Intelligent Safety Testing of Autonomous Vehicles With Implicit Traffic Distributions by Dense Reinforcement Learning

Kun Ren, Yizhou Xu, Jingxuan Yang, Yi Zhang, *Senior Member, IEEE*, Jianming Hu, *Senior Member, IEEE*, and Shuo Feng, *Member, IEEE*

***Abstract*—Reliable safety evaluation is essential for the development and large-scale deployment of autonomous vehicles (AVs). However, the complexity of high-dimensional traffic environments and the rarity of safety-critical events make direct testing extremely inefficient and costly, requiring an enormous number of simulation miles. Importance sampling has been employed to accelerate this process, but most existing methods rely on known or explicit traffic behavior distributions, which are often inaccessible in practice. To address this limitation, we propose Deep-IIS, which extends implicit importance sampling by utilizing rejection sampling to handle unknown distributions without requiring explicit density functions. By integrating dense deep reinforcement learning to adaptively optimize key sampling parameters, Deep-IIS intelligently concentrates sampling on high-risk regions, significantly improving testing efficiency while maintaining evaluation reliability. We validate the proposed method in two representative traffic scenarios, demonstrating its superior ability to uncover safety-critical cases and to produce accurate performance estimates under implicit distributions.**

***Index Terms*—Autonomous vehicles, importance sampling, dense reinforcement learning.**

I. Introduction

AUTONOMOUS vehicles (AVs) are envisioned to play a central role in the future of transportation, offering benefits such as enhanced safety, improved traffic flow, and reduced human error. However, validating the safety and reliability of AV systems remains a significant challenge. This difficulty stems from the high dimensionality, complexity, and stochastic nature of real-world driving environments, where traffic behavior arises from numerous uncertain and interacting factors. These factors contribute not only to the "curse of dimensionality" [1], but also to the difficulty of explicitly modeling the distribution of driving behaviors. Furthermore, the closed box nature of AV decision-making models limits their ability to handle out-of-distribution scenarios that deviate from their training data [2]. Of particular concern are safety-critical events [3], which, despite their rarity, have a disproportionately large impact on system safety and public trust [4], [5].

The most common evaluation frameworks, such as deploying AVs in naturalistic driving environments (NDEs), rely on collecting vast amounts of real or simulated driving data [1], [3], [6], [7], [8], [9]. Although these environments capture realistic traffic behaviors, they offer limited exposure to rare but critical events. AVs must drive billions of miles to encounter enough such critical scenarios. Reliable evaluation requires large-scale NDE testing [10], which is usually prohibitively inefficient.

Many recent researches have been made to improve the efficiency of AV testing and get unbiased estimates of safety metrics [1], [11], [12], [13], [14], [15], [16]. It has shown that importance sampling (IS) methods [17], [18], [19] can significantly improve testing efficiency by shifting the sampling focus toward rare, safety-relevant cases. By adjusting the scenario distribution while compensating with importance weights, IS allows for unbiased estimation with substantially fewer test runs. These methods can be applied to continuous naturalistic driving environments (NDEs), enabling fair comparisons between AVs and human drivers under the same distributional assumptions. Several studies have demonstrated the effectiveness of IS in traffic safety analysis, including applications in autonomous vehicle validation [1], [11], [20], [21], [22], [23]. In particular, intelligent testing environments [1] leverage IS to construct naturalistic and adversarial driving environments (NADEs), producing unbiased yet more efficient estimates compared to conventional NDE-based testing. Further improvements have introduced reinforcement learning to optimize the proposal distribution, effectively reducing variance and increasing test coverage [11].

However, one of the key limitations of the IS method is its reliance on an explicit distribution of the NDE, which is essential for modeling the proposal distribution and calculating

Received 16 August 2025; revised 3 February 2026 and 13 May 2026; accepted 16 June 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62473224 and Grant 62333015; in part by Beijing Natural Science Foundation under Grant 4244092, Grant L231014, and Grant L243025; and in part by Beijing Nova Program under Grant 20230484259 and Grant 20240484642. The Associate Editor for this article was B. Chen. *(Corresponding authors: Jianming Hu; Shuo Feng.)*

Kun Ren, Yizhou Xu, Jingxuan Yang, and Jianming Hu are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: rk23@mails.tsinghua.edu.cn; xu-yz24@mails.tsinghua.edu.cn; yangjx20@mails.tsinghua.edu.cn; hujm@mail.tsinghua.edu.cn).

Yi Zhang is with the Department of Automation, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China, also with the Tsinghua-Berkeley Shenzhen Institute (TBSI), Shenzhen 518055, China, and also with Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing 210096, China (e-mail: zhyi@mail.tsinghua.edu.cn).

Shuo Feng is with the Department of Automation, Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China (e-mail: fshuo@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2026.3707517

the importance weights. For example, some studies use probabilistic modeling methods to represent NDEs, particularly in simple scenarios [1], [20], [21], [22], [23]. In practice, however, this distribution can be difficult to obtain in complex traffic scenarios. To address such challenges, neural networks are increasingly used to model NDEs [6], [24], [25], [26]. While neural networks can handle the complexity of such environments, they do not always yield an explicit distribution. For instance, when a neural network models an NDE, it may output the mean and variance of traffic samples, allowing for the explicit calculation of the probability density. However, when the model combines neural networks with rule-based systems, the resulting distribution becomes even harder to define explicitly. Furthermore, when the model is treated as a closed box, we have no direct access to the underlying probability density values. In these cases, the distribution is considered implicit. This means that instead of directly modeling the probability density function of traffic behaviors, we only have access to sample outputs generated by NDE, without a clear functional form or value of the underlying probability density. When the distribution is implicit, traditional importance sampling methods become less effective and more difficult to apply.

Our previous work—implicit importance sampling (IIS)—attacks this challenge by extending the IS framework to settings where the underlying probability distribution of the environment is unknown or implicit [27]. IIS leverages accept-reject sampling to construct an unnormalized proposal distribution that biases the sampling towards higher-risk scenarios. This method allows us to sample effectively even when the distribution is not explicitly available, making it suitable for complex NDEs where traditional IS methods would struggle. However, the performance of IIS heavily relies on manually tuned parameters—such as K_1, which increases the sampling bias towards high-risk states, and K_2, which controls the magnitude of the estimation bias—to balance the trade-off between efficiency and accuracy. The manual tuning process is difficult to achieve optimal performance. Moreover, these parameters are fixed across all states, whereas achieving theoretically optimal testing results often requires adapting them dynamically according to every sampling state. This lack of adaptivity limits the effectiveness and reliability of IIS in complex environments.

To overcome these limitations and further enhance sampling efficiency and accuracy, we propose Deep-IIS, a deep reinforcement learning-based extension of IIS. In this framework, the key parameters of IIS are modelled as a state-dependent policy network, allowing the proposal distribution to be dynamically adjusted according to observed traffic states. The reward function of reinforcement learning is carefully designed and theoretically analyzed to encourage exploration of high-risk scenarios, thereby reducing testing variance, improving sampling efficiency, and mitigating estimation bias. By learning state-dependent parameters through reinforcement learning, Deep-IIS eliminates the need for manually tuned, fixed parameters, enabling more optimal and adaptive testing results. Extensive experiments demonstrate that Deep-IIS preserves the theoretical advantages of IIS while achieving superior trade-offs between efficiency and estimation accuracy compared to baseline IIS methods.

Figure 1 provides a schematic illustration of the relationship between the proposed Deep-IIS method and preliminary IIS method.

This paper presents the following contributions:

- We propose Deep-IIS, a deep reinforcement learning-enhanced extension of implicit importance sampling, which dynamically adjusts the proposal distribution based on observed traffic states in driving environments with implicit traffic behavior distributions.
- We theoretically analyze the reward function, showing that the proposed design promotes exploration of high-risk scenarios while controlling estimation variance and bias.
- We validate the effectiveness of Deep-IIS through experiments on two NDE models, demonstrating improved efficiency and accuracy over baseline IIS methods.

II. Preliminary Work

This section consists of four parts. Section II-A introduces the concept of naturalistic driving environment (NDE) and the standard crash rate estimation. Section II-B reviews naturalistic and adversarial driving environment (NADE), which improves sampling efficiency using importance sampling. Section II-C introduces our previously proposed IIS method that addresses the limitation of implicit traffic behavior distributions. Finally, Section II-D analyzes the estimation bias of IIS and motivates the need for adaptive strategies to adjust the sampling parameters.

A. Naturalistic Driving Environment (NDE)

A testing scenario is modeled as a Markov Decision Process (MDP) [28]:

$$\mathbf{s}(0) \rightarrow \mathbf{u}(0) \rightarrow \mathbf{s}(1) \rightarrow \mathbf{u}(1) \cdots \rightarrow \mathbf{u}(T-1) \rightarrow \mathbf{s}(T), \quad (1)$$

where $\mathbf{s}(j)$ and $\mathbf{u}(j)$ represent the state and action of all vehicles at time step j, including the autonomous vehicle (AV) and background vehicles (BVs).

We denote the NDE as $\mathbf{x} = (\mathbf{s}(0), \mathbf{u}(0), \mathbf{s}(1), \mathbf{u}(1), \cdots, \mathbf{u}(T-1), \mathbf{s}(T))$, and its probability distribution as $P(\mathbf{x})$:

$$P(\mathbf{x}) = P(\mathbf{s}(0)) \prod_{k=0}^{T} P(\mathbf{u}(k) | \mathbf{s}(k)). \quad (2)$$

To evaluate the performance of AV under NDE, the crash rate is defined as

$$P(A) = \sum_{\mathbf{x} \in \mathbf{X}} P(A|\mathbf{x}) P(\mathbf{x}) = \mathbb{E}_P [P(A|\mathbf{x})], \quad (3)$$

where A represents the accident event that involves AV during the testing scenarios. For every scenario $\mathbf{x} \in \mathbf{X}$, the occurrence of an accident is deterministic. Thus, the conditional probability $P(A|\mathbf{x})$ can be replaced by the indicator function $f(\mathbf{x})$:

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if event } A \text{ occurs} \\ 0, & \text{otherwise} \end{cases}. \quad (4)$$

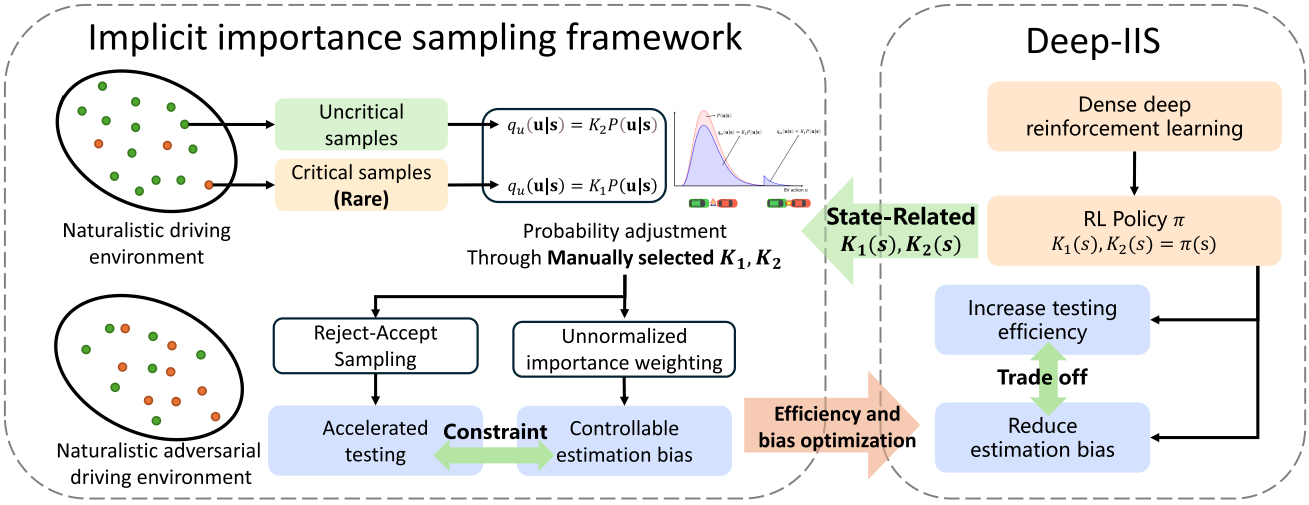


Fig. 1. The illustration of the relationship between deep-IIS and IIS.

