Comprehensive Safety Evaluation of Highly Automated Vehicles at the Roundabout Scenario

Xinpeng Wang, Graduate Student Member, IEEE, Songan Zhang, and Huei Peng

Abstract—A highly automated vehicle (HAV) is a safety-critical system. Therefore, a verification and validation (V&V) process that rigorously evaluates the safety of HAVs is necessary before their release to the market. In this paper, we propose an interaction-aware safety evaluation framework for the HAV and apply it to the roundabout entering scenario. Instead of assuming that the primary other vehicles (POVs) take predetermined maneuvers, we model the POVs as game-theoretic agents. To capture a wide variety of interactions between the POVs and the vehicle under test (VUT), we use level-k game theory and social value orientation (SVO) to characterize the interactive behaviors and train a diverse library of POVs using reinforcement learning. The game-theoretic library, together with initial conditions, form a rich testing space for the two-POV roundabout scenario. On the other hand, we propose an adaptive test case generation scheme based on adaptive sampling, stochastic optimization and upper confidence bound (UCB) algorithm to efficiently generate customized challenging cases for the VUT from the testing space. In simulations, the proposed testing space design captured a wide range of interactive patterns at the roundabout scenario. The proposed test case generation scheme was found to cover the failure modes of the VUT more effectively compared to other test case generation approaches.

Index Terms—Safety evaluation, highly automated vehicles, game theory, adaptive sampling, interaction.

I. INTRODUCTION

HIGHLY automated vehicles (HAVs) promise to make ground mobility safer, cleaner and more equitable. However, with the recent crashes from prototype and production HAVs [1], [2], the public trust and confidence have been severely impacted. A rigorous and transparent verification and validation (V&V) method is urgently needed. Scenario-based safety evaluation has become an accepted best practice for HAV V&V process, which decomposes common driving trips into simpler but representative driving scenarios. The safety efficacy of the VUT is then assessed with a variety of test cases under each scenario. This procedure has been studied for several scenarios including cut-in, unprotected left-turn and pedestrian crossing scenarios [3]–[5], etc. These scenarios are “reactive scenarios”, in which the VUT is challenged by a POV “in a surprise”, and shall react safely and responsively. A test case is fully determined by the initial conditions of the challenge. The POV in these scenarios follows a predetermined maneuver with no further interaction with the VUT.

For SAE level 3 and above automated vehicles [6], their operational design domain (ODD) may include interactive scenarios, including highway merging, roundabout entering, etc. In interactive scenarios, the POVs and the VUT will have mutual influence on the future decision-making and motion-planning of each other. Therefore, an interaction-aware agent needs to not only react to the motion of other agents, but also be aware of the impact of itself on others. This interaction can take place over a time horizon. More specifically, roundabouts, which is becoming popular in the US due to their significant benefits compared with stop signs and traffic signals [7], is a good example of interactive scenarios. Roundabout driving has been actively studied in recent literature [8]–[10], due to the diverse traffic patterns and complex interaction brought by multiple vehicles on different branches. The complex interactions make designing decision-making and path planning algorithms for HAVs at roundabouts challenging, let alone the evaluation of such algorithms. To design a comprehensive evaluation framework for interactive scenarios like roundabout entering, extra factors should be considered compared to evaluation for reactive scenarios. On the one hand, vehicles coming from neighboring entrances will predict and influence each other’s future motions. Therefore, the interaction between a POV and a VUT, and possibly between multiple POVs need to be modeled. On the other hand, in an interactive setting, different human drivers may exhibit different behaviors even with the same initial condition, including coasting,
accelerating, yielding, etc. The diverse behaviors pose challenges to the behavior prediction and decision-making features of the VUT, thus should be incorporated by the evaluation framework.

We propose an interaction-aware evaluation methodology in this paper. It consists of two modules: first, we create a testing space, in which we generate a library of interactive POVs using level-k game theory and social value orientation (SVO). Second, we propose an adaptive test case generation scheme to generate challenging test cases for a given VUT. We focus on the evaluation at the roundabout entering scenario. The POV library includes a set of templates for decision-making at roundabouts, which contributes to both assessment and design for HAVs planning algorithms. The test case generation scheme can identify diverse failure modes of the VUT, enabling efficient safety certification for HAV companies or third-party organizations. This paper is an extension from our previous work [11] with several new contributions:

1) We created the POV library for roundabout entering, a different interactive scenario.
2) We developed a two-phase game-theoretic decision-making framework that is scalable to multiple POVs and different roundabout layouts.
3) We enriched the adaptive sampling scheme with a set of carefully-designed case selection criteria, including a modified expected improvement criterion, and an exploration heuristic based on behavior mode boundary identification. Significant improvement is observed on failure mode coverage compared with [11].
4) We proposed two different methods for sample allocation between different POV categories, one is based on stochastic optimization, the other is based on a modified UCB algorithm. Significant improvement is observed on the failure case number compared with [11].
5) We conducted comprehensive testing at two roundabout scenarios with two POVs in the simulation; we demonstrated the efficacy of the adaptive test case generation scheme to identify failure modes of the VUT in a high-dimensional testing space.

The paper is organized as follows: Section II gives a brief literature review; Section III provides the problem formulation; Section IV introduces the POV library creation based on game-theoretic approaches; Section V discusses the adaptive test case generation procedure; simulation results are presented in Section VI; finally, concluding remarks are made in Section VII.

II. LITERATURE REVIEW

A. Scenario-Based Safety Evaluation for HAVs

Scenario-based safety evaluation of HAVs has been an active research area in recent years [12]. There are two key research questions under this topic: 1) scenario selection and modeling (what to test?), and 2) test case selection (how to test?). For scenario selection and modeling, after the functional scenario has been selected (e.g. lane-change, car-following or roundabout entering), the focus lies on the parameterization of the scenario and organizing structure of all possible test cases. [3], [4] selected initial conditions to describe the scenario and used probabilistic models learned from naturalistic driving data to describe the structure of the test cases. [13], [14] used a combination of exposure frequency and maneuver challenge to assign a criticality to each test case. These methods considered the patterns of naturalistic driving, while the interactions between vehicles were not captured by the probabilistic models. [15], [16] used reachability analysis to categorize the risks of the scenario based on the size of the solution space for the VUT. These model-based analyses, however, could only be applied to scenarios where risks are determined by the initial states of vehicles, to which the interactive scenarios do not belong. On the other hand, [17] applied level-k game theory to create a library of POVs, which can be used to create different test cases for evaluation at interactive scenarios.

For test case generation, the test matrix has been used to evaluate advanced driver assistance system (ADAS) [18]. However, the VUT can be tuned to pass a set of predefined test cases, but may fail under broader conditions in the real world. Monte-Carlo sampling based evaluation methods have been proposed to estimate the real-world performance of the VUT. Importance sampling [3], [4], [13], [14] and subset simulation [19] were used to efficiently estimate the collision/injury rate. However, the amount of samples required to reach convergence for the estimation is still prohibitively large for real-world testing, especially when the target scenario has high-dimensional parameter space like the roundabout scenario. Moreover, importance sampling methods require prior knowledge on the failure modes of the VUT, which could be unavailable for interactive scenarios. Meanwhile, falsification-based evaluation methods attempt to generate initial conditions or POVs behaviors that force the VUT to violate the safety requirements with limited test runs. Corner cases have been generated using simulated annealing [20], rapid-exploring random tree (RRT) [21], evolutionary algorithms [22], Bayesian optimization [23], adaptive sampling [24], reinforcement learning [25], [26], etc. In [20], [21], the falsifying POVs might behave adversarially, which is not a reasonable representation of the real-world driving environment. In [26], reinforcement learning was used to create adversarial yet socially acceptable POV behaviors, while the diversity of the corner cases was not discussed. In [23]–[25], the diversity of identified failure modes was addressed using region elimination, reward augmentation and performance boundary heuristic respectively. However, they have not considered the presence of categorical parameters in the testing space. Moreover, the coverage of the failure modes has not been formally defined or quantified. These issues will be addressed in this paper.

B. Modeling Driver Interactions

Modeling the interaction between human drivers has been a crucial problem for multiple key areas of the self-driving communities including behavior prediction, motion planning,
V&V, etc. Existing approaches can be categorized into three groups [9]:

1) Rule-based models, e.g., intelligent driver model (IDM) [27], MOBIL model [28].
2) Learning-based models, e.g., variational auto-encoder (VAE) [29], generative adversarial network (GAN) [30].
3) Game-theoretic models [9], [17], [31]–[36].

