Transportation Research Part C 174 (2025) 105106

Contents lists available at ScienceDirect

Transportation Research Part C

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

Intelligent testing environment generation for autonomous vehicles with implicit distributions of traffic behaviors

Kun Ren, Jingxuan Yang, Qiujing Lu, Yi Zhang, Jianming Hu *, Shuo Feng *

Department of Automation, Tsinghua University, 100084, Beijing, China

ARTICLE INFO

Keywords:
Autonomous vehicles
Importance sampling
Accelerated testing

ABSTRACT

The advancement of autonomous vehicles hinges significantly on addressing safety concerns and obtaining reliable evaluation results. Testing the safety of autonomous vehicles is challenging due to the complexity of the high-dimensional traffic environment and the rarity of safety-critical events, often requiring billions of miles to achieve comprehensive validation, which is inefficient and costly. Current approaches, such as accelerated testing using importance sampling, aim to provide unbiased estimates of the performance of autonomous vehicles by generating a new distribution of background vehicles' behaviors based on an initial nominal distribution. However, these methods require knowledge of the original distribution of traffic behaviors, which is often difficult to obtain in practice. In response to these challenges, we introduce a novel methodology termed implicit importance sampling (IIS). Unlike traditional methods, IIS is designed to generate intelligent driving environments based on implicit distributions of traffic behaviors where the true distributions are unknown or not explicitly defined. IIS method leverages accept-reject sampling to construct an unnormalized proposal distribution, which increases the likelihood of sampling adversarial cases. Through applying importance sampling technique with unnormalized proposal distribution, IIS enhances testing efficiency and obtains reliable and representative evaluation results as well. The bias caused by unnormalization is also proved to be controlled and bounded.

1. Introduction

The development and deployment of autonomous vehicles (AVs) are expected to revolutionize transportation by enhancing safety and reducing traffic congestion. However, ensuring the safety and reliability of AVs remains a critical challenge due to several factors. First, the high-dimensionality, complexity, and stochastic nature of traffic environments can lead to the "curse of dimensionality" (Feng et al., 2021c), making it difficult to explicitly model a traffic environment. Second, the black-box nature of AV models makes their decision-making processes hard to predict and limits their ability to handle scenarios beyond their training experience (Filos et al., 2020). Third, the presence of long-tail events, which are rare but critical, plays a significant role, as these low-probability events are often the ones that can lead to accidents (Liu and Feng, 2024). These events are referred to as safety-critical cases (Ding et al., 2023) or corner cases (Sun et al., 2021). Traditional testing methods, which require AVs to drive billions of miles to encounter a wide range of scenarios, are prohibitively time-consuming and inefficient (Kalra and Paddock, 2016).

* Corresponding authors.
E-mail addresses: rk23@mails.tsinghua.edu.cn (K. Ren), yangjx20@mails.tsinghua.edu.cn (J. Yang), qiujinglu@mails.tsinghua.edu.cn (Q. Lu), zhyi@mail.tsinghua.edu.cn (Y. Zhang), hujm@mail.tsinghua.edu.cn (J. Hu), fshuo@tsinghua.edu.cn (S. Feng).

https://doi.org/10.1016/j.trc.2025.105106
Received 29 September 2024; Received in revised form 10 March 2025; Accepted 11 March 2025
Available online 23 March 2025

To address this problem, researchers have developed several advanced testing methods. The most commonly applied method is testing AVs in simulation of naturalistic driving environments (NDEs), which are often modeled through rules or naturalistic driving data (NDD) that sampled from the real world (Feng et al., 2021c; Yan et al., 2023; Duan et al., 2024). Many high-fidelity simulators integrate driving models as well, such as CARLA (Dosovitskiy et al., 2017), AADS (Li et al., 2019), and SUMO (Krajzewicz, 2010). However, NDD and NDE alone are usually not sufficient to evaluate the performance of AVs due to the limitations in scenario diversity and the lack of rare, safety-critical events. Therefore, methods such as clustering (Kruber et al., 2018; Wang and Zhao, 2018; Sun et al., 2021) or random perturbation (Scanlon et al., 2021; Fang et al., 2020; Lu et al.; Liu and Feng, 2024) have been used to augment datasets to obtain more safety-critical data from NDE or NDD. While preserving the naturalism of generated scenarios, these methods face challenges in efficiency and diversity. Adversarial attacks generate AV testing data by purposely controlling background vehicles (BVs) to challenge AVs. Many RL-based methods have been proposed to diversely search safety-critical traffic scenarios (Koren and Kochenderfer, 2019; Lee et al., 2020; Corso et al., 2019; Niu et al., 2023). Applying specific disturbances to BVs is also a feasible approach (Hanselmann et al., 2022; Rempe et al., 2022; Hao et al., 2023). Additionally, scenario library construction methods, such as using Genetic Algorithms (GA), efficiently create diverse and critical traffic scenarios (Zhao et al., 2023; Jiang et al., 2024).

However, there is an expectation that we could efficiently evaluate the performance of AVs in contrast with human drivers, for example, the accident rate under the same NDE. The aforementioned methods cannot achieve this because only scenarios in a small area with a small number of vehicles can be generated instead of a continuous traffic flow. Recent studies have introduced innovative approaches to address these issues (Riedmaier et al., 2020; Nalic et al., 2020). For example, surrogate-based optimization methods leverage naturalistic driving data to create models that simulate real-world scenarios, improving risk assessment accuracy (Zhang et al., 2022, 2023). The splitting technique estimates small probability events by dividing the probability space into manageable segments, enhancing accuracy and efficiency in rare safety scenarios (Cancela et al., 2009). Importance sampling has shown significant potential for addressing the problem of rare events by focusing on generating critical cases that are more likely to lead to safety violations, thus reducing the number of miles needed for testing (Owen, 2013; Cancela et al., 2009; Morris et al., 1996). Works such as Zhao et al. (2017), Feng et al. (2021c, 2023), Arief et al. (2022), Huang et al. (2019) and Jiang et al. (2022) have demonstrated the capacity of importance sampling to provide unbiased safety estimates, using the NDE modeled as discrete distributions extracted from NDD. Feng et al. (2021c) introduced the concept of an intelligent testing environment, which utilizes importance sampling methods to create naturalistic and adversarial driving environment (NADE) for AV testing. This approach results in unbiased estimates of AV performance with greater efficiency than NDE. Building upon this foundation, Feng et al. (2023) further improve testing efficiency through using reinforcement learning to optimize the proposal distribution in importance sampling, thereby reducing sampling variance.

Despite the advantages of importance sampling, one of its key limitations is the requirement for a well-defined nominal distribution to compute the proposal distribution and associated weights. In practice, however, it is often challenging to obtain explicit models of the nominal distribution, especially in more complex NDEs where the distribution may be implicit. This issue arises because NDEs are usually difficult to model explicitly due to the high-dimensionality, complexity, and stochastic nature of traffic environments. In some simple cases, for example, a fixed road network, NDE can be modeled explicitly (Zhao et al., 2017; Feng et al., 2021c; Arief et al., 2022; Huang et al., 2019; Jiang et al., 2022). However, in most cases, NDEs are modeled using complex models, rules, or neural networks (Yan et al., 2023; Rempe et al., 2022; Mo et al., 2021; Liu et al., 2022). For instance, if NDE is constructed using a neural network, and the network outputs both the mean and variance of the samples, then we can utilize the Gaussian distribution to obtain the probability density values. In this case, the distribution becomes explicit. However, if the neural network is treated as a black-box model, where we only obtain the sampled outputs but not the underlying probability density value, then NDE is an implicit distribution. Furthermore, if NDE is constructed by combining a neural network with certain rules, where the network's output is adjusted according to these rules, the probability density values of the output are also unknown, making it an implicit distribution. As a result, the exact probabilities of events or behaviors are unknown, and we can only draw samples from the environments under an underlying yet unknown probability distribution. This challenge is not unique to autonomous driving but is common across various domains that involve testing of intelligent systems such as robotics.

Also, in the domain of importance sampling, several approaches have been suggested to overcome this challenge, such as approximating the weights through convex quadratic optimization rather than relying on the exact likelihood ratio (O'Hagan, 1987; Henmi et al., 2007; Delyon and Portier, 2016; Liu and Lee, 2017; Oates et al., 2016). While These methods have the advantage of reducing the variance of importance sampling and producing unbiased estimates, they still depend on certain necessary information about the nominal distribution, such as the form of the distribution or the derivatives of the probability density function. Therefore, there is a need for more generalizable approaches that can handle the situations that only data can be sampled following the implicit distribution without any other information related to probability density. Due to the rarity of safety-critical events, it is also prohibitively inefficient to estimate the distribution by only sampling the data.

In response to these challenges, we introduce a novel approach called implicit importance sampling (IIS). Unlike traditional methods, IIS is designed to generate the intelligent testing environment under implicit nominal distributions. By identifying critical cases and leveraging accept-reject sampling (Neumann, 1951), the probability of sampling critical cases is increased. Our method constructs an unnormalized proposal distribution using accept-reject sampling, along with an associated unnormalized weight. This approach enables the efficient evaluation of AV performance, even in the absence of a fully specified nominal distribution. Although this introduces some bias due to unnormalization, we prove that this bias can be restricted to a controllable range, ensuring that the evaluation results remain reliable. This allows for significantly accelerating testing, reducing the required number of test miles by several orders of magnitude. Fig. 1 provides a schematic illustration of the proposed method.