For simplicity, we use μ to denote the crash rate, i.e., $\mu = P(A)$:

$$\mu = \mathbb{E}_P\left[f(\mathbf{x})\right]. \tag{5}$$

According to the Crude Monte Carlo (CMC) method [29], the estimated crash rate can be estimated by

$$\hat{\mu}_P = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i), \mathbf{x}_i \sim P(\mathbf{x}), \tag{6}$$

where n is the number of tests, and $\mathbf{x}_i$ is sampled from naturalistic distribution $P(\mathbf{x})$.

B. Naturalistic and Adversarial Driving Environment (NADE)

Due to the low accident rate in NDE, the estimated crash rate μ is very small, which leads to a large variance of the estimated result. To address this issue, naturalistic and adversarial driving environment (NADE) [1] was proposed, which incorporates importance sampling to estimate crash rate more efficiently. Specifically, a proposal distribution $q(\mathbf{x})$ is designed to replace the nominal distribution $P(\mathbf{x})$, so that high-risk scenarios can be sampled more frequently. Here the nominal behavior distribution $P(\mathbf{u}|\mathbf{s})$ is assumed to be known and tractable, so that the modified proposal $q(\mathbf{x})$ can be explicitly constructed.

Similar to the definition of nominal distribution $P(\mathbf{x})$, a proposal function is designed as:

$$q(\mathbf{x}) = q(\mathbf{s}(0))\prod_{k=0}^{T} q(\mathbf{u}(k)|\mathbf{s}(k))\Big), \tag{7}$$

where $q(\mathbf{s}(0)) = P(\mathbf{s}(0))$.

The key is to design the distribution $q(\mathbf{u}(k)|\mathbf{s}(k))$ to increase the likelihood of sampling scenarios $\mathbf{x}$ that may lead to accidents. The criticality of action $\mathbf{u}$ under the state $\mathbf{s}$ is represented as $P(A|\mathbf{u},\mathbf{s})$. And the criticality of state $\mathbf{s}$ is defined as [30], [31]:

$$C(\mathbf{s},\mathbf{u}) = P(A|\mathbf{u},\mathbf{s})P(\mathbf{u}|\mathbf{s}). \tag{8}$$

If $C(\mathbf{s},\mathbf{u})$ is non-zero, the probability of corresponding critical action $\mathbf{u}$ is increased, namely higher $q(\mathbf{u}|\mathbf{s})$.

Under NADE, μ can be estimated more efficiently:

$$\begin{aligned}\hat{\mu}_q &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\frac{P(\mathbf{x}_i)}{q(\mathbf{x}_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)w(\mathbf{x}_i), \mathbf{x}_i \sim q(\mathbf{x}),\end{aligned} \tag{9}$$

where $\mathbf{x}_i \sim q(\mathbf{x})$ means that $\mathbf{x}_i$ is sampled from distribution $q(\mathbf{x})$, and $w(\mathbf{x}_i)$ is the importance weight with sample $\mathbf{x}_i$.

While increasing the probability of sampling critical events, the estimation result of NADE has been demonstrated that is unbiased:

$$\begin{aligned}\mathbb{E}_q\left(\frac{1}{n}\sum_{i=1}^{n} P(A|\mathbf{x}_i)\frac{P(\mathbf{x}_i)}{q(\mathbf{x}_i)}\right) &= \mathbb{E}_q\left(P(A|\mathbf{x})\frac{P(\mathbf{x})}{q(\mathbf{x})}\right) \\ &= \mathbb{E}_P\left(P(A|\mathbf{x})\right).\end{aligned} \tag{10}$$

C. Implicit Importance Sampling (IIS)

While NADE provides a structured way to construct testing environments based on explicitly defined traffic behavior distributions, it becomes ineffective when such distributions are unavailable in closed form. In many realistic autonomous driving scenarios, the distribution $P(\mathbf{x})$ is only implicitly defined — we can draw samples from it but cannot evaluate its density function directly. This reflects real-world settings where agent behaviors are observed through simulation or data collection, without a tractable analytical model.

To address this challenge, our preliminary work [27] proposes an implicit importance sampling (IIS) method. It constructs an unnormalized proposal distribution $q_u(\mathbf{u}|\mathbf{s})$ to favor critical driving actions:

$$\begin{aligned}q(\mathbf{u}|\mathbf{s}) &= \frac{q_u(\mathbf{u}|\mathbf{s})}{\int q_u(\mathbf{u}|\mathbf{s})\mathrm{d}\mathbf{u}}, \\ q_u(\mathbf{u}|\mathbf{s}) &= K(\mathbf{u},\mathbf{s})P(\mathbf{u}|\mathbf{s}),\end{aligned} \tag{11}$$

where $K(\mathbf{u},\mathbf{s})$ adjusts the sampling weight based on criticality. The function is defined as:

$$K(\mathbf{u},\mathbf{s}) = \begin{cases} K_0 = 1, & \text{if } C(\mathbf{s}) = 0 \\ K_1 > 1, & \text{if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) > 0 \\ K_2 < 1, & \text{if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) = 0 \end{cases}. \quad (12)$$

Here, $C(\mathbf{s})$ denotes the criticality of state $\mathbf{s}$, determined through trajectory prediction. Specifically, we generate multiple samples of actions $\mathbf{u}$ at a given state $\mathbf{s}$ to simulate possible future outcomes. If any predicted outcome results in a collision, the state is labeled as critical ($C(\mathbf{s}) > 0$), and the corresponding action $\mathbf{u}$ is also marked as critical ($P(A|\mathbf{u},\mathbf{s}) > 0$).

As shown in Eq. (12), K_1 increases the probability of sampling critical actions, while K_2 decreases the probability of sampling non-critical actions. This design effectively increase the testing efficiency for rare but safety-relevant events.

Since $q_u(\mathbf{u}|\mathbf{s})$ is unnormalized and cannot be directly sampled, we instead apply accept-reject sampling technique [32] to generate valid samples:

$$\begin{aligned} P_{\text{acc}}(\mathbf{u}|\mathbf{s}) &= \frac{q_u(\mathbf{u}|\mathbf{s})}{K_1 P(\mathbf{u}|\mathbf{s})} = \frac{K(\mathbf{u}|\mathbf{s})}{K_1} \\ &= \begin{cases} 1, & \text{if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) > 0 \\ \dfrac{K_2}{K_1}, & \text{if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) = 0 \end{cases}. \end{aligned} \quad (13)$$

Accordingly, the crash rate μ can be estimated under IIS using importance sampling:

$$\begin{aligned} \hat{\mu}_{q_u} &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\frac{P(\mathbf{x}_i)}{q_u(\mathbf{x}_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i) w_u(\mathbf{x}_i), \mathbf{x}_i \sim q_u(\mathbf{x}). \end{aligned} \quad (14)$$

D. Bias Analysis of IIS-Testing

Since the proposal distribution $q_u(\mathbf{x})$ is unnormalized, the estimator $\hat{\mu}_{q_u}$ in Eq. (14) is generally biased. This bias arises because the unnormalized importance weights $w_u(\mathbf{x}) = \frac{P(\mathbf{x})}{q_u(\mathbf{x})}$ do not account for the unknown normalization constant of $q_u(\mathbf{x})$. Let this normalization constant be denoted as $c(\mathbf{x})$, so that the normalized proposal becomes $q(\mathbf{x}) = \frac{q_u(\mathbf{x})}{c(\mathbf{x})}$.

Accordingly, the true importance weight should be:

$$w(\mathbf{x}) = \frac{P(\mathbf{x})}{q(\mathbf{x})} = \frac{P(\mathbf{x})}{q_u(\mathbf{x})} \cdot c(\mathbf{x}) = w_u(\mathbf{x}) \cdot c(\mathbf{x}). \quad (15)$$

The estimator $\hat{\mu}_{q_u}$ thus underestimates or overestimates the true value μ depending on the value of $c(\mathbf{x})$.

To quantify this deviation, we analyze bounds on $c(\mathbf{x})$ based on the design of $q_u(\mathbf{x})$.

Note that at non-critical time steps, we have $K(\mathbf{u},\mathbf{s}) = 1$, and hence $q_u(\mathbf{u}|\mathbf{s}) = P(\mathbf{u}|\mathbf{s})$. In such cases, $q_u(\mathbf{u}|\mathbf{s})$ is normalized and the correction coefficient of $q_u(\mathbf{u}|\mathbf{s})$ is 1. Therefore, the deviation arises only from the critical time steps. Let $T_{i,C}$ denote the set of critical time steps in trajectory $\mathbf{x}_i$. Then the overall correction coefficient can be expressed as a product over these steps:

$$c(\mathbf{x}_i) = \prod_{k \in T_{i,C}} c_{ik}, \quad (16)$$

where c_{ik} denotes the correction coefficient at time step k in trajectory $\mathbf{x}_i$.

To obtain bounds on $c(\mathbf{x}_i)$, we define:

$$\begin{aligned} c_{\min}(\mathbf{x}_i) &\overset{\text{def}}{=} (\min c_{ik})^{|T_{i,C}|}, \\ c_{\max}(\mathbf{x}_i) &\overset{\text{def}}{=} (\max c_{ik})^{|T_{i,C}|}, \end{aligned} \quad (17)$$

where $|T_{i,C}|$ denotes the number of critical steps.

