Comprehensive Safety Evaluation of Highly Automated Vehicles at the Roundabout Scenario

Xinpeng Wang¹, Songan Zhang², Huei Peng¹

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, a highly interactive driving scenario with various traffic situations. 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 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 and stochastic optimization 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 situations 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 [1] and production [2] HAVs, 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 situations, i.e. scenarios. The safety efficacy of the vehicle under test (VUT) can be 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 primary other vehicle (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 to the future decision-making and motion-planning of each other. 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]–[11], due to the myriad traffic situations 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 behavioral 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 based on adaptive sampling and stochastic optimization 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 method is able to identify diverse failure modes of the VUT, enabling efficient safety certification for OEMs or third-party organizations. This paper is an extension from our previous work [12] with several new contributions:

- 1) We developed a two-phase game-theoretic decision-making framework for the roundabout scenario that includes is scalable to multiple POVs.
- 2) We enriched the adaptive sampling scheme with several case selection schemes, including a modified expected

¹Xinpeng Wang (xinpengw@umich.edu) and Huei Peng are with the Department of Mechanical Engineering, the University of Michigan, Ann Arbor, MI 48109, U.S.

²Songan Zhang is with Ford Motor Company, Dearborn, MI 48126, U.S.

improvement criterion, and an exploration scheme based on behavior mode boundary identification.

- 3) We proposed a stochastic optimization scheme for sample allocation between different POV categories.
- 4) We conducted testing at a two-POV roundabout scenario in simulation; we demonstrated the efficacy of the adaptive test case generation scheme to identify failure modes of the VUT with 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 given 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 [13]–[15]. According to [13], 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. [16] 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. [17], [18] 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, [19] 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, test matrix has been used to evaluate advanced driver assistance system (ADAS) [20]. 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], [21] and subset simulation [22] 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. Meanwhile, falsification-based evaluation methods attempt to

generate initial conditions or POV behaviors that force the VUT to violate the safety requirements with limited test runs. Corner cases have been generated using simulated annealing [23], rapid-exploring random tree (RRT) [24], evolutionary algorithms [25], Bayesian optimization [26], adaptive sampling [27], reinforcement learning [28], [29], etc. In [23], [24], the falsifying POVs might behave adversarially, which is not a reasonable representation of real-driving situations. In [29], reinforcement learning was used to create adversarial yet socially acceptable POV behaviors, while the diversity of the corner cases was not discussed. In [26]–[28], the diversity of identified failure modes was addressed using region elimination, reward augmentation and performance boundary heuristics respectively. In this paper, we take an approach similar to [27], while we enrich the adaptive sampling scheme with multiple exploration and exploitation criteria, enhance the scheme with an extra stochastic optimization scheme for categorical parameters, and provide a formal definition of the failure mode coverage.

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) [30], MOBIL model [31].
- 2) Learning-based models, e.g., variational auto-encoder (VAE) [32], generative adversarial network (GAN) [33].
- 3) Game-theoretic models [9], [19], [34]–[39].

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 [37] or Stackelberg [35], [36] 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 and applied level- k game theory [19], [34], quantal response [38] or cumulative prospect theory [9] to model human driving behaviors. On the other hand, [37] and [39] 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

Fig. 1. Pipeline of the interaction-aware evaluation framework.

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 formulates the POV library (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 schemes (Section V). The overall concept of the proposed interaction-aware evaluation method 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., [40], 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 the green vehicle. Therefore, the previous evaluation scenario formulation [12], 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.

The geographical layout of the example roundabout for this work is based on a 4-way roundabout inside Mcity, an HAV test facility at the University of Michigan, as shown in Figure 2(a). Each vehicle is assumed to drive along a predetermined reference path, as shown in Figure 2(b). Frenet frame [41] 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, the vehicles are modelled as double integrator as they move along their reference path, which are shown in Figure 2(c). Denoting

Fig. 2. (a) The configuration of the three-vehicle roundabout scenario for testing. (b) The reference path of the two POVs and one VUT. (c) Definition of variables.

the two vehicles as A and B, the equation of motion can be written as (1).

$$\begin{cases} x_A(t+1) = x_A(t) + v_A(t)\Delta t \\ v_A(t+1) = v_A(t) + a_A(t)\Delta t \\ x_B(t+1) = x_B(t) + v_B(t)\Delta t \\ v_B(t+1) = v_B(t) + a_B(t)\Delta t \end{cases} \quad (1)$$

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. x_A, x_B are the longitudinal position and v_A, v_B are the longitudinal speed of vehicles A and B respectively. The input for each vehicle is the longitudinal acceleration, i.e. 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 formulation