Our contributions are threefold:

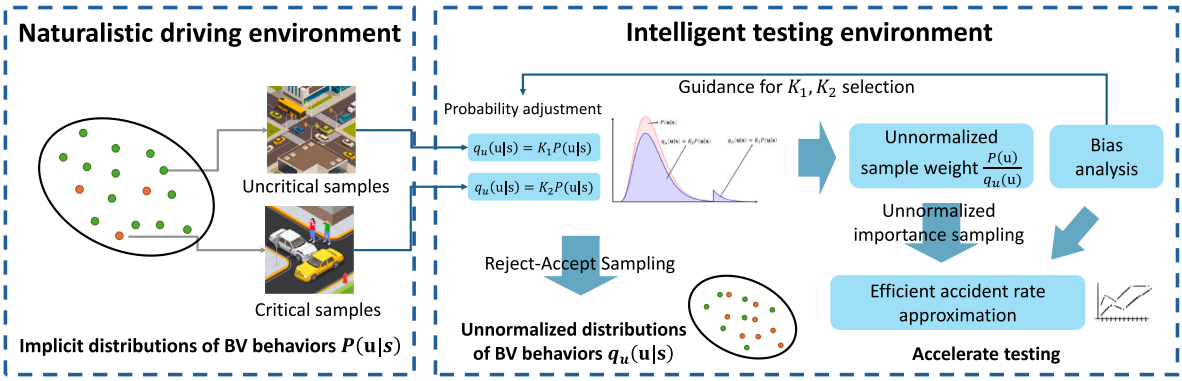


Fig. 1. Overview of implicit importance sampling framework.

- We present a method capable of generating intelligent testing environments for NDEs with implicit distributions of traffic behaviors.
- We prove that our method facilitates accelerated testing with controllable bias, providing a reliable range for bias that ensures the reliability of evaluation results.
- We conducted experiments on two distinct NDE models to demonstrate the generalizability of our approach. The results demonstrated significant acceleration in testing while maintaining a controlled level of bias, thereby validating the effectiveness of IIS in diverse scenarios.

2. Preliminary work

This section provides an overview of existing methods related to NDE and NADE models, which are essential for understanding the subsequent development of our approach.

2.1. Naturalistic driving environment (NDE)

Our algorithm for generating NADE is based on existing NDE models for sampling. A fundamental approach for testing autonomous vehicles (AVs) is to construct NDE models. In the context of our work, we aim for the vehicle driving behaviors and data distribution within the NDE model to closely resemble real-world traffic scenarios. This enables us to perform simulation tests under these conditions, where the results obtained can provide a reliable representation of the AV's performance in realistic environments.

As discussed in prior research (Feng et al., 2021c), NDE is represented by the combination of variables in a traffic scenario, which may include the position and velocity of vehicles and the parameters of road or weather. The NDE with N vehicles and T time steps can be represented as

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{1,1} & \cdots & \mathbf{x}_{1,T} \\ \ddots & \cdots & \ddots \\ \mathbf{x}_{N,1} & \cdots & \mathbf{x}_{N,T} \end{bmatrix}, \mathbf{x} \in \mathbf{X}, \tag{1}$$

where $\mathbf{x}_{i,j}$ represents the variables of the ith vehicle at the jth time step. This representation results in an extremely high-dimensional variable. To handle the complexities associated with this high dimensionality, Markov Decision Process (MDP) is used to simplify the distribution of $\mathbf{x}$. In a dynamic traffic scenario, the state and action of the ith vehicle at the jth time step are written as $\mathbf{s}_i(j)$ and $\mathbf{u}_i(j)$ respectively. The state and action of all vehicles at the jth time step are denoted as

$$\begin{aligned} \mathbf{s}(j) &= [\mathbf{s}_0(j), \mathbf{s}_1(j), \ldots, \mathbf{s}_N(j)], \\ \mathbf{u}(j) &= [\mathbf{u}_0(j), \mathbf{u}_1(j), \ldots, \mathbf{u}_N(j)]. \end{aligned} \tag{2}$$

So a scenario can be represented as a MDP (Puterman, 1990):

$$\mathbf{s}(0) \to \mathbf{u}(0) \to \mathbf{s}(1) \to \mathbf{u}(1) \cdots \to \mathbf{u}(T-1) \to \mathbf{s}(T). \tag{3}$$

The distribution $P(\mathbf{x})$ can be simplified as

$$P(\mathbf{x}) = P(\mathbf{s}(0)) \prod_{k=0}^{T} P(\mathbf{u}(k)|\mathbf{s}(k)) \mathcal{T}(\mathbf{s}(k+1)|\mathbf{s}(k), \mathbf{u}(k)), \tag{4}$$

where $\mathcal{T}(\cdot)$ is the state transition distribution.

Finally, our purpose is to evaluate the performance of the AV. To achieve this goal, we measure the AV's performance by calculating the accident rate. Specifically, the testing process involves simulating scenarios within the constructed NDE model,

generating various test scenarios, and then calculating the probability of accidents occurring involving the AV. By running these simulations over a large number of scenarios, we can estimate the accident rate as:

$$P(A)=\sum_{\mathbf{x}\in\mathbf{X}}P(A|\mathbf{x})P(\mathbf{x})\approx\frac{1}{n}\sum_{i=1}^{n}P\left(A|x_i\right), \tag{5}$$

where A indicates the accident between AV and BVs, and given a driving environment $\mathbf{x}_i$, $P\left(A|\mathbf{x}_i\right)$ is estimated by counting the number of accident events occurring during the test.

2.2. Naturalistic and adversarial driving environment (NADE)

To obtain more accidents and accelerate the AV testing process, NADE has been proposed in Feng et al. (2021c). The main idea is to increase the probability to sample $\mathbf{x}$ that may lead to accidents. A proposal distribution $q(\mathbf{x})$ is used to replace $P(\mathbf{x})$ to achieve this goal. And the key of this section is to design the distribution $q(\mathbf{x})$. The form of $q(\mathbf{x})$ resembles that of $P(\mathbf{x})$:

$$q(\mathbf{x})=q(\mathbf{s}(0))\prod_{k=0}^{T}q(\mathbf{u}(k)|\mathbf{s}(k))\mathcal{T}(\mathbf{s}(k+1)|\mathbf{s}(k),\mathbf{u}(k)), \tag{6}$$

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

To increase the likelihood of encountering accidents, we aim to sample $\mathbf{x}$ with a higher probability $p(\mathbf{x})$ when the conditional probability of an accident $P(A|\mathbf{x})$ is high, ideally close to 1. Accordingly, we aim to ensure that when $P(A|\mathbf{u},\mathbf{s})$ is higher, the corresponding action $\mathbf{u}$ can be sampled with a higher probability, namely higher $q(\mathbf{u}|\mathbf{s})$.

So the first step in obtaining $q(\mathbf{x})$ is to calculate the criticality given state $\mathbf{s}$. The criticality (Feng et al., 2021a,b) is calculated as

$$C(\mathbf{s})=\sum_{\mathbf{u}}P(A|\mathbf{u},\mathbf{s})P(\mathbf{u}|\mathbf{s}). \tag{7}$$

Then adjust distribution $P(\mathbf{u}|\mathbf{s})$ to obtain an expected $q(\mathbf{u}|\mathbf{s})$. Because $P(\mathbf{u}|\mathbf{s})$ is an explicit distribution whose probability can be obtained, $q(\mathbf{u}|\mathbf{s})$ can be designed by changing the value of probability directly according to the criticality $C(\mathbf{s})$.

2.3. AV performance evaluation in NADE

Accident rate is used to evaluate the performance of AV. The most direct method is testing an AV in NDE and estimating $P(A)$ by the Crude Monte Carlo (CMC) method (Mooney, 1997):

$$\begin{aligned}P(A)&=\mathbb{E}_P(P(A|\mathbf{x}))\\&\approx\frac{1}{n}\sum_{i=1}^{n}P\left(A|\mathbf{x}_i\right)\approx\frac{m}{n},\mathbf{x}_i\sim P(\mathbf{x}),\end{aligned} \tag{8}$$

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

However, due to the extremely low accident rate in NDE, a huge number of tests are required to estimate $P(A)$. Based on NADE, $P(A)$ can also be estimated more efficiently through the importance sampling method:

$$\begin{aligned}P(A)&=\mathbb{E}_P(P(A|\mathbf{x}))\\&=\mathbb{E}_q\left(P(A|\mathbf{x})\frac{P(\mathbf{x})}{q(\mathbf{x})}\right)\\&\approx\frac{1}{n}\sum_{i=1}^{n}P\left(A|\mathbf{x}_i\right)\frac{P\left(\mathbf{x}_i\right)}{q\left(\mathbf{x}_i\right)},\mathbf{x}_i\sim q\left(\mathbf{x}_i\right),\end{aligned} \tag{9}$$

where $\mathbf{x}_i\sim q\left(\mathbf{x}_i\right)$ means that $\mathbf{x}_i$ is sampled from naturalistic and adversarial distribution $q(\mathbf{x})$. Here proposal distribution $q(\mathbf{x})$ is referred to as the importance distribution. The ratio of $P(\mathbf{x})$ to $q(\mathbf{x})$ is called weight:

$$w(\mathbf{x})=\frac{P(\mathbf{x})}{q(\mathbf{x})}. \tag{10}$$

$P(\mathbf{x})$ and $q(\mathbf{x})$ have been simplified through MDP, so Eq. (9) can also be simplified as

$$P(A)\approx\frac{1}{n}\sum_{i=1}^{n}P\left(A|\mathbf{x}_i\right)\prod_{k=1}^{T_i}\frac{P(\mathbf{u}(k)|\mathbf{s}(k))}{q(\mathbf{u}(k)|\mathbf{s}(k))},\mathbf{x}_i\sim q\left(\mathbf{x}_i\right), \tag{11}$$

where T_i denotes the time-steps of the ith test. While increasing the probability of sampling critical events, previous work has demonstrated that the estimated result in Eq. (9) is unbiased. According to importance sampling technique:

$$\mathbb{E}_q\left(\frac{1}{n}\sum_{i=1}^{n}P\left(A|\mathbf{x}_i\right)\frac{P\left(\mathbf{x}_i\right)}{q\left(\mathbf{x}_i\right)}\right)=\mathbb{E}_q\left(P(A|\mathbf{x})\frac{P(\mathbf{x})}{q(\mathbf{x})}\right)=\mathbb{E}_P(P(A|\mathbf{x})). \tag{12}$$

So we can get an accurate testing result of AV performance in NADE.

3. Methodology

3.1. Generation of testing environment

It can be seen that the preliminary method relies on an explicit distribution $P(\mathbf{x})$. However, obtaining an explicit distribution $P(\mathbf{x})$ may be challenging. In such cases, $P(\mathbf{x})$ becomes an implicit distribution, from which we can only obtain samples, but cannot directly calculate the probability. To address this limitation, we propose a method for handling implicit distributions, allowing for greater flexibility and adaptability in generating NADE and accelerating testing. This section will introduce the details of this method.