Among them, game-theoretic models blend interpretability, data-efficiency of 1) and the flexibility of 2), which becomes the basis of our approach. Game-theoretic models represent driving as a game. Human drivers are assumed to be rational players that behave (near) optimally according to some utility functions. Nash [31] or Stackelberg [32], [33] equilibrium models have been applied to model human driving behaviors. However, they rely on the assumption that each player has an infinite level of rationality, which could be too strict considering that human drivers have to make quick decisions in a complex environment. Therefore, other researchers assumed bounded rationality of human drivers. To model this non-ideal nature of human driver behaviors, researches have applied level-k game theory [17], [34], quantal response [35] or cumulative prospect theory [9], etc. Among them, level-k game theory has been shown to outperform the equilibrium model on predicting human decision-making behaviors [37]. On the other hand, [31] and [36] considered the altruism of human drivers in a game-theoretic setting.

III. PROBLEM FORMULATION

A. Interaction-Aware Evaluation Pipeline

In this work, the research goal is to systematically evaluate the safety performance of a given VUT at interactive scenarios, specifically the roundabout entering scenario. The problem can be decomposed into two tasks. First, a testing space will be defined, which determines all the possible test cases (with interactions) to be evaluated. Second, a mechanism to select test cases from the testing space will be developed. In the first task, the testing space can be characterized by two sets of attributes: the first set defines the initial condition of the scenario; the second set describes the interactive and behavioral properties of the POV, which then forms the testing space (Section IV). In the second task, the test case generation procedure aims to evaluate the safety performance of a black-box VUT by adaptively discovering the failure modes of it through efficient sampling methods (Section V). The overall concept of the proposed interaction-aware evaluation framework is shown in Figure 1.

B. Roundabout Scenario Formulation

The roundabout scenario is more complex than other scenarios mainly due to the multiple entrances and exits, which result in a variety of possible interaction patterns between vehicles. According to common US traffic rules, e.g., [38], the entering vehicle should yield to all other vehicles already in the roundabout and approaching from upstream. For example, in a 4-way roundabout shown in Figure 2(a), the red vehicle should yield not only to the closest vehicle to its left, i.e., the blue vehicle, but also to the green vehicle. On the other hand, if the green or blue vehicle offers to yield the right-of-way to the red vehicle, it should also proceed responsively. Therefore, the previous evaluation scenario formulation [11], where only one POV is present, could not represent the diverse situations encountered by a real HAV. In this research, we focus on a roundabout scenario with two POVs and one VUT to generate more diverse test cases to challenge the VUT. The proposed framework in Section IV-D can scale to more POVs.

A running example throughout this paper is a 4-way roundabout inside Mcity, an HAV test facility at the University of Michigan, for which the geographical layout is shown.
in Figure 2(a). Each vehicle is assumed to drive along a predetermined reference path. Frenet frame [39] is used to represent the coordinate of each vehicle along the path.

Though three vehicles are present, we only consider pairwise interaction at any time for simplicity. For each pair, vehicles are modeled as double integrators as they move along their reference path, which is shown in Figure 2. The joint state of the two-vehicle system is \( [x_A, v_A, x_B, v_B]^T \). \( x_A, x_B \) are the longitudinal position and \( v_A, v_B \) are the longitudinal speed of vehicles A and B respectively. Here, all states are in the respective Frenet frame. The origin for both frames is the conflict point \( M \), the intersection point of the two reference paths. The input for each vehicle is the longitudinal acceleration, denoted as \( a_A \) and \( a_B \).

IV. POV LIBRARY CONSTRUCTION

The POV library should consist of driver models that capture the diverse driving styles and behaviors of human drivers for the target scenario. To approximate the decision-making procedure of human drivers, we assume that a POV is a bounded-rational game-theoretic agent, which takes the (near) optimal action with respect to its utility function and assumptions on the opponents.

A. Markov Game (MG) Formulation

Solving the optimal policy for one rational agent can be modeled as solving an MDP, which is defined by \( M = (\mathcal{X}, \mathcal{U}, \mathcal{P}, r, \gamma) \), with the state space \( \mathcal{X} \subseteq \mathbb{R}^n \), the action space \( \mathcal{U} \subseteq \mathbb{R}^m \), the transition dynamics of the environment \( \mathcal{P} : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X} \), the reward function \( r : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R} \), and the discount factor \( \gamma \in (0, 1) \).

When there are multiple rational agents interacting with each other, they can be modeled as a MG, which is a generalization of MDP [40], defined by the tuple \( \mathcal{G} = (\mathcal{N}, \mathcal{X}, \{\mathcal{U}^i\}_{i \in \mathcal{N}}, \mathcal{P}, \{r^i\}_{i \in \mathcal{N}}, \gamma) \). Other than what is defined in the MDP formulation, \( \mathcal{N} = \{1, 2 \ldots N_g\} \) denotes the collection of indices of \( N_g \) agents, \( \mathcal{U}^i \) and \( r^i \) denotes the action space and reward function of the \( i^{th} \) agent respectively. A policy for agent \( i \) is a state-action mapping, i.e. \( \pi^i : \mathcal{X} \rightarrow \mathcal{U}^i \). Then goal of \( i^{th} \) agent is to find a policy \( \pi^i \ast \) that maximizes its expected cumulative reward from any initial state \( x \):

\[
\pi^i \ast (x) = \arg\max_{\pi^i} V(x, \pi^i, \pi^{-i}) = \arg\max_{\pi^i} \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r^i(x_t, u_t) \right] = \pi^i \ast (x_t), u^{-i} = \pi^{-i} (x_t), x_0 = x
\]

In Eq. (1), \( x_t, u_t \) represent the state and action at time \( t \) respectively; \(-i\) represents the indices of all agents in \( \mathcal{N} \) except agent \( i \).

It is desirable for the POV library to cover a wide range of possible driving behaviors. With the MG formulation, we are able to achieve such modeling capability by either making the POV agent have different assumptions on the opponents’ policy \( \pi^{-i} \), or use different reward functions \( r_i \). Specifically, in this research, we adopt the idea of level-\( k \) game theory (for modeling different opponents) and SVO (for designing different rewards) to describe the diversified POVs. For simplicity, it is assumed that each POV is involved in a two-player MG, i.e. \( N_g = 2 \), with one opponent vehicle (OV), which could be the VUT or another POV.

B. Level-k Game Formulation

The level-\( k \) game theory model [41] is based on the idea that intelligent agents (such as human drivers) have finite level of reasoning depth. For a two-player game with agents A and B, instead of reaching an equilibrium with the opponent assuming that they are both infinitely rational, each agent assumes that her/himself is “one-level-smarter” than the opponent. The model first assumes that a level-0 policy \( \pi_0 \) is known a priori, which is a naïve policy that behaves in a non-interactive way. Then, a level-\( k \) agent (\( k > 0 \)) follows a utility-maximizing policy assuming that the opponent is a level-(\( k - 1 \)) agent. Using the level-0 policy as the starting point, the optimal policy for a level-\( k \) agent can be generated recursively. The optimal policy of a level-\( k \) agent A, denoted as \( \pi^A_k \ast \), can be calculated from:

\[
\pi^A_k \ast (x) = \arg\max_{\pi^A} V(x, \pi^A, \pi^B_{k-1})
\]

where \( \pi^B_{k-1} \) denotes the policy of a level-(\( k - 1 \)) agent B. Since \( \pi^B_{k-1} \) is already known and fixed when computing \( \pi^A_k \ast \), the two-player MG degenerates to an MDP that is easier to solve. On the other hand, since agents at different levels have different assumptions about their opponents, they represent agents with different thinking styles and complexity levels, contributing to the diversity of the POV library. The effectiveness of the level-\( k \) game formulation on explaining human driving behaviors has been validated in [34] using real traffic data. According to an experimental study in economics [42], human decision-makers are usually as high as level-2 thinkers. Therefore, we only consider agents that are up to level-2 in this research.
C. Social Value Orientation (SVO)

To systematically capture a set of diverse reward functions for the POV, we incorporate the SVO in the design of the POV library. SVO is a concept from the social psychology literature, which characterizes the degree of selfishness of an agent [43]. It quantifies the preference of an agent regarding the outcome for itself versus for others, which could be represented as an angle $\psi$ on a 2-dimension plane as shown in Figure 3(a). Here, different $\psi$ represent a range of personalities including egoism, altruism, competitiveness, etc. In the original game-theoretic setting, an agent is egoistic and will solely optimize for its own utility function, i.e. $\psi = 0$. When combining variable SVO with a game-theoretic driver model, as shown in [31], it could improve the accuracy of trajectory prediction, i.e., explain human driving behaviors better. Moreover, agents with different SVO can represent a continuous spectrum of human drivers, which complement the level-$k$ framework where drivers have discrete types and enrich the POV library. In this work, the SVO is combined with the level-$k$ game theory to model POV behaviors.

