Research Article

Heterogeneous Driver Modeling and Corner Scenarios Sampling for Automated Vehicles Testing

Jingwei Ge,1 Huile Xu,2,3 Jiawei Zhang,1,4 Yi Zhang,1,4,5 Danya Yao,1,5 and Li Li1

1Department of Automation, Tsinghua University, Beijing 100084, China
2National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
3Momenta, Beijing, China
4Tsinghua-Berkeley Shenzhen Institute (TBSI), Shenzhen 518055, China
5Jiangsu Province Collaborative Innovation Center of Modern Urban Traffic Technologies, Nanjing 210096, China

Correspondence should be addressed to Li Li; li-li@tsinghua.edu.cn

Received 18 January 2022; Accepted 2 April 2022; Published 11 May 2022

Academic Editor: Yuchuan Du

Virtual simulation-based testing of autonomous vehicles (AVs) needs massive challenging corner cases to reach high testing accuracy. Current methods achieve this goal by finding testing scenarios with low sampling frequency in the empirical distribution. However, these methods neglect modeling heterogeneous driving behavior, which actually is crucial for finding corner cases. To fill this gap, we propose an interpretable and operable method for sampling corner cases. Firstly, we initialize a testing scenario and allocate testing tasks to AVs. Then, to simulate the variability in driving behaviors, we design utility functions with several hyperparameters and generate aggressive, conservative, and normal driving strategies by adjusting hyperparameters. By changing the heterogeneous driving behavior of surrounding vehicles (SVs), we can sample the challenging corner cases in the scenario. Finally, we conduct a series of simulation experiments in a typical lane-changing scenario. The simulation results reveal that by adjusting the occurrence frequency of heterogeneous SVs in the testing scenario, more corner cases can be found in limited rounds of simulations.

1. Introduction

Autonomous vehicle (AV) testing is defined as the technique for testing whether an AV possesses the intelligence to cope with a complex traffic environment and make the right decisions [1]. The testing approach focuses on observing the behavior of an AV performing a fixed strategy in a selected testing scenario and evaluating its level of intelligence or functionality [2, 3]. As the degree of challenge in the testing scenario increases, the difficulty of the test increases proportionally [4].

The most straightforward testing approach is on-road testing [5], where AVs may directly encounter challenging scenarios of varying difficulty. Hence, the test results are highly realistic. However, the inherent problem of on-road testing is to manually create challenging scenarios and the inability to estimate or control the timing and frequency of obtaining challenging scenarios. According to the report [6], we need more than 8 billion miles of road testing of autonomous vehicles to achieve safety performance assessment results with a 95 percent confidence level. The report directly points out the fact that most scenarios encountered by autonomous vehicles in on-road tests are repetitive and worthless because of the inability to control for the emergence of challenging scenarios.

Luckily, virtual simulation testing can overcome the problems encountered in on-road testing [7, 8]. We can manually create testing scenarios that resemble the real traffic environment in the virtual simulation platform. However, we obviously cannot enumerate all real scenarios because this process is often troublesome and time-consuming, given the well-known combinatorial explosion problem [9]. We need to test AVs by sampling a small number of challenging scenarios that are representative of all possible scenarios.
Then, a question naturally arises, which is as follows: is it possible to efficiently and reliably test the capabilities of intelligent vehicles with a limited number of sampling scenarios?

To better illustrate this problem, Li et al. proposed the probably approximately correct (PAC) testing theory as the basic theory of intelligence testing [10]. PAC states that two types of errors need to be minimized when sampling a limited number of testing scenarios: empirical error and generalization error. Empirical error measures the influence of the hypothetical data distribution instead of the real data distribution, while generalization error measures how closely the mappings that start in chosen hypothesis can approximate the optimal hypothesis.

Current work mainly focuses on reducing empirical error. For example, Zhao et al. inspirationally designed a testing method called accelerated evaluation, which has dramatically driven the development of sample generation for simulation testing scenarios [11, 12]. Accelerated evaluation is to select some parametric scenario model as a man-made priori information and learn the distribution of model parameters from the empirical dataset. Testing scenarios with low sampling frequency can be well-obtained by the important sampling method. Then, the testing process is accelerated by testing the performance of AVs in sampled scenarios instead of the entire testing dataset. They believe that the scenarios that occur least frequently in the realistic environment under a certain metric are most worth testing.

However, the shortcoming of accelerated evaluation lies in the restricted man-made hypotheses. The method only assumes that the driving behaviors of SVs are homomorphic and lacks the modeling of differences in driving behaviors. A testing scenario library that considers only homomorphic SVs cannot cover all challenging corner cases. Thus, it is significant to consider how to sample challenging corner cases more effectively.

Based on [12], Feng et al. improved the accelerated evaluation by proposing an adversarial-based reinforcement learning approach [13]. The method constructs a natural adversarial driving environment through reinforcement learning and improves the testing efficiency by 10,000 times. However, the problem of poor interpretability of reinforcement learning is inescapable [14]. Meanwhile, the above accelerated evaluation methods lack flexible interactive operability. It is impossible to directly adjust the testing difficulty and the specific content of the scenario for different AVs with limited interactions. Therefore, accelerated evaluation has limitations on sampling more targeted corner cases for AVs.

To this end, this paper points out that we can flexibly test AV by carrying out the abundant interactions between AV and SVs, where different behaviors and strategies of SVs play a crucial role during this progress and should, therefore, be carefully considered.