Solving the optimal policy for one rational agent can be modelled as an Markov Decision Process (MDP) problem, which is defined by $\mathcal{M} = (\mathcal{X}, \mathcal{U}, \mathcal{P}, r, \gamma)$, 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 modelled as a Markov game (MG), which is a generalization of MDP [42], 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, $\mathcal{N} = \{1, 2, \dots, N_g\}$ 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 π^{i*} that maximizes its expected cumulative reward from any initial state x :

$$\begin{aligned} \pi^{i*}(x) &= \arg \max_{\pi^i} V(x, \pi^i, \pi^{-i}) \\ &= \arg \max_{\pi^i} \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r^i(x_t, u_t) \right] \\ &\quad \left[u^i = \pi^i(x_t), u^{-i} = \pi^{-i}(x_t), x_0 = x \right] \end{aligned} \quad (2)$$

In (2), 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 π^{-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 social value orientation (for designing different rewards) to describe the diversified POVs. For simplicity, it is assumed that each POV is involved in a two-player Markov game, 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 [43] 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 π_0 is known a priori, which is a naive 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 π_k^{A*} , can be calculated from:

$$\pi_k^{A*}(x) = \arg \max_{\pi^A} V(x, \pi^A, \pi_{k-1}^{B*}) \quad (3)$$

Where π_{k-1}^{B*} denotes the policy of a level- $(k-1)$ agent B. Since π_{k-1}^{B*} is already known and fixed when computing π_k^{A*} , the two-player MG degenerates to an MDP that is easier to solve. On the other hand, since agents at different

Fig. 3. (a) The SVO ring: we will focus on $\psi \in [0, \pi/2)$. (b) The sequence to compute level- k agents. The level-2 POVs have an extra parameter ψ characterizing their SVO angle.

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 [44], 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

To systematically capture a set of diverse reward functions for the POV, we incorporate the social value orientation (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 [45]. It quantifies the preference of an agent regarding the outcome for itself versus for others, which could be represented as an angle ψ on a 2-D plane as shown in Figure 3(a). Here, different ψ 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 [37], 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 social value orientation. 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.

Fig. 4. (a) Phase-1 interaction for POV #1; the opponent is the green vehicle. (b) Phase-2 interaction for POV #1; the opponent is the red vehicle.

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], [46] 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. Therefore, we divide 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 is outside of the roundabout ($x_{POV} < 0$). Since it needs to yield to vehicles from upstream according to traffic rules, 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 ($x_{OV} = 0$).

Therefore, a POV will be in one of the two-player games at any time, but with different opponents according to its location. 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 is assumed to follow 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 phase-2 POV will regulate the distance to the preceding vehicle with a rule-based algorithm, in addition to interacting with a phase-1 vehicle.

We first assume a pair of known level-0 policies: a level-0 phase-2 POV follows a constant speed; a level-0 phase-1 POV decelerates at $1m/s^2$ if $v_{POV} > v_{max}$, then follows a constant speed. Next, we can alternately generate the level- k phase-1 and phase-2 policies for POVs from the level- $(k-1)$ policies, and thus we can compute any level- k phase-1 or phase-2 POV policies by induction. The full POV policy is acquired by combining phase-1 and phase-2 policies with the same level and SVO. The procedure of constructing the POV library is demonstrated in Figure 3(b).

Finally, 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:

- 1) POV #1 is level-0; POV #2 is level-1;
- 2) POV #1 is level-1; POV #2 is level-0;
- 3) POV #1 is level-1; POV #2 is level-2 with any ψ ;
- 4) POV #1 is level-2 with any ψ ; POV #2 is level-1.

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 ψ 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 ($\mathcal{X} = X$). 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

TABLE I
PARAMETERS FOR REWARD DESIGN

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

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:

$$r(x, u) = W^T \Phi(x, u) = \sum_{i=1}^K w_i \phi_i(x, u) \quad (4)$$

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.
- 2) $\phi_2 = \phi_{step} = -1$: constant step cost to encourage shorter time to finish.
- 3) $\phi_3 = \phi_{vmax} = -\mathbf{1}_{v_{POV}(t) > v_{max}}$: penalty on over-speeding inside the roundabout; $\mathbf{1}$ is the indicator function.
- 4) $\phi_4 = \phi_{Dist} = \mathbf{1}_{dist(t) < d_{critical}}(d_{critical} - dist(t))$: $dist$ is the distance to the OV (in Cartesian frame); penalty for being too close.
- 5) $\phi_5 = \phi_{crash} = \mathbf{1}_{dist(t) < d_{crash}}$: penalty on collision.
- 6) $\phi_6 = \phi_{TTC} = \mathbf{1}_{TTC < TTC_{min}}(TTC - TTC_{min})$: penalty for time gap to the OV (in Frenet frame) that is too small. Here we define:

$$\begin{aligned} \Delta x_1 &= x_{POV}(t_1) - x_{OV}(t_1) \\ \Delta v_1 &= v_{POV}(t_1) - v_{OV}(t_1) \\ TTC &= \begin{cases} \frac{\Delta x_1}{-\Delta v_1} & \text{when } \Delta x_1 \Delta v_1 < 0 \\ \infty & \text{otherwise} \end{cases} \end{aligned}$$

where t_1 is the terminal time of the episode. All the reward parameter values are shown in Table I.

