Accelerated Verification of Autonomous Driving Systems based on Adaptive Subset Simulation

Ye Tian, Member, *IEEE,* Aohui Fu, He Zhang, Lanyue Tang, Jian Sun*

***Abstract*—A throughout safety verification of an Autonomous Driving System (ADS) is essential to ensure its capability to drive safely in naturalistic traffic environment. Commonly used test automation methods such as Important Sampling (IS) do not perform well in distribution fitting efficiency in high-dimensional scenarios. In this paper, we proposed the Adaptive Subset Simulation (ADSS) method for safety verification for ADS. Two different systems under test were adopted: Baidu Apollo and Intelligent Driver Model (IDM). ADSS finds scenarios that are more and more critical iteratively and is capable to derive the accurate collision rate in naturalistic driving environment with fewer testing resources. A 3-dimensional car-following scenario and a 6-dimensional cut-in scenario were constructed in LGSVL to evaluate the performance of ADSS. The results showed that as the number of scenario dimensions increases, ADSS exhibits a more significant advantage over IS and vanilla Subset Simulation (SS) in terms of result accuracy and acceleration efficiency. In the 6-dimensional cut-in scenario, the efficiency of ADSS was 112 times faster than that of IS and 2 times faster than that of SS. Notably, ADSS demonstrated excellent testing effectiveness for both the complex ADS Apollo and the simple ADS IDM. These findings highlight the significance of utilizing ADSS in the safety verification of ADS.**

***Index Terms*—Autonomous Driving System, Subset Simulation, Accelerated Testing Method, Safety Verification, Test Automation**

I. INTRODUCTION

Autonomous vehicles (AVs) equipped with ADS will play an increasingly important role in the future, resulting in ever more complex traffic systems [1-3]. Due to the increase in complexity of the traffic system and the possible traffic scenarios, the safety and reliability verification of the ADS has become a serious challenge [4, 5], which is crucial for the large-scale deployment of the ADS on open roads.

Due to the fact that the probability of the occurrence of high-risk events or corner cases is extremely low in a naturalistic driving environment, it is extremely time-consuming and monetary-intensive to verify the safety of AVs on open roads [6]. To compensate for its limitations, simulated scenario-based testing methods have been proposed in recent years [7]. From the perspective of efficiency, although the number of traffic scenarios is theoretically infinite, customized scenario-based testing allows us to only focus on test-worthy scenarios and avoid wasting time on inconsequential cases. It also avoids any physical damage to vehicle under test completely. From the perspective of authenticity and reliability, state-of-the-art simulation test platforms have functions such as environment rendering and vehicle dynamics simulation. They can be adopted to evaluate ADS from a full-stack, end-to-end perspective. It is capable of testing driver-assistant functions such as Autonomous Emergency Braking (AEB), Adaptive Cruise Control (ACC), as well as fully functional ADSs such as Baidu Apollo [8] and Autoware [9].

Simulated scenario-based testing methods hold the advantages of high efficiency. However, it can still be intractable when dealing with all possibilities of driving environment. In order to maintain consistency with the naturalistic driving environment, Monte Carlo simulation is widely used to sample scenarios from the real-world data distribution. However, rare safety-critical cases in the real world cause the inefficiency of Monte Carlo simulation. To solve such limitations, variation reduction techniques is commonly adopted to help reduce the required simulation test runs.

Two possible variation reduction methods to achieve a more efficient sampling are Importance Sampling (IS) and Subset Simulation (SS). The idea of IS is to transfer the original probability density distribution of the scenario to a region where rare events are more likely to occur, so to achieve unbiased estimation by adjusting the samples' weight. Arief et al. proposed a framework called Deep Probabilistic Accelerated Evaluation to design statistically guaranteed IS, to achieve accurate estimation of bounds on the safety-critical event probability, and proved its accelerated effectiveness in the safety test of ADS [10-12]. In addition, IS has been proven to be an effective accelerated evaluation method in lane-following scenarios [13] and cut-in scenarios [14]. However, IS requires prior knowledge of SUT's failure region beforehand to determine the important sampling distribution, which is challenging since ADS is generally a black box to testers. Furthermore, IS performs unsatisfactorily in high-dimension scenarios, where it loses efficiency and correctness in estimating the qualified IS distribution.

SS is a method used to estimate the extremely low probability of failure in engineering systems. It is widely used in structural reliability problems in buildings, aerospace, and nuclear systems [15-17]. The concept of SS is shown in **Figure 1**. The

*Research supported by the National Natural Science Foundation of China under Grant number [52422215] and [52172391].

The authors are all with the Department of Traffic Engineering and Key Laboratory of Road and Traffic Engineering, Ministry of Education, Tongji University. Shanghai, China. 201804. (Corresponding author: Jian Sun, sunjian@tongji.edu.cn)

parameter space is iteratively explored via the stratified sampling method. A new subset is created in each iteration and the samples gradually moves closer to the region of interests (i.e., the failure region). Compared with Monte Carlo methods and IS, SS requires fewer tests and shows better performance when dealing with high-dimensional models and black-box models [18].

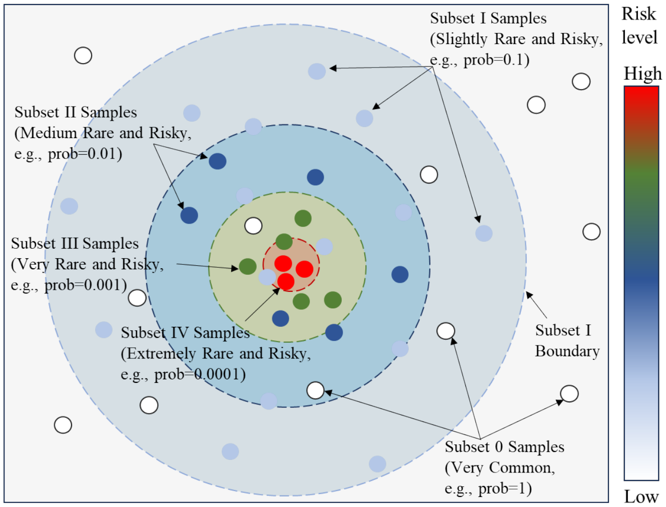


Figure 1 Illustration of subset simulation. In each iteration, subset containing rarer and more risky scenarios is sampled. Eventually, samples converge to the region with the highest risk level (i.e., the failure events).

Despite its advantages, SS also encounters issues with lower efficiency and accuracy in high-dimensional problems. Because of the limited subset selection in high-dimensional space and the complex characteristics and boundaries of ADS, SS will result in inefficient sampling. Due to the fixed sampling parameters used in the sampling process of SS, the acceptance rate of samples decreases and sample correlation increases as the dimensionality increases. These issues lead to biased estimations of the probability of rare events.

To overcome the limits of vanilla SS and address the high-dimensional issues, a more efficient and precise approach is required. We propose the idea of Adaptive SS (ADSS). It can address challenges in high-dimensional problems and explore the parameter space more effectively by dynamically altering subsets, enhancing estimating efficiency and accuracy. During the sampling process, the variance of the sampling distribution can be dynamically adjusted through scaling coefficients to optimize the acceptance rate. ADSS's flexibility and intelligence allow it to better deal with complexity in high-dimensional space, making it more useful in ADS testing.