Because $P(\mathbf{u}|\mathbf{s})$ is an implicit distribution whose probability cannot be calculated, we design and sample from $q(\mathbf{u}|\mathbf{s})$ with the help of accept-reject sampling (Neumann, 1951). The distribution $q(\mathbf{u}|\mathbf{s})$ is designed as

$$\begin{aligned} q(\mathbf{u}|\mathbf{s}) &= \frac{q_u(\mathbf{u}|\mathbf{s})}{\int q_u(\mathbf{u}|\mathbf{s})\mathrm{d}\mathbf{u}}, \\ q_u(\mathbf{u}|\mathbf{s}) &= K(\mathbf{u},\mathbf{s})\,P(\mathbf{u}|\mathbf{s}), \\ K(\mathbf{u},\mathbf{s}) &= \begin{cases} K_0 = 1, \text{ if } C(\mathbf{s}) = 0 \\ K_1 > 1, \text{ if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) > 0, \\ K_2 < 1, \text{ if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) = 0 \end{cases} \end{aligned} \tag{13}$$

where $q_u(\mathbf{u}|\mathbf{s})$ represents the unnormalized distribution because $\int K(\mathbf{u},\mathbf{s})\,P(\mathbf{u}|\mathbf{s})\mathrm{d}\mathbf{u}$ cannot be ensured to equal to 1. $C(\mathbf{s})$ denotes the criticality of state $\mathbf{s}$ and $C(\mathbf{s}) \neq 0$ means there is a possibility of an accident.

To sample from $q_u(\mathbf{u}|\mathbf{s})$ while $C(\mathbf{s}) \neq 0$, accept-reject sampling method is used to just continually sample $\mathbf{u} \sim P(\mathbf{u}|\mathbf{s})$, until a sample result is accepted with the accept probability:

$$\begin{aligned} P_{\text{acc}}(\mathbf{u}|\mathbf{s}) &= \frac{q_u(\mathbf{u}|\mathbf{s})}{K_1 P(\mathbf{u}|\mathbf{s})} = \frac{K(\mathbf{u}|\mathbf{s})}{K_1} \\ &= \begin{cases} 1, \text{ if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) > 0 \\ \dfrac{K_2}{K_1}, \text{ if } C(\mathbf{s}) \neq 0 \text{ and } P(A|\mathbf{u},\mathbf{s}) = 0 \end{cases} \end{aligned} \tag{14}$$

This can ensure that the sample results obey distribution $q_u(\mathbf{u}|\mathbf{s})$.

To calculate criticality $C(\mathbf{s})$ and $P(A|\mathbf{u},\mathbf{s})$, we conduct trajectory prediction utilizing the outputs of NDE models in this paper. Based on the predicted trajectories, we assess whether an accident involving AV is likely to occur in the future. If such an accident is predicted, the corresponding state and action are then classified as critical, namely $C(\mathbf{s}) \neq 0$ and $P(A|\mathbf{u},\mathbf{s}) \neq 0$. It is sufficient to determine whether these values are non-zero, without needing the exact values. The detailed design of the trajectory prediction and the calculation of these criticality values will be presented in the experimental section.

Fig. 2 illustrates how to adjust $P(\mathbf{u}|\mathbf{s})$ to obtain expected $q_u(\mathbf{u}|\mathbf{s})$. The red and blue curves in the figure illustrate the distributions $P(\mathbf{u}|\mathbf{s})$ and $q(\mathbf{u}|\mathbf{s})$, respectively. To ensure the driving environment is both adversarial and naturalistic, only if criticality $C(\mathbf{s})$ is not zero will the action $\mathbf{u}$ be sampled in $q(\mathbf{u}|\mathbf{s})$, else in $P(\mathbf{u}|\mathbf{s})$. So, as shown in Fig. 2(a), $q_u(\mathbf{u}|\mathbf{s})$ is equal to $P(\mathbf{u}|\mathbf{s})$ at uncritical state. On the other hand, while $C(\mathbf{s}) \neq 0$, $K_1 > 1$ will act on the part of $P(A|\mathbf{u},\mathbf{s}) > 0$, just as shown in Fig. 2(b). So the probability of $\mathbf{u}$ that satisfy $P(A|\mathbf{u},\mathbf{s}) > 0$ can be increased, meaning that the probability of event A can be increased.

Meanwhile, K_1 also causes the integral area of the probability density function to not be equal to 1. So $K_2 < 1$ are used for balancing. However, the value of $\int K(\mathbf{u},\mathbf{s})\,P(\mathbf{u}|\mathbf{s})\mathrm{d}\mathbf{u}$ varies for different states $\mathbf{s}$. Coupled with the implicit feature of $P(\mathbf{u}|\mathbf{s})$, it can be challenging to choose appropriate values for K_1 and K_2. Therefore, while sampling $\mathbf{u}$, the unnormalized distribution $q_u(\mathbf{u}|\mathbf{s})$ is used to replace $q(\mathbf{u}|\mathbf{s})$, as shown in Eq. (14). Unnormalization affects only the accident rate statistics, not the sampling results, which will be discussed in the next section.

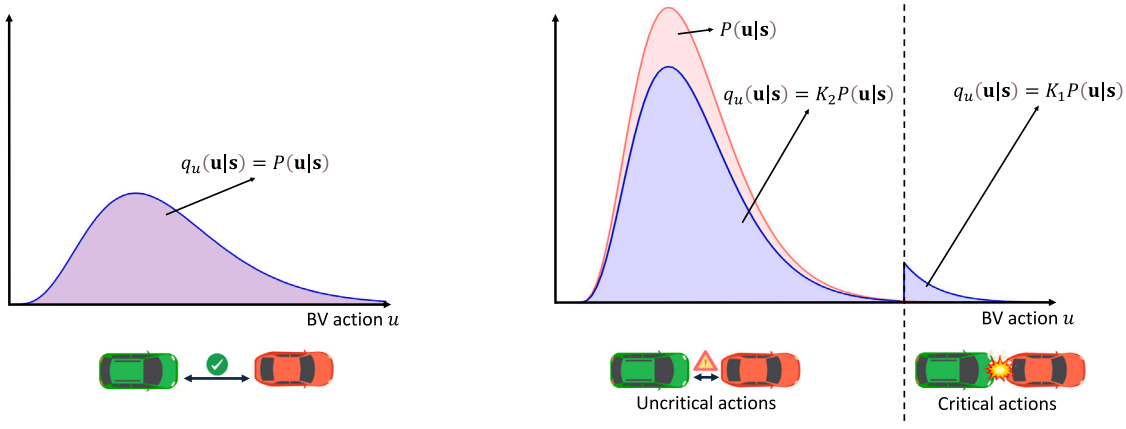
3.2. AV performance evaluation

Importance sampling relies on explicit distributions to calculate weights. But in this paper we can only get an unnormalized weight $w_u(\mathbf{x})$. The unnormalized weight at the ith test is calculated as:

$$\begin{aligned} w_u(\mathbf{x}_i) &= \frac{P(\mathbf{x}_i)}{q_u(\mathbf{x}_i)} = \prod_{k=1}^{T_i} \frac{P(\mathbf{u}(k)|\mathbf{s}(k))}{q_u(\mathbf{u}(k)|\mathbf{s}(k))} = \prod_{k=1}^{T_i} w_{u,ik}, \\ w_{u,ik} &= \frac{P(\mathbf{u}(k)|\mathbf{s}(k))}{q_u(\mathbf{u}(k)|\mathbf{s}(k))} = \frac{1}{K(\mathbf{u}(k),\mathbf{s}(k))}, \end{aligned} \tag{15}$$

where T_i denotes the time-steps of the ith test and w_{ik} represents the weight at kth time step during ith test.

At an uncritical state $\mathbf{s}$ whose criticality $C(\mathbf{s})$ is 0, we have $P(\mathbf{u}|\mathbf{s}) = q(\mathbf{u}|\mathbf{s})$ and so that its weight is 1. So only critical states need to be considered when calculating weight. $T_{i,C}$ denotes the set of critical time steps during ith test, then



(a) Uncritical state (b) Critical state

Fig. 2. A illustration of how to design proposal distribution.

$$w_u(\mathbf{x}_i) = \prod_{k \in T_{i,C}} w_{u,ik}. \tag{16}$$

Therefore, we can get an estimation of $P(A)$:

$$\mu_{q_u} = \mathbb{E}_q\left(P(A|\mathbf{x})\, w_u(\mathbf{x_i})\right). \tag{17}$$

We use $\hat{\mu}_{q_u}$ to represent importance sampling estimation of μ_{q_u}:

$$\hat{\mu}_{q_u} = \frac{1}{n}\sum_{i=1}^{n} P(A|\mathbf{x}_i)\, w_u(\mathbf{x_i}), \mathbf{x_i} \sim q(\mathbf{x}). \tag{18}$$

While sampling efficiency can be ensured to be higher than NDE, it cannot be ensured that $\hat{\mu}_{q_u}$ is an unbiased estimation of $P(A)$ because μ_{q_u} is obviously not equal to $P(A)$ in Eq. (9). But we have found that given suitable parameters K_1 and K_2, we can obtain evaluation results almost identical to Eq. (9) efficiently. The next part will elaborate on this and analyze the bias and variance of μ_{q_u}.

3.3. Theoretical analysis of bias

In this section, we discuss the bias between the estimation μ_{q_u} and $P(A)$ and how to choose suitable parameters K_1, K_2 to decrease the bias. The source of bias between $P(A)$ and μ_{q_u} is the difference between $q_u(\mathbf{x})$ and $q(\mathbf{x})$. We define the coefficient $c(\mathbf{x})$ as the ratio of $q_u(\mathbf{x})$ to $q(\mathbf{x})$:

$$c(\mathbf{x}) = \frac{q_u(\mathbf{x})}{q(\mathbf{x})}. \tag{19}$$

Then we present the following theorems.

