

Self-Improving Safety Performance of Reinforcement Learning Based Driving with Black-Box Verification Algorithms

Resul Dagdanov*, Halil Durmus, and Nazim Kemal Ure

Abstract—In this work, we propose a self-improving artificial intelligence system to enhance the safety performance of reinforcement learning (RL)-based autonomous driving (AD) agents using black-box verification methods. RL algorithms have become popular in AD applications in recent years. However, the performance of existing RL algorithms heavily depends on the diversity of training scenarios. A lack of safety-critical scenarios during the training phase could result in poor generalization performance in real-world driving applications. We propose a novel framework in which the weaknesses of the training set are explored through black-box verification methods. After discovering AD failure scenarios, the RL agent's training is re-initiated via transfer learning to improve the performance of previously unsafe scenarios. Simulation results demonstrate that our approach efficiently discovers safety failures of action decisions in RL-based adaptive cruise control (ACC) applications and significantly reduces the number of vehicle collisions through iterative applications of our method. The source code is publicly available at <https://github.com/data-and-decision-lab/self-improving-RL>.

Index Terms—Deep Reinforcement Learning, Autonomous Driving, Black-Box Verification

I. INTRODUCTION

Fully reliable and comfortable autonomy applications increasingly prioritize safety-critical implementations in autonomous systems. The advancement of autonomous driving (AD) technologies is largely due to the widespread use of the black-box systems such as deep neural networks in numerous control [1] and planning [2] problems. To ensure both environmental and human safety, black-box systems must undergo comprehensive testing and validation in a variety of edge scenarios before they can be deployed in real-world use cases.

In a recent study [3], researchers reduced unsafe scenarios of a black-box system by guiding exploration samples along predefined trajectory classes. However, without verification through rare-event simulation [4] and generalized importance sampling on a continuous action and observation space, the safety of the AD system could not be guaranteed. In another safety-critical system, researchers proposed a safety-constrained collision avoidance approach [5] with prediction-based reachability analysis. Nevertheless, these approaches

do not verify the designed models on the continuous feature domain of scenarios. As the operating environment of the black-box system becomes more complicated, testing the control policy in all potential circumstances becomes impractical. Hence, innovative techniques are needed to verify the safety performance of the black-box system. In the context of AD, stress testing methods for rare-event probability verifications are used, such as adaptive multilevel splitting (AMS) [6], reinforcement learning (RL) [7], Bayesian optimization [8], cross-entropy [9], unsupervised clustering [10], neural bridge sampling [11], and genetic algorithm [12]. Although these applications identify potential black-box system failures, they do not make use of the discovered occurrences to improve or repair the safety-critical system. In this work, our main focus is on leveraging rare-event sampling approaches to gain better insights into failure instances, which could be made feasible with a lesser number of iterations.

Consider a typical AD scenario where the EGO vehicle comfortably and safely pursues the lead vehicle or most important object (MIO). In ideal adaptive cruise control (ACC) settings, an EGO vehicle is expected to follow the MIO vehicle with a comfortable braking distance, the least amount of jerk, and pleasant acceleration. Without any prior knowledge of the mathematical model of the system, ACC is effectively approached using model-free deep RL through trial and error on the safety and comfort parameters in interaction with the simulation environment [13]. Similarly, another study [14] used deep RL to safely enable lane-changing maneuvers and demonstrated the advantages of model-free algorithms compared to rule-based baselines [15]. However, the safety of these trained RL agents is not fully verified over the continuous range of rare events, where time-to-collision (TTC) measurements are low and the probability of collision is high. In another conducted study [16], the effects of TTC on the high likelihood of collisions were analyzed using driver participants in simulation environments.

In this work, we investigate whether the rare-event assessment and validation samples of a safety-critical system can be utilized for automatic self-improvement (healing) of the RL agents. Our specific contribution is the development of a framework (see Fig. 1) where the RL-based AD system is continuously subjected to probabilistic Black-box verification methods to discover failure scenarios. The proposed framework incrementally adjusts the training scenarios of the RL agent to compensate for the failures in these scenarios. We select ACC as a case study for the application of our method and show that the proposed system significantly reduces the number of safety violations.

* The corresponding author.

R. Dagdanov is with ITU Artificial Intelligence and Data Science Research Center and the Department of Aeronautical Engineering, Istanbul Technical University, Turkey dagdanov21@itu.edu.tr

H. Durmus is with Eaton Technologies and the Department of Electronics and Communication Engineering, Istanbul Technical University, Turkey durmush@itu.edu.tr

N.K. Ure is with ITU Artificial Intelligence and Data Science Application and Research Center and Department of Computer Engineering, Istanbul Technical University, Turkey ure@itu.edu.tr