Each c_{ik} is linearly dependent on a term $H_1(\mathbf{s}(k))$, which reflects the probability mass of critical actions under $P(\mathbf{u}|\mathbf{s}(k))$. Let the minimum and maximum of this term over all encountered states be:

$$\begin{aligned} H_1^{\min} &\overset{\text{def}}{=} \min_{\mathbf{s}(k)} H_1(\mathbf{s}(k)), \\ H_1^{\max} &\overset{\text{def}}{=} \max_{\mathbf{s}(k)} H_1(\mathbf{s}(k)). \end{aligned} \quad (18)$$

Then the corresponding bounds for c_{ik} are:

$$\begin{aligned} \min c_{ik} &= (K_1 - 1) \cdot H_1^{\min} + K_2, \\ \max c_{ik} &= (K_1 - 1) \cdot H_1^{\max} + K_2. \end{aligned} \quad (19)$$

Here, $H_1^{\min}$ and $H_1^{\max}$ are estimated empirically by sampling actions from $P(\mathbf{u}|\mathbf{s}(k))$ and evaluating the portion of probability mass that lies in the critical action region.

With these bounds, we can estimate lower and upper limits of the crash rate $\hat{\mu}_{q_u}$ using:

$$\begin{aligned} \hat{\mu}_{q_u,\min} &\overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}) w_u(\mathbf{x}_i) c_{\min}(\mathbf{x}_i), \\ \hat{\mu}_{q_u,\max} &\overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}) w_u(\mathbf{x}_i) c_{\max}(\mathbf{x}_i). \end{aligned} \quad (20)$$

These estimates define confidence bounds for $P(A)$:

$$\begin{aligned} P(A) &\geq \hat{\mu}_{q_u,\min} - \lambda \cdot \text{std}(\hat{\mu}_{q_u,\min}), \\ P(A) &\leq \hat{\mu}_{q_u,\max} + \lambda \cdot \text{std}(\hat{\mu}_{q_u,\max}). \end{aligned} \quad (21)$$

These bounds provide a confidence interval for the estimation of $P(A)$, where λ determines the confidence level, and std$(\cdot)$ denotes the standard deviation.

The performance of IIS depends on two parameters, K_1 and K_2, which control the testing efficiency and estimation bias. Theoretically, the optimal values of K_1 and K_2 should achieve two goals: (i) make the coefficient $c(\mathbf{s})$ equal to 1 to eliminate estimation bias, and (ii) minimize the variance of the estimator $\hat{\mu}_{q_u}$ to improve testing efficiency. Achieving these goals generally requires the parameters to vary across different states $\mathbf{s}$, since the optimal trade-off between bias and variance depends on the specific traffic states during sampling. However, in our preliminary work, these parameters are manually tuned based on empirical results, and the parameters remain fixed throughout the testing process once chosen. This prevents state-dependent adaptation and limits the overall performance of IIS.

In this paper, we further propose Deep-IIS to adaptively adjust K_1 and K_2 during testing, aiming to improve sampling efficiency while reducing estimation bias.

III. Methodology

Building upon our prior work on IIS, we now propose a dynamic policy-based optimization of the key sampling parameters, leveraging deep reinforcement learning. This section presents the methodology of our proposed approach, which comprises four main parts. Section III-A introduces the dense reinforcement learning framework employed to train the agent in complex, high-dimensional traffic environments. Sections III-B and III-C detail the parameter optimization strategy for implicit importance sampling, which focuses on dynamically adjusting the key parameters K_1 and K_2 to improve testing efficiency and reduce estimation bias. This analysis offers insights into the conditions under which the proposed method achieves both reliable and efficient evaluation results.

A. Dense Reinforcement Learning

We adopt a reinforcement learning (RL) framework to adaptively optimize the sampling parameters (K_1, K_2) in the implicit importance sampling (IIS) framework. The entire testing process is modeled as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \rho, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, $\mathcal{T}$ is the transition probability, ρ is the initial state distribution, and $\gamma \in [0, 1)$ is the discount factor.

The sampling parameters (K_1, K_2) are treated as the actions of an agent, whose behavior is determined by a policy function π_θ, parameterized by θ:

$$\mathbf{a} = \pi(\mathbf{s}|\theta), \forall \mathbf{s} \in \mathcal{S}, (K_1, K_2) = \mathbf{a} \in \mathcal{A}. \tag{22}$$

Here, $\mathbf{s}$ denotes the current state of the testing scenario, and $\mathbf{a}$ denotes the action taken by the agent, corresponding to the value of K_1 and K_2.

To guide policy learning, we define a reward function $r(\mathbf{s}, \mathbf{a}) \in \mathcal{R}$, which measures the effectiveness of the parameters in improving sampling efficiency and reducing estimation bias. A detailed design of this reward is provided in Section III-C. The goal of the agent is to learn an optimal policy π^* that maximizes the expected accumulative discounted reward:

$$\begin{aligned} J(\theta) &= \mathbb{E}_\zeta \left[\sum_{t=0}^{T} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right], \\ \pi^* &= \underset{\pi}{\operatorname{argmax}} J(\theta), \end{aligned} \tag{23}$$

where $\zeta = \{\mathbf{s}_0, \mathbf{a}_0, r_0, \mathbf{s}_1, \ldots, \mathbf{s}_T\}$ is a trajectory generated by policy π.

To solve this optimization, we adopt Proximal Policy Optimization (PPO) algorithm [33]. The policy parameters are updated iteratively using the policy gradient:

$$\theta_{k+1} = \theta_k + l_k \nabla_\theta J(\theta_k), \tag{24}$$

where l_k is the learning rate at the iteration episode k.

Policy gradient is estimated by:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\zeta \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) A_t \right], \tag{25}$$

where A_t denotes the advantage function at timestep t, which is closely related to the accumulated reward.

In the context of autonomous vehicle testing, BVs typically exhibit safe and conservative driving behaviors. Only in rare critical states do BVs induce challenging interactions with AV under test. Consequently, most sampled trajectories yield zero reward signals, making standard RL inefficient and unstable. This phenomenon is known as the curse of rarity [4].

To address this challenge, dense reinforcement learning [11] is employed to enhance the learning efficiency by improving the density of critical states in dataset. Specifically, the Markov chain generated during testing is edited by retaining only the data corresponding to states whose risk level exceeds a predefined threshold. This leads to a modified policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\zeta \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A_t \mathbb{I}(\mathbf{s}_t \in \mathbb{S}_c) \right], \tag{26}$$

where $\mathbb{S}_c$ denotes the set of critical states, consistent with the definition in Eq. (12), i.e., the set of states $\mathbf{s}$ with $C(\mathbf{s}) > 0$. Accordingly, $\mathbb{I}(\mathbf{s}_t \in \mathbb{S}_c)$ is an indicator function that equals 1 if $\mathbf{s}_t$ is a critical state, and 0 otherwise.

It is worth noting that the use of dense reinforcement learning does not introduce policy training bias caused by sample distribution shift. This is because both the training and deployment of the policy are strictly restricted to the same subset of the state space, namely the critical state set $\mathbb{S}_c$. Only in critical states are the sampling parameters adjusted to perform adversarial sampling. Consequently, the policy π_θ is trained exclusively on trajectories filtered by the indicator function $\mathbb{I}(\mathbf{s}_t \in \mathbb{S}_c)$, and it is also invoked only when the traffic environment enters a critical state during testing. For non-critical states $\mathbf{s} \notin \mathbb{S}_c$, the default sampling strategy is adopted by setting $K_1 = K_2 = K_0 = 1$. These states are excluded from the policy gradient update and do not participate in policy inference after training. As a result, the policy does not generalize across heterogeneous state distributions, and the sample distribution seen during training remains consistent with that during deployment.

Also, the dense deep reinforcement learning approach has been theoretically proven to produce unbiased gradient estimates while significantly reducing the variance of the gradient estimation [11]. In this paper, we adopt this framework for parameter optimization.

B. Off-Policy Optimization of IIS Sampling Parameters

In this work, we adopt an off-policy reinforcement learning framework to optimize the parameters (K_1, K_2) in implicit importance sampling. Unlike on-policy methods that require new data to be collected for each policy update, off-policy methods enable learning from data generated by a fixed behavior policy. This is especially beneficial in safety-critical domains like autonomous driving, where failure cases (e.g., collisions) are rare and costly to sample.

Let π denote the target policy to be optimized, and π_b represent the behavior policy used to collect training data. The samples generated under π_b are stored in a replay buffer and reused for training the target policy π. In our context, the

output of π is the value pair $(\tilde{K}_1, \tilde{K}_2)$, whose range is $[0, 1]$. Then we use the linear function to map $\tilde{K}_1$ and $\tilde{K}_2$ to K_1 and K_2, which controls the sampling distribution in the IIS framework. Due to $K_1 \geq 1$ and $K_2 \leq 1$ is required, we use the following mapping function:

$$\begin{aligned} K_1 &= \tilde{K}_1 \times K_1^{\max} + 1, \\ K_2 &= 1 - \tilde{K}_2 \times (1 - K_2^{\min}), \end{aligned} \tag{27}$$

where $K_1^{\max}$ and $K_2^{\min}$ are the maximum value of K_1 and the minimum value of K_2, respectively. The values of $K_1^{\max}$ and $K_2^{\min}$ will be discussed in experimental section.

The input to the policy network π consists of the current state (position, speed, and heading angle) of the autonomous vehicle (AV) and a critical background vehicle (BV). Critical BV is defined as a BV whose predicted trajectory indicates potential collision risk with the AV. Leveraging dense deep reinforcement learning, we only include critical states in the training dataset, as description in section III-A, thus guaranteeing the input to π always incorporates a critical BV.

Correspondingly, we define $q_\pi(\mathbf{x})$ and $q_{\pi_b}(\mathbf{x})$ as the proposal distributions induced by the target and behavior policies, respectively. Their unnormalized forms, $q_{\pi,u}(\mathbf{x})$ and $q_{\pi_b,u}(\mathbf{x})$, are as defined in Eq. (11). Based on these, the unnormalized and normalized importance sampling weights are:

$$\begin{aligned} w_{\pi,u}(\mathbf{x}) &= \frac{P(\mathbf{x})}{q_{\pi,u}(\mathbf{x})}, \quad w_{\pi_b,u}(\mathbf{x}) = \frac{P(\mathbf{x})}{q_{\pi_b,u}(\mathbf{x})}, \\ w_\pi(\mathbf{x}) &= \frac{P(\mathbf{x})}{q_\pi(\mathbf{x})}, \quad w_{\pi_b}(\mathbf{x}) = \frac{P(\mathbf{x})}{q_{\pi_b}(\mathbf{x})}. \end{aligned} \tag{28}$$

Using the target policy π, the crash rate $P(A)$ can be estimated via IIS as defined in Eq. (14):

$$\begin{aligned} \hat{\mu}_{q_{\pi,u}} &= \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}_i) \frac{P(\mathbf{x}_i)}{q_{\pi,u}(\mathbf{x}_i)} \\ &= \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}_i) w_{\pi,u}(\mathbf{x}_i), \mathbf{x}_i \sim q_{\pi,u}(\mathbf{x}). \end{aligned} \tag{29}$$

Following Eq. (20) and Eq. (21), the lower and upper bounds of $P(A)$ are given by:

$$\begin{aligned} P(A) &\geq \hat{\mu}_{q_{\pi,u},\min} - \lambda \cdot \mathrm{std}(\hat{\mu}_{q_{\pi,u},\min}), \\ P(A) &\leq \hat{\mu}_{q_{\pi,u},\max} + \lambda \cdot \mathrm{std}(\hat{\mu}_{q_{\pi,u},\max}), \end{aligned} \tag{30}$$

where:

$$\begin{aligned} \hat{\mu}_{q_{\pi,u},\min} &\overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}) w_{\pi,u}(\mathbf{x}_i) c_{\pi,\min}(\mathbf{x}_i), \\ \hat{\mu}_{q_{\pi,u},\max} &\overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}) w_{\pi,u}(\mathbf{x}_i) c_{\pi,\max}(\mathbf{x}_i). \end{aligned} \tag{31}$$

Here $c_{\pi,\min}$ and $c_{\pi,\max}$ denote the lower and upper bounds of the correction coefficient $c(\mathbf{x})$ with policy π, as defined in Eq. (17) (18) (19).