Based on this, we propose an interpretable and operable sampling method. We contemplate that the different driving behaviors of SVs constitute hypothetical factors in the testing scenarios. To do so, the driving behavior of SVs demands a better description, and many approaches [15, 16] have provided in-depth models to represent them. We employ the utility function to model them since the utility function-based approach can combine multiple driving decisions to model driving behavior and is thus more easily extended to describe abundant and complex behavior [17]. Also, we can use the utility function to describe different driving strategies, which are used as classification criteria for the “intrinsic reality” of the vehicle.

The proposed approach is general and can be applied in various scenarios. In this paper, we take a frequently studied lane-changing scenario as an example. In this typical scenario, we can modulate the way the SVs interact with the AV by adjusting the sampling frequency of the SV’s behavior to get more challenging events.

To give a better presentation of our findings, the rest of this paper is arranged as follows: section 2 is an introduction to related work. In section 3, we formulate the testing scenario generation problem from five aspects: (1) components of the testing scenario are introduced in section 3.1, (2) testing tasks are explained in section 3.2, and the subject of the test, AV, is presented in section 3.3. Driving strategies, performance metrics, and testing evaluation are then described in section 3.4, section 3.5, and section 3.6, respectively. Section 4 focuses on the utility function of driving behaviors in testing scenarios. Finally, the simulation results validate that the proposed method can obtain more difficult testing scenarios in section 5.

2. Related Work

In this section, we provide a brief review of related work from two research areas. Firstly, the driving strategies of traffic participants in realistic environments are presented. Then, we offer an introduction to the application of utility functions in driving behavior modeling.

2.1. Driving Strategies. Driving strategy is an intrinsic property of the vehicle that directly affects its decision making and behavior. To show the “intrinsic realism” of vehicle behavior, it is imperative to consider the variety of driving strategies of traffic participants in testing scenarios. This view can be reflected in the impact that SVs with different strategies have on AV.

In an aggressive strategy, traffic participants usually focus more on enhancing the benefit of their journey, while treating other traffic participants as rational drivers [18]. In a real-world scenario, aggressive human-driven cars would be the cause of most traffic accidents, and aggressive human drivers often try to get the greatest benefit as their first consideration when engaging in traffic. For AVs, there are many algorithms that regard participation in traffic as a “noncooperative dynamic game” in the existing intelligent driving studies. Most of these algorithms are based on learning, such as deep learning [19] and deep reinforcement learning [20]. For instance, Wolf et al. designed an adaptive behavior algorithm for AVs based on deep reinforcement learning [21]. In training, the vehicle model continuously calculates the optimal path and uses it as a reward until it finds the most favorable path, i.e., the locally optimal solution.
A conservative driving strategy is also known as a defensive driving strategy. The core task of defensive driving strategies is to “prevent all potential danger against uncertainties” [22, 23]. Human-driven vehicles that use conservative driving strategies account for a large ratio in real life. Such strategies are often adopted by drivers who are new to the road or inexperienced in emergency management. AVs with this driving strategy are more acceptable to the public.

Google self-driving vehicle is designed as a typical conservative vehicle [24]. At the green light at an intersection, the Google self-driving vehicle is set to pause for a brief period before crossing the intersection [25]. In doing so, it increases the likelihood of avoiding a collision with the offending vehicle. In contrast, at the stop line of an intersection with no signal, it was set to intentionally move forward a short distance before crossing the intersection. This is a type that does not want to gamble and will make decisions with negative assumptions about the surrounding traffic environment. The implication is that Google’s self-driving vehicle is wary of participants who seize the right-of-way in some unreasonable way (e.g., quick braking, no warning, and queue jumping). It will yield early to mitigate or avoid potential risks [26].

The responsibility sensitive safety (RSS) model proposed by Mobileye is another representative model for conservative driving strategies. For example, in the car-following scenario, the following vehicle is assumed to decelerate at the minimum deceleration speed regardless of the deceleration speed of the vehicle in front. Hence, this approach dramatically increases the safety distance between two vehicles.

2.2. Utility Function. To describe the driving behaviors of vehicles based on driving strategies, the utility function can be applied to model the heterogeneous driving behavior of vehicles.

The utility function was first implemented in economics to measure consumer welfare or satisfaction as a function of the consumption of actual goods, such as food or clothing [27, 28]. Utility functions are widely used in the rational choice theory to analyze human behavior, as well as in transportation for analyzing traffic volume allocation decisions [29] and vehicle driving decisions [30], among other common applications.

Toledo et al. proposed the use of a utility function to model the integrated lane change behavior in microscopic traffic simulation. The model is divided into two submodels: the target lane selection model and the distance acceptance model. The trade-off between mandatory lane change and discretionary lane change in the lane change model is addressed by establishing the utility of the driver’s choice of the target lane [17].

After that, Altendorf and Flemisch presented a method for analyzing the behavior of the vehicle, which is controlled by both a human driver and collaborative automation, using utility functions. They pointed out that the utility functions of different drivers are heterogeneous and that the values of the variables need to be determined from empirical data. They also provided a basic utility formula that they believed is reasonable. To allow the analysis of the behavior of the human-machine system, they considered that the utility function is influenced by the vehicle’s own position, speed, and the position of obstacles and other vehicles [31].

Li et al. applied the utility functions to implement a more complex automatic lane change model for AVs. Their algorithm employed a dynamic bicycle model that combines the longitudinal and lateral motion of the vehicle to automatically determine the target lane using the utility functions. To achieve safe and smooth lane-changing maneuvers, the utility functions considered factors, such as the average speed of different lanes, the time interval between surrounding vehicles, and the remaining travel time of the vehicles in each lane [32].