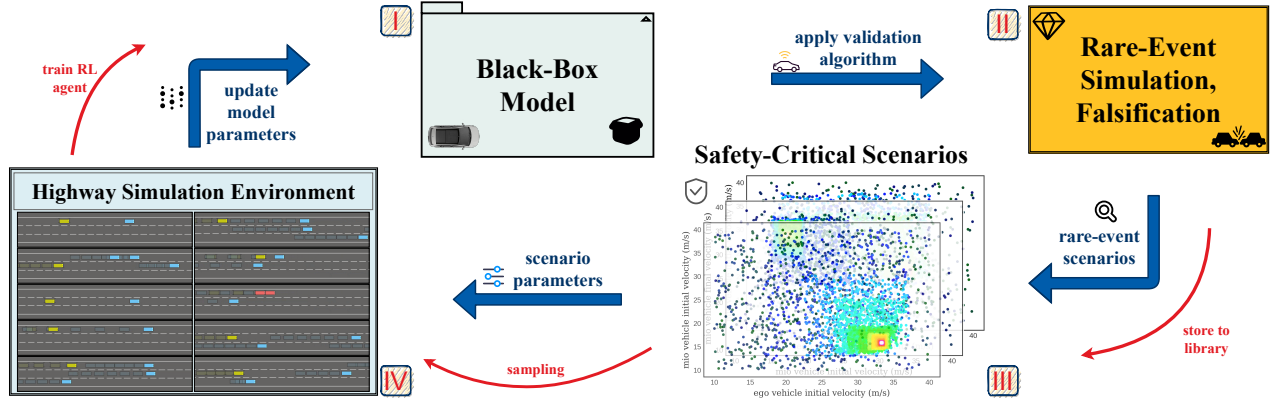


Fig. 1. **Proposed Method Overview.** **Stage-I:** A black-box model is trained in a simulation environment for predefined time steps. **Stage-II:** The trained RL agent is validated using a rare-event simulation algorithm. **Stage-III:** Estimated scenario parameters are stored in the library of safety-critical scenarios for future generations. **Stage-IV:** Collected rare-event scenarios are sampled with the proposed method to update model parameters for safer RL agent.

II. BACKGROUND

This section contains background information on the verification methodologies, deep RL algorithms, and model-based baseline approach utilized in this study.

A. Rare-Event Simulation

The verification and testing of a black-box system in adversarial, safety-critical scenarios are conducted in a risk-based simulation framework as shown in Fig. 1. Let $\mathbf{x} \sim \mathbb{P}_0$ be scenario parameters where \mathbb{P}_0 is a base distribution of the simulations. We define the safety function as $F : \mathcal{X} \rightarrow \mathbb{R}$ with safety metric τ and evaluate the probability of rare events.

$$p_\tau := \mathbb{P}_0(F(\mathbf{x}) \leq \tau) \quad (1)$$

1) *Grid-Search Approach:* Grid-Search (GS) is a simple parameter search approach across a bounded search space. In GS, the continuous search space is periodically partitioned into discrete grids. The interval value and dimension of the search space heavily influence the performance. However, the GS method suffers from the curse of dimensionality.

2) *Monte-Carlo Sampling:* The naïve Monte-Carlo (MC) estimator probability $\mathbb{P}(\phi)$ can be defined as in Eq. (2), where ϕ is a sampling scenario event, N is the total number of samples from the normal distribution \mathcal{N} .

$$p(\phi) \approx \frac{1}{N} \sum_{i=1}^N \mathbb{P}(\phi|\mathbf{x}_i), \quad \mathbf{x}_i \stackrel{\text{iid}}{\sim} \mathbb{P}(\mathbf{x}) \quad (2)$$

As shown in Eq. (3), the conditional naïve MC method can be applied to calculate the rare-event simulation probability \hat{p}_τ , where τ is a constant TTC threshold that reflects an ACC safety/comfort metric [4]. The continuous objective function $F : \mathcal{X} \rightarrow \mathbb{R}$ reflects the safety of the scenarios, where low values of $F(\mathbf{x})$ correlate to a higher likelihood of a collision.

$$\hat{p}_\tau \approx \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{F(\mathbf{x}_i) \leq \tau\}, \quad \mathbf{x}_i \in \mathcal{X} \quad (3)$$

By using a naïve MC method, all rare events can be found when $N \rightarrow \infty$. However, as the number N and the search space dimension increase, both the GS and MC simulation techniques become inadequate despite their simplicity.

3) *Bayesian Optimization:* Adversarial AD scenarios are effectively found via Bayesian Optimization (BO) [8]. An acquisition function $\mathbb{A}(\phi)$ in Eq. (4) is maximized at each iteration while taking the mean μ and variance σ^2 of the most recent BO predictions into account using Gaussian Process (GP) [17]. At each simulation rollout, scenario prediction is sampled as $y = F(\phi) + \mathcal{N}$ and the optimality is discovered by minimizing an objective function as $\phi^* = \underset{\phi}{\operatorname{argmin}}[F(\phi)]$.

$$\mathbb{A}(\phi) = \mathbb{E}_{\mathbb{P}(y|\phi)}[I(\phi)], \quad I(\phi) = \max(0, F^* - y) \quad (4)$$

where $I(\phi)$ is an improvement function and F^* is a minimum value of an objective function given ϕ scenario sample [18].