For level-2 POV with SVO angle ψ , the feature ϕ_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 ϕ_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. ϕ_3 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 technique [47]. First, the action-value function Q is defined as:

$$Q(x, u | \pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{t_1} \gamma^t r_t | x_0 = x, u_0 = u \right] \quad (5)$$

Q-learning uses temporal difference to estimate the optimal Q function, i.e. $Q^*(x, u | \pi^*)$, and learn the optimal policy π^* . For details, please refer to [47]. In this work, the reinforcement learning algorithm we use is Double Deep-Q network (DDQN) [48]. DDQN is based on the Deep-Q network (DQN) [49] 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 ψ and level- k . For the two-POV roundabout scenario, the initial condition x_0 is defined as:

$$x_0 = [x_{POV1}^0, x_{POV2}^0, x_{VUT}^0, v_{POV1}^0, v_{POV2}^0, v_{VUT}^0]^T \quad (6)$$

By combining x_0 with the SVO of each POV (ψ_1, ψ_2), the category c of two-POV combination, we build the testing space, denoted as $\mathcal{S}$, where each case s is:

$$s = [x_0^T, \psi_1, \psi_2, c]^T \quad (7)$$

A test case generation scheme is needed to pick N test cases $s = [s_1, \dots, s_N]$ 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:

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

where τ 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; μ_1, μ_2, μ_3 are weighting factors. The failure modes of a VUT are defined as: $\mathcal{S}_f(\lambda) = \{s | s \in \mathcal{S}, P(s) < \lambda\}$, where λ is the performance score threshold for a failure. All cases belong to $\mathcal{S}_f(\lambda)$ are failure cases. The key objectives can be translated to maximizing the failure mode coverage.

B. Adaptive testing method overview

We will generate N test cases in batches, with batch size n_b . For the testing space $\mathcal{S}$, 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: $\mathcal{S} = S \times C$, where S is the subspace with continuous variables.

Fig. 5. Measuring the failure mode coverage (FMC) of test samples on an 1-D continuous testing space. The blue curve shows the performance score $P(s)$, the red dashed line represents the score threshold λ , and the regions of curve under λ are the failure modes. For this example, $\text{FMC}(s, \rho, \lambda) = l_1 + l_2 + l_3 + l_4$.

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 for each batch can be divided into two stages, as shown in the lower part of Figure 1. In the 1st stage, we allocate the number of samples into different POV categories by solving a stochastic optimization problem, i.e. assign n_c^i cases to be tested with category c at batch i . In the 2nd stage, we sample new test cases within each category from S using an adaptive sampling scheme 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 the subspace S .

$$\text{FMC}(s_c, \rho, \lambda) = \int_{\bigcup B_{\dim}(\rho, s_\lambda)} \mathbf{1} dv, \quad (9)$$

where 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 ρ . $s_\lambda = \mathcal{S}_f(\lambda) \cap s_c$, which 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-D.

Within each POV category, we conduct adaptive sampling with a Gaussian process regression (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 [50]. GPR is a non-parametric probabilistic model [51] that is popular for its ability to model complex functions and infer the value around unexplored regions [52]. 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 (10).

$$f(s) \sim GP(m(s), k(s, s')) \quad (10)$$

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

$$P(s) = \delta + f(s), \text{ where}$$

$$f(s) \sim GP(0, k(s, s'|\theta)), \delta \sim \mathcal{N}(\beta, \sigma^2) \quad (11)$$

In (11), θ is the kernel parameter, (β, σ, θ) 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 Equation (12), where $\theta = [\theta_1, \theta_2]^T$. Hyper-parameters are optimized using maximum likelihood estimation.

$$k(s, s'|\theta) = \theta_1^2 \exp\left(-\frac{(s - s')^T(s - s')}{2\theta_2^2}\right) \quad (12)$$

With a GPR model $\hat{P}(\cdot)$ based on existing testing cases s and their results y , 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:

$$\begin{aligned} \mathbb{E}(\hat{P}(s^0|s, y)) &= k(s^0, s)(k(s, s) + \sigma^2 I)^{-1}(y - \beta) \\ \text{Var}(\hat{P}(s^0|s, y)) &= k(s^0, s^0) - \\ &\quad k(s^0, s)(k(s, s) + \sigma^2 I)^{-1}k(s, s^0) \end{aligned} \quad (13)$$

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.

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 $\mathcal{C}(s)$ 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) criteria:* Expected improvement (EI) is a popular acquisition function for choosing the most promising samples in Bayesian optimization [53], a closely-related method with the GPR-based adaptive sampling. EI indicates the possible improvement on optimal value brought by the new query.

$$\begin{aligned} \mathcal{C}(s) &= \mathbb{E}(I(s)), \text{ where} \\ I(s) &= \max(P_{\min} - \hat{P}(s), 0) \end{aligned} \quad (14)$$

Algorithm 1 Adaptive sampling for one step

Input: batches number i ; batch size n_c^i ; previous GPR model $\hat{P}_c^{i-1}$; exploration factor ϵ_0 .

Output: test cases with POVs of category c , s_c^i , and test results y_c^i ; updated GPR model $\hat{P}_c^i$.