The purpose of this paper is to choose different utility functions [17] to describe heterogeneous driving behaviors so that we can generate a more challenging testing scenario in limited rounds of simulations.

3. Problem Formulation

In this section, the heterogeneous testing scenario sampling problem is formulated.

Figure 1 shows the structure diagram of the designed sampling method. The method is mainly divided into 3 parts. Firstly, simulation-based testing requires a virtual testing scenario as the testing environment where we need to construct the spatial-temporal layout of the scenario and specify the composition of the scenario. After that, the testing tasks to be accomplished by AV should be described, based on which the behavior of the SVs is included. To better describe the behavior of traffic participants, we predesign multiple driving strategies. Finally, by utilizing the designed performance metrics, we monitor whether the AV encounters challenging events in response to the different behaviors of SVs, and thus evaluate whether the operation strategy of an AV is reasonable.

Based on the above description, the testing scenario is defined as a 6-tuple: \( \Omega = (\text{Scene}^t, \Psi, V, U, M, E)_{t \in T} \), where \( \text{Scene}^t \) is the testing scene at time \( t \), \( \Psi \) represents the testing tasks AV needs to accomplish, \( V \) refers to the AV, \( U \) is the functions that describe the behaviors of the SVs, \( M \) means the testing metric, and \( E \) shows the testing evaluation to AV.

This definition reflects the fact that by controlling \( U \), a testing scenario can be constructed to obtain the testing evaluation \( E \), where the ability of an AV \( V \) to complete a sequence of testing tasks \( \Psi \) in the Scene\(^t\) in the time set \( T \) is tested under testing metrics \( M \). The subsequent contents describe each of the six tuples in detail.

3.1. The Mathematical Definition of the Testing Scene. We refer to [33, 34] for a mathematical description of the testing scenario. The semantic method is selected to define the scenario by splitting it into a sequence of scenes by time. Then, the static and dynamic objects and the association between objects and the state of objects within the scene are defined. That is to say, a testing scene is a frame of the testing scenario, which is analogous to the snapshot, including static and dynamic elements.
We define $S$ as a portion of space in the real world. The testing scene can be written as $\text{Scene}^t (S) = \{E^t (S), \text{CE}^t (S), \text{SD}^t (S)\}$, where $E^t (S)$ is the set of element information, $\text{CE}^t (S)$ represents the set of connection elements, and $\text{SD}^t (S)$ is the set of state description.

The set of element information can be rewritten as $E^t (S) = E^t_S (S) \cup E^t_d (S)$, where $E^t_S (S)$ is the set of static elements and $E^t_d (S)$ is the set of dynamic elements. Static elements are objects that have been stationary for a prolonged enough period or whose motion is barely perceptible. It includes static elements in geospatial space, such as metrics, semantics, topology, and classification information for roads and all subcomponents (e.g., lanes, lane markings, and road types). $E^t_S (S)$ includes the representations of all stationary objects in $S$ space, i.e., it only indicates that the scene contains such elements, however, it does not indicate their location and state information. Similarly, the set $E^t_d (S)$ is the set of all temporally changing objects located in $S$ at time $t$. The dynamic elements, which are objects that change in displacement with time, are the main traffic participants in the testing scenario, including pedestrians, cyclists, manually driven vehicles, and AVs.

We use the semantic explanation to represent connection elements. $\text{CE}^t (S)$ can be divided into static and dynamic relationship elements. Static connection elements $CE^t_S (S)$ describe the relationships between elements in terms of distance, adjacency, and direction, while dynamic connection elements $CE^t_d (S)$ describe the relationships between dynamic elements, including proximity, stopping, following, overtaking, and other modes.

The set of the state description $\text{SD}^t (S)$ in the initial scene includes the speed state $SS^t (S)$ (acceleration, deceleration, etc.) and behavior state $BS^t (S)$ (left lane change, free driving, etc.) of the traffic participants. Thus, connection elements can be rewritten as $\text{SD}^t (S) = SS^t (S) \cup BS^t (S)$.

In this paper, we design a typical lane-changing scenario in a one-way two-lane road, not considering support from the roadside unit or other equipment. The distance between the starting position of the AV and the reference line is set as $R$, while the distance to the nearest lane line is set as $R'$.

In Figure 2, the relationship description adopts the polar coordinate method with the measured vehicle as the axis and sets the distance between the four vehicles as $(R_i, D_i)$, $(R_i, D_i)$, $(R_i, D_i)$, where $R_i$ denotes the absolute distance between the vehicle centers, and $D_i$ denotes the angular deviation between the driving direction of $V^{(i+1)}$ and the driving direction of $AV$.

The state information of the traffic participants in the initial scene includes the vehicle speed, acceleration, and other driving states of the vehicle. For $V^{(i)}$, the speed and acceleration are $(v_i, a_i)$, where $i = 1, 2, 3, 4$; $V^{(2)}$ is set to change lanes to its left lane. $V^{(1)}$, $V^{(3)}$, and $V^{(4)}$ are in the free driving state.

3.2 Testing Tasks. The testing scenario is a series of testing scenes whose starting and ending is with the implementation of the tasks and the completion of the tasks by the AVs, i.e., the composition of the scenario requires a clear definition of the testing tasks completed by the AVs. The PAC testing theory states that by sampling several tasks, the complete evaluation of AVs can be made. Thus, we define the set of tasks as $\Psi = \{\psi_1, \psi_2, \ldots, \psi_m\}$, which denotes the $m$ tasks to be completed in the testing scenario.