4) *Cross-Entropy Search:* A Cross-Entropy (CE) method is one of the widely used parametric adaptive importance-sampling (AIS) strategies [19].

$$\hat{p}_\tau = \frac{1}{N} \sum_{i=1}^N \frac{p_0(\mathbf{x}_i)}{p^*(\mathbf{x}_i)} \mathbb{1}\{F(\mathbf{x}_i) \leq \tau\}, \quad \mathbf{x}_i \in \mathcal{X} \quad (5)$$

where $p_0 = p_\tau \cdot p^*$ is a probability density function (pdf) of a base distribution \mathbb{P}_0 under random variable \mathcal{X} . Despite being an exact solution, Eq. (5) is impossible to compute since we are not sure of the definite result of likelihood ratio p_τ . By repeatedly minimizing Kullback-Leibler projection of p^* onto the θ parameterized probability distribution p_θ as $\theta^* \in \underset{\theta}{\operatorname{argmin}}[D_{KL}(p^*||p_\theta)]$, proximal value of the probability in Eq. (5) is computed [9].

5) *Adaptive Multilevel Splitting:* The sample inefficiency of naïve MC is addressed by the non-parametric Adaptive Multilevel Splitting (AMS) algorithm [20], which computes the rare-event probability p_τ by dividing the safety threshold into J intermediate levels, with $\infty =: \tau_0 > \tau_1 > \dots > \tau_{J-1} > \tau_J := \tau$. The probability of each level is calculated as shown in Eq. (6).

$$\mathbb{P}_0(F(\mathbf{x}) < \tau) = \prod_{j=1}^J \mathbb{P}(F(\mathbf{x}) < \tau_j | F(\mathbf{x}) < \tau_{j-1}) \quad (6)$$

As proposed in [6], each intermediate level is determined by specifying a constant fraction $\delta \in (0, 1)$ that is dropped at each intermediate iteration phase j .

B. Deep Reinforcement Learning

The main objective of RL under parameter θ is to maximize the future cumulative reward over trajectories $\omega = \{s_0, a_0, r_0, \dots, s_T, a_T, r_T\}$ with policy $\pi_\theta(a_t|s_t)$, where action $a_t \in A$, state $s_t \in S$, and \mathbf{R}_t is the reward at time t . The objective function is given by $J(\theta)$.

$$J(\theta) = \mathbb{E}_{\omega \sim \pi} \left[\sum_{t=0}^T \gamma^t \cdot \mathbf{R}_t \right] \quad (7)$$

An advantage estimator function \hat{A}_t is defined as in Eq. (8).

$$\hat{A}_t^{\pi_k} \approx \sum_{i=0}^{T-1} [\gamma^i \cdot \mathbf{R}_{t+i}] + \gamma^n \cdot V(s_{t+T}) - V(s_t) \quad (8)$$

where $V(s_t) = \mathbb{E}[G_t|s_t]$ is a value function, G_t is a discounted sum of rewards and $\gamma \in [0, 1)$ is a constant discount factor that controls the weight given to future rewards.

In this work, we evaluated our system by applying the continuous action policy-gradient-based algorithm, on-policy Proximal Policy Optimization (PPO) [21]. In an on-policy RL, the policy for choosing actions and the policy for updating the target neural network model are the same, while they are different in an off-policy RL. For our case study, we chose to use PPO exclusively as an on-policy RL algorithm.

1) *Proximal Policy Optimization*: The original version of the clipped objective function of PPO is given in Eq. (9).

$$J_{PPO}(\theta_k) = \mathbb{E}_{\omega \sim \pi_k} \left[\sum_{t=0}^T \left[\min(r_t(\theta) \cdot \hat{A}_t^{\pi_k}(s_t, a_t), \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \cdot \hat{A}_t^{\pi_k}(s_t, a_t)) \right] \right] \quad (9)$$

where $r_t(\theta) = \frac{\pi_{\theta_k}(a_t|s_t)}{\pi_{\theta_{k-1}}(a_t|s_t)}$ represents the ratio between the current policy and the previous policy. The effect of deviation from the previous policy π_{k-1} is controlled by $\epsilon = 0.2$ value.

C. Intelligent-Driver Model

One of the model-based methods used in ACC scenarios is the Intelligent Driver Model (IDM) [15]. The IDM generates an instantaneous longitudinal acceleration α of the vehicle using a continuous model, as specified in Eq. (11).

$$d^*(v, \Delta v) = d_0 + v \cdot T_{safe} + \frac{v \cdot \Delta v}{2 \cdot \sqrt{b} \cdot \alpha_{max}} \quad (10)$$

$$\alpha = \frac{dv}{dt} = \alpha_{max} \cdot \left[1 - \left(\frac{v}{v_d} \right)^\delta - \left(\frac{d^*(v, \Delta v)}{d} \right)^2 \right] \quad (11)$$

where $\alpha_{max} = 4m/s^2$ is the maximum acceleration, $\delta = 4$ is the acceleration exponent, $d_0 = 3.0m$ is the minimum distance gap, $b = -2.0m/s^2$ is the desired deceleration, $T_{safe} = 1.5s$ is the safe headway time, d is the distance to the front vehicle if it exists, and v and v_d are the current and desired velocities in m/s . The maximum deceleration of the IDM is manually limited to $a_{min} = -4m/s^2$. Other constant parameters of the model are set to be utilized as in the default simulation environment configurations [22].