D. Constructing the POV Library for the Roundabout Scenario

Based on the level-$k$ game theory and SVO, we create a library with the following types of POV agents: the level-0 POV, the level-1 POV and the level-2 POV with varying SVO. Due to the non-competitive nature of driving tasks, we only consider the SVO angle in the 1st quadrant, i.e. $0 \leq \psi < \pi/2$. The SVO is only considered for level-2 POV because a level-0 POV is non-interactive and a level-1 POV assumes its opponent to be level-0, thus the SVO is undefined.

To construct the POV library, we first design the policy for a level-0 POV as a baseline. It is a non-interactive policy with a fixed speed profile, which captures the behavior of inattentive drivers. Next, to generate the policy for a level-$k$ POV ($k > 0$), a level-$(k-1)$ OV is needed in advance. The level-0 OV policy can be computed the same way as the level-0 POV policy. Therefore, the procedure starts with computing level-0 policies for both POV and OV, and then level-$k$ ($k > 0$) POV and OV are generated sequentially by assuming a level-$(k-1)$ opponent is known respectively. Although the targets are level-$k$ POVs, level-$k$ OVs are needed as the stepping stones to obtain higher-level POVs.

When multiple POVs are present in the scenario, a key problem is to model the interaction between POVs. Previous work on decision-making at roundabout [8], [44] assumes that each pair of interactive vehicles are in a two-player game with each other throughout the scenario. This formulation not only makes the algorithm not scalable to more vehicles, but also deviates from human driving behavior. In reality, a vehicle at the roundabout will shift its focus throughout the scenario. Before entering the roundabout, it will first look for incoming traffic from upstream of the roundabout; after entering, it will ensure safety by looking at downstream vehicles that are entering. In [10], the roundabout entering is decomposed into 3 zones: decision zone, transition zone and ring zone. In this work, we simplify that idea by dividing the POV policy into two phases, in which the POV will be involved in different games with different agents.

- Phase-1: the POV is in phase-1 when it has not entered the roundabout ($x_{POV} < 0$). Since it needs to yield to vehicles from upstream according to traffic rules [38], it is involved in a two-player game with the closest vehicle upstream, as shown by the blue (POV) and green (OV) vehicles in Figure 4(a). It assumes that the OV is either non-interactive with the POV (when OV is outside the roundabout) or in its phase-2 (when OV is inside the roundabout). This phase ends when the POV enters the roundabout ($x_{POV} = 0$).
- Phase-2: the POV is in phase-2 when it is inside the roundabout ($x_{POV} > 0$). It is involved in a two-player game with the vehicle outside the roundabout downstream in the nearest branch, as shown by the blue (POV) and red (OV) vehicles in Figure 4(b). The opponent is assumed to be in its phase-1. This phase ends when the OV enters the roundabout ($x_{OV} = 0$) or when POV passes OV.

Therefore, a POV is involved in one of the two-player games at any given time according to its location with respect to the roundabout. Each phase corresponds to a set of driving policies. Since the combination of level-$k$ game theory and SVO is used to model the POVs, a level-$k$ phase-1 POV assumes that its opponent is following a level-$(k-1)$ phase-2 policy; by the same token, a level-$k$ phase-2 POV assumes that its opponent is following a level-$(k-1)$ phase-1 policy.
In addition, if the POV is in phase-1 but there is no vehicle upstream inside the roundabout, then the POV follows a level-0 phase-1 policy. If the POV is in phase-2 but there is no vehicle downstream in the roundabout, then the POV follows a level-0 phase-2 policy. If there are multiple vehicles in phase-2 (inside the roundabout), each of the following POVs in phase-2 will regulate the distance to the preceding vehicle with a rule-based car-following algorithm, multiple vehicles in phase-2 (inside the roundabout), each of the following POVs in phase-2 will regulate the distance to the preceding vehicle with a rule-based car-following algorithm.

Finally, we can combine this two-phase framework with the aforementioned training procedure for level-k agents to compute the policies for all phases and construct the POV library. The procedure is demonstrated in Figure 3. We first assume that the level-0 policies for phase-1 and phase-2 are both known: a level-0 phase-2 POV follows a constant speed; a level-0 phase-1 POV decelerates at \(1\,m/s^2\) if \(v_{POV} > v_{max}\), then follows a constant speed. Next, we can generate the level-1 phase-1 policy from the level-0 phase-2 policy, and generate the level-1 phase-2 policy from the level-0 phase-1 policy. In this way, we can compute any level-k phase-1 or phase-2 POV policies from the lower-level policies by induction. Finally, a full POV policy is acquired by combining phase-1 and phase-2 policies with the same level and SVO.

In addition, we need to ensure that the behaviors of multiple POVs are compatible with each other. If two POVs have the same level, or one is level-0 and the other is level-2, then both POVs will have wrong assumptions on the opponents, thus collisions may happen. Therefore, we pick four compatible combinations of the two POVs as the “POV categories” for the testing space, which form the POV library that will be considered in this paper:

1. Level-0 POV #1; level-1 POV #2;
2. Level-1 POV #1; level-0 POV #2;
3. Level-1 POV #1; level-2 POV #2 with SVO \(\psi\);
4. Level-2 POV #1 with SVO \(\psi\); level-1 POV #2.

### E. POV Behavior Generation Using Reinforcement Learning

To compute the driving policy for a level-k agent (\(k > 0\)), we use single-agent reinforcement learning (RL) to solve the MDP. To train a level-k POV, we model it as an agent operating in an environment consisting of level-\((k-1)\) OV. The same procedure applies to OV. To incorporate the factor of SVO, we consider the SVO angle \(\psi\) as an extra state of the model when a level-2 POV is trained to generate a continuum of level-2 POVs.

1) **Reinforcement Learning Formulation:** For a level-k agent, the state space includes continuous physical states of the POV and the OV, denoted as \(X = \mathcal{X}\). For a level-2 POV, the SVO angle is considered a constant state sampled from \(\mathcal{X} = X \times [0, \pi/2)\) and does not change during the whole scenario. The action space is a set of discrete acceleration input, ranging between \([a_{min}, a_{max}]\).

The reward function reflects the goal of driving for each agent. We assume that the reward function can be represented as a linear combination of \(K\) reward feature terms:

\[
r(x,u) = W^T \Phi(x,u) = \sum_{i=1}^{K} w_i \phi_i(x,u)
\]

where \(w_i\) is the weight of each term, \(\phi_i(x,u)\) represents each feature. Their definitions are as follow:

1. \(\phi_1 = \phi_{acc} = -a_{POV}(t)^2\): negative reward on acceleration action of the POV.
2. \(\phi_2 = \phi_{step} = -1\): constant step cost to encourage shorter time to finish.
3. \(\phi_3 = \phi_{vmax} = -1_{v_{POV}(t)}>v_{max}\): penalty on over-speeding inside the roundabout; \(1\) is the indicator function.
4. \(\phi_4 = \phi_{dist} = 1_{dist(t)<d_{critical}}(d_{critical}-dist(t))\): penalty for being too close. \(dist\) is the distance to the OV (in the Cartesian frame); \(d_{critical}\) is the safety-critical distance between POV and OV.
5. \(\phi_5 = \phi_{crash} = 1_{dist(t)<d_{crash}}\): penalty on collision. \(d_{crash}\) is the threshold distance for defining a collision between POV and OV.
6. \(\phi_6 = \phi_{TTC} = 1_{TTC(t_1)>TTC_{min}}(TTC(t_1) - TTC_{min})\): penalty for the time-to-collision between POV and OV (in the Frenet frame) being too small at the terminal time \(t_1\).