In the lane-changing scenario constructed above, the testing task in this scenario is divided into two semi-tasks: the AV needs to keep a safe distance when $V^{(2)}$ cuts in, and the AV does not appear to have a dangerous behavior before following $V^{(2)}$ smoothly. The testing scenario starts with the vehicles driving in the initial states and ends with the AV steadily following $V^{(2)}$.

3.3 AV under Test. The testing scenario revolves around the AV under test, while its difficulty is reflected by the behavior of the AV as well. Therefore, AV is one of the key factors affecting the sampling of the testing scenarios. Besides, different AVs behave differently when faced with the same environment. Thus, the difficulty of the testing scenario varies from AV to AV. It reveals that the differences between
the AV under test need to be fully considered when sampling the testing scenarios. The differences between AVs depend on the algorithm that realizes the functionality of automatic driving. Current algorithms are mainly based on learning methods, such as deep learning and reinforcement learning. Different algorithm structures, different testing datasets, and even different training durations can affect the decisions and behaviors of AVs when they respond to their surroundings. Thus, to show that the proposed scenario sampling method can automatically adapt to the AV and search for more targeted critical scenarios, we design different AVs in subsequent experiments.

3.4. Operable Components—SVs’ Driving Strategies. The purpose of AV testing is not limited to evaluating the intelligence of AVs. We can also use the results of evaluations to continuously design AVs with higher intelligence. However, because of the black-box nature of the artificial intelligence (AI) utilized in AVs, we can only observe whether the AV could behave as expected by setting specific testing conditions. Then, based on this, we can continuously adjust the testing conditions to help us design a more intelligent AV. At this point, testing scenarios with interpretability and operability can help us achieve continuous adjustments.

The operable component is the critical factor for AV testing to be interactive and interpretable. However, to the best of our knowledge, few papers mention interactive and interpretable testing scenario generation methods. To fill this gap, we set the driving strategies of the SVs as the operable components of the scenario and divide the driving strategies of the SVs into 3 categories: aggressive mode, conservative mode, and normal mode. Aggressive and conservative strategies are introduced in section 2. The normal strategy is a strategy that is intermediate between the aggressive strategy and the conservative strategy. For example, when following a vehicle, the vehicle will maintain an appropriate distance from the vehicle in front, neither too close nor too far. It is worth noting that most current testing studies compare their methods with the normal strategy, i.e., this strategy is commonly used as a baseline strategy. It is possible to utilize these three driving strategies to generate scenarios of different difficulty for different AVs. For example, we consider a well-trained AV being tested in a familiar driving scenario, where the familiar scenario refers to the scenario that the AV has experienced during the learning process. It has a better understanding of the strategy and behavior of the SVs. Hence, it will know how to behave well based on the behavior of the SVs to meet the test requirements. Conversely, if the AV is surrounded by an unfamiliar scenario where SVs are with unfamiliar strategies, it may pose greater difficulties.

We apply the driving strategy to both AVs and human-driven vehicles. Since most AVs are using AI technology to imitate the driving behavior of human drivers, their behaviors are very similar. Therefore, this paper does not distinguish between human-driven vehicles and AVs and neglects external attributes, such as the brands and car models. To vividly describe the role of driving strategies, we give one possible situation, where we can adjust the driving strategy of an SV \((V^{(2)})\) that prepares to change lanes, as shown in Figure 3. If \(V^{(2)}\) equips an aggressive driving strategy, it will treat the surrounding vehicles as rational traffic participants. It may not wait for a safe time to make the change. At this point, the resulting scene is challenging. At the same time, AV is exposed to dangerous spacing gap. However, if \(V^{(2)}\) is regulated as a normal vehicle, it will choose to change lanes in a safer state, while \(V^{(2)}\), with a conservative strategy, may not choose to change lanes at this time. Because of the limitation of space, we only unfold from different strategies of \(V^{(2)}\), while the strategies of other SVs will also have an impact on AV, which are not described here.

3.5. Performance Metrics. Performance metrics are the testing perspectives that are selected for testing AV, for example, driving safety, driving comfort, mobility, etc. Driving safety measures the degree of risk to the traffic participants, being an essential metric and fundamental basis for determining whether an AV is ready for the road. Driving comfort is the psychological and physiological comfort of the driver and passengers in the tested vehicle, while mobility is more focused on testing the driving efficiency of an AV when completing continuous tasks.

In this paper, driving safety is used as the performance metric. The time-to-collision (TTC) and headway are two widely used metrics to determine whether a vehicle is at risk of collision and be reasonable for testing the safety of vehicles. Thus, these metrics are selected as performance metrics for testing the safety of AVs.
When the performance metric is higher than the predefined threshold, there is no challenging event, i.e., the AV is in a safe state. Conversely, the AV is in a state of risk. We define the relative distance and relative speed between $V^{(1)}$ and $V^{(m)}$ as $R_m(t)$ and $\dot{R}_m(t)$, respectively, the mathematical expression of TTC at time $t$ is written as follows:

$$TTC_m(t) = \frac{R_m(t)}{\dot{R}_m(t)}$$  \hspace{1cm} (1)

Headway is usually defined as the time between two successive vehicles as they pass the same point on the roadway, measured from the same common feature of both vehicles. Headway can be expressed mathematically at time $t$ as follows:

$$H_m(t) = \frac{R_m(t)}{v_1(t)}$$  \hspace{1cm} (2)

where $v_1(t)$ is the speed of $V^{(1)}$ at time $t$.