In this paper, both the vanilla SS and ADSS is applied to the accelerated safety verification of Baidu Apollo and the Intelligent Driver Model (IDM). They are two representative ADSs. Baidu Apollo is one of the most advanced data-driven ADSs and one of the most widely used open-source ADSs in the industry. While IDM is a widely used rule-based driving model in academia, especially in adaptive cruise control systems [19, 20]. Two Systems Under Test (SUTs) are connected with the simulator LGSVL [21], which can provide a high-fidelity test environment. In the simulation environment, the measured SUT is bridged with the ego vehicle and undertakes tasks such as perception, planning, prediction, and control. ADSS, SS are evaluated with IS as benchmark. Their performances in scenarios of different dimensions (a 3-dimensional car-following scenario and a 6-dimensional cut-in scenario) are assessed.

The contributions of this work is bi-folded:

- The co-simulation of mainstreamed ADS based on a high-fidelity simulation platform is developed to emulate the real road traffic situation, which ensures the high reliability of the simulation testing results.
- The idea of ADSS is proposed, which explores the parameter space more effectively by dynamically altering subsets. The ADSS's and SS's performance is compared with the IS as the benchmark. We prove that the advantage of ADSS and SS on accuracy and acceleration effect is more significant when the number of scenario dimensions increases based on multiple replicate experiments.

The paper is organized as follows: In Section II, the concept and simulation procedure of SS and ADSS are presented. Section III introduces the design of the scenario-based testing method and simulation platform. Section IV discusses simulation results using ADSS, SS and IS on two SUTs. The last section concludes our findings.

II. Scheme For Adaptive Subset Simulation

In this section, the scheme of ADSS method is reviewed. The original and best-known stochastic simulation algorithm for estimating expectation is Monte Carlo Simulation (MCS). In MCS, the failure event probability, which is the probability p_F of AV collision in this paper, is estimated by the sample mean:

$$p_F \approx \hat{p}_F^{MC} = \frac{1}{N}\sum_{i=1}^{N} I_F\left(\theta^{(i)}\right) \qquad (1)$$

where $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(N)}$ are the independent distributed samples from base distribution $p(\theta)$, and I_F is the indicator function about whether the vehicle has collided. It is worth mentioning that the sample $\theta^{(i)} = (x_1, x_2, \dots, x_n)$ is the scenario to be tested, n represents the variable dimension of the scenario input. The basic distribution $p(\theta)$ is the distribution obtained from natural driving scenario data fitting. The main advantage of MCS is that its efficiency does not depend on the dimension d of the parameter space. Thus, it can even operate in high-dimensional parameter space. The measure of accuracy in usual interpretation of MCS is coefficient of variation (c.o.v), which is calculated by:

$$\delta(\hat{p}_F^{MC}) = \sqrt{\frac{1-p_F}{Np_F}} \qquad (2)$$

However, MCS has a serious drawback: when the probability of the events p_F is extremely small, number of samples N needed to achieve an acceptable level of confidence is extremely large. Therefore, accelerated testing method is needed.

The basic concept of SS is as follows: regard an extremely small failure probability p_F as a product of relatively larger probabilities $p_F = \prod_{j=1}^{m} p_j$, where $p_j(j = 1{:}m)$ are estimated sequentially. Then an estimate $\hat{p}_F^{SS}$ for p_F could be obtained as $\hat{p}_F^{SS} = \prod_{j=1}^{m} \hat{p}_j$. To reach this goal, a function $Y(\theta)$ for expectation is considered and let $F = \{\theta{:}\, Y(\theta) < 0\}$ denote the set of failure events (also known as the collision events in this study). Then the intermediate event sets can be expressed as $F = \{\theta{:}\, Y(\theta) < b_m\}$, where b_m represents intermediate failure threshold of each subset and $Y(\theta)$ is an indicator of the vehicle's level of danger during the test. The specific calculation steps of $Y(\theta)$ are shown in Eq.(3). If a collision occurs, $Y(\theta)$ equals to -1. If a collision does not occur, $Y(\theta)$ equals to the minimum time to collision (TTC) of the vehicle within the entire simulation run. Note that TTC is adopted herein as the safety indicator since such indicator has been widely adopted to represent the driving risk. The generalizability of the proposed ADSS method could be show by using TTC as the safety indicator. Note that TTC can be replaced by other safety indicators such as $mTTC$ (modified TTC), TTB (Time to Brake), etc.

$$Y(\theta) = \begin{cases} -1, & \text{if Ego collides} \\ \text{TTC}, & \text{if Ego does not collide} \end{cases} \tag{3}$$

Next, we take a decreasing sequence of nested subsets of the parameter space, starting from the entire space and shrinking to the failure region F:

$$\mathbb{R}^d = F_0 \supset F_1 \supset \cdots \supset F_{m-1} \supset F_m = F. \tag{4}$$

Subsets $F_1, \ldots, F_{m-1}$ are termed as intermediate failure regions. Then the failure probability p_F can be written as a product of conditional probabilities:

$$p_F = \prod_{j=1}^{m} P\left(F_j \mid F_{j-1}\right) = \prod_{j=1}^{m} p_j \tag{5}$$

where $p_j = P\left(F_j \mid F_{j-1}\right)$ is the conditional probability at the $j - 1^{th}$ conditional level. With that, the original problem (i.e., estimation of the small failure probability p_F) is turned into a sequence of m intermediate problems corresponding to evaluating larger conditional probabilities. Each conditional probability is denoted as:

$$p_j \approx \hat{p}_j^{MC} = \frac{1}{N}\sum_{i=1}^{N} I_{F_j}\left(\theta_{j-1}^{(i)}\right), \theta_{j-1}^{(i)} \overset{\text{i.i.d.}}{\sim} \pi\left(\cdot \mid F_{j-1}\right) \tag{6}$$

In SS, the first probability $p_1 = P(F_1 \mid F_0) = P(F_1)$ is estimated by MCS directly. Then, for $j \geq 2$, one needs to sample from conditional distribution $\pi\left(\cdot \mid F_{j-1}\right)$, which is not a trivial task. Markov chain Monte Carlo (MCMC) is commonly used to resolve this problem at the expense of generating dependent samples. The procedures of SS is illustrated in **Figure 2**.

A. Modified Metropolis Algorithm

MCMC is a class of algorithms for sampling from multi-dimensional target probability distributions that cannot be directly sampled or not efficiently sampled [22]. The Modified Metropolis algorithm (MMA) is widely adopted as the MCMC method due to the following advantages: (1) The MMA utilizes an adaptive acceptance criterion to determine whether to accept new sampled scenarios. This criterion can dynamically adjust based on the importance and probability distribution of the scenarios, enabling a balance between global exploration and local focus. As a result, the algorithm can generate representative test cases efficiently. (2) The MMA circumvents the issue of detailed balance in the original Metropolis algorithm [23] by sampling from a custom probability distribution. This allows for smoother sampling during the generation of test scenarios, better reflecting the distribution characteristics of scenarios in a natural driving environment.