C. Reward Function Design

The design of the reward function is crucial for deep reinforcement learning. In our formulation, each complete testing scenario $\mathbf{x}$ corresponds to a full trajectory in the Markov decision process. Instead of defining a timestep-wise reward $r(\mathbf{s}_t, \mathbf{a}_t)$, we adopt an episodic reward structure defined over the entire trajectory:

$$\mathbf{r}_t = \begin{cases} r(\mathbf{x}), & \text{if } t = T, \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

Here T is the episode horizon. This design reflects the fact that the effectiveness of a parameter setting can only be evaluated after observing the full scenario.

The total reward is composed of two terms: one targeting the reduction of variance in the estimator, and the other penalizing estimation bias:

$$r(\mathbf{x}) = r_{\text{var}}(\mathbf{x}) + \alpha r_{\text{bias}}(\mathbf{x}), \tag{33}$$

where α is a hyperparameter that balances the trade-off between variance reduction and bias correction.

To reduce the variance of the IIS estimator, we define the first term as:

$$\begin{aligned} r_{\text{var}}(\mathbf{x}) = f(\mathbf{x}) \cdot & \\ \big(& w_{\pi,u}(\mathbf{x}) c_{\pi,\max}(\mathbf{x}) \cdot w_{\pi,u_b}(\mathbf{x}) c_{\pi_b,\max}(\mathbf{x}) \\ & + w_{\pi_b,u}(\mathbf{x}) c_{\pi,\min}(\mathbf{x}) \cdot w_{\pi_b,u_b}(\mathbf{x}) c_{\pi_b,\min}(\mathbf{x}) \big). \end{aligned} \tag{34}$$

This reward formulation is motivated by the goal of minimizing the variance of the estimator. But w_π is unknown due to the implicit nature of $P(\mathbf{x})$ and $q_\pi(\mathbf{x})$, and only unnormalized weight $w_{\pi,u}$ is available. Therefore, we rely on upper and lower bounds of the variance to guide the learning:

$$\begin{aligned} \sigma_\pi^2 \leq \mathbb{B}_{q_{\pi_b}} \big[& f(\mathbf{x}) \cdot w_{\pi,u}(\mathbf{x}) c_{\pi,\max}(\mathbf{x}) \\ & \cdot w_{\pi_b,u}(\mathbf{x}) c_{\pi_b,\max}(\mathbf{x}) \big] - \mu^2, \\ \sigma_\pi^2 \geq \mathbb{B}_{q_{\pi_b}} \big[& f(\mathbf{x}) \cdot w_{\pi,u}(\mathbf{x}) c_{\pi,\min}(\mathbf{x}) \\ & \cdot w_{\pi_b,u}(\mathbf{x}) c_{\pi_b,\min}(\mathbf{x}) \big] - \mu^2. \end{aligned} \tag{35}$$

Here, μ denotes the theoretical estimate of the crash rate $P(A)$. The reward term $r_{\text{var}}(\mathbf{x})$ in Eq. (34) approximates these bounds and encourages the policy to minimize the estimation variance. Detailed derivations are provided in Section III-D.

The second component penalizes deviation from unbiased estimation. Since the correction coefficient $c(\mathbf{x})$ should ideally equal 1 to ensure unbiasedness, we define:

$$r_{\text{bias}}(\mathbf{x}) = \left| c_{\pi,\min}(\mathbf{x}) - 1 \right| + \left| c_{\pi,\max}(\mathbf{x}) - 1 \right|. \tag{36}$$

This term penalizes both underestimation and overestimation introduced by imperfect correction, driving the policy toward configurations with minimal bias. We theoretically analyze in Section III-D to show that this reward term $r_{\text{bias}}(\mathbf{x})$ helps to minimize the bias of the estimator.

Together, two parts of the proposed reward function guide the reinforcement learning process to optimize the policy toward more efficient and reliable sampling in the context of implicit importance sampling.

D. Theoretical Analysis

In this section, we provide a theoretical analysis of the reward function proposed in Section III-C. Specifically, we

analyze how it contributes to reducing estimation bias and improving sampling efficiency through variance reduction.

First, the following theorem establishes the condition under which the estimator becomes asymptotically unbiased:

Theorem 1: If $c_{\pi,\min}(\mathbf{x}) \to 1$ and $c_{\pi,\max}(\mathbf{x}) \to 1$ for all $\mathbf{x} \sim q_u(\mathbf{x})$, then the estimator $\hat{\mu}_{q_\pi,u}$ becomes asymptotically unbiased for estimating $P(A)$.

Proof: Given $c_{\pi,\min}(\mathbf{x}) \to 1$ and $c_{\pi,\max}(\mathbf{x}) \to 1$, we have $c_\pi(\mathbf{x}) \to 1$, and therefore $\frac{1}{c_\pi(\mathbf{x})} \to 1$. Recall that $c_\pi(\mathbf{x})$ reflects the discrepancy between the unnormalized proposal $q_{\pi,u}(\mathbf{x})$ and the normalized proposal $q_\pi(\mathbf{x})$, such that $q_\pi(\mathbf{x}) = \frac{q_{\pi,u}(\mathbf{x})}{c_\pi(\mathbf{x})}$.

Substituting into Eq. (29):

$$\begin{aligned}\hat{\mu}_{q_{\pi,u}} &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\frac{P(\mathbf{x}_i)}{q_{\pi,u}(\mathbf{x}_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\frac{P(\mathbf{x}_i)}{q_\pi(\mathbf{x}_i)}\frac{1}{c_\pi(\mathbf{x})} \\ &\to \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i)\frac{P(\mathbf{x}_i)}{q_\pi(\mathbf{x}_i)} \\ &= \hat{\mu}_{q_\pi}. \end{aligned} \tag{37}$$

Since $\hat{\mu}_{q_\pi}$ is an unbiased estimator of $P(A)$, it follows that $\hat{\mu}_{q_{\pi,u}}$ becomes asymptotically unbiased. ∎

Remark 1: Theorem 1 shows that estimation bias vanishes when the correction factor $c_\pi(\mathbf{x})$ approaches 1. This highlights the importance of aligning the unnormalized proposal distribution $q_{\pi,u}$ with the true normalized distribution q_π. As such, the bias term in the reward term (Eq. (36)) is designed to penalize large deviations of $c_\pi(\mathbf{x})$ from 1, thus encouraging unbiasedness in the sampling process.

We now analyze the variance of $\hat{\mu}_{q_\pi,u}$, which reflects the testing efficiency. The following theorem provides bounds on this variance:

Theorem 2: Under policies π and π_b, the variance of estimator $\hat{\mu}_{q_\pi,u}$ is bounded as:

$$\begin{aligned}\sigma_\pi^2 \le{}& \mathbb{E}_{q_{\pi_b}}\big[f(\mathbf{x})\cdot w_{\pi,u}(\mathbf{x})c_{\pi,\max}(\mathbf{x}) \\ &\cdot w_{\pi_b,u}(\mathbf{x})c_{\pi_b,\max}(\mathbf{x})\big] - \mu^2, \\ \sigma_\pi^2 \ge{}& \mathbb{E}_{q_{\pi_b}}\big[f(\mathbf{x})\cdot w_{\pi,u}(\mathbf{x})c_{\pi,\min}(\mathbf{x}) \\ &\cdot w_{\pi_b,u}(\mathbf{x})c_{\pi_b,\min}(\mathbf{x})\big] - \mu^2. \end{aligned} \tag{38}$$

Proof: The variance of $\hat{\mu}_{q_\pi,u}$ is given by:

$$\begin{aligned}\sigma_\pi^2 &= \mathbb{E}_{q_\pi}\left[\left(\frac{f(\mathbf{x})P(\mathbf{x}) - \mu q_\pi(\mathbf{x})}{q_\pi(\mathbf{x})}\right)^2\right] \\ &= \mathbb{E}_{q_\pi}\left[\left(\frac{f(\mathbf{x})P(\mathbf{x})}{q_\pi(\mathbf{x})} - \mu\right)^2\right] \\ &= \mathbb{E}_{q_\pi}\left[\left(\frac{f(\mathbf{x})P(\mathbf{x})}{q_\pi(\mathbf{x})}\right)^2\right] + \mu^2 - 2\mu\mathbb{E}_{q_\pi}\left[\frac{f(\mathbf{x})P(\mathbf{x})}{q_\pi(\mathbf{x})}\right] \\ &= \mathbb{E}_{q_\pi}\left[\left(\frac{f(\mathbf{x})P(\mathbf{x})}{q_\pi(\mathbf{x})}\right)^2\right] - \mu^2. \end{aligned} \tag{39}$$

Expanding the first term via importance sampling:

$$\mathbb{E}_{q_\pi}\left[\left(\frac{f(\mathbf{x})P(\mathbf{x})}{q_\pi(\mathbf{x})}\right)^2\right] = \mathbb{E}_P\left[\left(\frac{f(\mathbf{x})P(\mathbf{x})}{q_\pi(\mathbf{x})}\right)^2\frac{q_\pi(\mathbf{x})}{P(\mathbf{x})}\right]$$

$$= \mathbb{E}_{q_{\pi_b}}\left[f(\mathbf{x})\frac{P(\mathbf{x})}{q_\pi(\mathbf{x})}\frac{P(\mathbf{x})}{q_{\pi_b}(\mathbf{x})}\right]. \tag{40}$$