Theorem 1.

$$\mathbb{E}_q\left(\hat{\mu}_{q_u}\right) = \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}\right). \tag{20}$$

Proof.

$$\begin{aligned}\mathbb{E}_q\left(\hat{\mu}_{q_u}\right) &= \mathbb{E}_q\left(\frac{1}{n}\sum_{i=1}^{n} P(A|\mathbf{x_i})\, w_u(\mathbf{x_i})\right) = \mathbb{E}_q\left(P(A|\mathbf{x})\, w_u(\mathbf{x})\right) \\ &= \mathbb{E}_q\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}\frac{P(\mathbf{x})}{q(\mathbf{x})}\right) = \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}\right).\end{aligned} \tag{21}$$

End of proof. □

Remark 1. Theorem 1 demonstrates that $\hat{\mu}_{q_u}$ provides an unbiased estimation of $\mathbb{E}\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}\right)$ rather than $P(A)$. This indicates that while $\hat{\mu}_{q_u}$ may not exactly estimate $P(A)$, it estimates a quantity scaled by the ratio of $q_u(\mathbf{x})$ to $q(\mathbf{x})$. The bias here arises from the mismatch between the proposal distribution $q_u(\mathbf{x})$ and the true distribution $q(\mathbf{x})$. Ideally, to minimize this bias, $c(\mathbf{x})$ should be as close to 1 as possible, ensuring that $\hat{\mu}_{q_u}$ provides a better approximation of $P(A)$.

Theorem 2. *The bound of $P(A)$ can be calculated as*

$$\mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\min}(\mathbf{x})\right) \le P(A) \le \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\max}(\mathbf{x})\right), \tag{22}$$

where $c_{\min}(\mathbf{x})$ and $c_{\max}(\mathbf{x})$ denotes the bound of $c(\mathbf{x})$ during test $\mathbf{x}$.

Proof.

$$\mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\min}(\mathbf{x})\right) \le P(A) = \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c(\mathbf{x})\right) \le \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\max}(\mathbf{x})\right). \tag{23}$$

End of proof. □

Remark 2. Theorem 2 establishes bounds on $P(A)$ based on the minimum and maximum values of the coefficient $c(\mathbf{x})$. This theorem provides a range within which the true value of $P(A)$ lies. The bounds are calculated by incorporating $c_{\min}(\mathbf{x})$ and $c_{\max}(\mathbf{x})$, which represent the extremal values of $c(\mathbf{x})$. By estimating these bounds through sampling results, we can obtain a reliable range for $P(A)$.

In the following content, we will detail the methods used to calculate the bound of $P(A)$ as outlined in Theorem 2, including how to estimate the bound of $c(\mathbf{x})$. Further, we will describe the process for selecting the parameters K_1 and K_2 to control the bias in the estimation. This involves ensuring that $c(\mathbf{x})$ remains close to 1, as discussed in Theorem 1, to ensure that the bound of $P(A)$ is both accurate and reliable for evaluating the performance of AV.

We estimate the bound of $P(A)$ using sampling results based on $q(\mathbf{x})$:

$$\begin{aligned}\mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\min}(\mathbf{x})\right) &= \mathbb{E}_q\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}\frac{P(\mathbf{x})}{q(\mathbf{x})}c_{\min}(\mathbf{x})\right) = \mathbb{E}_q\left(P(A|\mathbf{x})\,w_u(\mathbf{x})\,c_{\min}(\mathbf{x})\right)\\ &\approx \frac{1}{n}\sum_{i=1}^{n} P\left(A|\mathbf{x_i}\right) w_u\left(\mathbf{x_i}\right) c_{\min}\left(\mathbf{x_i}\right), \quad \mathbf{x_i} \sim q(\mathbf{x}),\\ \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\max}(\mathbf{x})\right) &\approx \frac{1}{n}\sum_{i=1}^{n} P\left(A|\mathbf{x_i}\right) w_u\left(\mathbf{x_i}\right) c_{\max}\left(\mathbf{x_i}\right), \quad \mathbf{x_i} \sim q(\mathbf{x}).\end{aligned} \tag{24}$$

And we record these estimations as

$$\begin{aligned}\hat{\mu}_{q_u,\min} &\overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} P\left(A|\mathbf{x_i}\right) w_u\left(\mathbf{x_i}\right) c_{\min}\left(\mathbf{x_i}\right),\\ \hat{\mu}_{q_u,\max} &\overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} P\left(A|\mathbf{x_i}\right) w_u\left(\mathbf{x_i}\right) c_{\max}\left(\mathbf{x_i}\right).\end{aligned} \tag{25}$$

Introduce confidence level λ and standard deviation function $\mathrm{std}(\cdot)$, then the bound of $P(A)$ can be estimated as

$$\begin{aligned}P(A) &\ge \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\min}(\mathbf{x})\right)\\ &\ge \hat{\mu}_{q_u,\min} - \lambda\cdot\mathrm{std}\left(\hat{\mu}_{q_u,\min}\right),\\ P(A) &\le \mathbb{E}_P\left(\frac{P(A|\mathbf{x})}{c(\mathbf{x})}c_{\max}(\mathbf{x})\right)\\ &\le \hat{\mu}_{q_u,\max} - \lambda\cdot\mathrm{std}\left(\hat{\mu}_{q_u,\max}\right).\end{aligned} \tag{26}$$

For each test $\mathbf{x}_i$, $P\left(A|\mathbf{x}_i\right) w_u\left(\mathbf{x}_i\right)$ can be calculated based on the sample results. The key is to estimate the bound of $c\left(\mathbf{x_i}\right)$. For test $\mathbf{x}_i$, using c_{ik} to represent the normalization coefficient of $q_u\left(\mathbf{u}_k|\mathbf{s}_k\right)$ at the kth time step:

$$c_{ik} = \int q_u\left(\mathbf{u}_k|\mathbf{s}_k\right)\mathrm{d}\mathbf{u}_k. \tag{27}$$

It is straightforward to see that $q\left(\mathbf{u}_k|\mathbf{s}_k\right) = \frac{q_u(\mathbf{u}_k|\mathbf{s}_k)}{c_{ik}}$. At time steps that are not critical, i.e., $k \notin T_{i,C}$, we have $P\left(\mathbf{u}_k|\mathbf{s}_k\right) = q\left(\mathbf{u}_k|\mathbf{s}_k\right) = q_u\left(\mathbf{u}_k|\mathbf{s}_k\right)$ and therefore $c_{ik} = 1$. So for the entire sequence, $c\left(\mathbf{x}_i\right)$ is given by the product $\prod_{k\in T_{i,C}} c_{ik}$. Further, we calculate c_{ik} as

$$\begin{aligned}c_{ik} &= K_1\cdot H_1(\mathbf{s}(k)) + K_2\cdot H_2(\mathbf{s}(k))\\ &= \left(K_1 - 1\right)\cdot H_1(\mathbf{s}(k)) + K_2,\end{aligned} \tag{28}$$

where

$$\begin{aligned}H_1(\mathbf{s}(k)) &= \int_{\mathbf{U}_C} P(\mathbf{u}|\mathbf{s}(k))\mathrm{d}\mathbf{u},\\ H_2(\mathbf{s}(k)) &= 1 - H_1(\mathbf{s}(k)).\end{aligned} \tag{29}$$

$\mathbf{U}_c$ denotes the set of critical actions.

It is evident that H_1 and H_2 cannot be directly obtained, as the probability distribution $P(\mathbf{u}|\mathbf{s}(k))$ is an implicit distribution, making precise integration calculations difficult. Therefore, we relax the conditions and proceed with the maximum and minimum values of H_1 for subsequent computations. Under the assumption that the maximum and minimum values of H_1 are known, we

continue with the analysis. And the determination of the maximum and minimum values of H_1 will be discussed later. Due to c_{ik} being linear respect to H_1, the minimum and maximum values of c_{ik} are respectively

$$\begin{aligned} \min c_{ik} &= \left(K_1 - 1\right) \cdot \min H_1 + K_2, \\ \max c_{ik} &= \left(K_1 - 1\right) \cdot \max H_1 + K_2. \end{aligned} \tag{30}$$

Then

$$\begin{aligned} c\left(x_i\right) &= \prod_{k \in T_{i,C}} c_{ik} \geq \prod_{k \in T_{i,C}} \min c_{ik} = \left(\min c_{ik}\right)^{T_{i,C}}, \\ c\left(x_i\right) &= \prod_{k \in T_{i,C}} c_{ik} \leq \prod_{k \in T_{i,C}} \max c_{ik} = \left(\max c_{ik}\right)^{T_{i,C}}, \end{aligned} \tag{31}$$

where superscript $T_{i,C}$ denote the size of set $T_{i,C}$. Define

$$\begin{aligned} c_{\min}\left(\mathbf{x}_i\right) &\overset{\text{def}}{=} \left(\min c_{ik}\right)^{T_{i,C}}, \\ c_{\max}\left(\mathbf{x}_i\right) &\overset{\text{def}}{=} \left(\max c_{ik}\right)^{T_{i,C}}. \end{aligned} \tag{32}$$

Then the bound of $P(A)$ in Eq. (26) can be calculated.

Eqs. (26),(30) and (32) provide practical methods for calculating the bound of $P(A)$. And according to these three equations, we also find that the key to control the bound of $P(A)$ is to choose suitable K_1, K_2 to make $\min c_{ik}$ and $\max c_{ik}$ close to 1. Given the range of H_1, we need to choose K_1, K_2 by minimize $(\min c_{ik} - 1)$ and minimize $(\max c_{ik} - 1)$. However, these two optimization objectives are inherently conflicting: as one of values of $\min c_{ik}$ and $\max c_{ik}$ approaches 1, the other would deviate from 1. Therefore, a balance needs to be considered by carefully selecting the K_1 and K_2. In the experimental section, we will provide an explanation and clarification of the parameters used in this paper.