All the reward parameter values are shown in Table I.

| Parameter | Value | Parameter | Value |
|-----------|-------|-----------|-------|
| \(a_{min}\) | -4.0 m/s\(^2\) | \(a_{max}\) | 3.0 m/s\(^2\) |
| \(v_{max}\) | 10.0 m/s | \(TTC_{min}\) | 4.0 s |
| \(d_{critical}\) | 8.0 m | \(d_{crash}\) | 6.0 m |

For level-2 POV with SVO angle \(\psi\), the feature \(\phi_1\) is modified such that the acceleration of the OV is also considered:

\[
\phi_1 = -(\cos(\psi)a_{POV}(t)^2 + \sin(\psi)a_{OV}(t)^2)
\]

Since \(\phi_2\) will take effect on both vehicles equally in each episode, and \(\phi_4 \sim \phi_6\) are safety features shared by both vehicles, they will not be regulated by the SVO. \(\phi_5\) will not be considered for SVO because the speed violation of the OV is independent to the action of a POV.

2) **Learning Level-k Policies Using Q Learning:** To learn the optimal policy of a POV/OV, we apply the Q-learning method [45]. First, the action-value function \(Q\) is defined as:

\[
Q(x,u|\pi) = E_{\pi} \left[ \sum_{t=0}^{T} \gamma^t r_t | x_0 = x, u_0 = u \right]
\]

Q-learning uses temporal difference to estimate the optimal \(Q\) function, i.e. \(Q^*(x,u|\pi^*)\), and learn the optimal policy \(\pi^*\). For details, please refer to [45]. In this work, the reinforcement learning algorithm we use is Double Deep-Q network (DDQN) [46]. DDQN is based on the Deep-Q network (DQN) [47] method. It addresses the problem of overestimating future return of DQN by decoupling the action evaluation and action selection into max operations in two different Q-networks.

### V. ADAPTIVE TEST CASE GENERATION

#### A. Problem Formulation

In Section IV, we systematically generate a library of interactive POVs, which are characterized by the SVO \(\psi\).
and level-$k$. For the two-POV roundabout scenario, the initial condition $x_0$ is defined as:

$$x_0 = [x_{POV1}^0, x_{POV2}^0, x_{VUT}^0, b_{POV1}^0, b_{POV2}^0, b_{VUT}]^T$$  \hspace{1cm} (6)

By combining $x_0$ with the SVO of each POV $(\psi_1, \psi_2)$, the category $c$ of two-POV combination, we build the testing space, denoted as $S$, where each case $s$ is:

$$s = [x_0^T, \psi_1, \psi_2, c]^T$$  \hspace{1cm} (7)

A test case generation scheme is needed to pick $N$ test cases $s = [s_1, \ldots, s_N]$ from the testing space to locate low-score cases (“failure modes”) of the VUT. The main challenge is that different VUTs may have different performance profiles and weaknesses, and thus the failure modes are unknown at the beginning of the testing. Therefore, the proposed scheme should select new cases based on past test results to adaptively search for the weaknesses of each VUT as the testing proceeds. The goals of the test case generation scheme are two-fold:

1) Challenge: find test cases where the VUT performs poorly (i.e. identify the weakness).
2) Coverage: explore (possibly disjoint) regions of poor performance as many as possible.

For each test case $s$ and a given VUT, we define the performance score $P(s)$ as:

$$P(s) = \mu_1 I_{crash}(\tau) + \mu_2 P_{safety}(\tau) + \mu_3 P_{task}(\tau),$$  \hspace{1cm} (8)

where $\tau$ is the resulting joint trajectory of POVs and the VUT; $I_{crash}$ is the indicator for collision; $P_{safety}$ is the safety score; $P_{task}$ is the score of task accomplishment; $\mu_1, \mu_2, \mu_3$ are weighting factors. The failure modes of a VUT are defined as: $S_f(\lambda) = \{s|s \in S, P(s) < \lambda\}$, where $\lambda$ is the performance score threshold for failure. All cases belong to $S_f(\lambda)$ are failure cases. The key objectives can be translated to maximizing the failure mode coverage.

B. Adaptive Testing Scheme Overview

We will generate $N$ test cases in batches, with batch size $n_b$. For the testing space $S$, the last attribute for POVs category $c \in C$ is a categorical variable, while all others are continuous variables. Then, the testing space can be decomposed into two parts, written as: $S = S \times C$, where $S$ is the subspace with continuous variables.

Sampling from continuous and categorical attributes need to be treated differently, since samples in one category have little correlation with samples in another category. Therefore, the test case generation scheme can be divided into two stages for each batch, as shown in the lower part of Figure 1. In the 1st stage, we allocate the quota of samples into different POV categories, i.e. assign $n_i^c$ cases to category $c$ at batch $i$. In the 2nd stage, we sample new test cases within each category from $S$ using an adaptive sampling method based on the Gaussian process regression (GPR). We will introduce this two-stages scheme next.

C. Adaptive Sampling Within Each POV Category

In this section, we introduce the 2nd-stage of test case generation. First, to assess the quality of test samples in each POV category, we define the criterion FMC to formally characterize the failure mode coverage in each subspace $S$:

$$FMC(s_c, \rho, \lambda) = \int_{\cup B_{dim}(\rho, s)} 1 dv$$  \hspace{1cm} (9)

In Eq. (9), $s_c$ are the test cases within the $c$th POV category, $dim$ is the dimensionality of $S$, and $B_{dim}(\rho, s)$ is a $dim$-dimensional hyper-ball centered around case $s$ with radius $\rho$. $S_f(\lambda)$ consists of all the failure cases in $s_c$. The FMC evaluates the generalized volume of the union of hyper-balls centered around identified failure cases, which characterizes the coverage. Here, all dimensions of $S$ are normalized between $[0,1]$. Figure 5 is a graphic illustration of the FMC in 1-dimension.

Within each POV category, we conduct adaptive sampling with a GPR meta-model. The adaptive sampling method involves the alternation between two steps: updating meta-models with prior testing results and generating samples from the new meta-models [48]. GPR is a non-parametric probabilistic model [49] that is popular for its ability to model complex functions and infer the value around unexplored regions [50]. The key idea is to maintain and update a GPR-based meta-model according to existing samples, and use the meta-model to generate new samples.

Gaussian process (GP) is a stochastic process, for which the joint distribution of every finite collection of random variables follows a multivariate Gaussian distribution. A GP is characterized by its mean function $m(s)$ and covariance function $k(s,s')$ (kernel), as show in Eq. (10).

$$f(s) \sim GP(m(s), k(s,s'))$$  \hspace{1cm} (10)

In this work, we use GP as the surrogate model of the performance score profile of each VUT, as shown in Eq. (11). GP characterizes the prior belief of $P(s)$, which is updated by the new test samples and their test results.

$$P(s) = \delta + f(s), \text{ where} \hspace{1cm} f(s) \sim GP(0, k(s,s'|\theta)), \delta \sim N(\beta, \sigma^2)$$  \hspace{1cm} (11)

In Eq. (11), $\theta$ is the kernel parameter, $(\beta, \sigma, \vartheta)$ constitute the hyper-parameters of the model. In this work, we use a zero
mean function and a square exponential kernel function for the GPR model, as shown in Eq. (12), where \( \theta = [\theta_1, \theta_2]^T \). Hyper-parameters are optimized using maximum likelihood estimation.

\[
k(s, s' | \theta) = \theta_1^2 \exp \left(-\frac{(s - s')^T(s - s')}{2\theta_2^2}\right)
\]

(12)

With a GPR model \( \hat{P}(\cdot) \) based on existing testing cases \( s \) and their results \( y_i \), for any unobserved query case \( s^0 \), the joint distribution of \( \hat{P}(s^0 | s, y) \) and \( y \) is also a Gaussian distribution. Therefore, the conditional mean and variance of \( \hat{P}(s^0 | s, y) \) are:

\[
\mathbb{E}(\hat{P}(s^0 | s, y)) = k(s^0, s)k(s, s) + \sigma^2 I)^{-1}(y - \beta)
\]

\[
\text{Var}(\hat{P}(s^0 | s, y)) = k(s^0, s^0) - k(s^0, s)k(s, s) + \sigma^2 I)^{-1} k(s, s^0)
\]

(13)

The procedure of adaptive sampling is illustrated in Algorithm 1. Details about selecting new samples using the meta-model (lines 6-9) are explained in Section V-D.

Algorithm 1 Adaptive Sampling for One Step

Input: batches number \( i \); batch size \( n_i^c \); previous GPR model \( \hat{P}_i^{c-1} \); exploration factor \( \epsilon_0 \).
Output: test cases with POVs of category \( c \), \( s_i^c \), and test results \( y_i^c \); updated GPR model \( \hat{P}_i^c \).