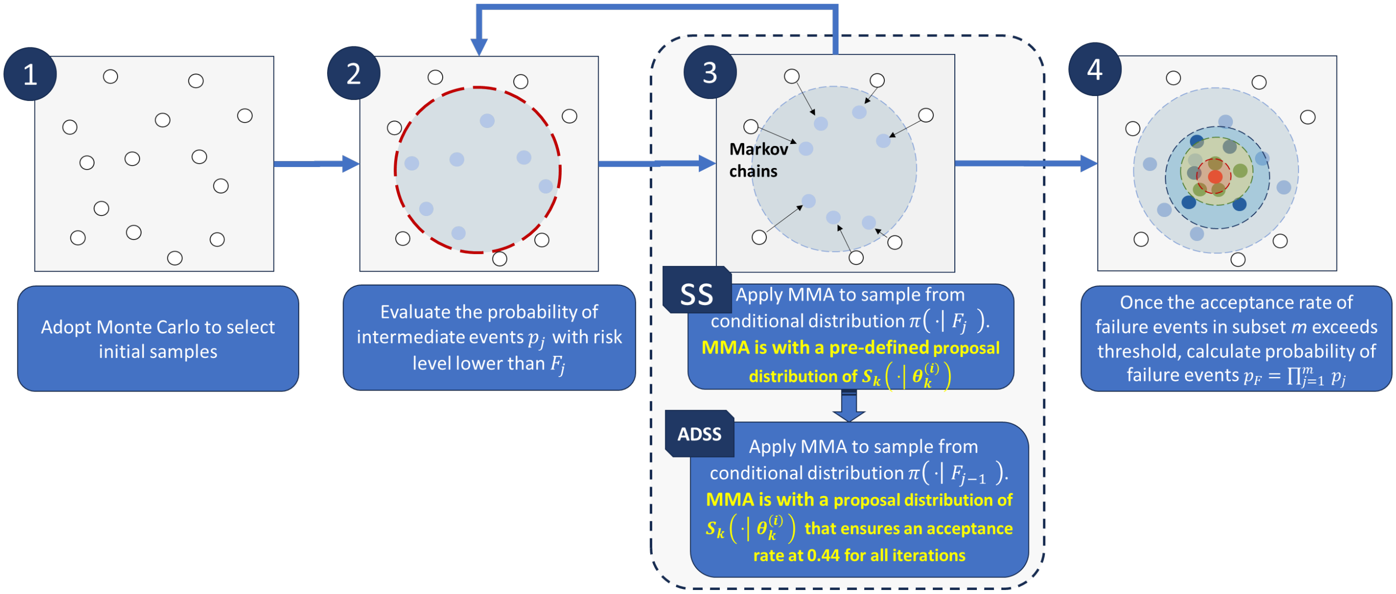

1
2
3
4
Markov chains
Adopt Monte Carlo to select initial samples
Evaluate the probability of intermediate events p_j with risk level lower than F_j
SS
Apply MMA to sample from conditional distribution $\pi(\cdot \mid F_j)$.
MMA is with a pre-defined proposal distribution of $S_k\left(\cdot \mid \theta_k^{(i)}\right)$
ADSS
Apply MMA to sample from conditional distribution $\pi(\cdot \mid F_{j-1})$.
MMA is with a proposal distribution of $S_k\left(\cdot \mid \theta_k^{(i)}\right)$ that ensures an acceptance rate at 0.44 for all iterations
Once the acceptance rate of failure events in subset m exceeds threshold, calculate probability of failure events $p_F = \prod_{j=1}^{m} p_j$

Figure 2: Procedures of vanilla SS and ADSS.

(3) The MMA exhibits high efficiency and flexibility when generating representative subsets of test scenarios. It can

explore large scenario spaces efficiently and can be expanded and adjusted according to specific requirements. Consequently, it is applicable to SS in various scales and complexities of autonomous driving scenarios.

During the calculation, the MMA algorithm uses the seed samples determined from the previous subset, and evolves according to proposal distribution $S_k\left(\cdot\mid\theta_k^{(i)}\right)$, which corresponds to k^{th} dimension of the original distribution $\pi_k(\cdot)$. To summarize, the Modified Metropolis algorithm proceeds as follows:

```
Modified Metropolis algorithm
Input: $\theta^{(1)} \in \mathbb{F}$, initial state of a Markov chain;
       $N^s$, total states of Markov Chain
       Original distribution: $\pi_k(\cdot)$ for dimension $k$;
       Proposal distribution: $S_k\left(\cdot\mid\theta_k^{(i)}\right)$ for dimension $k$;
Algorithm:
for $i = 1,\ldots,N^s - 1$ do
    % Generate a candidate state $\xi$:
    for $k = 1,\ldots,d$ do
        Sample $\tilde{\xi}_k \sim S_k\left(\cdot\mid\theta_k^{(i)}\right)$
        Compute the acceptance ratio
        $r = \frac{\pi_k(\tilde{\xi}_k)}{\pi_k\left(\theta_k^{(i)}\right)}$          (7)
        Accept or reject $\tilde{\xi}_k$ by setting
        $\xi_k = \begin{cases} \tilde{\xi}_k, & \text{with probability } min\{1,r\} \\ \theta_k^{(i)} & \text{with probability } 1 - min\{1,r\} \end{cases}$          (8)
    end for
    Check whether $\xi \in \mathbb{F}$ by test, accept or reject $\xi$ by:
        $\theta^{(i+1)} = \begin{cases} \xi, & \text{if } \xi \in \mathbb{F} \\ \theta^{(i)}, & \text{if } \xi \notin \mathbb{F}. \end{cases}$          (9)
end for
Output: $\theta^{(1)},\ldots,\theta^{(N)}$, $N$ states of a Markov chain.
```

From the perspective of the MMA sampling process, the probability of generating duplicate candidates in each dimension is non-zero and it depends on the one-dimensional proposal distribution. As the scenario dimension n increases, the probability of simultaneously having duplicate candidates across all dimensions decreases dramatically, resulting in candidates ξ that are consistently different from the current state. Therefore, MMA demonstrates a certain degree of applicability in high-dimensional problems. However, as the dimensionality increases, MMA often faces challenges such as reduced acceptance rates and increased correlations between samples. With the rise in dimensionality, the probability of accepting proposed states in the MMA algorithm rapidly diminishes, leading to a decrease in samples falling into the target region F and an increase in correlations between samples. These challenges can result in biasness in probability estimates, as illustrated in the following equation:

$$E\left[\frac{\hat{P}_f - P_f}{P_f}\right] = \sum_{i>j} \delta_i\delta_j E\left[\mathbb{Z}_i\mathbb{Z}_j\right] + \sum_{i>j>k} \delta_i\delta_j\delta_k E\left[\mathbb{Z}_i\mathbb{Z}_j\mathbb{Z}_k\right] + \cdots + \left(\prod_{i=1}^{m} \delta_i\right) E\left[\prod_{i=1}^{m} \mathbb{Z}_i\right] \tag{10}$$

where δ_i denotes the c.o.v. of $\widehat{p_\iota}$ and $\mathbb{Z}_i$ is calculated as $\mathbb{Z}_i = (\widehat{p}_i - p_i)/\delta_i$. Generally, $\left[\mathbb{Z}_i\mathbb{Z}_j\right], E\left[\mathbb{Z}_i\mathbb{Z}_j\mathbb{Z}_k\right] for\ i > j > k$, are not equal to zero due to the fact that Z_i is mutually correlated. As a result, $\widehat{P_f}$ is biased.

B. Adaptive Subset Simulation with Optimal Scaling

From the perspective of the sampling process in subset simulation, the performance of the algorithm relies on the choice of the variance in the proposal distribution $S_k\left(\cdot\mid\theta_k^{(i)}\right)$ within the MMA algorithm. A large variance of the proposal distribution will result in the rejection of many candidate samples, while a small variance will lead to high correlation between states. Therefore, enhancing the performance of the MMA algorithm can be achieved by adaptively adjusting the corresponding parameters during the sampling process.

To gain further insights into the performance of the MMA algorithm in high dimensions, it becomes necessary to introduce an efficiency metric for quantification. Typically, the efficiency of the MMA algorithm is defined in terms of diffusion velocity, i.e., the speed at which the Markov chain converges to genetic moment estimates [24]. For example, in MMA, when constructing a one-dimensional Markov chain process θ_t for each dimension of the variables, efficiency is computed by considering the reciprocal of the integral of the autocorrelation function of θ_t. This implies that maximizing efficiency is equivalent to minimizing the correlation of the chain [25].