- 1: **if** $i = 1$ **then**
 - 2: Sample initial test batch s_c^1 uniformly from S .
 - 3: **else**
 - 4: Randomly sample p queries $\tilde{s}$ from S based on uniform distribution ($p \gg n_c^i$).
 - 5: $\epsilon = \epsilon_0 \alpha^{i-1}$.
 - 6: Pick $(1 - \epsilon)n_c^i$ queries from $\tilde{s}$ according to $\mathcal{C}_{exploit}(s)$ as exploitation samples, denoted as $s_{exploit}$.
 - 7: Pick $\epsilon n_c^i(1 - r_{BMB})$ queries from $\tilde{s}$ according to $\mathcal{C}_{explore}(s)$ as the exploration samples, denoted as $s_{explore}^1$.
 - 8: Sample $\epsilon n_c^i r_{BMB}$ cases according to Algorithm 2, denoted as $s_{explore}^2$.
 - 9: $s_c^i = [s_{exploit}, s_{explore}^1, s_{explore}^2]$.
 - 10: **end if**
 - 11: Execute test cases s_c^i , acquire results $y_c^i = P(s_c^i)$.
 - 12: Fit/Update the GPR model: $y = \hat{P}_c^i(s) = \hat{P}(s|s_c^{1:i}, y_c^{1:i})$.
-

Fig. 6. (a) Computing expected FMC improvement through volume approximation. The yellow region on the left shows the real V_{add} brought by the new sample, while on the right shows the approximated $\hat{V}_{add}$. (b) Sampling along the performance boundary. Points with different colors belongs to different behavior modes. a., b., c., d. correspond to line 3, 4, 5, 6 in Algorithm 2 respectively.

where P_{min} is the minimal performance score achieved by existing test samples. To better suit our problem settings and goals, we modify the EI criteria to explicitly consider the possible FMC improvement from the new query:

$$\mathcal{C}_{exploit}(s) = \mathbb{E}(I(s)) \times V_{add}, \text{ where} \quad (15)$$

$$I(s) = \max(\lambda - \hat{P}(s), 0)$$

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

$$V_{add} \approx \hat{V}_{add} \propto \rho^{dim} - (\max(\rho - d_{min}(s), 0))^{dim} \quad (16)$$

where $d_{min}(s)$ is the distance from s to the nearest identified corner case.

2) *Exploration #1: Standard deviation criteria:* For exploration, a straightforward query criteria is the standard deviation.

$$\mathcal{C}_{explore}(s) = \hat{\sigma}(s) \quad (17)$$

where $\hat{\sigma}(s) = \sqrt{\text{Var}[\hat{P}(s)]}$. The higher $\mathcal{C}_{explore}(s)$ 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 scores. 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 illustrated in [27], where the behavior 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. Therefore, in each batch, we will identify the behavior mode boundaries of the VUT and generate samples nearby.

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

To sample test cases along BMBs, we propose the algorithm 2. The procedure is visualized in Figure 6(b).

Algorithm 2 Sampling at 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^s .

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

- 1: $s_{BMB} = []$
 - 2: **for** $k \leftarrow 1$ to n_{BMB} **do**
 - 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 $\hat{s}$ 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 schemes above. The percentage of cases for exploration and exploitation are determined by the parameter ϵ , which gradually decreases at the rate of α ($\alpha \in (0.9, 1)$), such that the procedure will start with more exploration, and bias towards exploitation as more data are collected and a better meta-model is built. Between

the two exploration schemes, we allocate a fixed ratio r_{BMB} of all exploration cases to be sampled from the BMBs, while the others are chosen based on the standard deviation criteria (e.g. $r_{BMB} = 1/4$). It is important to note that the proposed method has different goals compared to the Bayesian optimization methods though they share similar formulation and usages on the GPR model and the EI criteria. 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 schemes 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. The key objective has two aspects: (1) we hope to maximize the total number of identified corner cases from all categories; (2) we prefer corner cases that are scattered in more categories. Since the potential of finding corner cases in each category is unknown a priori, we design a sample allocation scheme that balances exploration and exploitation by solving a stochastic optimization problem.

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

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

$h(\cdot)$ 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 $h(\cdot)$ will encourage more corner cases in each category. On the other hand, the reward for each extra corner case of the same category will diminish ($dh/d\bar{n}_c^i$ decreases as $\bar{n}_c^i$ increases), which encourages finding corner cases in other categories rather than concentrated in a single category, i.e., exploration is encouraged.