1: if \( i = 1 \) then
2: Sample initial test batch \( s_i^1 \) uniformly from \( S \).
else
4: Randomly sample \( p \) queries \( \hat{s} \) from \( S \) based on uniform distribution (\( p \gg n_i^c \)).
5: \( \epsilon = \epsilon_0 \alpha^{i-1} \).
6: Pick \((1 - \epsilon)n_i^c \) queries from \( \hat{s} \) according to \( C_{\text{exploit}}(\lambda \hat{s}) \) as exploration samples, denoted as \( s_{\text{exploit}} \).
7: Pick \( en_i^c (1 - r_{\text{MBM}}) \) queries from \( \hat{s} \) according to \( C_{\text{explore}}(\lambda \hat{s}) \) as a part of exploration samples, denoted as \( s_{\text{explore-1}} \).
8: Sample \( en_i^c r_{\text{MBM}} \) cases according to Algorithm 2 as the rest of exploration samples, denoted as \( s_{\text{explore-2}} \).
9: \( s_i^c = [s_{\text{exploit}}, s_{\text{explore-1}}, s_{\text{explore-2}}] \).
10: end if
11: Execute test cases \( s_i^c \), acquire results \( y_i^c = P(s_i^c) \).
12: Fit/Update the GPR model: \( y = \hat{P}_i^c(s) = \hat{P}(s|s_i^c, y_i^c) \).

D. Test Case Selection

To achieve good coverage of the failure modes, we need to balance exploitation and exploration when choosing a new batch of samples. On the one hand, samples with low predicted \( \hat{P}(s) \) represent more challenging cases, which are preferred for the goal of challenge. On the other hand, it is desirable to explore regions with high uncertainty to pick more informative samples for a better meta-model, which helps coverage. We will describe the criteria \( C(s) \) that evaluate the potential quality of a query \( s \) for both exploration and exploitation, and how to balance between these two goals.

1) Exploitation: Modified Expected Improvement (MEI) Criterion: Expected improvement (EI) is a popular acquisition function for choosing the most promising samples in Bayesian optimization [51], a closely-related method with the GPR-based adaptive sampling. EI indicates the possible improvement on optimal value brought by the new query.

\[
C(s) = \mathbb{E}(I(s)), \text{ where } I(s) = \max(P_{\text{min}} - \hat{P}(s), 0)
\]

(14)

In Eq. (14), \( P_{\text{min}} \) is the minimal performance score achieved by existing test samples. To better suit our problem settings and goals, we modify the EI criterion to explicitly consider the possible FMC improvement from the new query:

\[
C_{\text{exploit}}(s) = \max(I(s) \times V_{\text{add}}), \text{ where } I(s) = \max(\lambda - \hat{P}(s), 0)
\]

(15)

In Eq. (15), \( V_{\text{add}} \) is the additional volume covered by \( B_{\text{dim}}(\rho, s) \), \( \lambda \) is the performance score threshold for a failure. Since the actual \( V_{\text{add}} \) is hard to compute due to its highly non-convex shape, we use an approximation of \( V_{\text{add}} \) for the above computation, as demonstrated in Figure 6(a) and Eq. (16):

\[
V_{\text{add}} \approx V_{\text{add}} \propto \rho_{\text{dim}} - (\max(\rho - d_{\text{min}}(s), 0))_{\text{dim}}
\]

(16)

where \( d_{\text{min}}(s) \) is the distance from \( s \) to the nearest identified failure case. A sample with high MEI means that it has a high chance of being failure case, and it is far from previous failure cases.

2) Exploration #1: Standard Deviation Criterion: For exploration, a straightforward query criterion is the posterior standard deviation:

\[
C_{\text{explore}}(s) = \hat{\sigma}(s)
\]

(17)

where \( \hat{\sigma}(s) = \sqrt{\text{Var}(\hat{P}(s))} \). The higher \( C_{\text{explore}}(s) \) indicates that the current GPR meta-model has higher uncertainty at \( s \), i.e., higher exploration return.
3) Exploration #2: Sampling on the Behavior Mode Boundary: In the aforementioned sampling schemes, the only information from past test cases that aids the adaptive sampling is the performance score. However, utilizing more contextual information of test cases for selecting new test queries can be beneficial, especially for a high-dimensional testing space. Therefore, we consider the behavior mode of a test case as additional information to guide adaptive sampling. The idea is inspired by [24], where the performance mode boundaries of a system under test are identified in hope of generating informative test cases. The intuition is that a switch between performance modes has the potential to induce confusion and fail the VUT.

We define the behavior mode boundary (BMB) for our test scenario. In the roundabout entering scenario with two POVs, the BMB is defined as the final passing order of the VUT. There are three BMBs: VUT being the first to pass, the second to pass or the last to pass.

The process of generating test cases using BMB is illustrated in Algorithm 2. Here, the BMB is estimated locally using existing test cases of different behavior modes. Then, new test cases are sampled along the estimated BMB. The procedure is visualized in Figure 6. Therefore, the BMB becomes another heuristic that guides the exploratory sampling in the testing space.

Algorithm 2 Sampling Near the Behavior Mode Boundary

Input: number of cases to be sampled along BMBs \( n_{BMB} \); all previous test cases \( s \), and their corresponding behavior modes \( BM^c \).

Output: \( s_{BMB} \): test cases sampled at the behavior mode boundaries.

1. \( s_{BMB} = [ ] \)
2. for \( k \leftarrow 1 \) to \( n_{BMB} \)
3. Randomly pick two behavior modes, \( BM_1 \) & \( BM_2 \)
4. Randomly pick a point from mode \( BM_1 \): \( \hat{s}_1 \)
5. Find the closest neighbor of \( \hat{s}_1 \) in mode \( BM_2 \): \( \hat{s}_2 \).
6. Find the closest neighbor of \( \hat{s}_2 \) in \( BM_1 \): \( \hat{s}_1 \).
7. Draw a sample \( s \) randomly on the bisection subspace of \( \hat{s}_1 \) and \( \hat{s}_2 \): \( s_{BMB} = [ s_{BMB}, \hat{s} ] \)
8. end for

4) Balancing Exploration and Exploitation: For each batch, we pick test cases according to the three criteria above. The percentage of cases for exploration and exploitation are defined as more exploration, and bias towards exploitation as more of the other cases are chosen based on the standard deviation criterion (e.g. \( r_{BMB} = 1/4 \)). It is important to note that the proposed method has different goals compared to the Bayesian optimization methods, which share similar formulation and usages on the GPR model and the EI criterion. The Bayesian optimization focuses on solving an optimization problem, i.e. finding one global optimal point on the performance surface, while the proposed adaptive sampling method focuses on identifying failure modes, i.e. finding more “valleys” of the performance surface. Therefore, the exploration criteria are added compared to standard Bayesian optimization methods.

E. Sample Allocation Between POV Categories

In this section, we discuss how to distribute test samples into different categories of POVs. Since the potential of finding failure cases in each category is unknown a priori, the sample allocation method needs to balance exploration (finding out which category is the most promising) and exploitation (sample more from the most promising category). The key objective can have two different forms: (1) to maximize the total number of identified failure cases from all categories; (2) to find more failure cases in total, while preferring failure cases that are scattered in more categories. Since both forms of the objective have their significance and applicability, we propose two sample allocation methods to handle them separately.

1) Balancing Failure Case Number and Diversity: To handle the trade-off between diversity and richness of the failure cases, we propose to find the best allocation policy by solving a stochastic optimization problem.

First, we define the failure case reward for each category as a function of the number of failure cases \( \bar{n}_c^i \) at batch \( i \) and category \( c \in C \), denoted as \( \beta_c^i \):

\[
\beta_c^i = h(\bar{n}_c^i) \quad (18)
\]

\( h() \) is a concave and monotonically increasing function defined over \([0, +\infty[\). In this research, we choose \( h(x) = x^\tau, \tau \in (0, 1) \). Such an \( h() \) will encourage more failure cases in each category. On the other hand, the reward for each extra failure case of the same category will diminish \((dh/d\bar{n}_c^i) \) decreases as \( \bar{n}_c^i \) increases), which encourages finding failure cases in other categories rather than concentrated in a single category, i.e., exploration is encouraged.