Using unnormalized weights $w_{\pi,u}, w_{\pi_b,u}$, and correction terms c_π, c_{π_b}, we get:

$$\begin{aligned}\sigma_\pi^2 &= \mathbb{E}_{q_{\pi_b}}\left[f(\mathbf{x})\frac{P(\mathbf{x})}{q_\pi(\mathbf{x})}\frac{P(\mathbf{x})}{q_{\pi_b}(\mathbf{x})}\right] - \mu^2 \\ &= \mathbb{E}_{q_{\pi_b}}\left[f(\mathbf{x})\cdot\frac{P(\mathbf{x})}{c_\pi q_\pi(\mathbf{x})}c_\pi(\mathbf{x})\cdot\frac{P(\mathbf{x})}{c_{\pi_b}q_{\pi_b}(\mathbf{x})}c_{\pi_b}(\mathbf{x})\right] - \mu^2 \\ &\le \mathbb{E}_{q_{\pi_b}}\Big[f(\mathbf{x})\cdot\frac{P(\mathbf{x})}{c_\pi q_\pi(\mathbf{x})}c_{\pi,\max}(\mathbf{x}) \\ &\qquad\cdot\frac{P(\mathbf{x})}{c_{\pi_b}q_{\pi_b}(\mathbf{x})}c_{\pi_b,\max}(\mathbf{x})\Big] - \mu^2 \\ &= \mathbb{E}_{q_{\pi_b}}\big[f(\mathbf{x})\cdot w_{\pi,u}(\mathbf{x})c_{\pi,\max}(\mathbf{x}) \\ &\qquad\cdot w_{\pi_b,u}(\mathbf{x})c_{\pi_b,\max}(\mathbf{x})\big] - \mu^2. \end{aligned} \tag{41}$$

By applying upper bounds $c_{\pi,\max}$ and $c_{\pi_b,\max}$, we arrive at the upper bound in Eq. (38). The lower bound follows similarly by applying $c_{\pi,\min}$ and $c_{\pi_b,\min}$. ∎

Remark 2: Theorem 2 confirms that the variance of the IIS estimator is tightly related to both the sampling weights and correction coefficients. The reward term (Eq. (34)) is constructed to minimize the upper and lower bound of this variance. Notably, the bounds are related to the correction coefficients $c_\pi(\mathbf{x})$ and $c_{\pi_b}(\mathbf{x})$. Keeping these two values close to 1 not only reduces bias (Theorem 1) but also contributes to variance reduction, by controlling values not too large. This dual effect further justifies our reward formulation.

In summary, the proposed reward function in Deep-IIS is theoretically grounded. It effectively balances estimation bias and variance, thereby ensuring reliable and efficient safety evaluation through reinforcement learning–based sampling policy optimization.

IV. Experimental Studies

In this section, we first present the overall experimental setup in Section IV-A, including the experimental pipeline and detailed descriptions of the two naturalistic driving environments (NDEs) used in our experiments.

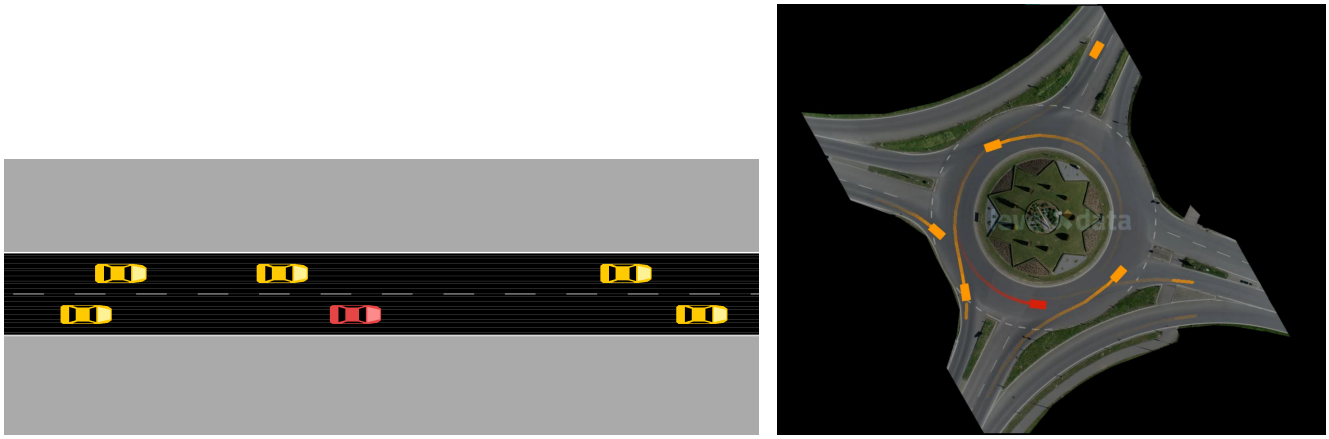
Sections IV-B and IV-C present the experimental results under two distinct NDE settings respectively.

A. Experimental Setting

To evaluate the effectiveness of our proposed method, we applied it to two distinct naturalistic driving environment (NDE) models. We denote these two testbeds as NDE-I and NDE-II in this paper. The testing process consisted of running large-scale simulations, during which test scenarios were collectd and the crash rate was calculated.

The experiments we conducted consist of three main parts:

- We performed baseline experiments under the NDEs to get the ground-truth crash rate.
- We performed baseline experiments using the IIS method, where the parameters K_1 and K_2 were manually set based on empirical knowledge. This stage provided the baseline results of IIS and simultaneously collected training data for our proposed method.



(a) (b)

Fig. 2. Overview of traffic scenarios in two NDEs from a bird's-eye perspective.

- We performed experiments using our proposed Deep-IIS method. The training data obtained from the previous stage was used to train the policy π, which dynamically adjusts the parameters K_1 and K_2.

Through these experiments, we aim to validate the following two aspects:

- Our method can effectively accelerate safety testing by improving testing efficiency compared to the IIS method.
- While achieving higher efficiency, our method maintains nearly unbiased estimates of the ground-truth crash rate, with controlled bias that ensures the reliability of the results. Furthermore, it reduces the bias range compared to the IIS method, demonstrating a better balance between efficiency and accuracy.

The NDEs were constructed from real-world driving datasets, so the evaluation outcomes closely reflect practical traffic situations. Then we provide detailed introductions to the two NDE models respectively.

NDE-I is constructed based on a discrete distribution $P(\mathbf{u}|\mathbf{s})$, which is derived from the real-world driving dataset. The modeling process begins by analyzing a naturalistic driving dataset, in which both vehicle states and actions are discrete into state-action pairs. Using statistical analysis, the empirical probabilities of these state-action pairs are derived to estimate the conditional distribution.

At each time step k, the joint action distribution is modeled as:

$$P(\mathbf{u}(k)|\mathbf{s}(k)) = \prod_{i=1}^{N} P(\mathbf{u}_i(k)|\mathbf{s}(k)), \tag{42}$$

where N is the total number of vehicles.

To reduce modeling complexity, the distribution is further simplified by assuming spatial independence among vehicles:

$$P(\mathbf{u}(k)|\mathbf{s}(k)) = \prod_{i=1}^{N} P\left(\mathbf{u}_i(k)|\mathbf{s}_{N_i}(k)\right), \tag{43}$$

where $\mathbf{s}_{N_i}(k)$ denotes the local state information relevant to the i-th vehicle. The conditional probability $P\left(\mathbf{u}_i(k)|\mathbf{s}_{N_i}(k)\right)$ is then computed from the empirical frequency of corresponding state-action pairs in the dataset.

Although the discrete distribution $P(\mathbf{u}|\mathbf{s})$ is explicit for NDE-I, we treat it as an implicit distribution for the purpose of validating our method. All simulations in NDE-I were conducted on a two-lane highway, as shown in Figure 2a. The red vehicle denotes the tested AV and the yellow vehicles represent BVs.

Based on the modeling approach from previous work D2RL [11], the experimental scenario features the lane width of 4 meters and a total length of 1200 meters. The AV is initialized at the 400-meter position, with the remaining sections of the road fully populated by BVs. For the initialization of BVs, a data-driven method is adopted, leveraging naturalistic driving data (NDD) to derive empirical distributions of vehicle speeds and following distances. Specifically, we utilize the distribution modeling results from D2RL: initial speeds of vehicles are sampled within the range of 20–40 m/s, and following distances are sampled from the range of 16–170 meters. Specific values are determined by random sampling from the constructed distributions. Detailed information regarding the distribution modeling process and data sources can be found in the referenced work [11].

Each simulation episode ended either when the AV reached the 800-meter mark or when a collision involving the AV occurred. The AV under test follows the Intelligent Driver Model (IDM) [34] for longitudinal control and the Minimizing Overall Braking Induced by Lane Changes (MOBIL) model [35] for lane-changing behavior.

NDE-II model is built using a neural network trained on naturalistic driving data. It consists of a transformer-based backbone network and a safety-layer that integrates learned predictions with rule-based adjustments. The backbone model F_M calculates the predicted trajectory over the next five time steps, with each step representing 0.4 seconds.

The safety-layer is designed to ensure the statistical consistency of the accident rate with naturalistic driving dataset. If a collision is predicted in the next time step, a conflict critic module F_C is triggered to judge whether the collision would