3.6. Testing Evaluation. Testing evaluation is utilized to assess the performance of AV in the testing scenario and is the ultimate purpose of AV testing. After the AV performs the testing task, we, firstly, check the completion situation of the task and then provide a comprehensive analysis of the AV’s intelligence or capabilities based on the results of testing metrics in each testing scene. For example, we can evaluate AV using the number of challenging events or the proportion of challenging events to the total events in the testing scenario.

Nevertheless, the main goal of this paper is to highlight the significance of sampling testing scenarios to AV testing. To do so, we evaluate and compare the performance of the same AV in different testing scenarios instead of evaluating the intelligence of the AV specifically.

4. Behavioral Utility Function

This section focuses on portraying the utility function of vehicle behavior. Firstly, the general formulation of the behavioral utility is described. Then, to interpret the behavioral utility function in detail, we unfold it for the utility of the lane-changing and car-following behaviors.

4.1. The Basic Formula of Behavioral Utility. The heterogeneous behavior of vehicles needs to be modeled separately in virtual simulations. Given the high complexity of vehicle behaviors, it is often impossible to give a fidelity model of behaviors in simulation. Thus, we resort to the utility function approach since it can simulate different behaviors of vehicles through utility without explicitly specifying a concrete model.

Meanwhile, the behavioral utility of vehicles can reflect different choices more directly, such as whether to change lanes or the appropriate value of the following distance, compared to other deterministic models. There are different utility functions for the behaviors of vehicles in different testing scenarios.

The behavioral utility function takes the external environment and its own parameters as inputs and outputs the utility value of the vehicle’s behavior. We assume the inputs to the utility function to be a set of explanatory variables: $x_1, x_2, \ldots, x_N$. For any participant $V^{(i)}$, $i = 1, 2, \ldots, N$, define $U_i$ as the behavioral utility. We consider the behavioral utility as a weighted sum of these independent factors. Then, the basic utility function can be written as:

$$U_i = \sum_{k=1}^{N} \alpha_{i,k} x_k,$$

where $\alpha_{i,k}$ is the weight parameter.

For ease of presentation, the vehicle under test is assumed to be $V^{(1)}$, and other traffic participants are $V^{(2)} \ldots V^{(N)}$.
4.2. Utility Function for Lane-Changing Behavior and Car-Following Behavior. Here, we depict the utility function of the vehicles in the testing scenario (e.g., $V^{(2)}$ in Figure 2) to perform the lane-changing behavior, along with the utility function used to perform the car-following behavior, by giving a semiquantitative and semiquantitative model.

In terms of lane-changing behavior, the utility value reflects the choice of whether to change the lanes of a vehicle at a given moment. The utility is influenced by many factors, and these independent factors constitute the explanatory variables mentioned earlier. They can be interpreted in two categories: self-influence variables and environmental influence variables [17].

Take $V^{(2)}$ in Figure 2 again as an example. $V^{(2)}$ is about to decide to change lanes. The behavioral utility function of $V^{(2)}$ is affected by its self-influence variables and environmental influence variables. Based on the basic utility function, we can get the following:

$$U_2 = a_1 h_{lc} + a_2 \text{gap}_1 + a_3 \text{gap}_2 + \sum_{k=4}^{n} a_k x_k,$$

where $h_{lc}$ is the headway between the two vehicles, $\text{gap}_1$ is the gap between $V^{(2)}$ and $V^{(3)}$, and $\text{gap}_2$ is the gap between $V^{(1)}$ and $V^{(4)}$. The above three variables are all environmental influence variables, while $\sum_{k=4}^{n} a_k x_k$ represents the self-influence variables.

However, the specific utility function of the lane-changing behavior is not a critical part of our method and does not influence the analysis of this paper. The model parameters can be obtained by learning-based methods or directly set by expert experience. Then, these different parameters can yield abundant lane-changing behaviors. Because of the length limitation of the paper, we will explain the design of our behavioral utility function in detail in our subsequent paper. In this paper, we care more about characterizing the heterogeneous output of the behavioral model utility function to verify the effect of heterogeneous behaviors on the difficulty of the testing scenarios. We can control the output of the model by tuning the hyperparameter family of the utility function model $H = \{h_1, h_2, \ldots, h_m\}, m \in Z^+$. The choice of lane-changing is an important characterization quantity for measuring safety, and the decision also reflects different lane-changing strategies. Thus, we set up an interpretable and tunable hyperparameter in the hyperparameter family of the utility function model of the lane-changing vehicle, such as $h_n$. This hyperparameter controls the probability of whether the vehicle will change lanes when receiving the current inputs. By controlling $h_n$, it is feasible to construct traffic participants with different driving strategies, thus sample testing scenarios with different difficulties.

Current studies have done few studies on the utility for vehicles to change lanes with different driving strategies. Despite the prevalence of drivers with aggressive and conservative strategies in everyday life, there is still a lack of datasets in academia that characterize these two types of drivers separately. Therefore, our utility functions are mainly derived from our lab’s previous experience in conducting autonomous driving research. We give a form of utility based on our observation, as shown in Figure 4. We will later conduct a more in-depth analysis of the driving data for different strategies in a subsequent article. Refining the behavioral utility under different driving strategies will bring the simulation closer to reality, however, it will not affect the emphasis of this paper.