We define the optimization problem below. Since \( \bar{n}_c^i \) is unknown beforehand, we will estimate it according to the results of previous iterations, which can be written as \( \bar{n}_c^i = \eta_c^i n_b q_c^i \). Here, \( \eta_c^i \) is the variable being optimized, the ratio of samples assigned to category \( c \) at batch \( i \): \( n_b \) is batch size, and \( q_c^i \in [0, 1] \) is a random variable describing the belief on the failure case probability (FCP) of \( c \)th category in batch \( i \). Specifically, the test-case sampling from each category can be approximated as independent Bernoulli trials, with the “probability of successfully finding a failure case in each trial” being \( q_c^i \). The goal is to maximize the overall expected failure case reward \( R_{fc} \), as stated in Eq. (19).

\[
\begin{align*}
& \max \sum_{q_c^i} \mathbb{E}_{q_c^i} (R_{fc}) = \sum_{c=1}^{|C|} \mathbb{E}_{q_c^i} (\beta_c^i) \\
& \text{s.t.} \sum_{c=1}^{|C|} \eta_c^i = 1 \\
& \eta_c^i \geq 0, \quad c = 1 \ldots |C| \\
& |C| = \text{the cardinality of the set of categories } C. \quad q_c^i \sim Beta(\alpha_c^i, \beta_c^i), \text{ where}
\end{align*}
\]
\(a_i^t, \beta_i^t\) are the parameters that control the shape of the Beta distribution [52]. Without loss of generality, we assume \(q_i^1 \sim \text{Beta}(1, 1)\), i.e. \(q_i^1\) has a uniform prior. Assuming that for batch \(i\) and category \(c\), \(a_i^t\) failure cases and \(\beta_i^t\) non-failure cases were found, and we denote the observation \(D_i^t\) as \(D_i^t = [a_i^t, \beta_i^t]\). Then, we can conduct a Bayesian update on \(q_i^t\) according to the existing testing results:

\[
\begin{align*}
Pr(q_i^t|D_i^t) &\propto Pr(D_i^t|q_i^t) Pr(q_i^t) \\
Pr(q_i^{t+1}) &= Pr(q_i^t|D_i^t), 
\end{align*}
\]  

(20)

where the 1st line is the Bayes theorem; the 2nd line states that the the prior of \(q_i^{t+1}\) is defined as the posterior of \(q_i^t\). Since Beta distribution is the conjugate prior of the binomial distribution [52], the posterior distribution of \(q_i^t\) will have the same form, i.e. a Beta distribution. Therefore, we can compute the distribution of \(q_i^{t+1}\) with the following updating rules on its parameters:

\[
\begin{align*}
q_i^{t+1} &\sim \text{Beta}(a_i^{t+1}, \beta_i^{t+1}), \\
&\text{where } a_i^{t+1} = a_i^t + \hat{\gamma}(a_i^t - 1) \\
&\beta_i^{t+1} = \beta_i^t + \hat{\gamma}(\beta_i^t - 1) 
\end{align*}
\]  

(21)

When the constant \(\hat{\gamma} = 1\), Eq. (21) is equivalent to Eq. (20). However, in the adaptive sampling procedure, we acquire more knowledge about the failure modes with more test cases, and subsequently, the FCP \(q_i^c\) will likely not stay stationary with respect to \(i\). Therefore, we assume \(\hat{\gamma} \in (0, 1)\) and use it as a discounting factor to attenuate earlier information.

In each iteration, we solve the optimization problem Eq. (19) to acquire an optimal allocation of samples in each category, i.e. \(\pi_i^c\). This strategy will invest more samples in better-performing categories, while maintaining some samples in all categories.

2) Greedy Optimization for Total Failure Case Number: When the only goal is to find as many failure cases as possible, we model the reward of each sample as a binary variable, i.e. 1 if the sample is a failure case, 0 if not. Then, this problem is akin to the stochastic multi-arm bandit (MAB) problem [53]. In a MAB problem, one needs to allocate limited resources to competing choices to maximize the expected reward (minimize the total regret). Each choice has a stochastic reward, whose distribution is unknown a priori. There have been many classical algorithms for solving the MAB problem, including upper confidence bound (UCB) [54], Thompson sampling [55], etc. Both methods can achieve sub-linear regret with respect to the number of samples [53]. In standard MAB formulation, the reward for each choice follows a stationary distribution. However, in our problem setting, the reward for each category (i.e. \(q_i^c\)) will change over time due to more informed sampling, as explained in section V-E.1. Therefore, directly applying the above methods will not yield ideal results.

In this paper, we propose to implement the UCB algorithm for sample allocation with some modifications. At each timestep \(t\), we pick a new sample from category \(c_i^{UCB}\) based on the following criterion:

\[
c_i^{UCB} = \arg \max_{c \in C} \hat{Q}_i(c) + U_i(c)
\]

where \(U_i(c) = \sqrt{\frac{2 \log t}{N_i(c)}}\), \(\hat{Q}_i(c) = q_i^c\).

(22)

Here, \(i\) is the batch number that the \(i^{th}\) sample belongs to; \(\hat{Q}_i(c)\) and \(U_i(c)\) are the estimated reward and upper confidence bound on that reward estimation respectively for category \(c\); \(N_i(c)\) is the number of total samples in category \(c\) as of time \(t\). The algorithm starts by exploring the under-discovered categories \((U_i(c)\) dominated). Then it gradually turns to exploit the category with the highest estimated reward \((\hat{Q}_i(c)\) dominated). There are several modifications we made compared to the standard UCB algorithm:

1) In standard UCB, \(\hat{Q}_i(c)\) equals to the sample mean of the reward, i.e. the cumulative ratio of failure cases for each category. In our setting, since the reward distribution for each category is not stationary, we set \(\hat{Q}_i(c) = q_i^c\) to ensure that the estimated reward adapts to the dynamics of the FCP.

2) Since we generate and execute test samples in batches, \(\hat{Q}_i(c)\) will stay constant throughout each batch, whereas \(\hat{Q}_i(c)\) is updated at every new sample in the standard UCB.

VI. SIMULATION RESULTS

In this section, we first evaluate the performance of the proposed sample allocation methods using simplified experiments. Then, we conduct interaction-aware testing for a rule-based baseline VUT in MATLAB simulations to validate the effectiveness of the POM library and the performance of the proposed test case sampling scheme.

A. Comparison Between Sample Allocation Methods

We evaluate the two proposed sample allocation methods in four simplified experiments. During test case sampling, the sample allocation (1st stage) is followed by adaptive sampling within each category (2nd stage), which introduces extra uncertainty due to the randomness in the initialization and in the complex sampling process. Therefore, to reduce the impact of the 2nd-stage process and to compare the performance of the sample allocation methods in isolation, we abstract the outcome of the 2nd stage into simple Bernoulli trials with different non-stationary parameters for each category. Here is the setup for the simplified experiments:

- Each test sample is an independent trial with a binary outcome, either 0 (not a failure case) or 1 (failure case). The probability of a case being a failure case is \(p_c(N_c) \in [0, 1]\), which depends on the property of the category \(c\), and the number of samples so far in that category, denoted as \(N_c\).
- There are four categories \((C = \{c_1, c_2, c_3, c_4\})\), each with a different FCP dynamics \(p_c(N_c), c \in C\). Each category has a fixed maximum FCP.
- We conduct four experiments. In all of them, no failure cases could be found in \(c_1\) \((p_{c_1} = 0)\); \(p_{c_2}\) is set to be the same linear function; \(p_{c_3}, p_{c_4}\) are linear in the first two experiments, and quadratic in the last two experiments, each with different parameters. The presented FCP...
The indices of experiments are shown in the parentheses in the legend. The FCP dynamics for each category for the 4 simplified experiments are shown in Figure 7.

B. Rule-Based VUT Algorithm

For the roundabout scenario, we design a rule-based speed planning algorithm for the VUT. This is a flawed algorithm by design such that it has failure modes to be discovered. Its decision-making process has 3 modes:

1) The VUT starts by coasting at a constant speed. Go to mode 2 when it is \( x^{(b1)} \) close to point \( M \) in Figure 2.

