc, D.J., Nobukawa, K., Pan, C.S., 2016. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. IEEE Trans. Intell. Transp. Syst. 18 (3), 595–607.

Zhou, J., Wang, L., Wang, X., 2023. Online adaptive generation of critical boundary scenarios for evaluation of autonomous vehicles. IEEE Trans. Intell. Transp. Syst. 24 (6), 6372–6388.

Zhou, J., Wang, L., Wang, X., Meng, Q., 2024. Adaptive deep reinforcement learning for critical boundary scenario generation. In: 2024 43rd Chinese Control Conference. CCC, IEEE, pp. 6615–6620.