III. METHODOLOGY

In this section, the introduction of the problem setting, RL model structure, evaluation metrics, and continuous self-improvement methodology, shown in Fig. 1, is carried out.

A. Problem Setting

We constructed the problem of finding safety-critical scenarios in an ACC driving application using rare-event simulation techniques and proposed an effective and sample-efficient continuous self-improving framework. Initially, we visualized certain features in an ACC scenario, such as the time-gap (\mathcal{T}^{gap}) in Eq. (12) and time-to-collision (τ^{ttc}) in Eq. (13) between the lead (MIO) and following (EGO) vehicles, which are illustrated in Fig. 2.

$$\mathcal{T}_t^{gap} = \frac{d_t^{mio}}{v_t^{ego}} \quad \forall t \in (0, 1, \dots, T) \quad (12)$$

As acknowledged in previous studies [6], [9], we chose TTC, given in Eq. (13), as an objective metric to minimize in rare-event simulations. We formulated risk-based evaluation to minimize the objective function $F(\mathbf{x}_i)$, as low values of τ_i^{ttc} are rare and dangerous. $F(\mathbf{x}_i)$ is evaluated by simulating the black-box model, which is, in our case, an RL agent.

$$\tau_t^{ttc} = \frac{d_t^{mio}}{v_t^{ego} - v_t^{mio}} \quad v_t^{ego} > v_t^{mio}, \quad t \in (0, 1, \dots, T) \quad (13)$$

The objective TTC value τ_i^{ttc} for scenario event \mathbf{x}_i is determined by simulating a black-box model until the episode is done (terminated), as defined in Eq. (14).

$$F(\mathbf{x}_i) = \tau_i^{ttc} = \arg \min_t (\tau_t^{ttc}), \quad \forall t \in (0, 1, \dots, T) \quad (14)$$

We selected $\mathbf{x}_i = \{(v_i^{ego}, d_i^{mio}, v_i^{mio}, v_{iT}^{mio}) \in \mathbb{R}^4 : v_i^{ego}, d_i^{mio}, v_i^{mio}, v_{iT}^{mio} \in \mathbb{R}\}$ as a validation parameter vector to represent simulation parameters, where v_i^{ego} is the initial velocity of an EGO vehicle, v_i^{mio} is the initial velocity of the MIO vehicle, d_i^{mio} is the initial distance to the MIO vehicle, and v_{iT}^{mio} is the desired/target velocity of the lead vehicle in a particular scenario sample i . The validation parameter vector elements are visually explained in Fig. 2. For our case study, we fixed the lateral movement of the vehicles at $y^{ego} = y^{mio} = 0.0m$. Additionally, the range limits of the validation parameter vector elements are given in Table I.

In this work, we used IDM to control the longitudinal motion of the MIO vehicle. The configuration parameters of the IDM are fixed throughout all experiments in this study.

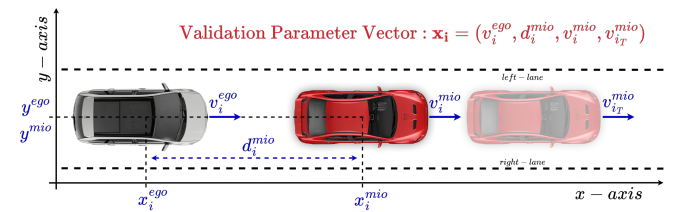


Fig. 2. Case study illustration of validation parameters in an ACC scenario, where v_i^{ego} is an initial velocity of the EGO vehicle, d_i^{mio} is an initial MIO vehicle distance relative to an EGO vehicle, v_i^{mio} and v_{iT}^{mio} are initial and target velocities of the MIO vehicle for a scenario parameter vector \mathbf{x}_i .

B. Black-Box Deep RL Model

A black-box RL agent is trained with a continuous action policy algorithm PPO to learn longitudinal low-level control behavior in safety-critical ACC scenarios.

TABLE I

THE OBSERVATION AND ACTION SPACES OF THE BLACK-BOX MODEL.

Observation Space (S)		
v_t^{ego}	longitudinal velocity	[10m/s, 40m/s]
$\dot{v}_t^{ego} = \alpha_t$	longitudinal linear acceleration	[-4m/s ² , 4m/s ²]
$\ddot{\alpha}_t$	EGO vehicle jerk in x-axis	[-8m/s ³ , 8m/s ³]
d_t^{mio}	MIO vehicle relative distance	[10m, 120m]
$v_t^{mio} - v_t^{ego}$	MIO vehicle relative velocity	[-40m/s, 40m/s]
Action Space (A)		
a_t	throttle & brake combination	[-1, +1]

1) *State Space*: The observation state s_t of our RL agent at each step $t \in (0, 1, \dots, T)$, where T is the episode horizon, can be represented as a generalized vector space format in Table I as $\mathbf{S} = \{s_t \in \mathbb{R}^5 : v_t^{ego}, \alpha_t, \ddot{\alpha}_t, d_t^{mio}, v_t^{mio} - v_t^{ego} \in \mathbb{R}\}$. Normalization is applied before feeding an observation state to a simple dense 2-layer hidden multi-layer perceptron (MLP) model with $2 \times [64, \text{ReLU}]$ number of neurons for both the actor and critic networks.