2) The speed of the VUT is regulated by a PID controller to a target speed \( v_{tar} \). The VUT predicts the projected final gap with both POVs, and makes an online decision between three options: yield (to both), slip in (between two POVs), or pass (in front of both). If “yield” or “slip in”, then decelerate (set \( v_{tar} = 0 \)); Else, \( v_{tar} = v_0 \), where \( v_0 \) is a default desired speed. Go to mode 3 when the distance from VUT to point \( M \) is less than \( x^{(b2)} \)

3) Execute the last decision from mode 2. If “yield”, wait for both POVs to pass, and then perform car-following with the last POV; if “slip in”, wait for the 1st POV to pass, then perform car-following with the first POV; if “pass”, then maintain the PID control in mode 2 with \( v_{tar} = v_0 \).

This same VUT algorithm will be the subject VUT of the experiments in the following sections.

C. Various Interactive Test Cases

In the roundabout scenario, the diversity of interaction patterns is reflected by the different passing orders of the involving vehicle, i.e. the behavior modes. The order is jointly determined by the timing of entering the roundabout and the yielding/passing decision for each pair of vehicles during their right-of-way negotiations. In this section, we present a variety of simulated test cases generated by the proposed POV library that demonstrates such diversity.

In Figure 8, we present two test cases with the same initial condition but with different POV properties. In the 1st case, as shown in Figure 8(a), the level-1 POV #2 accelerates to pass POV #1, then passes the VUT. The level-2 POV #1 has a cooperative SVO value. It first decelerates to yield the POV #2, then yields the VUT as well. The VUT first slows down to yield to the POV #2, then accelerate to enter the roundabout ahead of the POV #1.

In the 2nd case (Figure 8(b)), all the initial conditions and attributes are the same as in the 1st case, except that the SVO of the POV #1 decreases to 0.2\( \pi \), i.e. POV #1 is less cooperative. Major difference in their behaviors starts when the VUT interacts with POV #1 after \( t = 6s \): the VUT still chooses to slip in between the two POVs, while POV #1 chooses to pass ahead of the VUT. Then a collision happens between POV #1 and VUT. A slight change in the SVO parameter incurs completely different interactive behaviors and reveals a failure mode of this VUT.

In Figure 9, we present two test cases with different initial conditions. In the 3rd case (Figure 9(a)), the level-1 POV #1 first passes the POV #2, then yields to the approaching VUT. The VUT first reduces speed, and then accelerates to pass first after observing the yielding of POV #1.
In the 4th case (Figure 9(b)), the initial position of the POV #2 is closer than the previous case. The POV #1 first accelerates harder to pass the POV #2, then it accelerates again to pass the VUT. The POV #2 keeps a steady speed and also passes the VUT. The VUT stops before the roundabout to yield both POVs before it proceeds. By altering only the initial conditions, we also generate different interaction patterns and passing orders of vehicles.

In addition, to demonstrate that the proposed POV library is scalable to different roundabout layouts, we present two additional test cases at another 3-branch roundabout from the INTERACTION dataset [56] in Figure 10. Each vehicle follows a reference path defined along the lane center, while the POVs follow the same driving policy computed in Section IV to plan their longitudinal actions.

In the 5th case (Figure 10(a)), all vehicles enter the roundabout from different branches and exit at the same branch (top left). The POV #1 first accelerates to pass the POV #2, then keeps constant speed to pass the VUT; the POV #2 accelerates to pass the VUT; the VUT yields to both POVs. Thus, we replicate the interaction pattern of the 4th case in a different roundabout scenario.

In the last case, (Figure 10(b)), the initial states for all vehicles are the same as in the 5th case, while the POV #2 exits the roundabout one intersection earlier, thus there is no potential conflict between POV #2 and VUT. The POV #1 first accelerates to pass the POV #2, then passes the VUT;
The sampling space for each experiment, where each sample is denoted as \( V_{\text{UT}} \), is given to us. The VUT category \( c \) is also fixed in each experiment. Only a subset of the testing space will be sampled, i.e. \( s^* = [x_{\text{POV1}}^0, x_{\text{POV2}}^0] \) for the ground truth, which is generated with 40000 samples using uniform sampling. As shown in Fig. 11(a), the ground truth has several disjoint regions with red colors, i.e. the failure modes. For each method from Fig. 11(b) to 11(e), the total number of test cases is \( N = 400 \). Specifically, for the proposed method, the batch size is \( n_b = 20 \), and it runs for 20 batches.

Uniform sampling locates one failure region with very few failure cases; simulated annealing identifies many failure modes, but all concentrated around two failure modes at the lower-left corner, which results in low FMC value; subset simulation performs better than simulated annealing, but failure cases are also not distributed in multiple failure modes. The proposed method stands out by identifying failure cases at more diverse failure modes and achieving the highest FMC value. Specifically, failure cases in Fig. 11(e) are not too concentrated thanks to the MEI exploitation criterion, which explicitly discourages failure cases that are too close. With only 1% of the sample sizes of the ground truth, the proposed method discovers most of the failure modes qualitatively.

Then, we present the comparison of quantitative results when the sampling space has a higher dimension in Table III. Two VUT categories are evaluated:

- Category #1: level-0 POV1 and level-1 POV2, with \( s^* = [x_{\text{POV1}}, x_{\text{POV2}}] \) (4-dimension);
- Category #4: level-2 POV1 and level-1 POV2, with \( s^* = [x_{\text{POV1}}, x_{\text{POV2}}, v_{\text{POV1}}, v_{\text{POV2}}, \psi] \) (5-dimension).

The high dimensionality of the sampling space makes the test case generation results sensitive to the initialization of the proposed and comparison methods. Therefore, we run 100 repeated test runs for each method (except uniform sampling) and compare their average performance. For uniform sampling, to ensure that each dimension is discretized in the same resolution, we allow for more test samples (625 cases for category #1, 1024 for category #4). Each experiment with all other methods has 400 test cases. The adaptive sampling method from our previous work [11] is also added for comparison. Even with the advantage on the case number, the uniform sampling method only achieves slightly better results than simulated annealing. Subset simulation performs

\[ v_{\text{UT}} = 8.0 \text{ m/s} \]
Fig. 11. Testing experiment results with level-1 POV #1 and level-0 POV #2. \( v_{POV1}^0 = v_{POV2}^0 = 9 \text{ m/s}, v_{VUT}^0 = 10 \text{ m/s} \). The samples are color coded by the performance score (truncated at 0 and -600), where red means lower score.

TABLE III
RESULTS COMPARISON BETWEEN DIFFERENT TEST CASE GENERATION METHODS IN ONE CATEGORY

| Method                        | FMC(\(s^*, 0.05, -500\)) |
|-------------------------------|-----------------------------|
|                               | L-0 POV1                    | L-2 POV1 | L-1 POV2 | L-1 POV2 |
| Uniform sampling              | \(0.92 \times 10^{-4}\)     | \(1.97 \times 10^{-5}\) | |
| Simulated annealing           | \(0.79 \times 10^{-4}\)     | \(0.68 \times 10^{-5}\) | 1.54 \times 10^{-4} | 2.21 \times 10^{-5} |
| Subset simulation             | \(4.07 \times 10^{-4}\)     | \(3.55 \times 10^{-5}\) | |
| Adaptive sampling in [11]     | \(4.38 \times 10^{-4}\)     | \(3.59 \times 10^{-5}\) | 4.86 \times 10^{-4} | 4.09 \times 10^{-5} |
| Proposed w/o BMB              |                             |          |          |
| Proposed method               |                             |          |          |

better than the above two methods, but fall short compared to the three variant of the GPR-based adaptive sampling methods. By modifying the sample selection techniques, the proposed method achieves significant FMC improvement compared to the scheme in [11] in both experiments. Moreover, By adding the step of BMB identification, the proposed method also achieves better results than the variant without it consistently.

2) Results for Sampling From All Categories: Finally, we simulate the adaptive test case generation procedure with all four POV categories. The goal is to identify more failure cases given a fixed number (\(N = 1200\)) of cases. Figure 12 shows the change of sample allocation across different POV categories, and the ratio of failure cases for each category. The results are presented for both of the proposed sample allocation methods. Here, the sample allocation in batch \(i\) is determined by the failure case ratios in batch \((i-1)\) and in earlier batches.

Fig. 12. Testing experiment results with the full POV library; batch size \(n_b = 60\), run for 20 batches. The sample allocation and the failure case ratio across POV categories in all batches are demonstrated for the stochastic optimization method in (a), (b), for the UCB-dynamic method in (c),(d).