Up to this point, we have presented the methodology of this paper. However, the process of obtaining the minimum and maximum values of H_1 was not fully explained. To address this, we now provide a detailed description of how these values are derived. Specifically, we perform simulations based on NDE prior to the accelerated testing, and identify the critical states where $C(\mathbf{s}) > 0$ in the simulations. Once a critical state is identified, we sample extensively from NDE to obtain actions $\mathbf{u}$ given the state, and thus approximate the empirical distribution of $P(\mathbf{u}|\mathbf{s})$ when $C(\mathbf{s}) > 0$. Then, based on the estimated criticality value, we identify the critical actions that $P(A|\mathbf{u}, \mathbf{s})$ is no-zero. Finally, using these critical actions and their probabilities, we approximate the value of H_1

$$H_1\left(\mathbf{s}\left(k\right)\right) = \int_{\mathrm{U}_C} P\left(\mathbf{u}|\mathbf{s}\left(k\right)\right) \mathrm{d}\mathbf{u} \approx \sum_{\mathbf{u} \in \mathbf{u}_c} P\left(\mathbf{u}|\mathbf{s}\left(k\right)\right), \tag{33}$$

and then obtain the approximation of the minimum and maximum values of H_1. The process of calculating H_1 is independent of the testing process and does not occur concurrently with the testing. Therefore, it does not affect the testing efficiency.

To enhance the process of bias analysis, a flowchart is provided to visually represent the key variables and the computational formulas used in our method, as shown in Fig. 3.

4. Experimental studies

To validate our method, we adapted our implicit importance sampling approach to two different NDE models based on existing works (Feng et al., 2023; Yan et al., 2023). We refer to these environments as NDE-I and NDE-II in this paper. Since the NDEs were generated based on naturalistic driving data, the evaluation results are representative of real-world scenarios. And the adaptability and generalizability of our approach were verified, as different types of NDE models and varied road geometries were utilized in these experiments.

The testing method involved conducting extensive simulations on the NDEs, gathering test scenarios, and calculating the proportion of scenarios where the AV encountered accidents. Additionally, naturalistic and adversarial scenarios were generated using our proposed method, and the probability of accidents occurring for the AV was weighted and estimated. We compared the accident rate results and testing efficiency from both methods, validating the effectiveness of our method.

NDE-I is modeled with a discrete distribution $P(\mathbf{u}|\mathbf{s})$ and we assume that the distribution is unknown by the accelerated testing approach. Specifically, data analysis is conducted using a naturalistic driving dataset, where both vehicle state values and action values are discretized into state–action pair. Then the empirical probability of state–action pair can be obtained from naturalistic driving dataset using statistical methods. $P(\mathbf{u}|\mathbf{s})$ at time step k is modeled as

$$P\left(\mathbf{u}\left(k\right)|\mathbf{s}\left(k\right)\right) = \prod_{i=1}^{N} P\left(\mathbf{u}_i\left(k\right)|\mathbf{s}\left(k\right)\right), \tag{34}$$

where N is the number of vehicles. The $P\left(\mathbf{u}_i\left(k\right)|\mathbf{s}\left(k\right)\right)$ is further simplified by assuming spatial independence:

$$P\left(\mathbf{u}\left(k\right)|\mathbf{s}\left(k\right)\right) = \prod_{i=1}^{N} P\left(\mathbf{u}_i\left(k\right)|\mathbf{s}_{N_i}\left(k\right)\right), \tag{35}$$

where N_i denotes the vehicles that have dependencies with ith vehicle. Finally, $P\left(\mathbf{u}_i\left(k\right)|\mathbf{s}_{N_i}\left(k\right)\right)$ is calculated by the empirical probability of state–action pair.

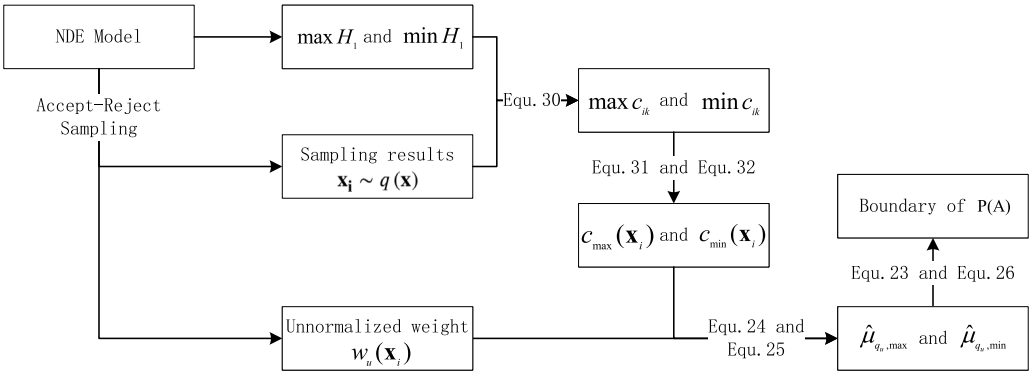


Fig. 3. Computational Flowchart for bias analysis.

According to explicit and discrete distributions, the true value of the distribution $q(\mathbf{u}|\mathbf{s})$ can be easily obtained through normalization, allowing us to calculate the true weight values. We compared the experimental results using both the true weights and the unnormalized estimated weights from IIS in NDE to demonstrate the validity of our method. All experiments on NDE-I were conducted on a two-lane highway with a 400-meter driving distance for the AV. The tested AV was constructed based on the Intelligent Driver Model (IDM) (Treiber et al., 2000) for longitudinal control and the Minimizing Overall Braking Induced by Lane Changes (MOBIL) model (Kesting et al., 2007) for lane-changing behavior. Each test episode concluded either when the AV reached the 400-meter mark or an accident occurred. The road network and representative scenarios of the NDE are illustrated in Fig. 4(a). In the figure, the red vehicle represents the tested AV, while the yellow vehicles represent the background vehicles.

On the other hand, NDE-II model is constructed using a neural network, which is trained on naturalistic driving data. The model consists of a backbone network built on the transformer architecture, along with a safety-layer that combines the transformer network with complex rules to adjust the output of the backbone. Specifically, backbone network F_M takes the current scenario state as input and outputs the predicted trajectory for the next 5 steps, with a time step of 0.4 s. A Conflict Critic Module F_C evaluates the predicted trajectory to determine if an accident occurs at the next time step. If no accident occurs, the trajectory is accepted. If an accident occurs, an acceptance probability is used to decide whether to accept the trajectory. This acceptance probability is based on naturalistic driving dataset and ensures consistency with the accident rate in the real data. This intricate network structure makes it difficult to calculate the probability density values.

As the exact distribution value is not accessible, only the sample results can be obtained, making implicit importance sampling the sole applicable approach. We compared the experimental results using unnormalized estimated weights from IIS with results in NDE to confirm that our method is effective for problems involving implicit distributions. We conducted experiments in a roundabout environment, where the AV was controlled using vehicle trajectories directly generated from the NDE. Each test episode lasted for 36 s, concluded when an accident occurred, or when the AV exited the roundabout. The corresponding road network and scenario illustration are shown in Fig. 4(b), representing the Ann Arbor roundabout.

For performance metrics, we assessed both adversarialism and naturalism by comparing crash rates and key data distributions between IIS and NDE. In the case of NDE-I, only risky data, which constitutes a small proportion, was collected and analyzed to demonstrate that IIS generates more adversarial driving environments. For NDE-II, all driving data was collected and analyzed to prove that, despite the small amount of risk data, the generated NADE closely resembles the original NDE. Also, we compared accident rates to verify the results of accelerated testing, showing that our method can estimate accident rates more efficiently with controllable and acceptable biases.

In summary, our experiments validated three key aspects of our method:

- IIS is capable of generating NADE that increase adversarial scenarios while preserving the naturalistic properties of the original driving environment.
- IIS achieves accelerated testing by efficiently estimating accident rates, particularly in cases where the nominal distribution $P(\mathbf{x})$ is implicit.
- Our method allows for controlled bias during estimation, and with appropriate parameter selection, we demonstrate that the bias can be kept within a reliable range. Additionally, we provide guidelines for choosing parameters to ensure reliable results.

4.1. Results analysis for NDE-I

Before discussing the results, we first introduce the criticality measure $C(\mathbf{s})$ for NDE-I. We utilized a tree-search method for simple trajectory prediction. Specifically, a one-second tree search is performed for both the AV and the surrounding BVs at the current moment to evaluate whether BVs' actions could result in a traffic accident involving the AV. If an action is determined to potentially cause an accident, the corresponding BVs' action at the current moment is considered risky, with $P(A|\mathbf{u},\mathbf{s}) \neq 0$; otherwise, $P(A|\mathbf{u},\mathbf{s}) = 0$.

(a) (b)

Fig. 4. Overview of traffic scenarios in two NDEs from a bird's-eye perspective.

As for parameters K_1 and K_2, we set $K_1 = 100, K_2 = 0.99$ and $K_1 = 500, K_2 = 0.99$ respectively, to ensure that c_{ik} approaches 1. And the range of H_1 was determined to be $\left(1 \times 10^{-7}, 5 \times 10^{-5}\right)$. The corresponding values of $\left(\min c_{ik}, \max c_{ik}\right)$ being approximately $(0.9900099, 0.99495)$ for $K_1 = 100$ and $(0.9900499, 1.01495)$ for $K_1 = 500$, according to Eq. (30). Given that c_{ik} remains close to 1, we can ensure that the bias is kept within a controllable range.

Although both K_1 and K_2 are involved, K_2 is consistently set to 0.99, so the analysis in the paper focuses primarily on the variations in K_1. For validation, we conducted approximately 5×10^6 test episodes for the parameter setting $K_1 = 100$, and 6×10^5 test episodes for the setting $K_1 = 500$, in the naturalistic and adversarial driving environments, while the corresponding number of episodes in NDE was approximately 2×10^8. Fig. 5 visualizes the key data distributions with $K_1 = 500, K_2 = 0.99$, comparing NDE with the NADE generated by IIS.

The first column of figures shows the distribution of crash types and near-miss incidents involving AV. We use time-to-collision (TTC) and bumper-to-bumper distance to assess proximity between AV and background vehicles (BVs). It is evident that the crash rate in NADE reached 2.45×10^{-05}, significantly higher than the NDE's crash rate of 1.58×10^{-7}. This result demonstrates that the environment generated by IIS is more adversarial. Additionally, the near-miss distances and TTC values in the IIS-generated environment tend to cluster at lower values, further explaining the increased frequency of accidents. Traffic flow distributions were altered, as the sample distribution $q(\mathbf{x})$ replaced the nominal distribution $P(\mathbf{x})$.