2) *Action Space*: A control action a_t of the black-box model is denoted as a combination of throttle ($a_t > 0$) and brake ($a_t < 0$) pedal openings. The output of the actor (policy) network with \tanh activation function is directly applied to the simulator at each episodic step t . We restricted our action space to a continuous domain of longitudinal control without any steering motion as our case study only involves ACC scenarios on a straight-road highway.

3) *Reward Function*: The custom reward function in Eq. (16) is designed to express the vehicle following scenario with a safety metric (\mathcal{T}_t^{gap}). A positive reward is given when the black-box agent follows the lead vehicle (MIO) with $0.8s \leq \mathcal{T}_t^{gap} \leq 2.0s$. However, when the agent fails to drive safely and crashes with the front vehicle, a significant penalty (negative reward) is given, and the episode is terminated. The reward function $\mathbf{R}_t(s_t, a_t)$ for RL training is as follows:

$$\mathbf{R}_t^{gap} = \begin{cases} -\frac{1}{\mathcal{T}_t^{gap}} & \text{if } \mathcal{T}_t^{gap} \in (0.0, 0.8) \\ +\mathcal{T}_t^{gap} & \text{if } \mathcal{T}_t^{gap} \in [0.8, 2.0] \\ -\mathcal{T}_t^{gap} & \text{if } \mathcal{T}_t^{gap} > 2.0 \end{cases} \quad (15)$$

$$\mathbf{R}_t(s_t, a_t) = \begin{cases} -10.0 & \text{collision} \\ \mathbf{R}_t^{gap} + c_j \cdot \dot{\alpha}_t + \frac{c_{sp} \cdot v_t}{v_{max}} & \text{otherwise} \end{cases} \quad (16)$$

where $c_j = -0.5$ is a penalty coefficient for the vehicle jerk $\dot{\alpha}_t$ and $c_{sp} = 5.0$ is a reward coefficient for the velocity v_t at the episodic time-step t . These weight values are determined through experiments and account for the safety and comfort emphasis in the ACC scenarios. When the black-box agent experiences a jerk, a negative reward is provided since it closely resembles a lack of driving comfort.

C. Self-Improving Method

We proposed a continuous learning pipeline in which a black-box model is periodically tested with rare-event simulations and improved by effectively leveraging safety-critical scenarios during training stages. All verification samples are stored in a library to be sampled during training of the black-box agent policy. The probability of sampling from the library of rare-event simulations is given in Eq. (17) as:

$$P(X^{\pi_k} | \pi_k(\theta)) = \frac{1}{2} \cdot \frac{k+1}{\sum_{g=0}^{G+1} [g]}, \quad k \in (0, 1, \dots, G) \quad (17)$$

where G is the total number of generations while training a black-box model. $X^{\pi_k} = \{\mathbf{x}_i : \forall i \in (1, 2, \dots, N)\}$ is a set of validation scenario events of the policy π_k , where k is a generation number and N is a total number of rare-event simulations. As illustrated in Fig. 3, at each generation, the trained model parameters θ are transferred to the next generation policy to distill learned knowledge and gained experience from previous generations so the continuous transfer learning is carried out. One of the rare-event simulation methods, detailed in Section II-A, is applied for black-box model verification after each generation stage separately. A set of scenario simulation parameters X^{π_k} of the verified/tested policy π_k (black-box model) is added to the library of rare-event simulations (safety-critical scenarios) as depicted at Stage-III in Fig. 1. After computing the probability in Eq. (17), an initial state observation s_0 is determined by selecting scenario event parameter $\mathbf{x}_i \in X^{\pi_k}$ from the uniform discrete random distribution $U\{\mathbf{x}_0, \mathbf{x}_N\}$.

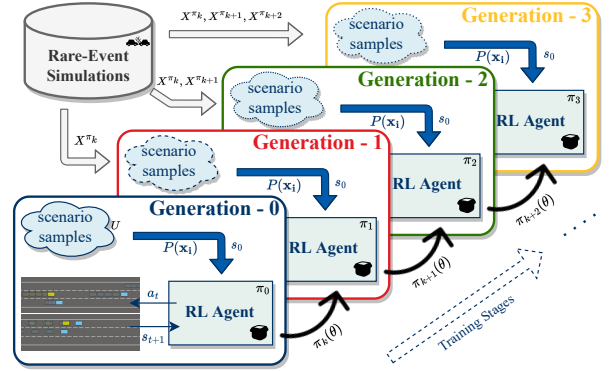


Fig. 3. **Proposed Improvement Stages**. The safety performance of the black-box model is increased with self-improvement during RL training.

In the Generation-0 stage of black-box policy training, as shown in Fig. 3, the scenario initial condition for each training episode is constructed from an initial state observation s_0 that is sampled from the uniform continuous random distribution within the state space ranges provided in Table I. However, in subsequent generations of black-box model training, scenarios are sampled from a combination of both a uniform distribution with a probability of $P_U = 1/2$ and the library of rare-event simulations as in Eq. (17). The likelihood of sampling a safety-critical scenario $P(X^{\pi_k})$ from generation k is increased compared to previous generations, as explained in Eq. (17), as the aim is to enable the training agent to learn from its mistakes and safety-critical failures.