We describe the probability of executing the lane-changing behavior with the same lane-changing behavior utility for vehicles with aggressive, conservative, and normal strategies in our testing scenario, respectively. It can be seen that in the testing scenario, vehicles with aggressive strategies have a greater probability of performing a lane change than normal vehicles for the same utility value being calculated, while vehicles with conservative strategies are more cautious about the lane-changing behavior.

Moreover, similar to the utility of the lane-changing behavior, Figure 5 describes the utility of the car-following behavior. We can adjust the hyperparameters of the vehicle’s utility function to get the vehicle’s car-following utility under different strategies. The choice of the desired following distance varies according to the designed strategy for the same following utility value.

The aggressive vehicle expects to maintain a closer distance than the normal vehicle does when they share the same utility value, while conservative vehicles tend to maintain a longer desired following distance.

5. Simulations

5.1. The Preparations for the Simulation. The designed road environment is a one-way two-lane road, where each lane is 4 m wide. We set the AV and SVs to be the same size and model, the length and width of which are 4 m and 2 m, respectively. The specific locations and other parameters of each vehicle at the initial scene are shown in Table 1. In addition, the expected speed limit of all vehicles is set to 9 m/s.

The car-following model is provided in the literature [35] for generating continuous trajectories for autonomous vehicles, which is also an extension of the collision avoidance model [36]. We assume that the vehicle at the back maintains a suitable and adjustable distance from the vehicle in front. The following vehicle, firstly, calculates the final distance $L(t)$ since time $t$, if its ac/decelerates to reach the same speed as the leading vehicle with a fixed ac/deceleration max $a$. 

$$L(t) = x_{lead}(t) - x_{follow}(t) + \frac{v_{lead}^2(t) - v_{follow}^2(t)}{2a_{max}},$$

where $x_{lead}(t)$, $v_{lead}(t)$ represent the location and speed of the vehicle in front, respectively, and $x_{follow}(t)$, $v_{follow}(t)$ are the location and speed of the following vehicle, respectively. Then, the desired distance $G$ is set to determine whether the final distance is appropriate. If $L(t)$ is less than $G$, the vehicle behind will slow down. If $L(t)$ is greater than $G$, the vehicle will accelerate until it travels at maximum speed.
Therefore, we can calculate the speed of the following vehicle as follows:

\[ v_{\text{follow}}(t + T_0) = \begin{cases} \max\{0, v_{\text{follow}}(t) - a_{\max}T_0\} & L(t) < G, \\ \min\{v_{\max}, v_{\text{follow}}(t) + a_{\max}T_0\} & L(t) \geq G, \end{cases} \]

where \( v_{\max} \) is the max speed (14 m/s in this paper) and \( T_0 \) is the time interval (0.1s in this paper).

5.2. Simulation of Challenging Testing Scenarios. We regulate the challenging degree of the testing scenario by modifying the driving strategy of the SVs. We design two experiments to demonstrate the theme of this paper. In each simulation, we repeat the simulation 200 times for the typical scenario (Figure 2) and utilize TTC and headway as performance metrics to record the challenge events that the AV might encounter.

In the first experiment, we set the AV to be the normal vehicle. The driving strategy of the \( V^{(2)} \) is adjusted to change its behavior, while we keep all other SVs as autonomous vehicles with normal strategy. In addition, to facilitate comparison, we set another testing scenario in which all SVs are autonomous vehicles with normal strategies (reflecting current research results in other papers) as the experimental benchmark.

Figures 6 and 7 demonstrate the difficulty of the designed testing scenarios for the AV under TTC metric and headway metric, respectively, in one testing scenario. It can be seen intuitively that when \( V^{(2)} \) adopts an aggressive driving strategy (set to Scenario 1.1), TTC and headway will bring more challenges to AV than when \( V^{(2)} \) adopts a normal strategy in the whole testing process, while when \( V^{(2)} \) is a conservative vehicle (set as Scenario 1.2), it will be more cautious in changing lanes, with fewer challenging testing events for the AV.

To verify that our sampling method is not contingent, we represent the results of 200 replicate experiments in box plots. In this case, we choose a threshold of 2s for the TTC metric and 1.5s for the Headway metric. It is set to determine that the AV encounters a challenging event when the metric is below the threshold. In this paper, the starting and ending points of the testing scenario are from the start of the task to the completion of the testing task. It implies that in each testing scenario, the time to complete the task varies and the total number of events is inconsistent as well. Therefore, we cannot measure the testing scenarios by comparing the number of challenging events, and we choose the frequency of challenging events to compare different predefined scenarios. As can be seen from Figure 8, the designed scenario 1.1 obtains more challenging events than the baseline scenario, both under the TTC metric and the Headway metric.

It is set to determine that the AV encounters a challenging event when the metric is below the threshold. In this paper, the starting and ending points of the testing scenario are from the start of the task to the completion of the testing task. It implies that in each testing scenario, the time to complete the task varies and the total number of events is inconsistent as well. Therefore, we cannot measure the testing scenarios by comparing the number of challenging events, and we choose the frequency of challenging events to compare different predefined scenarios. As can be seen from Figure 8, the designed scenario 1.1 obtains more challenging events than the baseline scenario, both under the TTC metric and the Headway metric.

For better persuasiveness, we test the AV with aggressive strategies in the second comparison simulation. By
Figure 6: The results for the first simulation under TTC: (a) the probability density of TTC in scenario 1.1, (b) the probability density of TTC in scenario 1.2, (c) the probability density of TTC in the benchmark scenario, and (d) the comparison of the three scenarios.