When using the stochastic optimization method, the sample sizes start evenly. Then, since more failure cases have been found within category #1, #3, #4, and none for category #2 POVs, the sample sizes increase for the other three categories while reducing gradually for category #2. This helps to focus on the more promising categories, while maintaining some exploration in the under-performing ones. Finally, we find 271 failure cases in total: 47 failure cases in category #1, 203 in category #3, and 21 in category #4.

When applying the UCB-dynamic method, category #3 is quickly identified as the most promising one, and is then exploited extensively. Finally, we find 518 failure cases in total: 516 in category #3, 2 in category #4. More failure cases are found compared to the previous method, while the diversity of samples is not as good. The two presented methods have their own strengths, which can be utilized by testers with different goals.

VII. CONCLUSION

In this paper, we develop techniques to evaluate black-box HAVs at the roundabout entering scenario. We apply two game-theoretic formulations, level-\(k\) game theory and SVO, to generate a library of interactive POVs. We formulate a realistic two-phase planning algorithm for each POV, making the POV models scalable to scenarios with multiple POVs. We also designed a two-stage adaptive test case generation framework: a sample allocation procedure that distributes samples into different scenario categories, and an adaptive sampling procedure for a single category. For sample allocation, we propose two methods based on different objectives. For adaptive sampling, multiple exploration and exploitation criteria are designed and implemented. We propose the metric FMC to measure the test sample quality. Finally, We verify the proposed method by running testing in simulation with a rule-based VUT. Our POV library is able to capture a wide variety of interactive behaviors for different roundabout
scenarios. The UCB-dynamic method can generate the most failure cases across several simplified experiments among sample allocation methods. The adaptive sampling method can discover failure cases across several simplified experiments among scenarios for collision avoidance of road vehicles,” in *IEEE Trans. Intell. Vehicility Sys.* (IV), Jun. 2018, pp. 1326–1333.

17. N. Li, D. W. Wyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, “Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems,” *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 5, pp. 1782–1797, Sep. 2018.

18. “European new car assessment programme—TEST PROTOCOL—AEB systems,” NCAP, Leuven, Belgium, Tech. Rep., 2017.

19. S. Zhang, H. Peng, D. Zhao, and H. E. Tseng, “Accelerated evaluation of autonomous vehicles in the lane change scenario based on subset simulation technique,” in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3935–3940.

20. C. E. Tuncali, T. P. Pavlic, and G. Fainekos, “Utilizing S-TauLiRo as an automated test generation framework for autonomous vehicles,” in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1470–1475.

21. C. E. Tuncali and G. Fainekos, “Randomly-exploring trees for testing automated vehicles,” in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 661–666.

22. M. Klischat and M. Althoff, “Generating critical test scenarios for automated vehicles with evolutionary algorithms,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2352–2358.

23. B. Gangopadhay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, “Identification of test cases for automated driving systems using Bayesian optimization,” in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2019, pp. 1341–1347.

24. G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta, “Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles,” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1443–1450.

25. A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, “Adaptive stress testing with reward augmentation for autonomous vehicle validation,” in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 163–168.

26. S. Zhang, H. Peng, S. Nageshrao, and H. E. Tseng, “Generating socially acceptable perturbations for efficient evaluation of autonomous vehicles,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1341–1347.

27. M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations,” *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, p. 1805, Aug. 2000.

28. A. Kesting, M. Treiber, and D. Helbing, “General lane-changing model MOBIL for car-following models,” *Transp. Res. Rec.*, vol. 1999, pp. 86–94, Jan. 2007.

29. Y. Hu, W. Zhan, L. Sun, and M. Tomizuka, “Multi-modal probabilistic prediction of interactive behavior via an interpretable model,” in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 557–563.

30. J. Li, H. Ma, and M. Tomizuka, “Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6658–6664.

31. W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, “Social behavior for autonomous vehicles,” *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 50, pp. 24972–24978, 2019.

32. J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, “Hierarchical game-theoretic planning for autonomous vehicles,” in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 9590–9596.

33. J. H. Yoo and R. Langari, “A Stackelberg game theoretic driver model for merging,” in *Proc. ASME Dyn. Syst. Control Conf. (DSCC)*, vol. 2, 2013, pp. 1–8.

34. B. M. Albaba and Y. Yildiz, “Driver modeling through deep reinforcement learning and behavioral game theory,” *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 2, pp. 885–892, Mar. 2022.

35. A. Sarkar and K. Czamecki, “A behavior driven approach for sampling rare event situations for autonomous vehicles,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 6407–6414.

36. L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, “Courteous autonomous cars,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 663–670.

37. T.-H. Ho and X. Su, “A dynamic level-k model in sequential games,” *Manage. Sci.*, vol. 59, no. 2, pp. 452–469, Feb. 2013.

38. SOS What Every Driver Must Know. Accessed: Nov. 7, 2022. [Online]. Available: https://www.michigan.gov/sos/resources/forms/what-every-driver-must-know

39. M. Werling, J. Ziegler, S. Kammel, and S. Thrun, “Optimal trajectory generation for dynamic street scenarios in a Frenét frame,” in *Proc. IEEE Intell. Conf. Robot. Autom.*, May 2010, pp. 987–993.
[40] K. Zhang, Z. Yang, and T. Ba¸sar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” in Handbook of Reinforcement Learning and Control. Springer, 2021, pp. 321–384.

[41] R. Nagel, “Unraveling in guessing games: An experimental study,” Amer. Econ. Rev., vol. 85, no. 5, pp. 1313–1326, 1995.

[42] M. A. Costa-Gomes, V. P. Crawford, and N. Iriberri, “Comparing models of strategic thinking in van Huyck, Battalio, and Beil’s coordination games,” J. Eur. Econ. Assoc., vol. 7, nos. 2–3, pp. 365–376, Apr./May 2009.

[43] C. G. McClintock and S. T. Allison, “Social value orientation and helping behavior,” J. Appl. Social Psychol., vol. 19, no. 4, pp. 353–362, Mar. 1989.

[44] L. Wang, L. Sun, M. Tomizuka, and W. Zhan, “Socially-compatible behavior design of autonomous vehicles with verification on real human data,” IEEE Robot. Autom. Lett., vol. 6, no. 2, pp. 3421–3428, Apr. 2021.

[45] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.

[46] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 2094–2100.

[47] V. Mnih et al., “Playing Atari with deep reinforcement learning,” 2013, arXiv:1312.5602.

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

[49] C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning, Cambridge, MA, USA: MIT Press, 2006.

[50] M. J. Kochenderfer and T. A. Wheeler, Algorithms for Optimization. Cambridge, MA, USA: MIT Press, 2019.

[51] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, “Taking the human out of the loop: A review of Bayesian optimization,” Proc. IEEE, vol. 104, no. 1, pp. 148–175, Jan. 2015.

[52] C. M. Bishop, Pattern Recognition and Machine Learning. Cham, Switzerland: Springer, 2006.

[53] T. Lattimore and C. Szepesvári, Bandit Algorithms. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[54] R. Agrawal, “Sample mean based index policies by O(\log n) regret for the multi-armed bandit problem,” Adv. Appl. Probab., vol. 27, no. 4, pp. 1054–1078, Dec. 1995.

[55] O. Chapelle and L. Li, “An empirical evaluation of Thompson sampling,” in Proc. Adv. Neural Inf. Process. Syst., vol. 24, 2011, pp. 2249–2257.

[56] W. Zhan et al., “INTERACTION dataset: An INTERnational, adversarial and cooperative moTION dataset in interactive driving scenarios with semantic maps,” 2019, arXiv:1910.03088.

Xinpeng Wang (Graduate Student Member, IEEE) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree in mechanical engineering with the University of Michigan, Ann Arbor. His research interests include human driving behavior modeling and the control and evaluation of highly automated vehicles.

Songan Zhang received the B.S. and M.S. degrees in automotive engineering from Tsinghua University in 2013 and 2016, respectively, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2021. She is currently a Research Scientist at Ford Motor Company. Her research interests include decision-making of the mobile robot and perception of the mobile robot. Her specialties are reinforcement-learning and meta reinforcement learning.

Huei Peng received the Ph.D. degree in mechanical engineering from the University of California at Berkeley, Berkeley, in 1992. He is currently a Professor with the Department of Mechanical Engineering, University of Michigan. His current research interests include adaptive control and optimal control, with emphasis on their applications to vehicular and transportation systems, design and control of electrified vehicles, and connected/automated vehicles. He is a SAE Fellow and an ASME Fellow.