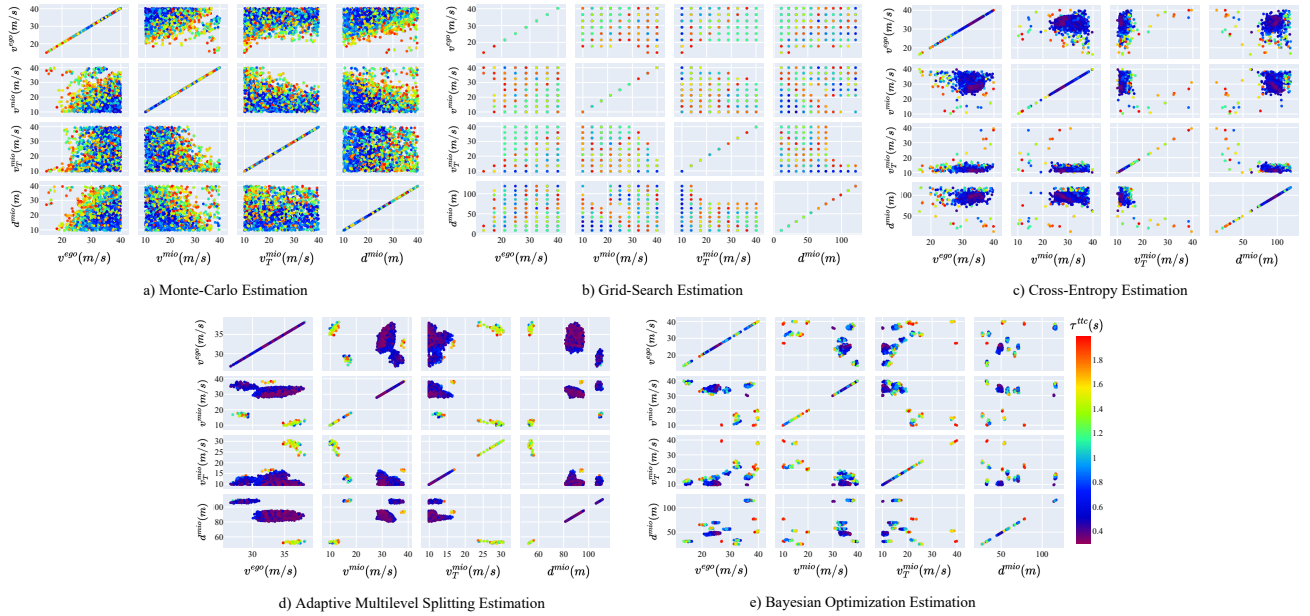


Fig. 4. **Generating Safety-Critical Scenarios.** The results of the rare-event scenario estimation algorithms for $\tau^{ttc} \in [0.0s, 2.0s]$ are visualized in matrices. Naïve MC (a) and GS (b) sampling methods are inefficient for estimating the rare-event probability p_τ . A parametric CE (c) method converges to one local minimum, while non-parametric AMS (d) and BO (e) approaches estimate importance sampling distribution with multiple local minimums.

IV. EXPERIMENTS & RESULTS

In this section, the performance outcomes of several verification techniques, detailed in Section II-A, are discussed along with the rare-event probabilities p_τ of the algorithms. The effectiveness of each rare-event algorithm on the safety performance of the self-improving approach, for building safe and reliable black-box systems, is analyzed in detail.

A. Simulation Setup

The training and validation simulations are performed in a Highway-Env Gym environment [22] since the simulator is robust, parallelizable, and sample efficient for conducting multiple-parameter traffic scenarios. As clarified in Section III-A and in Fig. 2, the simulator is initialized with scenario parameters \mathbf{x}_i at each verification iteration $i \in (1, 2, \dots, N)$ during application of the black-box validation algorithm. Every simulation is carried out at a $10hz$ frequency with $T = 250$ time-steps episode length, yielding $25s$ of scenario events. We limited the number of vehicles in an environment to two, as the ACC scenarios with only longitudinal control are studied in this work.

B. Training RL Agent

The continuous action PPO policies are trained using the Ray-RLlib library [23], with default configuration hyperparameters ($\gamma = 0.95, \alpha_{lr} = 0.0003, \epsilon = 0.2, \beta_{KL} = 0.01, \lambda = 1.0$). The RL agent operates as a black-box model, subject to continual verification and testing using rare-event estimation algorithms, as shown in Fig. 1. At each continual generation, the trained parameters of the black-box model are loaded and the training process continues. The training and validation environment of the black-box model is depicted in Stage-IV of Fig. 1, and one generation is defined as a single continual loop of self-improvement, as illustrated in Fig. 3.