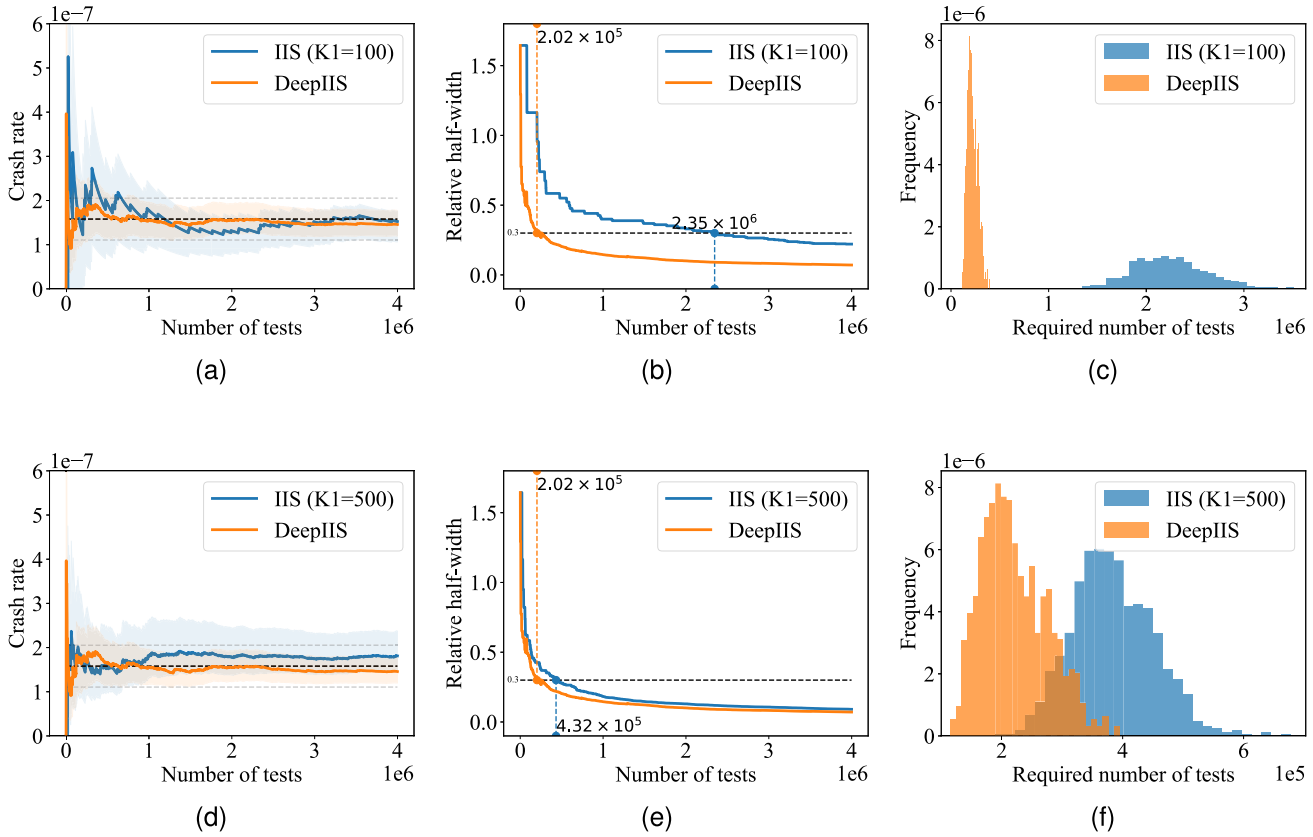


Fig. 3. Evaluation results for NDE-I using Deep-IIS, compared with IIS under ($K_1 = 100, K_2 = 0.99$) and ($K_1 = 500, K_2 = 0.99$). Subfigures 3a and 3d show the estimated crash rates and their corresponding bounds, where solid lines denote the crash rates, and shaded regions indicate the estimation bounds. Subfigures 3b and 3e show RHW curves. Subfigures 3c and 3f show frequency distribution of bootstrapped required number of tests.

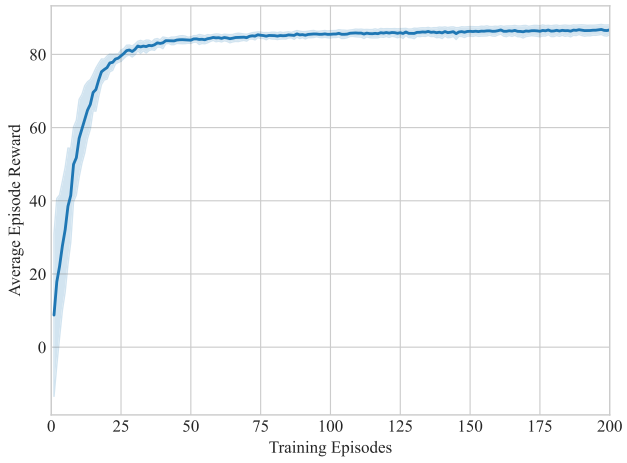


Fig. 4. Evolution of the reward function during Deep-IIS training for NDE-I.

actually occur. If the collision is deemed avoidable, the safety-layer adjusts the predicted trajectory to prevent it.

Due to the combination of neural network and rule-based filtering, the distribution $P(\mathbf{u}|\mathbf{s})$ is implicit for NDE-II. As a result, only sample-based evaluations can be conducted, making conditional importance sampling-based testing method [11] invalid. Consequently, only implicit importance sampling (IIS) is applicable for estimating the crash rate.

All experiments in NDE-II were conducted within a roundabout environment as shown in Figure 2b. The red vehicle denotes the tested AV and the yellow vehicles represent BVs. The roundabout has two lanes, and each lane in the roundabout has a width of approximately 4.5 meters. The inner edge radius of the roundabout is 16 meters, while the outer edge radius is 25 meters. There are four exits and four entrances to the roundabout, with each entrance having two lanes and each exit having one lane. The traffic flow entering each entrance lane is set to 200 vehicles per hour. These settings are based on previous work [6].

Each simulation episode ran for up to 36 seconds and terminated either when the AV exited the roundabout or when a collision involving the AV occurred. The AV's motion is controlled using vehicle trajectories sampled from the NDE model.

B. Results Analysis for NDE-I

We first conducted large-scale Monte Carlo simulations in NDE-I to estimate the ground-truth crash rate. A total of 2.3×10^8 test episodes were simulated, and the estimated ground-truth crash rate was 1.580×10^{-7}.

Following the IIS method, we conducted experiments using two sets of parameters, ($K_1 = 100, K_2 = 0.99$) and ($K_1 = 500, K_2 = 0.99$). According to equation (19), both parameter settings result in correction coefficients c_{ik}

TABLE I

COMPARISON OF IIS AND DEEP-IIS FOR NDE-I

Method	CR ($\times 10^{-7}$)	Bias ($\times 10^{-7}$)	Bound ($\times 10^{-7}$)	Speed-up
IIS ($K_1 = 100$)	1.525	0.055	0.694	87×
IIS ($K_1 = 500$)	1.812	0.232	0.903	470×
Deep-IIS	1.546	0.034	0.695	1005×

close to 1, ensuring the estimation bias remains within a controllable range. With the range of H_1 having been determined as $\left(1 \times 10^{-7}, 5 \times 10^{-5}\right)$, the values of $(\min c_{ik}, \max c_{ik})$ were approximately $(0.9900, 0.9950)$ for $K_1 = 100$ and $(0.9900, 1.0150)$ for $K_1 = 500$. To validate these settings, we ran 4×10^6 test episodes using the IIS method.

The minimum number of test episodes required to reach a precision threshold (RHW= 0.3) was computed. Under NDE testing, approximately 2.03×10^8 episodes were required to reach RHW= 0.3. As shown in Figure 3, this number was significantly reduced to 2.35×10^6 using IIS with $K_1 = 100$, corresponding to an 87× speed-up. When $K_1 = 500$, the required number of episodes dropped further to 4.32×10^5, yielding a 470× speed-up. However, the case with $K_1 = 100$ achieved more accurate estimation, with an estimation value of 1.525×10^{-7}, while the ground-truth was 1.580×10^{-7}. And the case with $K_1 = 500$ had an estimation value of 1.812×10^{-7}, which was further to the ground-truth. This reflects a fundamental trade-off in the IIS method, where a larger K_1 improves efficiency but potentially increases estimation bias.

We then used the data generated under ($K_1 = 100$, $K_2 = 0.99$) as the behavior policy π_b to train a Deep-IIS policy π. The reward function was balanced by hyperparameter $\alpha = 5 \times 10^{-5}$ and normalized to range $[-100, 100]$. The values of $K_1^{\max}$ and $K_2^{\min}$ were set to 5000 and 0.8, respectively.

To monitor the training progress, we tracked the evolution of the reward function over 200 training episodes. As shown in Figure 4, the solid curve depicts the mean reward averaged over 3 independent random seeds, and the shaded region around it denotes the corresponding standard deviation. The mean reward increases steadily over training, indicating the convergence of the policy after approximately 200 episodes.

Using the trained Deep-IIS policy, we performed 4×10^6 test episodes. The results are summarized in Figure 3. The estimated crash rate with Deep-IIS was 1.546×10^{-7}, closely matching the ground-truth. It achieved a speed-up of 1005×, outperforming both IIS configurations. Some numerical comparisons are presented in table I, including estimated crash rate (CR), bias relative to the ground-truth (Bias), estimated bound (Bound), and acceleration factor (speed-up). Compared to IIS with $K_1 = 100$, Deep-IIS increased the bound by only 0.001×10^{-7}, while improving speed-up by 11.69×. Compared to the $K_1 = 500$ case, Deep-IIS achieved both lower bias and narrower bound, along with a 2.14× improvement in speed-up. These comparisons further demonstrate that Deep-IIS offers a better balance between estimation accuracy and testing efficiency, outperforming IIS method in both dimensions.

The estimated crash rate with Deep-IIS was 1.546×10^{-7}, closely matching the ground-truth. It achieved a speed-up of 1005×, outperforming both IIS configurations. Some numerical comparisons are presented in Table I, including estimated crash rate (CR), bias relative to the ground-truth (Bias), estimated bound (Bound), and acceleration factor (speed-up).

Compared to IIS with $K_1 = 100$, Deep-IIS increased the bound by only 0.001×10^{-7}, while improving speed-up by 11.69×. Compared to the $K_1 = 500$ case, Deep-IIS achieved both lower bias and narrower bound, along with a 2.14× improvement in speed-up. These comparisons further demonstrate that Deep-IIS offers a better balance between estimation accuracy and testing efficiency, outperforming IIS method in both dimensions.

We also plot the histogram of the distributions of K_1 and K_2 values, as shown in fig 5. The distribution reveals a concentration around values near 1, which indicates that the trained agent focuses more on critical states, leading to adjustments in the distribution occurring within a smaller range. This helps to reduce the variance induced by importance sampling and the bias caused by unnormalized distribution adjustments. Meanwhile, when K_1 increases, K_2 tends to decrease, counteracting the bias introduced by a larger K_1 value. This behavior illustrates how the model adapts to minimize testing errors while optimizing efficiency.

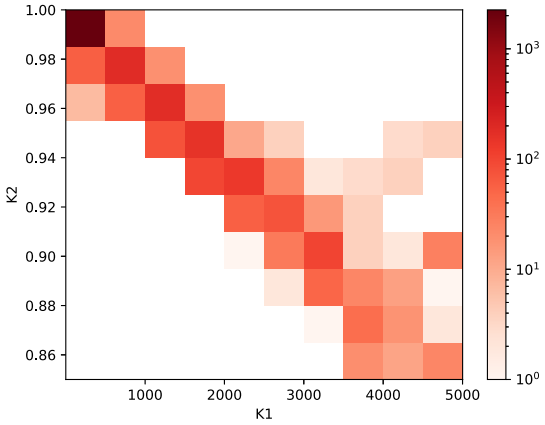


Fig. 5. Histogram of the distributions of K_1 and K_2 values output by the Deep-IIS policy for NDE-I.