In the context of SS, MMA is applied to estimate each conditional probability $Pr\left(F_j\mid F_{j-1}\right)$, where $j = 2,\ldots,M$. The coefficient of variation δ_j can be calculated using the following equation:

$$\delta_j = \sqrt{\frac{1-P_j}{NP_j}\left(1+\gamma_j\right)} \tag{11}$$

where:

$$\gamma_j = 2\sum_{k=1}^{N/N_S-1}\left(1-\frac{kN_s}{N}\right)\rho_j(k) \tag{12}$$

$N_s = p_0N$ represents the number of seeds in the MCMC sampling at subset level j, where p_0 is a fixed intermediate failure probability. $N/N_s = 1/p_0$ is the length of each chain, and $\rho_j(k)$ denotes the average k-lag autocorrelation coefficient of the stationary sequence $\left\{I_{F_j}\left(\theta_{j-1}^{((l-1)/p_0+t)}\right): t = 1,\ldots,N/N_f\right\}, l = 1,\ldots,N_s$. This sequence represents the autocorrelation of the Markov chain samples.

As γ_j increases, the autocorrelation of the Markov chain samples also increases, leading to a decrease in the accuracy of subset simulation. Therefore, we can define the following performance metric:

$$eff_{\Upsilon} = \left(1 + \gamma_j\right)^{-1} \tag{13}$$

This metric can be used to compare the efficiency of the MMA algorithm during the sampling process. Specifically, it indicates that to achieve the same variance as the Monte Carlo method, the variance of $\widehat{P_j}$ needs to be multiplied by the factor $\left(1+\gamma_j\right)^{-1}$. An efficiency factor closer to 1 suggests higher algorithm efficiency, as it implies a smaller variance adjustment factor is required. Maximizing efficiency is equivalent to minimizing the correlation of the chain, i.e., minimizing γ_j.

The MMA algorithm samples from the proposal distribution $S(\cdot|\,\theta)$, where the acceptance rate depends on whether the candidate state lies within the failure region, determined by the functional value of the candidate state. Gelman et al. proposed that the adjustment of acceptance rates should adhere to a fundamental principle: maintaining acceptance rates between 30% and 70%. The basic principle behind this is that a low acceptance rate implies a large number of redundant samples in the Markov chain, while a high acceptance rate indicates a extremely slow movement of the Markov chain [25]. As mentioned above, minimizing the factor γ_j corresponds to maximizing the efficiency measurement eff_{Υ}. Zuev et al. reported that the factor γ_j reaches optimal when the acceptance rate is between 0.3 and 0.5 [22]. Through numerical experiments, Iason et al. demonstrated that, under the condition of 0.1 intermediate failure probability, for the problem of using a one-dimensional normal distribution as the proposal distribution, the optimal acceptance rate is approximately 0.44 [26]. Experiments on 4 performance functions (a linear function, a convex function, a concave functions, and a hypersphere functions) were conducted to verify the optimal value of acceptance rate. It turns out 0.44 works well to find the failure cases in all 4 performance functions.

Based on the research on the efficiency metrics and optimal acceptance rate of the MMA algorithm mentioned above, we propose dynamically adjusting the standard deviation σ_k of the one-dimensional proposal distribution $S_k(\cdot|\,\theta_k)$ to maintain its acceptance rate close to the optimal value of 0.44.

In SS, we need to use samples $\left\{\theta_j^{(k)}: k = 1, \dots, N_s\right\}$ falling into F_j in simulation subset j as seeds to simulate N_s Markov chains to obtain samples of $\pi(\cdot|\,F_j)$ for subset $j+1$. The idea of ADSS is to simulate Markov chains step by step. In each step, N_a chains, occupying a portion of N_s, are simulated using the same standard deviation σ_k. After completing the simulation of N_a chains, the standard deviation σ_k of the proposal distribution is adjusted based on the estimated acceptance rates of the previous N_a chains. The seeds for simulating N_a chains are randomly selected (without replacement) from N_s seeds to ensure uniform convergence on the chain stepping, which guarantees the asymptotic unbiasedness of subset simulation.

In the process of ADSS, a set of initial values are selected for the standard deviation of the proposal distribution, denoted as $\sigma_{0k}, k = 1, \dots, n$. Additionally, an initial scaling parameter $\lambda_1 \in (0,1)$ is determined. The number of chains N_a after which the proposal distribution will be adapted is selected such that N_s/N_a is an integer. The adaptive process is carried out at each iteration step $iter = 1, \dots, N_s/N_a$. In each adaptive step $iter$, the standard deviation σ_k of the proposal distribution for each component is calculated by scaling the initial value σ_{0k} using the scaling coefficient λ_{iter}. However, each σ_k cannot exceed the standard deviation of the corresponding random variable, which is 1.0. Therefore, σ_k is adaptively adjusted at each step $iter$ using the following equation:

$$\sigma_k = \min(\lambda_{iter}\sigma_{0k}, 1.0) \tag{14}$$

Then N_a seeds are randomly selected from $\left\{\theta_j^{(i)}: i = 1, \dots, N_s\right\}$, and conditional sampling is conducted to simulate the corresponding Markov chains with standard deviation σ_k. Subsequently, the average acceptance rate of the Markov chains is evaluated using the following Eq. (15):

$$\hat{a}_{iter} = \frac{1}{N_a}\sum_{i=1}^{N_a} \hat{E}_{\xi}\left[a\left(\theta_j^{(i)}\right)\right] \tag{15}$$

where $\hat{E}_{\xi}\left[a\left(\theta_j^{(i)}\right)\right]$ is the average acceptance sample possessed by the chain of seed sample $\theta_j^{(i)}$. Then, adjustment of the scaling coefficient λ_{iter} is updated based on Eq. (16):

$$\log\lambda_{\mathrm{iter}+1} = \log\lambda_{\mathrm{iter}} + \zeta_{\mathrm{iter}}\left[\hat{a}_{\mathrm{iter}} - a^*\right] \tag{16}$$

where a^* is the pre-determined optimal acceptance rate which is set to 0.44. ζ_{iter} is a positive real number intended to ensure that the variation of λ_{iter} gradually diminishes and eventually converges to 0. Here, we set it as $\zeta_{iter} = iter^{-1/2}$. From Eq. (16), it can be inferred that if the chain's average acceptance rate is less than 0.44, the variance of each one-dimensional conditional normal distribution decreases, whereas if it exceeds 0.44, the variance of each one-dimensional normal distribution increases.

For the initial values of the standard deviation σ_{0k} during the iterative process of the conditional sampling algorithm, it is set as the sample variance of the seed samples in the corresponding subset. The sample mean and standard deviation for each dimension of the seed samples are calculated using the following equations:

$$\widehat{\mu_k} = \frac{1}{N_s}\sum_{i=1}^{N_s} \theta_{jk}^{(i)} \tag{17}$$

And

$$\hat{\sigma}_k^2 = \frac{1}{N_s - 1}\sum_{i=1}^{N_s}\left(\theta_{jk}^{(i)} - \hat{\mu}_k\right)^2 \tag{18}$$

The smaller the standard deviation $\widehat{\sigma}_k$ corresponding to a random variable, the greater its impact on the limit state function. By selecting the initial values σ_{0k} of the standard deviation in the conditional sampling algorithm adaptively, dimensions with larger impacts are sure to move over a wider range. Compared with SS using a fixed proposal distribution $S_k\left(\cdot|\,\theta_k^{(i)}\right)$ for sampling, using adaptive scaling coefficients helps maintain acceptance rates extremely close to the optimal value, 0.44.