In Section III-B, we thoroughly described the observation space \mathbf{S} and the action space \mathbf{A} of a black-box RL model, and clarify their respective ranges in Table I. During each generation cycle shown in Fig. 1, the RL agent undergoes transfer learning for $300K$ training steps. Initially, the black-box model parameters are loaded randomly, and the policy is trained by sampling from a continuous uniform distribution within the observation space ranges. After each generation, the previously trained model parameters and weights are loaded to continue optimizing the black-box agent on the upgraded rare-event scenario library, as visualized in Fig. 3. At the end of each episode, the environment is reset to the initial configurations of the ACC events, which are sampled from both the uniform distribution and the library of safety-critical scenarios (as described in Section III-C).

C. Agent Verification

Several rare-event estimation algorithms, including the BO method, CE search, and AMS method, are applied to evaluate the black-box model continuously. Additionally, for benchmarking and comparison purposes, GS and naïve MC sampling techniques are tested. The sample-efficient performance of generating safety-critical scenarios that cause a black-box system to fail is depicted in Fig. 4. The objective function chosen to minimize is TTC, as ACC scenarios with lower values of TTC have a higher probability of causing vehicle accidents. A fixed number of rare-event simulation samples, $N = 8192$, is used in each generation for all estimation techniques. In each generation, as visualized in Fig. 3, N estimation samples are added to the library of rare-event simulations. Each estimation sample is formulated as a vector \mathbf{x}_i and stored in the library separately for each black-box verification algorithm. A detailed explanation of the validation vector \mathbf{x}_i is provided in Section III-A.

D. Self-Improvement Results

We evaluated the proposed self-improvement approach by training four different RL agents over three generations, totaling 1.2M simulation time steps for each agent. The primary objective was to decrease the number of vehicle collisions and improve the safety of the black-box system. To demonstrate the effectiveness of rare-event simulation algorithms in a self-improvement approach for a safety-critical black-box system, we compared our proposed approach with GS and naïve MC sampling methods using black-box verification algorithms. Fig. 5 shows the normalized number of collisions with MIO vehicles in the ACC scenarios constructed from a continuous uniform distribution space. After each proceeding generation of self-improvement with rare-event simulation samples, the black-box agent experiences fewer collisions, as demonstrated in Fig. 5 and Table II.

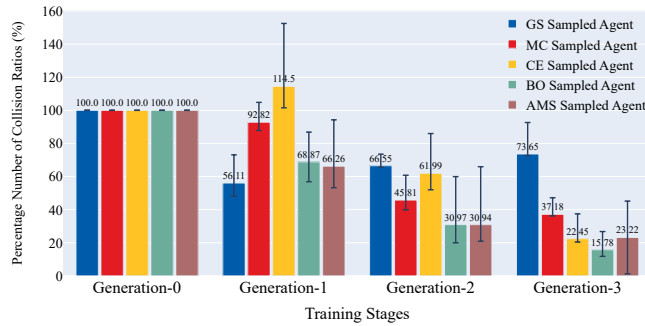


Fig. 5. **Percentage Ratios of Vehicle Collisions.** The self-improving black-box models are evaluated at each training generation stage under a uniform state distribution. The max-min performance with the average collision ratios, normalized w.r.t. the collisions in Generation-0, is illustrated for each RL agent. The evaluations are carried out with 7 random seeds.

To further investigate the safety and reliability performance of the proposed self-improvement approach, we evaluated the Generation-3 black-box RL agents on unseen safety-critical scenarios where $\tau^{ttc} < 5.0$ seconds. We chose a TTC threshold of $\tau^{ttc} < 5.0$ seconds to resemble the higher collision risk in the scenario event, as research in [16] considers $\tau^{ttc} \in [3.5s, 5.0s]$ as a safety-critical condition. We collected 1000 scenario events by applying CE, BO, and AMS algorithms to all Generation-3 trained RL agents and selecting $\tau^{ttc} < 5$ seconds samples randomly. We evaluated RL agents on these rare-event scenarios with 5 different random seeds, and the collision results with standard deviations are presented in Table II. Additionally, we evaluated a rule-based IDM, described in Section II-C, on these 1000 risky scenarios and visualized the time-gap and TTC performance comparisons in Fig. 6. The IDM vehicle did not crash in any of these ACC scenarios. However, the observed mean TTC values were lower than those of the trained RL agents.

TABLE II

THE NUMBER OF COLLISIONS IN SAFETY-CRITICAL SCENARIOS.

Self-Improved RL Agents in 1000 Samples : $\tau^{ttc} < 5s$				
GS	MC	CE	BO	AMS
328±21	182±15	105±25	137±17	27±12

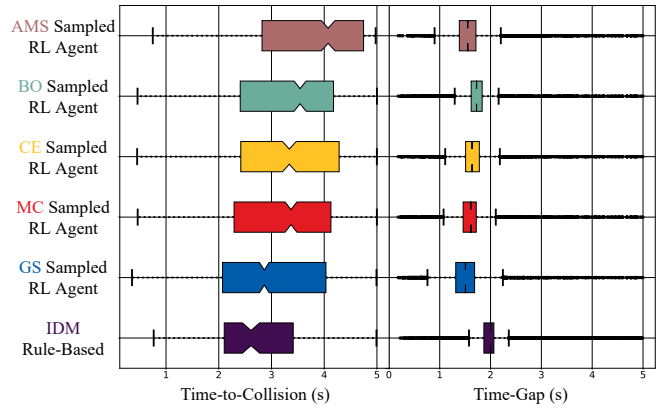


Fig. 6. **Time-Gap and Time-to-Collision Performance.** Evaluation comparisons of the Generation-3 RL agents and the rule-based IDM in randomly selected 1000 rare-event scenarios where $\tau^{ttc} < 5s$.