One advantage of importance sampling is its ability to provide unbiased estimates by applying weights to the samples. By applying the true weight $w(\mathbf{x})$, we modified the distributions as shown in the second column of figures, bringing them closer to the original NDE distribution. This indicates that importance sampling maintains unbiasedness within a certain level of accuracy. However, in our work, $P(\mathbf{x})$ is treated as an implicit distribution, and only approximate, unnormalized weights can be calculated. As a result, the estimated outcomes inherently carry bias. Despite this, our experiments show that the bias remains small under the given parameters, as demonstrated in the final column of figures.

We also evaluated the crash rate estimation using IIS. A confidence level of $\lambda = 0.95$ was chosen for calculating bounds and confidence intervals. Fig. 6 presents the evaluation results under two parameter settings. Fig. 6(a) and Fig. 6(d) depict the crash rate progression over time, with the shaded areas representing confidence intervals. Fig. 6(b) and Fig. 6(e) display the relative half-width (RHW) metric (Zhao et al., 2017), which is used to measure efficiency. The minimum number of test episodes required to reach a precision threshold (RHW=0.3) was computed. For NDE, approximately 2.03×10^8 episodes were required. With $K_1 = 100$, the number of tests required in the environment generated by IIS was reduced to 2.34×10^6, accelerating the evaluation by a factor of 87. And the experimental results showed that the estimated crash rate converges to the very similar value. When $K_1 = 500$, only 4.32×10^5 tests were needed, accelerating the evaluation by a factor of 470. However, this higher efficiency came at the cost of increased yet still controlled bias in the crash rate estimate.

The results above demonstrate that our method significantly accelerates the evaluation process while providing a reliable estimation of the crash rate, even with some degree of bias. Then we further analyze the efficiency, the range of bias, and the relationship between these outcomes and the parameter K_1. We computed the bounds for the estimated crash rate and displayed them in Fig. 7, where the shaded area represents the bounds of $P(A)$ calculated using Eq. (26). While a larger K_1 increases the probability of encountering riskier scenarios, it also results in a looser bound on $\min c_{ik}$, which eventually lead to large bound of $c(\mathbf{x})$ and result $P(A)$. Thus, $K_1 = 500$ achieved greater efficiency but with a slightly larger bias. The optimal value of K_1 should be selected based on the trade-offs between efficiency and accuracy.

4.2. Results analysis for NDE-II

NDE-II is based on a complex neural network, which can predict the vehicle states for the next five time steps. We calculate $P(A|\mathbf{u}, \mathbf{s})$ based on the trajectory prediction results. If a predicted trajectory indicates a collision with the AV, we consider that the model output at the current state and action satisfies $P(A|\mathbf{u}, \mathbf{s}) > 0$. Multiple samples are generated at the same state, and if any predicted trajectory results in a collision, the current state is considered risky, namely $C(\mathbf{x}) > 0$.

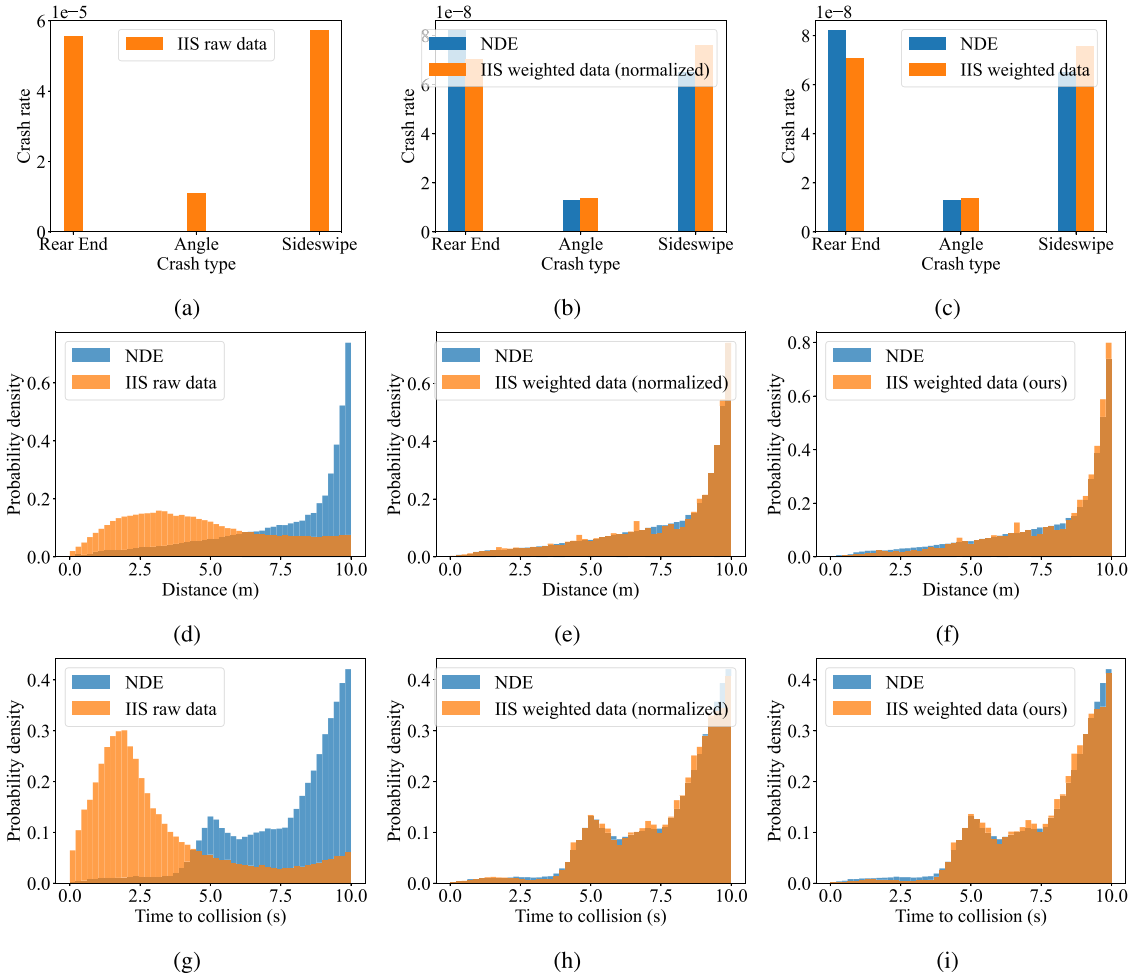


Fig. 5. Naturalistic and adversarial driving environment generation for NDE-I.

We set the parameters as $K_1 = 50, K_2 = 0.99$ and $K_1 = 100, K_2 = 0.99$, respectively. And the range of H_1 was determined to be $\left(4.5 \times 10^{-5}, 5 \times 10^{-3}\right)$. The corresponding values of $\left(\min c_{ik}, \max c_{ik}\right)$ are approximately $(0.992205, 1.137)$ for $K_1 = 50$ and $(0.994455, 1.287)$ for $K_1 = 100$. Given that c_{ik} remains close to 1, we can ensure that the bias is kept within a controllable range.

For validation, we collected approximately 4×10^5 test episodes under both parameter settings in NADE, while about 1.3×10^7 episodes were tested in NDE. In Fig. 8, we visualized the distribution of key data with $K_1 = 50, K_2 = 0.99$.

The crash rate in NADE reached 3.49×10^{-4}, significantly higher than in NDE (7.51×10^{-6}), demonstrating the adversarial nature of the generated environment. After applying the weighting adjustments, the crash rate in NADE was modified to 6.98×10^{-6}, as shown in Fig. 8(b). This modification results in a slight bias due to the use of unnormalized weights, but it remains within a reasonable range for practical purposes. Fig. 8(c), Fig. 8(b), and Fig. 8(e) show the distribution of distance between vehicles and vehicle speeds, respectively. The KL-divergence values of the distributions were 0.002, 0.004, and 0.003, respectively, indicating that the generated NADE closely resembles NDE. This is reasonable because only the behavior of critical vehicles at specific moments, representing an extremely small proportion of the data, was altered, while the overall driving environment remained consistent with naturalistic conditions.

Similar to the analysis in Section 4.1, we estimated the accident rates and plotted the results in Fig. 9. In NDE, approximately 4.26×10^6 tests were required to reach the desired precision. With $K_1 = 50$, our method required only 8.61×10^4 tests, accelerating the evaluation by a factor of 49. For $K_1 = 100$, only 5.10×10^4 tests were required, resulting in a speed-up of 83 times.

Although a larger K_1 leads to higher evaluation efficiency, it also introduces a slight bias in the results. We further calculated the bounds of the estimated crash rates, which are visualized in Fig. 10, showing the trade-off between efficiency and accuracy when choosing different values of K_1.