Figure 7: Continued.
modulating the driving strategies of the SVs, we verify that the scheme is implementable and can find more testing scenarios worth testing for different types of AV.

As the first comparison simulation, we used the scenario with SVs as normal traffic participants as the experimental benchmark. From Figures 9 and 10, it can be observed that the AV comes across a higher percentage of challenging events with all aggressive SVs (set to scenario 2.1) than the events the baseline provides, via the TTC metric. The second is the scenario (set to scenario 2.2) with all conservative SVs, while under the headway metric, a higher sampling frequency of challenging events is generated in scenario 2.2.

We also repeated the experiment 200 times and still set the threshold for the TTC metric to 2 s and for the headway metric to 1.5 s. As shown in the box plot in Figure 11, we obtained more challenging events in scenario 2.1 and Scenario 2.2 than the benchmark has under both metrics with Scenario 2.2, which is dominated by conservative vehicles, obtaining the most challenging events under the headway metric.

The results indicate that it is not just the SVs with aggressive strategies that bring more challenging events to the AV. When the AV adopts an aggressive strategy by appropriately increasing the frequency of the conservative SVs, the testing scenario will bring more challenging events to AV.

![Boxplots for the first simulation](image)

Figure 7: The results for the first simulation under the headway: (a) the probability density of the headway in scenario 1.1, (b) the probability density of the headway in scenario 1.2, (c) the probability density of the headway in the benchmark scenario, and (d) the comparison of the three scenarios under the headway.

![Boxplots for the first simulation](image)

Figure 8: The boxplots for the first simulation: (a) the frequency of challenging events in each scenario when TTC is less than 2 s and (b) the frequency of challenging events in each scenario when the headway is less than 1.5 s.
Figure 9: The results for the second simulation under TTC: (a) the probability density of TTC in scenario 2.1, (b) the probability density of TTC in Scenario 2.2, (c) the probability density of TTC in the benchmark scenario, and (d) the comparison of the three scenarios under TTC.

Figure 10: Continued.
6. Conclusion

In this paper, we design a testing scenario sampling method for AV testing. The testing scenarios we sample are with interpretability and operability by considering different behaviors of the SVs. Besides, we utilize utility functions to describe the driving behavior and use different strategies as classification criteria for the “intrinsic reality” of the vehicle. We also provide a typical lane-changing scenario, where we find more challenging testing events effectively by interactively controlling the occurrence frequency of the heterogeneous SVs. Furthermore, our proposed method can be explained that some manual modifications to the driving strategies of the SVs can result in more challenging events for the AV. Such events are interpretable and will help provide more tangible answers to subsequent upgrades and improvements to AV.

It is also worth noticing that according to the analysis of the simulation, we should make different adjustments based on different testing metrics to sample more challenging events. In the subsequent work, (1) we will compare different testing metrics in detail to match the real testing needs. (2) We will set more specific testing tasks to evaluate the AV more accurately. (3) We will perform in-depth research on how to optimize the means of modulating the SVs so that we can get more challenging events when targeting each specific AV.

Figure 10: The results for the second simulation under the headway: (a) the probability density of the headway in scenario 2.1, (b) the probability density of the headway in scenario 2.2, (c) the probability density of the headway in the benchmark scenario, and (d) the comparison of the three scenarios under the headway.

Figure 11: The boxplots for the second simulation: (a) the frequency of challenging events in each scenario when TTC is less than 2 s and (b) the frequency of the challenging events in each scenario when the headway is less than 1.5 s.
Data Availability
The data that supports the findings of this study are available from the corresponding author on reasonable request.

Conflicts of Interest
The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments
This work was supported in part by National Key Research and Development Program under Grant 2021YFB2501200 and National Natural Science Foundation of China under Grant 62133002.