We can define the optimization problem below. Since $\bar{n}_c^i$ is unknown beforehand, we will estimate it according to results of previous iterations, which can be written as $\bar{n}_c^i = \eta_c^i n_b q_c^i$. Here, η_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 corner-case probability 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 corner case in each trial" being q_c^i . The goal is to maximize the overall expected corner case reward R_{cc} , as stated in (19).

$$\begin{aligned} \max_{\eta_c^i} \quad & \mathbb{E}_{q_c^i}(R_{cc}) = \sum_{c=1}^{|C|} \mathbb{E}_{q_c^i}(\beta_c^i) \\ \text{s.t.} \quad & \sum_{c=1}^{|C|} \eta_c^i = 1 \\ & \eta_c^i \geq 0, \quad c = 1 \dots |C| \end{aligned} \quad (19)$$

$|C|$ is the cardinality of the set of categories C . q_c^i can be modelled by a Beta distribution: $q_c^i \sim \text{Beta}(\alpha_c^i, \beta_c^i)$, where α_c^i, β_c^i are the parameters that control the shape of the Beta distribution [54]. Without loss of generality, we assume $q_c^1 \sim \text{Beta}(1, 1)$, i.e. q_c^1 has a uniform prior. Assuming that for batch i and category c , a_c^i corner cases and b_c^i non-corner cases were found, and we denote the observation D_c^i as $D_c^i = [a_c^i, b_c^i]$. Then, we can conduct a Bayesian update on q_c^i according to the existing testing results:

$$\begin{aligned} Pr(q_c^i | D_c^i) &\propto Pr(D_c^i | q_c^i) Pr(q_c^i) \\ Pr(q_c^{i+1}) &= Pr(q_c^i | D_c^i), \end{aligned} \quad (20)$$

where the 1st line is the Bayes theorem; the 2nd line states that the prior of q_c^{i+1} is defined as the posterior of q_c^i . Since Beta distribution is the conjugate prior of the binomial distribution [54], the posterior distribution of q_c^i will have the same form, i.e. a Beta distribution. Therefore, we can compute the distribution of q_c^{i+1} with the following updating rules on its parameters:

$$\begin{aligned} q_c^{i+1} &\sim \text{Beta}(\alpha_c^{i+1}, \beta_c^{i+1}), \text{ where} \\ \alpha_c^{i+1} &= a_c^i + \hat{\gamma}(\alpha_c^i - 1) \\ \beta_c^{i+1} &= b_c^i + \hat{\gamma}(\beta_c^i - 1) \end{aligned} \quad (21)$$

When the constant $\hat{\gamma} = 1$, (21) is equivalent to (20). However, as more information is gathered with more samples in each category, q_c^i will 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 will solve the optimization problem (19) to acquire an optimal allocation of samples in each category, i.e. η_c^i . This strategy will invest more samples in better-performing categories, but will maintain some samples in all categories.

VI. SIMULATION RESULTS

In this section, We conduct interactive-aware testing for a rule-based VUT in MATLAB simulations to validate the performance of the proposed method.

A. Rule-based VUT algorithm

For the roundabout scenario, we design a rule-based speed planning algorithm for the VUT. Its decision-making has 3 stages:

- 1) The VUT starts by coasting at a constant speed. Go to stage 2 when it is x^{rb1} close to point M in Figure 2(c).
- 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 between (two POVs), or pass in front (of both). If "yield" or "slip in", then decelerate (set $v_{tar} = 0$); Else, $v_{tar} = v_0$, v_0 is a default desired speed. Go to stage 3 when the distance from VUT to point M is less than x^{rb2} ($x^{rb2} < x^{rb1}$).
- 3) Execute the last decision from stage 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 stage 2 with $v_{tar} = v_0$.

B. Various interactive test cases

In this section, we present exemplar test cases in simulations. In Figure 7,8,9,10 we present four test cases with different POV types as well as initial conditions.

In the 1st case, as shown in Figure 7, the level-1 POV #2 accelerates to pass POV #1, then passes the VUT. The POV #1 first decelerates to yield the POV #2, then keeps a steady speed to drive ahead of the VUT. The VUT slows down to yield to both vehicles, and enters the roundabout behind POV #1. The final order is: POV #2, POV #1, VUT.

In the 2nd case (Figure 8), all the initial conditions and attributes are the same as the 1st case, except that the SVO of the level-2 POV #1 increases from 0 to 0.2π , i.e. POV #1 is more cooperative. Major difference starts when the VUT interacts with POV #1 after $t = 2s$: the VUT switches from "yield to both" to "slip in", while POV #1 still chooses to drive ahead of the VUT. Then a collision happens between POV #1 and VUT. A slight change in the test parameters incurs completely different interactive behaviors and reveals a failure mode of the VUT.

In the 3rd case (Figure 9), all are unchanged except the SVO of POV #1 increases to 0.35π . Although the VUT still attempts to slip in ahead of the POV, the POV chooses to yield and there is no collision this time. The final order is: POV #2, VUT, POV #1.

In the last case, (Figure 10), the initial speed and types of the two POVs are different while the initial positions of all vehicles remain unchanged. The POV #1 first accelerates to pass the POV #2, then keeps constant speed to pass the VUT; the VUT yields to both POVs. The final order is: POV #1, POV #2, VUT. Though starting from the same positions, the four cases demonstrate very different interaction patterns and thus challenge the VUT in different ways. The richness of the proposed testing space is demonstrated.