All source code of vanilla SS and ADSS could be found at : https://github.com/f2133397/AdaptiveSubsetSimualtion

The conditional sampling method corresponding to ADSS is summarized as follows:

```
Adaptive Conditional Sampling Method
Input: $\theta^{(1)} \in \mathbb{F}$, initial state of a Markov chain;
        $N_s$, total states of Markov Chain
        $\lambda_1$(Initial scaling parameter)
        Original distribution: $\pi_k(\cdot)$ for dimension k;
        Proposal distribution: $S_k\left(\sigma_{jk} \mid \theta_k^{(i)}\right)$ for dimension k;
Algorithm:
Apply Eq. (14) and Eq. (15). Set $\sigma_{0k} = \hat{\sigma}_k, k = 1, \dots, d$
for $iter = 1, \dots, N_s/N_a$ do
    for $i = (iter - 1)N_a + 1, \dots, iter * N_a$ do
        for $k = 1, \dots, d$
            Sample $\tilde{\xi}_k \sim S_k\left(\sigma_{jk} \mid \theta_k^{(i)}\right)$
            Compute the acceptance ratio
                $r = \frac{\pi_k(\tilde{\xi}_k)}{\pi_k\left(\theta_k^{(i)}\right)}$    (19)
            Accept or reject $\tilde{\xi}_k$ by setting
                $\xi_k = \begin{cases} \tilde{\xi}_k, & \text{with probability } min\{1, r\} \\ \theta_k^{(i)} & \text{with probability } 1 - min\{1, r\} \end{cases}$    (20)
        end for
        Check whether $\xi \in \mathbb{F}$ by test, accept or reject $\xi$ by:
                $\theta^{(i+1)} = \begin{cases} \xi, & \text{if } \xi \in \mathbb{F} \\ \theta^{(i)}, & \text{if } \xi \notin \mathbb{F}. \end{cases}$    (21)
    end for
    Evaluate the average acceptance rate $\hat{a}_{iter}$ of the last $N_a$ chains using Eq. (15).
    Compute the new scaling parameter $\lambda_{\text{iter}+1}$ using Eq. (16).
end for
Output: $\theta^{(1)}, \dots, \theta^{(N_s)}$, $N_s$ states of a Markov chain.
```

III. Scenario-Based Test And Simulation Platform

A. Scenario-based Testing

Two typical scenarios: a 3-dimensional car-following scenario and a 6-dimensional cut-in scenario are constructed. The capabilities of different approaches (IS, SS, and ADSS) are compared from three aspects: accuracy, efficiency and robustness. Before the safety verification of the ADS, the parameter distribution of the scenario must conform to the naturalistic driving environment, that is, the base distribution *p(x)* must be fitted beforehand.

Two Gaussian Mixed Models (GMMs) were established using 993 cut-in scenarios and 30,292 car-following scenarios extracted from highD, which is a large-scale naturalistic vehicle trajectory dataset from German highways [27]. GMM is commonly used as a parametric model of the probability distribution for multivariate data with an arbitrarily complex probability density function (PDF).

1) 3-Dimensional Car-following Scenario: The car-following scenario is the most basic functional scenario that AVs must handle, and rear-end collision is one of the most common collision types in real world. Two vehicles are defined in the logic scenario, one is the ego vehicle equipped with ADS, while the other is NPC (non-player character) vehicles which moves with a fixed speed (see **Figure 3**).

2) 6-Dimensional Cut-in Scenario: A cut-in scenario is defined as a lane-change maneuver of a NPC vehicle that starts in an adjacent lane and ends in the ego's lane, which is commonly seen and may cause severe collisions. We introduce three participants in a Cut-in scenario: the Ego vehicle, the Cut-in NPC vehicle, and the leading NPC vehicle. The parameters contain the initial speeds of three vehicles (v_1, v_2, v_3), the initial longitudinal gaps (S_{1x}, S_{2x}) and the initial lateral gap of (dis_{1y}) (see **Figure 4**). During the cut-in scenario, the cut-in trajectory is fitted by a three-order Bezier curve.

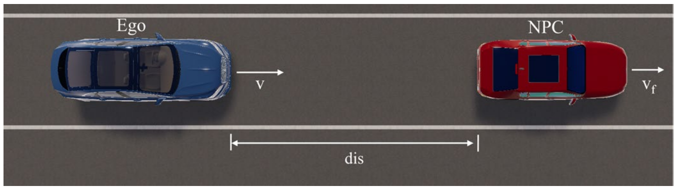


Figure 3. 3-Dimensional Car-following Logical Scenario.

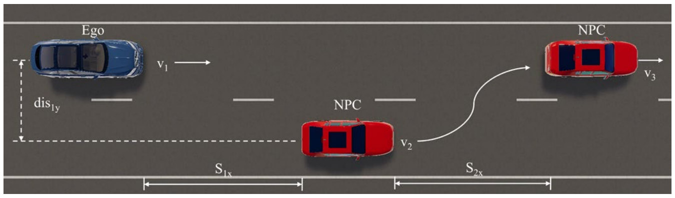


Figure 4. 6-Dimensional Cut-in Logical Scenario.

B. Design of the Simulation Platform

We aim to conduct the testing of ADSs in a sophisticated simulation environment. The following describes the ADS test framework we established, following the procedures of producing high-definition map, bridging between ADS and simulator, calibration of simulation parameter, and analyzing of simulation results.

1) Producing High-definition maps: High-definition maps play an important role in autonomous driving simulation testing. In this study, we used roadrunner and Unity to create various formats of high-definition maps, which can meet the requirement of the cross-platform usage between LGSVL and Apollo.

2) Co-Simulation Between ADS and Simulator: In the designed simulation experiment, the simulator provides a highly realistic simulation environment. As SUTs, ADSs need to complete specific driving tasks in the simulation scenario. We examine the risk levels of their driving behaviors during the simulation process. The simulator utilizes 3D rendering engine to accomplish the simulation of the real traffic environment. In addition, it simulates the functionality of the perception sensors of the autonomous vehicles, using which the surrounding environment including moving vehicles could be detected. During this process, ADSs receive the sensor perception results from the simulator, then the prediction, motion planning, control and other modules in the ADS workflow will return the corresponding control decisions back into the simulator to achieve co-simulation between environment and SUT. The interaction between the simulator and the ADSs under test is shown in **Figure 5**.