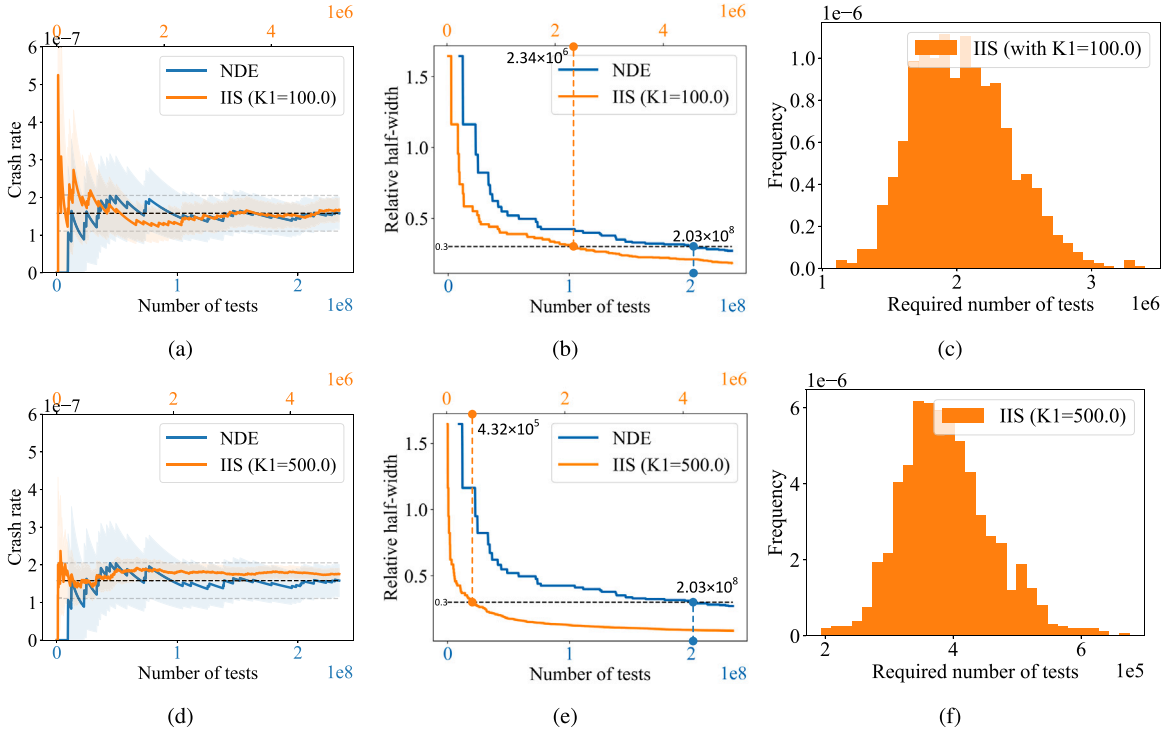


Fig. 6. AV performance evaluation for NDE-I.

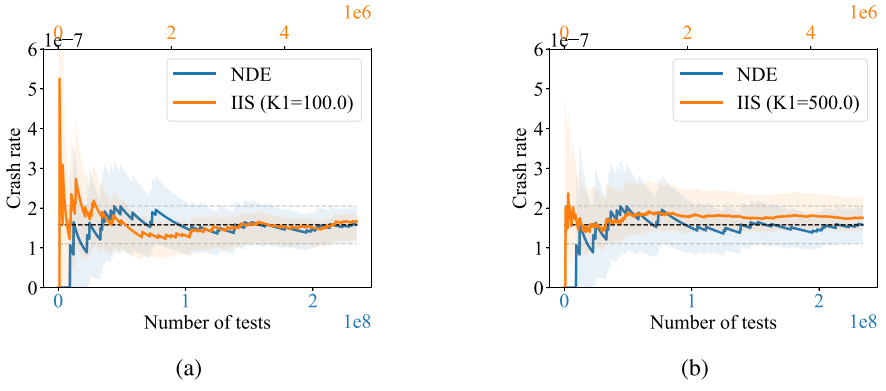


Fig. 7. Bias analysis of AV performance evaluation results for NDE-I.

5. Conclusion and future work

In this paper, we introduce a novel Implicit Importance Sampling (IIS) approach designed to enhance the intelligent testing environment for autonomous vehicles (AVs). Our method generates testing environments that are more adversarial, which allows for accelerated testing and provides reliable and representative evaluation results. We validated our approach through experiments on two different NDE models, demonstrating its scalability and effectiveness. Our method is effective for any NDE model that testing scenarios can be sampled, even if the NDE model is implicit or not well-defined. This versatility highlights the potential of IIS to address common challenges in testing not only autonomous vehicles but also other intelligent agents in complex environments.

Future work will focus on improving the proposal distribution design, including reducing bias by enhancing the parameter selection process. We plan to explore the integration of neural networks for the proposal distribution design in future studies. Specifically, we aim to investigate the use of neural networks to output parameters K_1 and K_2, and optimize the current proposal distribution. This enhancement will make our method more robust and accurate, leading to more reliable testing outcomes for autonomous vehicles and other intelligent systems.

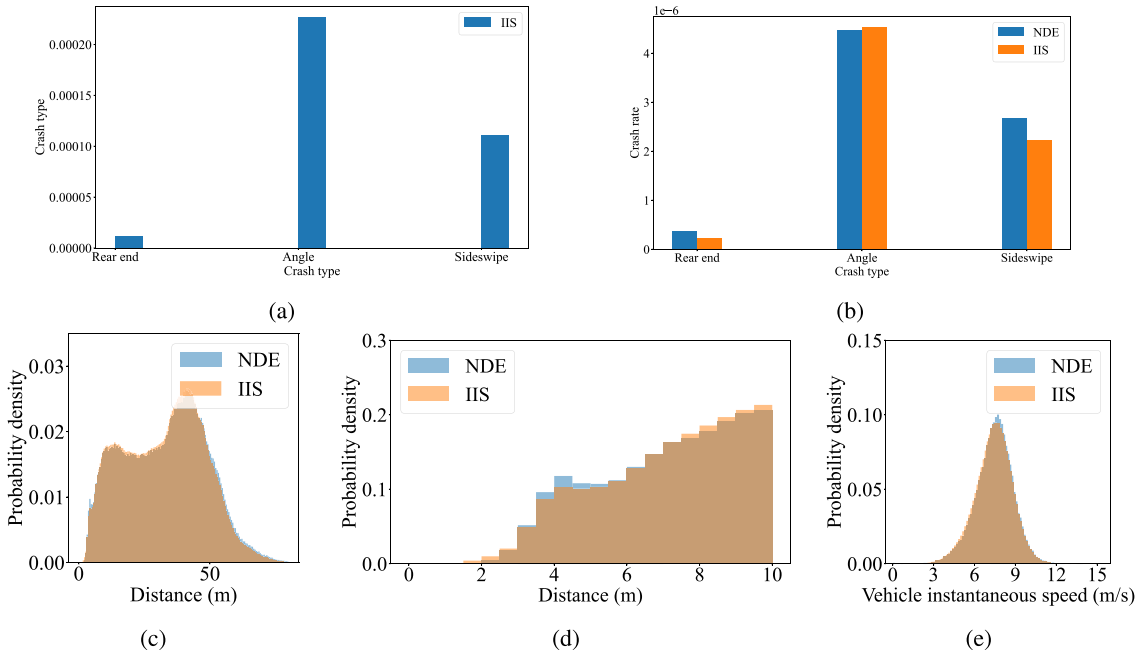


Fig. 8. Naturalistic and adversarial driving environment generation for NDE-II.

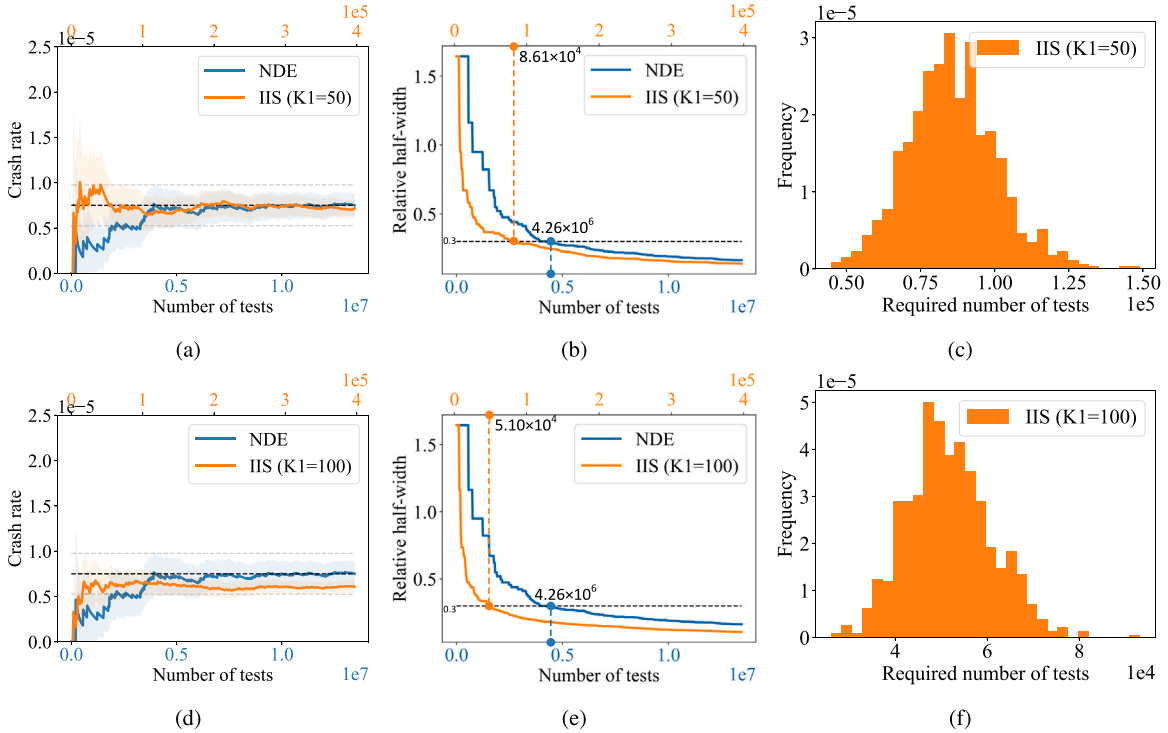


Fig. 9. AV performance evaluation for NDE-II.

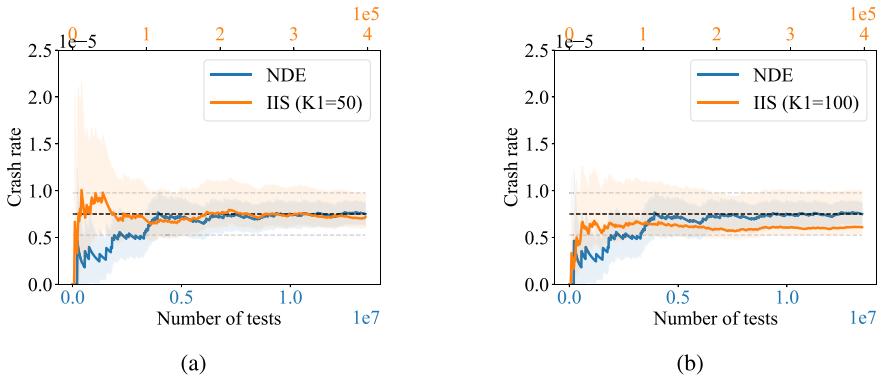


Fig. 10. Bias analysis of AV performance evaluation results for NDE-II.