C. Testing results comparison

A test case of the two-POV roundabout scenario is defined by (6) and (7). We fix the initial position for the VUT, $x_{VUT}^0 = -60m$ and assume the speed of the VUT v_{VUT}^0 is given to us. The POV category c is also fixed in each experiment. Only a subset of the testing space will be the sampling space for each experiment, where each sample is denoted as s^* . For the following experiments, the threshold for a failure case is set to $\lambda = -500$, which indicates that a collision had happened.

1) *Results within a single POV category:* We first show the results of testing experiments with a fixed POV category. Each VUT is put through $N = 400$ test cases. We compare the proposed GPR-based adaptive sampling scheme to other test case generation schemes, including uniform sampling, simulated annealing [55], and subset simulation [22]. The

Fig. 7. Testing experiment results with initial condition: Level-2 POV #1 with $\Psi = 0$, Level-1 POV #2, $x_{POV1}^0 = -30.5m$, $x_{POV2}^0 = -4.5m$, $v_{POV1}^0 = 8.0m/s$, $v_{POV2}^0 = 10.0m/s$, $v_{VUT}^0 = 8.0m/s$; blue for POV #1, green for POV #2, red for VUT.

Fig. 8. Testing experiment results with initial condition: Level-2 POV #1 with $\Psi = 0.2\pi$, Level-1 POV #2, $x_{POV1}^0 = -30.5m$, $x_{POV2}^0 = -4.5m$, $v_{POV1}^0 = 8.0m/s$, $v_{POV2}^0 = 10.0m/s$, $v_{VUT}^0 = 8.0m/s$.

FMC criterion is computed for all methods to compare their capability of discovering failure modes.

Figure 11 shows an exemplary testing result comparison when the sampling space is set to 2-D. Only the initial position of both POVs are sampled, i.e. $s^* = [x_{POV1}^0, x_{POV2}^0]$. All methods are compared against the ground truth, which is generated with 40000 samples using uniform sampling. The ground truth shows several disjoint regions with failure cases. With 400 cases, uniform sampling can locate one failure region with very few failure cases; simulated annealing can identify many failure cases, but all concentrated around two failure modes, making the resulting FMC value low; subset simulation performs better than simulated annealing, but it concentrates

Fig. 9. Testing experiment results with initial condition: Level-2 POV #1 with $\Psi = 0.35\pi$, Level-1 POV #2, $x_{POV1}^0 = -30.5m$, $x_{POV2}^0 = -4.5m$, $v_{POV1}^0 = 8.0m/s$, $v_{POV2}^0 = 10.0m/s$, $v_{VUT}^0 = 8.0m/s$.

Fig. 10. Testing experiment results with initial condition: Level-1 POV #1, Level-0 POV #2, $x_{POV1}^0 = -30.5m$, $x_{POV2}^0 = -4.5m$, $v_{POV1}^0 = 9.0m/s$, $v_{POV2}^0 = 7.0m/s$, $v_{VUT}^0 = 8.0m/s$.

around similar failure modes. The proposed method stands out by identifying failure cases at more failure modes and achieves the highest FMC value. With only 1% of the sample sizes compared with the ground truth, the proposed method discovers most of the failure modes qualitatively. In Figure 12, we superimpose the failure cases found in the ground truth with the behavior modes of all cases. It can be seen that failure cases mostly lie at the behavior mode boundaries, which matches our intuition to choose samples on BMB for exploration.

Then, we present the comparison of quantitative results when the sampling space has a higher dimension in table II. Two POV categories are evaluated:

TABLE II
RESULTS COMPARISON FOR ADAPTIVE TEST CASE GENERATION

Methods	FMC(s^* , 0.05, -500)	
	L-0 POV1 L-1 POV2	L-2 POV1 L-1 POV2
Uniform sampling	0.92×10^{-4}	$1.97 \times 10^{-5} *$
Simulated annealing	0.79×10^{-4}	0.68×10^{-5}
Subset simulation	1.54×10^{-4}	2.21×10^{-5}
Ours w/o performance boundary	2.58×10^{-4}	3.47×10^{-5}
Ours	2.49×10^{-4}	3.71×10^{-5}

- Category #1: level-0 POV1 and level-1 POV2, with $s^* = [x_{POV1}^0, x_{POV2}^0, v_{POV1}^0, v_{POV2}^0]$ (4-D);
- Category #4: level-2 POV1 and level-1 POV2, with $s^* = [x_{POV1}^0, x_{POV2}^0, v_{POV1}^0, v_{POV2}^0, \psi_1]$ (5-D).

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 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). Even with the advantage on case number, the uniform sampling method only achieves slightly better results than simulated annealing. Subset simulation performs better than the above two methods, but fall short compared to the two variant of the GPR-based adaptive sampling methods. By adding the BMB identification scheme, Our proposed method achieves marginal better results than the variant without for category #4, but is slightly inferior for category #1. It is probable that the BMB identification scheme is more helpful when the dimension of the sampling space is higher.

2) *Results with multiple POV levels:* 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 13 shows the change of sample allocation across different POV categories, and the ratio of failure cases for each category. 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 in later batches while reducing gradually for category #2. This helps to focus on the more promising categories, while maintains some exploration in the under-performing ones.