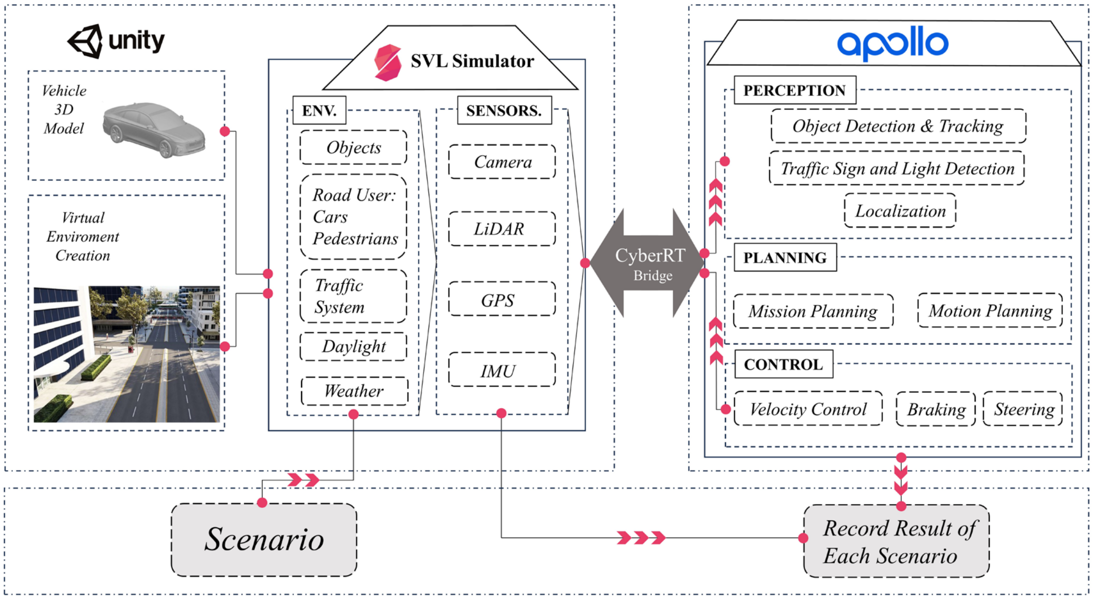


Figure 5. The Co-Simulation between LGSVL and SUT.

3) Simulation Setups: In the simulation experiments, the LGSVL simulator was adopted to emulate the real-world environment and vehicle dynamics. Baidu Apollo 7.0 and IDM were chosen as SUTs. During the simulation process, the vehicle under test can obtain the obstacle information (including the speed and position of the obstacle) at every moment through the sensors. In addition, in order to avoid the situation of collision-free caused by the braking principle of IDM, we set a parameter b_{max} as the hard-coded maximum deceleration, which is consistent with that of Apollo. Other parameters of IDM can be found in Table I [28].

TABLE I PARAMETERS OF IDM MODEL

Name	Description	Value
V_d	Desired velocity	21.7 m/s
T	Safe time gap	1.2 s
a	Maximum acceleration	2.22 m/s
b	Comfortable deceleration	2.4 m/s^2
δ	Acceleration exponent	4
S_0	Minimum distance in congested traffic	1 m
S_1	Safety distance exponent	2 m
b_{max}	Maximum deceleration	6 m/s^2

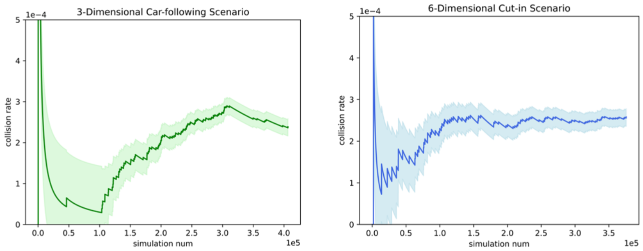
Experiment was run on a desktop with the following specifications:

- Ubuntu version 18.04
- Intel Core i7-11700 CPU @ 2.50GHz × 16
- 64GB RAM
- GeForce RTX 3080 PCIe4.0/DLSS
- LGSVL simulator 2021.3 (linux64) with modular testing setup (3D Ground Truth sensor and Signal sensor publish ground truth perception data to Apollo via CyberRT bridge)
- Baidu Apollo (7.0.0)

4) Collection of simulation results: With the help of LGSVL Python API, a collision callback function was configured to monitor if the ego vehicle collides with other objects during the simulation process. In the meantime, with the 3D-ground truth information obtained by the sensor, we can calculate TTC of the ego vehicle real-time. If it collided, the location of the collision and the speed of the colliding vehicle will be recorded. Otherwise, the TTC will be returned.

IV. CASE STUDY

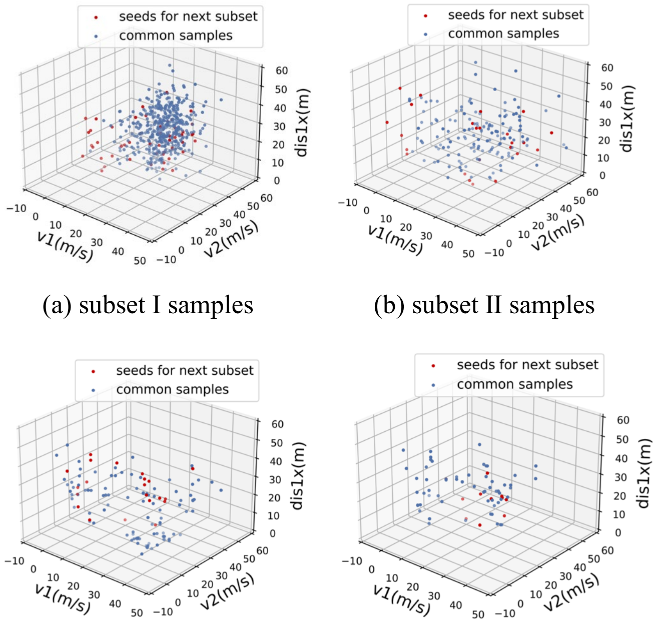
Using the base distribution previously established with GMM, we first performed MCS on the two SUTs to obtain the ground-truth collision rate and the required number of tests for the two SUTs in the two scenarios, with the relative half-width set to 0.2. For example, the performance of MCS for Apollo is shown in **Figure 6**.



(a) Car-following scenario (b) Cut-in scenario

Figure 6. Evolution of collision rates of Apollo obtained by MCS

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3449947



(a) subset I samples

(b) subset II samples

(c) subset III samples

(d) subset IV samples

Figure 7. Apollo samples tested by SS in cut-in scenario in each subset. The X-, Y-, Z- axis represent v_1, v_2, dis_{1y}, respectively.

A. Evaluate SUT using Subset Simulation

By applying initial MMA algorithm, we can sample from conditional distribution $\pi(\cdot|\, F_{j-1})$. The total number of sample in each subset is set to 500, intermediate failure threshold is set to 0.1 [22]. The proposal distribution $S_k(\cdot|\cdot)$ is normal distribution, which is the most well-studied candidate distribution. According to the test result $Y(\xi)$ of each scenario sample defined in **Eq.** (3), it can be concluded that the smaller the test results, the more dangerous it is, and the closer it is to the failure zone represented by the collision scenario. When all sample seeds of the next subset are located within the failure region, both SS and ADSS converge. The samples selected by SS in each iteration for Apollo in cut-in scenario is shown in the **Figure 7**.

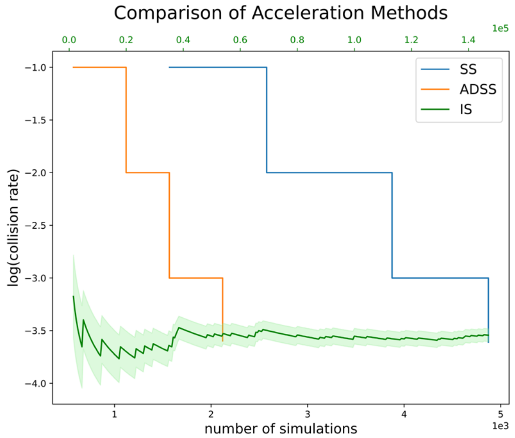


Figure 8. Test result of Apollo in cut-in scenario. ADSS outperforms SS and IS in terms of acceleration effect.

In general, v_2 increases and dis_{1x} decreases from subset I to IV. This indicates that as the speed of the cut-in vehicle increases, and the longitudinal distance between the vehicles decreases, the level of danger increases that collisions are more likely to occur.