CRediT authorship contribution statement

Kun Ren: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Jingxuan Yang:** Writing – review & editing, Visualization, Methodology. **Qiujing Lu:** Writing – review & editing, Visualization, Methodology. **Yi Zhang:** Writing – review & editing, Validation. **Jianming Hu:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Shuo Feng:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Acknowledgments

This work is supported by National Natural Science Foundation of China No. 62473224 and No. 62333015, Beijing Natural Science Foundation, China 4244092 and L231014, and Beijing Nova Program, China 20230484259 and 20240484642.

References

Arief, M., Cen, Z., Liu, Z., Huang, Z., Li, B., Lam, H., Zhao, D., 2022. Certifiable evaluation for autonomous vehicle perception systems using deep importance sampling (deep is). In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1736–1742.

Cancela, H., El Khadiri, M., Rubino, G., 2009. Rare event analysis by Monte Carlo techniques in static models. Rare Event Simul. Monte Carlo Methods 145–170.

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.

Delyon, B., Portier, F., 2016. Integral approximation by kernel smoothing. Bernoulli 22 (4), 2177–2208. http://dx.doi.org/10.3150/15-BEJ725.

Ding, W., Xu, C., Arief, M., Lin, H., Li, B., Zhao, D., 2023. A survey on safety-critical driving scenario generation—A methodological perspective. IEEE Trans. Intell. Transp. Syst. 24 (7), 6971–6988.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. In: Conference on Robot Learning. PMLR, pp. 1–16.

Duan, S., Bai, X., Shi, Q., Li, W., Zhu, A., 2024. Uncertainty evaluation for autonomous vehicles: A case study of AEB system. Automot. Innov. 1–14.

Fang, J., Zhou, D., Yan, F., Zhao, T., Zhang, F., Ma, Y., Wang, L., Yang, R., 2020. Augmented LiDAR simulator for autonomous driving. IEEE Robot. Autom. Lett. 5 (2), 1931–1938.

Feng, S., Feng, Y., Sun, H., Bao, S., Zhang, Y., Liu, H.X., 2021a. Testing scenario library generation for connected and automated vehicles, part II: Case studies. IEEE Trans. Intell. Transp. Syst. 22 (9), 5635–5647. http://dx.doi.org/10.1109/TITS.2020.2988309.

Feng, S., Feng, Y., Yu, C., Zhang, Y., Liu, H.X., 2021b. Testing scenario library generation for connected and automated vehicles, Part I: Methodology. IEEE Trans. Intell. Transp. Syst. 22 (3), 1573–1582. http://dx.doi.org/10.1109/TITS.2020.2972211.

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. http://dx.doi.org/10.1038/s41586-023-05732-2, Publisher: Nature Publishing Group. URL https://www.nature.com/articles/s41586-023-05732-2.

Feng, S., Yan, X., Sun, H., Feng, Y., Liu, H.X., 2021c. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. Nat. Commun. 12 (1), 748.

Filos, A., Tigkas, P., McAllister, R., Rhinehart, N., Levine, S., Gal, Y., 2020. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In: International Conference on Machine Learning. PMLR, pp. 3145–3153.

Hanselmann, N., Renz, K., Chitta, K., Bhattacharyya, A., Geiger, A., 2022. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII. Springer, pp. 335–352.

Hao, K., Cui, W., Luo, Y., Xie, L., Bai, Y., Yang, J., Yan, S., Pan, Y., Yang, Z., 2023. Adversarial safety-critical scenario generation using naturalistic human driving priors. IEEE Trans. Intell. Veh.

Henmi, M., Yoshida, R., Eguchi, S., 2007. Importance sampling via the estimated sampler. Biometrika 94 (4), 985–991.

Huang, Z., Arief, M., Lam, H., Zhao, D., 2019. Evaluation uncertainty in data-driven self-driving testing. In: 2019 IEEE Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 1902–1907.

Jiang, Z., Pan, W., Liu, J., Dang, S., Yang, Z., Li, H., Pan, Y., 2022. Efficient and unbiased safety test for autonomous driving systems. IEEE Trans. Intell. Veh. 8 (5), 3336–3348.

Jiang, Y., Zhao, D., Zhu, B., Liu, Z., Zhao, X., 2024. Driving segment embedding and patterns dictionary generation from real-world data using self-supervised learning. Automot. Innov. 1–12.

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.
Kesting, A., Treiber, M., Helbing, D., 2007. General lane-changing model MOBIL for car-following models. Transp. Res. Rec. 1999 (1), 86–94.
Koren, M., Kochenderfer, M.J., 2019. Efficient autonomy validation in simulation with adaptive stress testing. In: 2019 IEEE Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 4178–4183.
Krajzewicz, D., 2010. Traffic simulation with SUMO–simulation of urban mobility. Fundam. Traffic Simul. 269–293.
Kruber, F., Wurst, J., Botsch, M., 2018. An unsupervised random forest clustering technique for automatic traffic scenario categorization. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2811–2818.
Lee, R., Mengshoel, O.J., Saksena, A., Gardner, R.W., Genin, D., Silbermann, J., Owen, M., Kochenderfer, M.J., 2020. Adaptive stress testing: Finding likely failure events with reinforcement learning. J. Artificial Intelligence Res. 69, 1165–1201.
Li, W., Pan, C., Zhang, R., Ren, J., Ma, Y., Fang, J., Yan, F., Geng, Q., Huang, X., Gong, H., et al., 2019. AADS: Augmented autonomous driving simulation using data-driven algorithms. Sci. Robot. 4 (28), eaaw0863.
Liu, H.X., Feng, S., 2024. Curse of rarity for autonomous vehicles. Nat. Commun. 15 (1), 4808.
Liu, L., Feng, S., Feng, Y., Zhu, X., Liu, H.X., 2022. Learning-based stochastic driving model for autonomous vehicle testing. Transp. Res. Rec. 2676 (1), 54–64.
Liu, Q., Lee, J., 2017. Black-box importance sampling. In: Artificial Intelligence and Statistics. PMLR, pp. 952–961.
Lu, Q., Bai, R., Li, S., He, H., Feng, S., RACL: Risk Aware Closed-Loop Agent Simulation with High Fidelity.
Mo, Z., Shi, R., Di, X., 2021. A physics-informed deep learning paradigm for car-following models. Transp. Res. Part C: Emerg. Technol. 130, 103240.
Mooney, C.Z., 1997. Monte Carlo Simulation, no. 116, Sage.
Morris, R., Descombes, X., Zerubia, J., 1996. The ising/potts model is not well suited to segmentation tasks. In: 1996 IEEE Digital Signal Processing Workshop Proceedings. IEEE, pp. 263–266.
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, Vol. 10. p. 1.
Neumann, V., 1951. Various techniques used in connection with random digits. pp. 36–38, Notes By GE Forsythe.
Niu, H., Ren, K., Xu, Y., Yang, Z., Lin, Y., Zhang, Y., Hu, J., 2023. (Re)2H2O: Autonomous driving scenario generation via reversely regularized hybrid offline-and-online reinforcement learning. In: 2023 IEEE Intelligent Vehicles Symposium. IV, pp. 1–8. http://dx.doi.org/10.1109/IV55152.2023.10186559.
Oates, C.J., Girolami, M., Chopin, N., 2016. Control functionals for Monte Carlo integration. J. R. Stat. Soc. Ser. B Stat. Methodol. 79 (3), 695–718. http://dx.doi.org/10.1111/rssb.12185, arXiv:https://academic.oup.com/jrsssb/article-pdf/79/3/695/49215177/jrsssb_79_3_695.pdf.
O'Hagan, A., 1987. Monte Carlo is fundamentally unsound. Stat. 247–249.
Owen, A.B., 2013. Monte Carlo Theory, Methods and Examples. Stanford.
Puterman, M.L., 1990. Markov decision processes. Handbooks Oper. Res. Management Sci. 2, 331–434.
Rempe, D., Philion, J., Guibas, L.J., Fidler, S., Litany, O., 2022. Generating useful accident-prone driving scenarios via a learned traffic prior. In: Conference on Computer Vision and Pattern Recognition. CVPR.
Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., Diermeyer, F., 2020. Survey on scenario-based safety assessment of automated vehicles. IEEE Access 8, 87456–87477.
Scanlon, J.M., Kusano, K.D., Daniel, T., Alderson, C., Ogle, A., Victor, T., 2021. Waymo simulated driving behavior in reconstructed fatal crashes within an autonomous vehicle operating domain. Accid. Anal. Prev. 163, 106454.
Sun, H., Feng, S., Yan, X., Liu, H.X., 2021. Corner case generation and analysis for safety assessment of autonomous vehicles. Transp. Res. Rec. 2675 (11), 587–600.
Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E 62 (2), 1805.
Wang, W., Zhao, D., 2018. Extracting traffic primitives directly from naturalistically logged data for self-driving applications. IEEE Robot. Autom. Lett. 3 (2), 1223–1229.
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.
Zhang, H., Sun, J., Tian, Y., 2023. Accelerated risk assessment for highly automated vehicles: Surrogate-based monte carlo method. IEEE Trans. Intell. Transp. Syst..
Zhang, H., Zhou, H., Sun, J., Tian, Y., 2022. Risk assessment of highly automated vehicles with naturalistic driving data: A surrogate-based optimization method. In: 2022 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 580–585.
Zhao, S., Duan, J., Wu, S., Gu, X., Li, C., Yin, K., Wang, H., 2023. Genetic algorithm-based SOTIF scenario construction for complex traffic flow. Automot. Innov. 6 (4), 531–546.
Zhao, D., Lam, H., Peng, H., Bao, S., LeBlanc, D.J., Nobukawa, K., Pan, C.S., 2017. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. IEEE Trans. Intell. Transp. Syst. 18 (3), 595–607. http://dx.doi.org/10.1109/TITS.2016.2582208.