VII. CONCLUSIONS

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 social value orientation, 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 scheme: a sample allocation procedure based on stochastic optimization, and an adaptive sampling procedure based on Gaussian process regression. Multiple

Fig. 11. Testing experiment results with level-1 POV #1 and level-0 POV #2. $v_{POV1}^0 = v_{POV2}^0 = 9m/s$, $v_{VUT}^0 = 10m/s$. The samples are color coded by the performance score (truncated at 0 and -600), where darker means more dangerous. For (b), batch size $n = 20$, run for 20 batches.

Fig. 12. Behavior modes versus failure modes. Most failure cases lie near the behavior mode boundaries.

Fig. 13. Testing experiment results with the full POV library; batch size $n_b = 60$, run for 20 batches. (a) Sample allocation across POV categories in all batches. (b) The ratio of failure cases found in each category.

exploration and exploitation schemes are designed and implemented. We propose a metric, failure mode coverage (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. The sampling method can customize test cases to discover the failure modes of the VUT efficiently. It outperforms other sampling methods according to the FMC metric. For future work, we plan to extend the current framework to other interactive scenarios including crowded parking lots, unsignalized intersection, etc.

ACKNOWLEDGEMENT

We thank Shaobing Xu, Geunseob Oh, Yuanxin Zhong for their insightful suggestions and help.

REFERENCES

- [1] “Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk.” [Online]. Available: <https://www.nbcnews.com/tech/tech-news/self-driving-uber-car-hit-killed-woman-did-not-recognize-n1079281>
- [2] “Two Killed in Driverless Tesla Car Crash, Officials Say - The New York Times.” [Online]. Available: <https://www.nytimes.com/2021/04/18/business/tesla-fatal-crash-texas.html>
- [3] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and 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, 3 2017.
- [4] X. Wang, H. Peng, and D. Zhao, “Combining reachability analysis and importance sampling for accelerated evaluation of highly automated vehicles at pedestrian crossing,” in *ASME 2019 Dynamic Systems and Control Conference, DSCC 2019*, vol. 3. American Society of Mechanical Engineers, 10 2019.
- [5] X. Wang, Y. Dong, S. Xu, H. Peng, F. Wang, and Z. Liu, “Behavioral Competence Tests for Highly Automated Vehicles,” in *Accepted by IEEE Intelligent Vehicles Symposium*, 2020.
- [6] “Automated Vehicles for Safety | NHTSA.” [Online]. Available: <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>
- [7] “Roundabouts.” [Online]. Available: <https://www.iihs.org/topics/roundabouts>
- [8] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, “Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts,” in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 321–326.
- [9] L. Sun, W. Zhan, Y. Hu, and M. Tomizuka, “Interpretable Modelling of Driving Behaviors in Interactive Driving Scenarios based on Cumulative Prospect Theory,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 10 2019, pp. 4329–4335.
- [10] S. Masi, P. Xu, and P. Bonnifait, “A curvilinear decision method for two-lane roundabout crossing and its validation under realistic traffic flow,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1290–1296.
- [11] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, “The round dataset: A drone dataset of road user trajectories at roundabouts in germany,” submitted.
- [12] X. Wang, S. Zhang, K.-H. Lee, and H. Peng, “An interaction-aware evaluation method for highly automated vehicles,” *arXiv preprint arXiv:2102.11462*, 2021.
- [13] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, “Survey on scenario-based safety assessment of automated vehicles,” *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020.
- [14] “Pegasus Method - pegasus-EN.” [Online]. Available: <https://www.pegasusprojekt.de/en/pegasus-method>
- [15] “SAKURA Project.” [Online]. Available: <https://www.sakura-prj.go.jp/>
- [16] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, “Testing scenario library generation for connected and automated vehicles, part i: Methodology,” *IEEE Transactions on Intelligent Transportation Systems*, 2020.