IS with the same configuration as [14] was implemented in the above two scenarios. The performance of IS, SS, and ADSS in terms of acceleration rate in cut-in scenario are shown in **Figure 10**.

According to the experiment result, the collision rates of Apollo obtained by SS and IS in the cut-in scenario are 2.49×10^{-4} and 2.86×10^{-4}, respectively. According to the ground truth obtained by MCS, the collision rate is 2.58×10^{-4}, and the relative errors of SS and IS are 3.49% and 10.85%, respectively. SS achieves higher accuracy. From the number of tests required, the number of simulation runs required by SS is around 5,000. However, IS needs nearly 237,000 simulation runs to converge to the corresponding collision rate. It can be seen that compared with IS, the acceleration efficiency of SS increases by more than 47 times. Besides, ADSS achieved a superior acceleration effect compared to SS, approximately doubling the speed compared to SS. For ADS with simulation noise like Apollo, the acceleration effect of SS and ADSS is more obvious.

B. Robustness Comparison of Accelerated Methods

SS encounters challenges with diminishing sampling efficiency and result certainty as dimensionality of the scenario increases. Consequently, other than sampling efficiency, there is a necessity for ADSS to bolster its robustness in multiple trials to derive the collision rate.

TABLE II and TABLE III shows the performance comparison of the three test methods (IS, SS, ADSS). Results are based on 20 replications.

TABLE II PERFORMANCE OF DIFFERENT TEST METHODS FOR CAR-FOLLOWING SCENARIO

Method	**3-Dimensional Car-following**			
	SUT	***Mean of $\hat{P}_f$***	***$COV_{\hat{P}_f}$***	***Mean of N_{sim}***
MCS	Apollo	2.392×10^{-4}	0.009	4.064×10^5
SS	Apollo	2.351×10^{-4}	0.213	4.980×10^3
ADSS	Apollo	$\mathbf{2.347 \times 10^{-4}}$	**0.086**	2.681×10^3
IS	Apollo	2.013×10^{-4}	**0.026**	1.762×10^5
MCS	IDM	8.36×10^{-4}	0.004	2.351×10^5
SS	IDM	8.03×10^{-4}	0.152	1.6×10^3
ADSS	IDM	8.16×10^{-4}	**0.073**	1.533×10^3
IS	IDM	$\mathbf{8.23 \times 10^{-4}}$	**0.012**	9.569×10^4

* Mean of $\hat{P}_f$ denotes the obtained collision rate over 20 replications. $COV_{\hat{P}_f}$ denotes the variance of $\hat{P}_f$. A higher $COV_{\hat{P}_f}$ indicates lower result robustness. Mean of N_{sim} denotes the mean value of number of required simulation runs over 20 replications.

TABLE III PERFORMANCE OF DIFFERENT TEST METHODS FOR CUT-IN SCENARIO

Method	**6-Dimensional Cut-in**			
	SUT	***Mean of $\hat{P}_f$***	***$COV_{\hat{P}_f}$***	***Mean of N_{sim}***
MCS	Apollo	2.583×10^{-4}	0.013	3.412×10^{5}
SS	Apollo	2.517×10^{-4}	0.312	4.714×10^{3}
ADSS	Apollo	$\mathbf{2.546 \times 10^{-4}}$	**0.196**	2.103×10^{3}
IS	Apollo	2.142×10^{-4}	**0.097**	2.373×10^{5}
MCS	IDM	7.653×10^{-3}	0.011	2.687×10^{4}
SS	IDM	7.598×10^{-3}	0.257	5.634×10^{3}
ADSS	IDM	$\mathbf{7.613 \times 10^{-3}}$	**0.131**	2.743×10^{3}
IS	IDM	7.363×10^{-3}	**0.084**	1.765×10^{4}

One can tell that both SS and ADSS achieve higher accuracy as compared to IS, which can be shown from the smaller deviation from MCS on indicator "mean of $\hat{P}_f$". This is mainly due to the inability of IS to fully recognize the multiple separated failure regions that complex ADSs such as Apollo have. When the SUT is IDM, there is a decreasing trend in estimation error. Taking the cut-in scenario as an example, the relative error between IS and MCS has decreased from 17.1% to 3.8%. This is mainly because the test results obtained by IDM are more stable, and is not affected by simulation noise caused by communication delays between sensors, which is commonly seen in Apollo.

As the scenario dimension increases, the deviation of the IS estimation tends to increase. It can be seen from TABLE II and TABLE III that as the number of dimensions of the scenario increases from 3 to 6, the estimation error of IS increases from 15.8% to 17.1%. Meanwhile, the estimation errors of SS increase from 1.7% to 2.6%. When the actual number of conducted simulation runs, denoted as N_{sim}, serves as a metric for evaluating testing resource expenditure, discrepancies arise in the effectiveness of IS as compared to SS and ADSS. For instance, in three-dimensional car-following scenario where IS undertakes 3×10^3 simulation runs, resulting in mean of $\hat{P}_f$ at 5.793×10^{-4}, substantial deviations from MCS's estimation: 2.392×10^{-4} is observed. Consequently, under resource-constrained testing environments, IS may exhibit limited practical applicability.

From the perspective of the stability of the test results, the $COV_{\hat{P}_f}$ of IS is smaller than that of SS, indicating that the collision rate estimated by IS is more robust. In contrast, the collision rate estimated by SS fluctuates. However, the mean of $\hat{P}_f$ estimated by SS based on 20 replications is closer to MCS despite its $COV_{\hat{P}_f}$ is larger, which is mainly due to the fact it guarantees the stationary state of Markov Chain by 20 replications. Judging from the required number of simulation runs, the acceleration effect of SS is prominent as compared to IS. The higher the dimension of the scenario is, the more significant advantage of SS over IS is.

In between of SS and ADSS, from the perspective of the mean of $\hat{P}_f$, across different scenarios and different SUTs, ADSS consistently demonstrates estimates closer to those obtained from MCS. Moreover, ADSS exhibits smaller $COV_{\hat{P}_f}$, indicating greater robustness. In terms of the mean of N_{sim}, ADSS consistently outperforms SS, achieving a doubled acceleration in the 6-dimensional cut-in scenario. Compared to SS, ADSS achieves improved accuracy in estimating probabilities and significantly reduces the number of required simulations, which shows that it has achieved the goal of increasing the acceptance rate in the conditional sampling process. In comparison to SS, ADSS demonstrates superior performance by dynamically adjusting sampling parameters such as acceptance rates and standard deviations. This adaptability allows ADSS to optimize the sampling process, resulting in more accurate estimations of rare event probabilities.

V. CONCLUSIONS

In this paper, we proposed the idea of SS and ADSS and their performances were compared with SS and IS. Both ADSS and SS employ the concept of hierarchical sampling, resulting in significant acceleration effect. Compared to SS, ADSS dynamically adjusts the proposal distribution to optimize sample acceptance rates, achieving improvement in both acceleration and robustness. SS is 50 times faster as compared to IS in 6-dimensional cut-in scenario, while ADSS is 2 times faster as compared to SS.

In addition to safety validation through probability estimation of dangerous events, ADSS can also assist in identifying potential corner cases and safety hazards for autonomous driving systems. In this paper, the design of the functional function relies solely on the TTC as a measure, raising concerns about its applicability when facing other types of complex scenarios. Therefore, in various scenarios, the consideration of alternative indicators of risk severity becomes imperative. These indicators may encompass information such as vehicle speed, acceleration, road curvature, among others, offering a more comprehensive assessment of the danger level within traffic scenarios. This adjustment may enhance the accuracy and applicability of SS techniques. Besides, the inclusion of complex urban environments, uncertain interactions with non-vehicular traffic participants, and varied weather conditions would demonstrate ADSS' novelty by showcasing its adaptability to real-world driving complexities.