TABLE II

DEVIATIONS OF THE ESTIMATED CRASH RATE BOUNDS FROM THE GROUND-TRUTH CRASH RATE FOR NDE-I WITH DIFFERENT α VALUES

α	$R(\%)$	$R^{\max}(\%)$
5×10^{-6}	65.62	37.78
1×10^{-5}	49.92	38.42
5×10^{-5}	36.66	24.33
1×10^{-4}	45.00	24.43
1×10^{-2}	40.67	34.56

TABLE III

COMPARISON OF IIS AND DEEP-IIS FOR NDE-II

Method	CR ($\times 10^{-6}$)	Bias ($\times 10^{-6}$)	Bound ($\times 10^{-6}$)	Speed-up
IIS ($K_1 = 50$)	7.181	0.326	3.703	49×
IIS ($K_1 = 100$)	6.097	1.410	4.576	83×
Deep-IIS	7.081	0.426	4.085	108×

Moreover, the weighting coefficient α in the reward function is selected empirically based on experimental evaluation. Due to the high computational cost of repeated reinforcement learning training, we evaluate a limited set of candidate values of α and choose one according to the quality of crash-rate estimation bounds. Specifically, we adopt the deviation of the estimated crash-rate bounds from the ground-truth crash rate as the primary evaluation metric:

$$R = \frac{|\hat{\mu}_{q_\pi,u,\max} - \hat{\mu}_{q_\pi,u,\min}|}{\hat{\mu}_P},$$
$$R^{\max} = \max\left(\frac{|\hat{\mu}_{q_\pi,u,\max} - \hat{\mu}_P|}{\hat{\mu}_P}, \frac{|\hat{\mu}_{q_\pi,u,\min} - \hat{\mu}_P|}{\hat{\mu}_P}\right). \quad (44)$$

The estimation bounds jointly reflect both the estimation bias and variance introduced by the sampling process. The smaller the deviation, the better the estimation quality. Therefore, these bounds provide a balanced and informative criterion for assessing the overall quality of the testing results. We summarize the experimental results for different α values in table II.

The reported results demonstrate that the selected α values 5×10^{-5} achieve a favorable balance between estimation accuracy and testing efficiency, leading to relatively tight bounds. With a smaller α value, the RL agent prefers to reduce the variance, but ignore the biases introduced by the unnormalized distribution adjustments. In contrast, a larger α value also results in a wider bound, as the agent focuses less on reducing the variance.

C. Results Analysis for NDE-II

We conducted the same set of experiments in the NDE-II environment. To obtain the ground-truth crash rate, we simulated 1.3×10^7 test episodes, resulting in an estimated crash rate of 7.507×10^{-6}.

We applied $(K_1 = 50, K_2 = 0.99)$ and $(K_1 = 100, K_2 = 0.99)$, yielding correction coefficients c_{ik} with values close to 1, ensuring the bias remains bounded. With the range of H_1 having been determined as $\left(4.5 \times 10^{-5}, 5 \times 10^{-3}\right)$, the values of $(\min c_{ik}, \max c_{ik})$ were approximately $(0.992205, 1.137)$ for $K_1 = 50$ and $(0.994455, 1.287)$ for $K_1 = 100$. To validate the estimates, 4×10^6 test episodes were simulated for each setting.

Under NDE testing, approximately 4.26×10^6 episodes were required to achieve the precision threshold of RHW = 0.3. As shown in Figure 6, this number was reduced to 8.61×10^4 using IIS with $K_1 = 50$, corresponding to an $49\times$ speed-up. When $K_1 = 100$, the required number of episodes dropped further to 5.10×10^4, yielding a $83\times$ speed-up. As with NDE-I, the IIS method exhibited a trade-off between efficiency and estimation accuracy. The case with $K_1 = 50$ provided a more accurate estimation of 7.181×10^{-6}, which is closer to the ground-truth. And the case with $K_1 = 100$ had an estimation value of 6.097×10^{-7}, which was further to the ground-truth. This again reflects the fundamental trade-off: larger K_1 enhances efficiency at the potential cost of increased bias.

A Deep-IIS policy was then trained using the dataset generated with $(K_1 = 50, K_2 = 0.99)$. The values of $K_1^{\max}$ and $K_2^{\min}$ were set to 100 and 0.8, respectively. We note that the parameter settings differ between the two experimental setups. These parameters are selected empirically and subsequently validated through experiments. When configuring the parameters, a larger value of $K_1^{\max}$ can increase the upper bound of testing efficiency, but it may also introduce larger estimation bias. For NDE-I, the crash rate is lower and the rarity of safety-critical events is more pronounced; therefore, a larger $K_1^{\max}$ can be adopted to improve the sampling efficiency of rare events. In contrast, for NDE-II, the rarity of such events is less significant. Setting an excessively large $K_1^{\max}$ in this case may lead to an overly large unnormalized coefficient of the sampling distribution, resulting in increased sampling bias. The primary role of K_2 is to compensate for the sampling bias introduced by K_1. We set $K_2^{\min} = 0.8$; however, experimental results show that the learned values of K_2 are consistently greater than 0.9.

TABLE IV
DEVIATIONS OF THE ESTIMATED CRASH RATE BOUNDS FROM THE GROUND-TRUTH CRASH RATE FOR NDE-II WITH DIFFERENT α VALUES

α	$R(\%)$	$R^{\max}(\%)$
2×10^{-5}	54.23	40.79
2×10^{-3}	60.77	39.63
2×10^{-2}	54.42	31.10
3×10^{-2}	59.46	37.09
5×10^{-2}	64.89	34.43

The reward function was balanced by $\alpha = 0.02$ and normalized to the range $[-100, 100]$.

To monitor the training progress, we tracked the evolution of the reward function over 200 training episodes. Figure 7 further corroborates our results: the mean reward (3 random seeds) increases steadily, with the standard deviation shaded area indicating low performance fluctuation, and the policy achieves stable convergence around the 200-episode threshold.

Using the learned policy, 2×10^6 episodes were tested, yielding an estimated crash rate of 7.081×10^{-6} and achieving a speed-up of $108\times$.

As shown in Figure 6 and table III, Deep-IIS again demonstrated improved balance between estimation accuracy and testing efficiency. Compared to IIS with $K_1 = 50$, Deep-IIS achieved similar bias and bound, while improving efficiency by $2.20\times$. Compared to IIS with $K_1 = 100$, it substantially reduced both the bias and bound, while still achieving a $1.30\times$ gain in speed-up.

The histogram of the distributions of K_1 and K_2 values is shown in fig 8. As similar to NDE-I, some K_1 and K_2 values are near 1, limiting the extent and range of distribution adjustments. This indicates that the agent focuses more on critical states, leading to a more efficient sampling process while maintaining a balance between bias reduction and variance control. When K_1 increases, K_2 tends to decrease, counteracting the bias introduced by a larger K_1 value.

These findings are consistent with the results observed in NDE-I, further validating the effectiveness of the proposed Deep-IIS approach.

Moreover, the selection of weighting coefficient α in the reward function is as same as which was discussed in NDE I.

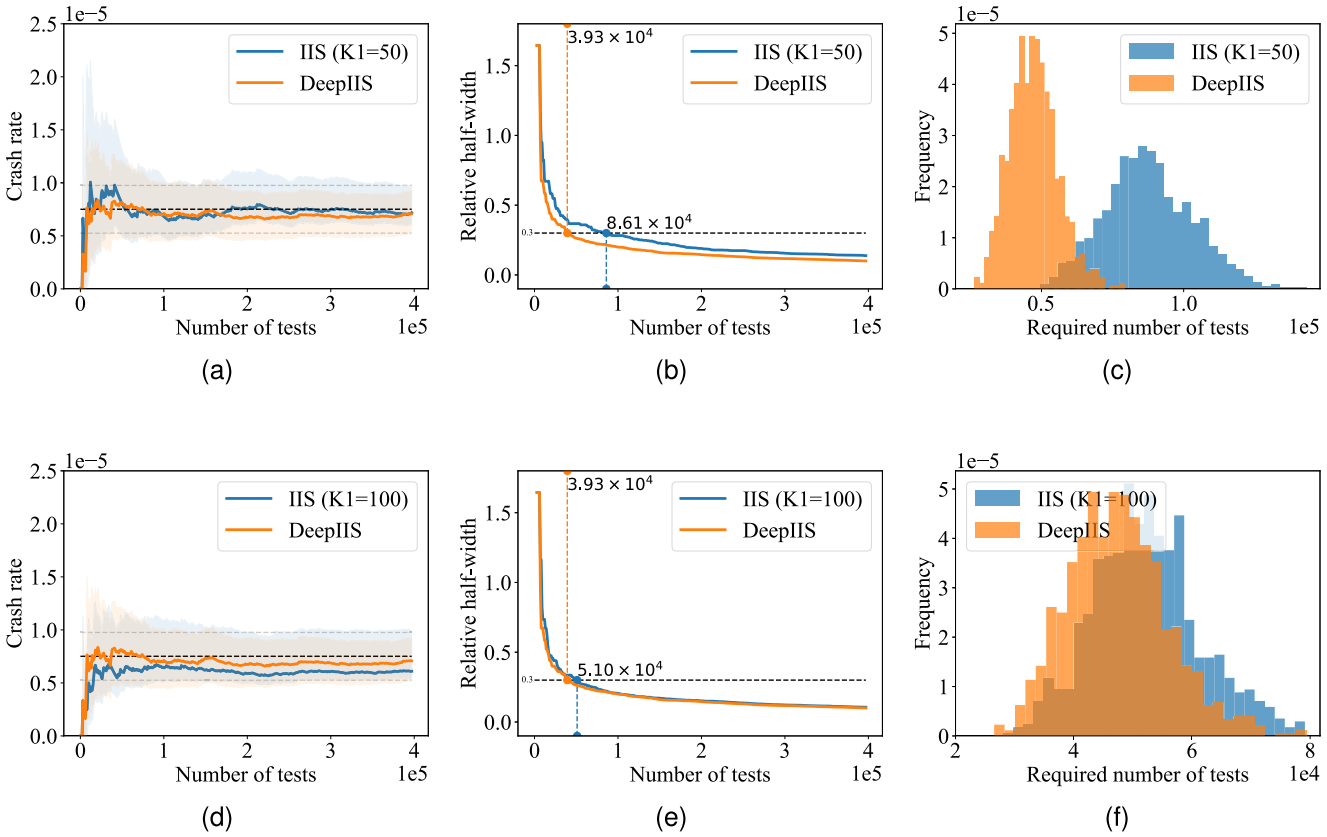


Fig. 6. Evaluation results for NDE-II using Deep-IIS, compared with IIS under ($K_1 = 50, K_2 = 0.99$) and ($K_1 = 100, K_2 = 0.99$). Subfigures 6a and 6d show the estimated crash rates and their corresponding bounds, where solid lines denote the crash rates, and shaded regions indicate the estimation bounds. Subfigures 6b and 6e show RHW curves. Subfigures 6c and 6f show frequency distribution of bootstrapped required number of tests.

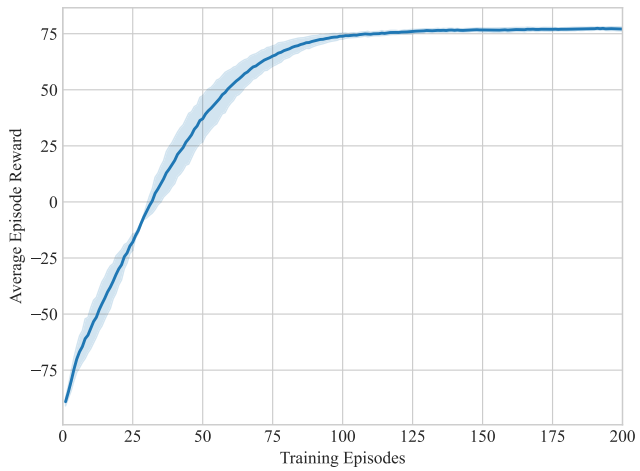


Fig. 7. Evolution of the reward function during Deep-IIS training for NDE-II.