References
[1] E. Thorn, K. C. Shawn, and C. Michelle, “A framework for automated driving system testable cases and scenarios,” D. Transp., Nat. Highway Traffic Saf., Washington, DC, USA, Tech. Rep.No. 812623, 2018.
[2] 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, 2018.
[3] S. Feng, Y. Feng, C. Yu, Z. Yi, and X. L. Henry, “Testing scenario library generation for connected and automated vehicles, Part I: methodology,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1573–1582, 2020.
[4] L. Li, Y.-L. Lin, N. Zheng et al., “Artificial intelligence test: a case study of intelligent vehicles,” Artificial Intelligence Review, vol. 50, no. 3, pp. 441–465, 2018.
[5] H. Liu and Y. Feng, “Development of an augmented reality environment for connected and automated vehicle testing,” University of Michigan Transportation Research Institute, Ann Arbor, MI, USA, 2019.
[6] L. Fraade-Blanar, S. B. Marjory, M. A. James, and K. Nidhi, Measuring Automated Vehicle Safety: Forging a framework, RAND Corporation, Santa Monica, CA, USA, 2018.
[7] L. Li, X. Wang, K. Wang et al., “Parallel testing of vehicle intelligence via virtual-real interaction,” Science Robot, vol. 4, 2019.
[8] Y. Feng, C. Yu, S. Xu, X. H. Liu, and H. Peng, “An Augmented Reality Environment for Connected and Automated Vehicle Testing and evaluation,” in Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1549–1554, IEEE, Chongshu, China, June 2018.
[9] E. Lamb, “Math’s proof smashes size record: supercomputer produces a 200-terabyte proof—but is it really mathematics?” Nature, vol. 534, no. 7605, pp. 17–19, 2016.
[10] L. Li, N. Zheng, and F. Y. Wang, “A Theoretical Foundation of Intelligence Testing and its Application for Intelligent vehicles,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 10, pp. 6297–6306, 2020.
[11] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, and K. Nobukawa, C. S. Pan, “Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 3, pp. 595–607, Mar. 2017.
[12] 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, 2018.
[13] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, “Intelligent driving intervention test for autonomous vehicles with naturalistic and adversarial environment,” Nature Communications, vol. 12, no. 1, pp. 1–14, 2021.
[14] A. Verma, V. Muruki, R. Singh, K. Pushmeet, and C. Swarat, “Programmatically Interpretable Reinforcement learning,” in Proceedings of the 35th International Conference on Machine Learning, pp. 5045–5054, PMLR, Stockholm, Sweden, April 2018.
[15] T. P. Wenzel and M. Ross, “The effects of vehicle model and driver behavior on risk,” Accident Analysis & Prevention, vol. 37, no. 3, pp. 479–494, 2005.
[16] W. J. Schakel, V. L. Knoop, and B. van Arem, “Integrated lane change model with relaxation and synchronization,” Transportation Research Record, vol. 2316, no. 1, pp. 47–57, 2012.
[17] T. Toledo, H. N. Koutsopoulos, and M. E. Ben-Akiva, “Modeling integrated lane-changing behavior,” Transportation Research Record, vol. 1857, no. 1, pp. 30–38, 2003.
[18] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, “Social behavior for autonomous vehicles,” National Academy of Sciences, vol. 116, no. 50, Article ID 24972, 2019.
[19] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F. Y. Wang, “Capturing car-following behaviors by deep learning,” IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 3, pp. 910–920, 2017.
[20] V. Mnih, K. Kavukcuoglu, D. Silver et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, p. 529, 2015.
[21] P. Wolf, K. Kurzer, T. Wingert, F. Kuhnt, and J. M. Zollner, “Adaptive behavior generation for autonomous driving using deep reinforcement learning with compact semantic states,” IEEE Intelligent Vehicles Symposium, vol. 2018, pp. 993–1000, 2018.
[22] J. Wei, J. M. Dolan, J. M. Snider, and B. Litkouhi, “A point-based mdp for robust single-lane autonomous driving behavior under uncertainties,” in Proceedings of the 2011 IEEE International Conference on Robotics and Automation, pp. 2586–2592, IEEE, Shanghai, China, May 2011.
[23] T. Q. Tang, J. He, S. C. Yang, and S. Hua-Yan, “A car-following model accounting for the driver’s attribution,” Physica A: Statistical Mechanics and Its Applications, vol. 413, pp. 583–591, 2014.
[24] M. Birdsell, “Google and ITE: the road ahead for self-driving cars[.].” Institute of Transportation Engineers, ITE Journal, vol. 84, no. 5, p. 36, 2014.
[25] C. Urmsen, “The View from the Front Seat of the Google Self-Driving Car,” 2015, https://www.wired.com/2015/05/the-view-from-the-front-seat-of-the-google-self-driving-car/.
[26] S. Noh, “Decision-making framework for autonomous driving at road intersections: safeguarding against collision, overly conservative behavior, and violation vehicles,” IEEE Transactions on Industrial Electronics, vol. 66, no. 4, pp. 3275–3286, 2018.
[27] W. E. Armstrong, “Uncertainty and the utility function[.].” The Economic Journal, vol. 58, no. 229, pp. 1–10, 1948.
[28] J. C. Harsanyi, “Cardinal utility in welfare economics and in the theory of risk-taking[.].” Journal of Political Economy, vol. 61, no. 5, pp. 434–435, 1953.
[29] L. Chen, B. Wang, and X. Chen, “Utility-based Resource Allocation for Mixed Traffic in Wireless networks,” in
[30] G. Arslan, J. R. Marden, and J. S. Shamma, “Autonomous Vehicle-Target Assignment: A Game-Theoretical formulation,” University of California, Los Angeles, CA, USA, Department of Mechanical and Aerospace, 2007.

[31] E. Altendorf and F. Flemisch, “Prediction of Driving Behavior in Cooperative Guidance and Control: A First Game-Theoretic approach,” 2014.

[32] Z. Li, J. Jiang, and W. H. Chen, “Automatic Lane Change Maneuver in Dynamic Environment Using Model Predictive Control method,” in Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2384–2389, IEEE, Las Vegas, NV, USA, October 2020.

[33] E. Andreotti, P. Boyraz, and S. Selpi, “Mathematical definitions of scene and scenario for analysis of automated driving systems in mixed-traffic simulations,” IEEE Transactions on Intelligent Vehicles, vol. 6, no. 2, pp. 366–375, 2020.

[34] X. Zhang, J. Tao, K. Tan et al., “Finding Critical Scenarios for Automated Driving Systems: A Systematic Literature Review,” 2021, https://arxiv.org/abs/2110.08664.

[35] Y. Meng, L. Li, F. Y. Wang, L. Keqiang, and L. Zhiheng, “Analysis of cooperative driving strategies for nonsignalized intersections,” IEEE Transactions on Vehicular Technology, vol. 67, no. 4, pp. 2900–2911, 2017.

[36] E. Kometani and T. Sasaki, “On the stability of traffic flow,” Journal of Operations Research Japan, vol. 2, no. 1, pp. 11–26, 1958.