REFERENCES

[1] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, "Did we test all scenarios for automated and autonomous driving systems?," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019: IEEE, pp. 2950-2955.

[2] J. Sun, H. Zhang, H. Zhou, R. Yu, and Y. Tian, "Scenario-Based Test Automation for Highly Automated Vehicles: A Review and Paving the Way for Systematic Safety Assurance," *IEEE Transactions on Intelligent Transportation Systems,* pp. 1-16, 2021, doi: 10.1109/TITS.2021.3136353.

[3] Y. Huang, Y. Ye, J. Sun, and Y. Tian, "Characterizing the Impact of Autonomous Vehicles on Macroscopic Fundamental Diagrams," *IEEE Transactions on Intelligent Transportation Systems,* pp. 1-12, 2023, doi: 10.1109/TITS.2023.3265647.

[4] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety,* vol. 4, no. 1, pp. 15-24, 2016.

[5] J. Lu, C. Cui, Y. Ma, A. Bera, and Z. Wang, "Quantifying Uncertainty in Motion Prediction with Variational Bayesian Mixture," 2024.

[6] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?," *Transportation Research Part A: Policy Practice,* vol. 94, pp. 182-193, 2016.

[7] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE access,* vol. 8, pp. 87456-87477, 2020.

[8] Baidu. "Apollo open platform." https://www.apollo.auto/ (accessed.

[9] t. A. Foundation. "Autoware." https://www.autoware.org/ (accessed.

[10] M. Arief *et al.*, "Deep probabilistic accelerated evaluation: A robust certifiable rare-event simulation methodology for black-box safety-critical systems," in *International Conference on Artificial Intelligence and Statistics*, 2021: PMLR, pp. 595-603.

[11] H. Zhang, J. Sun, and Y. Tian, "Accelerated Risk Assessment for Highly Automated Vehicles: Surrogate-Based Monte Carlo Method," *IEEE Transactions on Intelligent Transportation Systems,* pp. 1-10, 2023, doi: 10.1109/TITS.2023.3335104.

[12] H. Zhang, J. Sun, and Y. Tian, "Accelerated Safety Testing for Highly Automated Vehicles: Application and Capability Comparison of Surrogate Models," *IEEE Transactions on Intelligent Vehicles,* pp. 1-10, 2023, doi: 10.1109/TIV.2023.3319158.

[13] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 3, pp. 733-744, 2017.

[14] Y. Xu, Y. Zou, and J. Sun, "Accelerated testing for automated vehicles safety evaluation in cut-in scenarios based on importance sampling, genetic algorithm and simulation applications," *Journal of intelligent connected vehicles,* vol. 1, no. 1, pp. 28-38, 2018.

[15] N. Pedroni and E. Zio, "An Adaptive Metamodel-Based Subset Importance Sampling approach for the assessment of the functional failure probability of a thermal-hydraulic passive system," *Applied Mathematical Modelling,* vol. 48, pp. 269-288, 2017/08/01/ 2017, doi: https://doi.org/10.1016/j.apm.2017.04.003.

[16] Y. Zhao and Z. Wang, "Subset simulation with adaptable intermediate failure probability for robust reliability analysis: An unsupervised learning-based approach," *Structural Multidisciplinary Optimization,* vol. 65, no. 6, p. 172, 2022.

[17] E. Zio and N. Pedroni, "Sensitivity analysis of the model of a nuclear passive system by means of Subset Simulation," *Procedia-Social Behavioral Sciences,* vol. 2, no. 6, pp. 7778-7779, 2010.

[18] F. Blandfort, "Subset Simulation and Interpolation: Efficient Reliability Estimation under Model-Dynamics for Complex Civil Engineering Structures," Technische Universität Kaiserslautern, 2021.

[19] M. Pourabdollah, E. Bjärkvik, F. Fürer, B. Lindenberg, and K. Burgdorf, "Calibration and evaluation of car following models using real-world driving data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 16-19 Oct. 2017 2017, pp. 1-6, doi: 10.1109/ITSC.2017.8317836.

[20] Y. Mei, T. Nie, J. Sun, and Y. Tian, "Bayesian Fault Injection Safety Testing for Highly Automated Vehicles with Uncertainty," *IEEE Transactions on Intelligent Vehicles,* pp. 1-15, 2024, doi: 10.1109/TIV.2024.3401051.

[21] G. Rong *et al.*, "Lgsvl simulator: A high fidelity simulator for autonomous driving," in *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)*, 2020: IEEE, pp. 1-6.

[22] K. M. Zuev, J. L. Beck, S.-K. Au, and L. S. Katafygiotis, "Bayesian post-processor and other enhancements of Subset Simulation for estimating failure probabilities in high dimensions," *Computers & Structures,* vol. 92-93, pp. 283-296, 2012/02/01/ 2012, doi: https://doi.org/10.1016/j.compstruc.2011.10.017.

[23] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," 1970.

[24] A. Gelman, W. R. Gilks, and G. O. Roberts, "Weak convergence and optimal scaling of random walk Metropolis algorithms," *The Annals of Applied Probability,* vol. 7, no. 1, pp. 110-120, 2/1 1997, doi: 10.1214/aoap/1034625254.

[25] G. A, G. O. Roberts**, and W. R. Gilks***, "Efficient Metropolis Jumping Rules," *Bayesian Statistics 5,* vol. 5, pp. 599-608, 1996-05 1996, doi: 10.1093/oso/9780198523567.003.0038.

[26] I. Papaioannou, W. Betz, K. Zwirglmaier, and D. Straub, "MCMC algorithms for Subset Simulation," *Probabilistic Engineering Mechanics,* vol. 41, pp. 89-103, 2015/07/01/ 2015, doi: https://doi.org/10.1016/j.probengmech.2015.06.006.

[27] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018: IEEE, pp. 2118-2125.

[28] J. Sun, H. Zhou, H. Xi, H. Zhang, and Y. Tian, "Adaptive design of experiments for safety evaluation of automated vehicles," *IEEE Transactions on Intelligent Transportation Systems,* 2021.

Ye Tian, IEEE member, he received the Ph.D. degree in transportation engineering from The University of Arizona, Tucson, AZ, USA, in 2015. He is currently an Associate Professor of transportation engineering with Tongji University, Shanghai, China. He serves as an Associate Editor of IEEE Transactions of Intelligent Transportation Systems. His research interests include active demand management, dynamic traffic assignment, mesoscopic traffic simulation, and safety assurance of automated vehicles.

Aohui Fu received the B.S. degree from Southeast university. He is currently a master student at Tongji University. His research interests include autonomous driving system virtual simulation, reliability verification, and autonomous driving system acceleration test.

He Zhang received the B.S. degree in transportation engineering from Southwest Jiaotong University, Chengdu, Sichuan, China. Now she is currently pursuing the Ph.D. degree with the Department of Traffic Engineering in Tongji University, Shanghai, China. Her main research interests include safety test on highly automated vehicle.

Lanyue Tang received the B.S. degree from Southeast university, she is currently a master student at Tongji University. Her research interests include human-machine interaction, artificial intelligence, autonomous vehicle, and traffic simulation.

Jian Sun received the Ph.D. degree in transportation engineering from Tongji University, Shanghai, China. He is currently a Professor of transportation engineering with Tongji University. His research interests include intelligent transportation systems, traffic flow theory, AI in transportation, and traffic simulation.