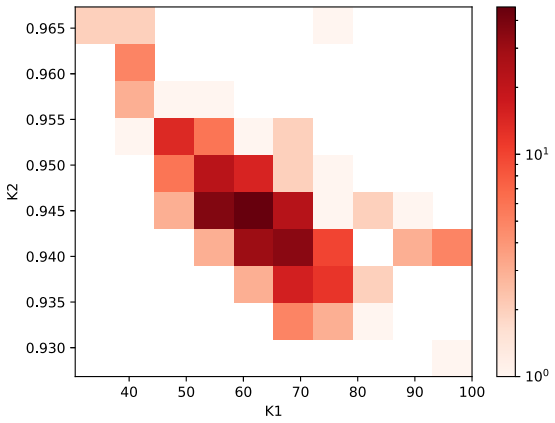


Fig. 8. Histogram of the distributions of K_1 and K_2 values output by the Deep-IIS policy for NDE-II.

The experimental results for different α values is in table IV. The reported results demonstrate that the selected α values 0.02 achieve a favorable balance between estimation accuracy and testing efficiency, leading to relatively tight bounds.

We note the significant difference in α between NDE-I and NDE-II arises from the distinct rarity characteristics of safety-critical events in the two environments. In NDE-I, crash events are rarer, and larger values of K_1 are encouraged to improve sampling efficiency, which reduces the relative contribution of the variance-related term in the reward. In contrast, NDE-II exhibits less pronounced rarity of critical events, leading to relatively smaller learned values of K_1. As a result, the variance-related component in the reward becomes more dominant, and a larger α is required to properly balance the variance and bias terms in the reward function.

V. Conclusion

This paper presents Deep-IIS, a deep reinforcement learning–enhanced framework for accelerating autonomous vehicle testing under implicit traffic behavior distributions. Deep-IIS constructs an unnormalized proposal distribution via rejection sampling and models its key sampling parameters as a state-dependent policy network. This enables adaptive adjustment of the proposal distribution based on observed traffic states, eliminating manual parameter tuning and improving robustness across scenarios. Experiments on two representative naturalistic driving environment (NDE) models demonstrate that Deep-IIS achieves superior sampling efficiency and estimation reliability compared with baseline IIS methods.

Nevertheless, the current evaluation is limited to relatively small-scale traffic environments. Future work will extend Deep-IIS to larger, more complex traffic environments, integrate richer NDE models.

References

[1] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Commun.*, vol. 12, no. 1, p. 748, Feb. 2021.

[2] A. Filos, P. Tigkas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3145–3153.

[3] H. Sun, S. Feng, X. Yan, and H. X. Liu, "Corner case generation and analysis for safety assessment of autonomous vehicles," *Transp. Res. Record, J. Transp. Res. Board*, vol. 2675, no. 11, pp. 587–600, Nov. 2021.

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

[5] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—A methodological perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 6971–6988, Jul. 2023.

[6] X. Yan, Z. Zou, S. Feng, H. Zhu, H. Sun, and H. X. Liu, "Learning naturalistic driving environment with statistical realism," *Nature Commun.*, vol. 14, no. 1, p. 2037, 2023.

[7] S. Duan, X. Bai, Q. Shi, W. Li, and A. Zhu, "Uncertainty evaluation for autonomous vehicles: A case study of AEB system," *Automot. Innov.*, vol. 7, no. 4, pp. 644–657, Nov. 2024.

[8] F. Kruber, J. Wurst, and M. Botsch, "An unsupervised random forest clustering technique for automatic traffic scenario categorization," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2811–2818.

[9] W. Wang and D. Zhao, "Extracting traffic primitives directly from naturalistically logged data for self-driving applications," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1223–1229, Apr. 2018.

[10] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transp. Res. A, Policy Pract.*, vol. 94, pp. 182–193, Dec. 2016.

[11] S. Feng et al., "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, Mar. 2023. [Online]. Available: https://www.nature.com/articles/s41586-023-05732-2

[12] J. Yang, R. Bai, H. Ji, Y. Zhang, J. Hu, and S. Feng, "Adaptive testing environment generation for connected and automated vehicles with dense reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 4, pp. 5135–5145, Apr. 2025.

[13] R. Bai, J. Yang, W. Gong, Y. Zhang, Q. Lu, and S. Feng, "Accurately predicting probabilities of safety-critical rare events for intelligent systems," in *Proc. IEEE 20th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2024, pp. 3243–3249.

[14] S. Li, H. He, J. Yang, J. Hu, Y. Zhang, and S. Feng, "Few-shot testing of autonomous vehicles with scenario similarity learning," 2024, *arXiv:2409.14369*.

[15] S. Li, J. Yang, H. He, Y. Zhang, J. Hu, and S. Feng, "Few-shot scenario testing for autonomous vehicles based on neighborhood coverage and similarity," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2024, pp. 620–626.

[16] J. Yang, Z. Wang, D. Wang, Y. Zhang, Q. Lu, and S. Feng, "Adaptive safety performance testing for autonomous vehicles with adaptive importance sampling," *Transp. Res. C, Emerg. Technol.,*, vol. 179, 2025, Art. no. 105256.

[17] A. B. Owen, "Monte Carlo theory, methods and examples," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2013.

[18] H. Cancela, M. El Khadiri, and G. Rubino, "Rare event analysis by Monte Carlo techniques in static models," in *Rare Event Simulation Using Monte Carlo Methods*. Hoboken, NJ, USA: Wiley, 2009, pp. 145–170.

[19] R. D. Morris, X. Descombes, and J. Zerubia, "The Ising/Potts model is not well suited to segmentation tasks," in *IEEE Digit. Signal Process. Workshop Proc.*, Sep. 1996, pp. 263–266.

[20] D. Zhao et al., "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 595–607, Mar. 2017.

[21] M. Arief et al., "Certifiable evaluation for autonomous vehicle perception systems using deep importance sampling (deep IS)," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 1736–1742.

[22] Z. Huang, M. Arief, H. Lam, and D. Zhao, "Evaluation uncertainty in data-driven self-driving testing," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1902–1907.

[23] Z. Jiang et al., "Efficient and unbiased safety test for autonomous driving systems," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 5, pp. 3336–3348, May 2022.

[24] D. Rempe, J. Philion, L. J. Guibas, S. Fidler, and O. Litany, "Generating useful accident-prone driving scenarios via a learned traffic prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17284–17294.

[25] Z. Mo, R. Shi, and X. Di, "A physics-informed deep learning paradigm for car-following models," *Transp. Res. C, Emerg. Technol.*, vol. 130, Sep. 2021, Art. no. 103240.

[26] L. Liu, S. Feng, Y. Feng, X. Zhu, and H. X. Liu, "Learning-based stochastic driving model for autonomous vehicle testing," *Transp. Res. Record, J. Transp. Res. Board*, vol. 2676, no. 1, pp. 54–64, Jan. 2022.

[27] K. Ren, J. Yang, Q. Lu, Y. Zhang, J. Hu, and S. Feng, "Intelligent testing environment generation for autonomous vehicles with implicit distributions of traffic behaviors," *Transp. Res. C, Emerg. Technol.*, vol. 174, May 2025, Art. no. 105106. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X2500110X

[28] M. L. Puterman, "Markov decision processes," *Handbooks Oper. Res. Manage. Sci.*, vol. 2, pp. 331–434, 1990.

[29] C. Z. Mooney, *Monte Carlo Simulation*. Newbury Park, CA, USA: Sage, 1997.

[30] S. Feng, Y. Feng, H. Sun, S. Bao, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part II: Case studies," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5635–5647, Sep. 2021.

[31] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part I: Methodology," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1573–1582, Mar. 2021.

[32] V. Neumann, "Various techniques used in connection with random digits," Stanford Univ., Stanford, CA, USA, Tech. Rep., 1951, pp. 36–38.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[34] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.

[35] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Record: J. Transp. Res. Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007.

Kun Ren received the bachelor's degree from the Department of Automation, Tsinghua University, Beijing, China, in 2023, where he is currently pursuing the master's degree with the Department of Automation. His current research interests include testing and evaluation of intelligent systems.

Yizhou Xu received the bachelor's degree from the Department of Automation, Tsinghua University, Beijing, China, in 2024, where he is currently pursuing the master's degree with the Department of Automation. His current research interests include simulation and testing of autonomous driving.

Jingxuan Yang received the bachelor's degree from the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China, in 2020, and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2025. He is currently a Post-Doctoral Research Fellow with the Department of Automation, Tsinghua University. His current research interests include curse of rarity, variance reduction, dense learning, and adaptive testing.

Yi Zhang (Senior Member, IEEE) received the B.S. and M.S. degrees from Tsinghua University, China, in 1986 and 1988, respectively, and the Ph.D. degree from the University of Strathclyde, U.K., in 1995. He is currently a Professor in control science and engineering with Tsinghua University. His current research interests focus on intelligent transportation systems. His active research areas include intelligent vehicle-infrastructure cooperative systems, analysis of urban transportation systems, urban road network management, traffic data fusion and dissemination, urban traffic control and management, the advanced control theory and applications, advanced detection and measurement, and systems engineering.

Jianming Hu (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in 1995, 1998, and 2001, respectively. He is currently an Associate Professor with the Department of Automation (DA), Tsinghua University. He has presided and participated in more than 20 research projects granted from the Ministry of Science and Technology of China, National Science Foundation of China, and other large companies, with more than 30 journal articles and more than 100 conference papers. His recent research interests include networked traffic flow, large-scale traffic information processing, intelligent vehicle infrastructure cooperation systems (V2X or connected vehicles), and urban traffic signal control.

Shuo Feng (Member, IEEE) received the bachelor's and Ph.D. degrees from the Department of Automation, Tsinghua University, China, in 2014 and 2019, respectively. He was a Post-Doctoral Research Fellow with the Department of Civil and Environmental Engineering and also an Assistant Research Scientist with the University of Michigan Transportation Research Institute (UMTRI), University of Michigan, Ann Arbor. He is currently an Associate Professor with the Department of Automation, Tsinghua University. His research interests lie in the development and validation of safety-critical machine learning, particularly for connected and automated vehicles. He was a recipient of the Best Ph.D. Dissertation Award from the IEEE Intelligent Transportation Systems Society in 2020 and the ITS Best Paper Award from the INFORMS TSL Society in 2021. He is an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT VEHICLES and an Academic Editor of the *Automotive Innovation*.