- [17] G. Chou, Y. E. Sahin, L. Yang, K. J. Rutledge, P. Nilsson, and N. Ozay, "Using control synthesis to generate corner cases: A case study on autonomous driving," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2906–2917, 2018.
- [18] M. Althoff and S. Lutz, "Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, vol. 2018-June. IEEE, 6 2018, pp. 1326–1333.
- [19] N. Li, D. W. Oyler, 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 Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1782–1797, 2018.
- [20] NCAP, "European New Car Assessment Programme - TEST PROTOCOL – AEB systems," Tech. Rep., 2015.
- [21] S. Feng, Y. Feng, H. Sun, Y. Zhang, and H. X. Liu, "Testing Scenario Library Generation for Connected and Automated Vehicles: An Adaptive Framework," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 3 2020.
- [22] 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 *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 11 2018, pp. 3935–3940.
- [23] C. E. Tuncali, T. P. Pavlic, and G. Fainekos, "Utilizing S-TaLiRo as an automatic test generation framework for autonomous vehicles," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, no. ii, pp. 1470–1475, 2016.
- [24] C. E. Tuncali and G. Fainekos, "Rapidly-exploring random trees-based test generation for autonomous vehicles," *arXiv preprint arXiv:1903.10629*, 2019.
- [25] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 2352–2358.
- [26] B. Gangopadhyay, S. Khastgir, S. Dey, P. Dasgupta, G. Montana, and P. Jennings, "Identification of test cases for automated driving systems using bayesian optimization," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1961–1967.
- [27] G. E. Mullins, P. G. Stankiewicz, and S. K. Gupta, "Automated generation of diverse and challenging scenarios for test and evaluation of autonomous vehicles," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1443–1450, 2017.
- [28] A. Corso, P. Du, K. Driggs-Campbell, and M. J. Kochenderfer, "Adaptive stress testing with reward augmentation for autonomous vehicle validation," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 163–168.
- [29] S. Zhang, H. Peng, S. Nagesh Rao, and H. E. Tseng, "Generating socially acceptable perturbations for efficient evaluation of autonomous vehicles," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020-June, pp. 1341–1347, 2020.
- [30] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.
- [31] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.
- [32] Y. Hu, W. Zhan, L. Sun, and M. Tomizuka, "Multi-modal probabilistic prediction of interactive behavior via an interpretable model," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 557–563.
- [33] J. Li, H. Ma, and M. Tomizuka, "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6658–6664.
- [34] B. M. Albaba and Y. Yildiz, "Modeling cyber-physical human systems via an interplay between reinforcement learning and game theory," *Annual Reviews in Control*, vol. 48, pp. 1–21, 2019.
- [35] J. F. Fisac, E. Bronstein, E. Stefansson, D. Sadigh, S. S. Sastry, and A. D. Dragan, "Hierarchical game-theoretic planning for autonomous vehicles," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, 2019, pp. 9590–9596.
- [36] J. H. Yoo and R. Langari, "A stackelberg game theoretic driver model for merging," *ASME 2013 Dynamic Systems and Control Conference, DSCC 2013*, vol. 2, pp. 1–8, 2013.
- [37] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 116, no. 50, pp. 2492–2497, 2019.
- [38] A. Sarkar and K. Czamecki, "A behavior driven approach for sampling rare event situations for autonomous vehicles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11 2019, pp. 6407–6414.
- [39] L. Sun, W. Zhan, M. Tomizuka, and A. D. Dragan, "Courteous Autonomous Cars," *IEEE International Conference on Intelligent Robots and Systems*, pp. 663–670, 2018.
- [40] "SOS - What Every Driver Must Know." [Online]. Available: <https://www.michigan.gov/sos/0,4670,7-127-1642-103522-,00.html>
- [41] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 987–993.
- [42] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *arXiv preprint arXiv:1911.10635*, 2019.
- [43] R. Nagel, "Unraveling in guessing games: An experimental study," *The American Economic Review*, vol. 85, no. 5, pp. 1313–1326, 1995.
- [44] M. A. Costa-Gomes, V. P. Crawford, and N. Iriberri, "Comparing Models of Strategic Thinking in Van Huyck, Battalio, and Beil's Coordination Games," *Journal of the European Economic Association*, vol. 7, no. 2-3, pp. 365–376, 4 2009.
- [45] C. G. McClintock and S. T. Allison, "Social Value Orientation and Helping Behavior," *Journal of Applied Social Psychology*, vol. 19, no. 4, pp. 353–362, 3 1989.
- [46] L. Wang, L. Sun, M. Tomizuka, and W. Zhan, "Socially-compatible behavior design of autonomous vehicles with verification on real human data," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3421–3428, 2021.
- [47] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [48] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 2094–2100.
- [49] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
- [50] 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," *Journal of Systems and Software*, vol. 137, pp. 197–215, 2018.
- [51] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: MIT Press, 2006.
- [52] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for optimization*. Mit Press, 2019.
- [53] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.
- [54] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [55] L. C. W. Dixon and G. P. Szegő, *Towards global optimisation*. North-Holland Amsterdam, 1978, vol. 2.

Xinpeng Wang received his B.S. degree in automation from Tsinghua University, Beijing, China in 2017. He is currently pursuing the Ph.D. degree in Mechanical Engineering at 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 B.S. degree and M.S. degree in automotive engineering from Tsinghua University in 2013 and 2016 respectively. She received her Ph.D. in mechanical engineering at University of Michigan, Ann Arbor in 2021.

She is currently a research scientist in Ford Motor Company. Her research interests include decision-making of the mobile robot, perception of the mobile robot. Her specialties are reinforcement learning and meta reinforcement learning.

Huei Peng received his Ph.D. in Mechanical Engineering from the University of California, Berkeley in 1992.

He is now a Professor at the Department of Mechanical Engineering at the University of Michigan, and the Director of Mcity. His research interests include adaptive control and optimal control, with emphasis on their applications to vehicular and transportation systems. His current research focuses include design and control of electrified vehicles, and connected/automated vehicles. He is both an

SAE fellow and an ASME Fellow.