The trajectories in Fig. 7 depict the MIO vehicle following performance of Generation-3 agents on some failure events of Table II.

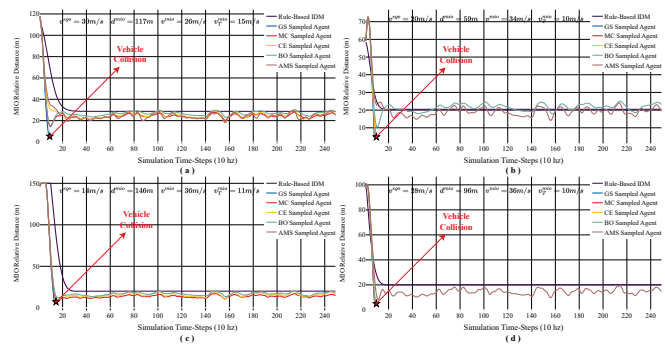


Fig. 7. **Comparison of Agent Collision Scenarios.** Lead vehicle relative positions with respect to the EGO vehicle are compared at each time step. Equations in each plot indicate the sampled scenario parameter vector \mathbf{x}_i elements (rounded here).

V. CONCLUSION & FUTURE WORK

In this work, we have proposed a novel self-improving framework using black-box verification algorithms for enhancing the safety performance of reinforcement learning-based autonomous driving agents. We have demonstrated that the weaknesses of the exploration nature of RL agents could be inspected by leveraging rare-event simulations. With the proposed methodology, safety-critical scenarios of the black-box system are detected in the early stages of the model training. Simulation results show that with our proposed iterative self-improvement approach, the RL-based black-box AD system significantly reduces failures and vehicle collisions.

The future work will expand to a higher-dimensional scenario vector space, including multiple vehicle parameters. Additionally, steering control will be added to the action space to enable more generalized driving maneuvers.

ACKNOWLEDGEMENT

This work is supported by Istanbul Technical University BAP Grant NO: MOA-2019-42321. We gratefully thank Eaton Technologies for their technical support.

REFERENCES

- [1] S. Kuutti, S. Fallah, and R. Bowden, "Training adversarial agents to exploit weaknesses in deep control policies," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 108–114.
- [2] M. Kahn, A. Sarkar, and K. Czarnecki, "I know you can't see me: Dynamic occlusion-aware safety validation of strategic planners for autonomous vehicles using hypergames," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 11 202–11 208.
- [3] Y. Bicer, A. Alizadeh, N. K. Ure, A. Erdogan, and O. Kizilirmak, "Sample efficient interactive end-to-end deep learning for self-driving cars with selective multi-class safe dataset aggregation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2629–2634.
- [4] S. Asmussen and P. W. Glynn, *Stochastic simulation: algorithms and analysis*. Springer, 2007, vol. 57.
- [5] A. Li, L. Sun, W. Zhan, M. Tomizuka, and M. Chen, "Prediction-based reachability for collision avoidance in autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 7908–7914.
- [6] J. Norden, M. O'Kelly, and A. Sinha, "Efficient black-box assessment of autonomous vehicle safety," *CoRR*, vol. abs/1912.03618, 2019. [Online]. Available: <http://arxiv.org/abs/1912.03618>
- [7] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1–7.
- [8] Y. Abeyirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8271–8277.
- [9] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/file/653c579e3f9ba5c03f2f2f8cf4512b39-Paper.pdf>
- [10] 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. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0164121217302546>
- [11] A. Sinha, M. O'Kelly, R. Tedrake, and J. C. Duchi, "Neural bridge sampling for evaluating safety-critical autonomous systems," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6402–6416, 2020.
- [12] A. Corso and M. J. Kochenderfer, "Interpretable safety validation for autonomous vehicles," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.
- [13] L. C. Das and M. Won, "Saint-acc: Safety-aware intelligent adaptive cruise control for autonomous vehicles using deep reinforcement learning," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 2445–2455. [Online]. Available: <https://proceedings.mlr.press/v139/das21a.html>
- [14] A. Alizadeh, M. Moghadam, Y. Bicer, N. K. Ure, U. Yavas, and C. Kurtulus, "Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1399–1404.
- [15] 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.
- [16] X. Li, A. Rakotonirainy, and X. Yan, "How do drivers avoid collisions? a driving simulator-based study," *Journal of Safety Research*, vol. 70, pp. 89–96, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022437518307333>
- [17] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.
- [18] J. B. Mockus and L. J. Mockus, "Bayesian approach to global optimization and application to multiobjective and constrained problems," *Journal of optimization theory and applications*, vol. 70, no. 1, pp. 157–172, 1991.
- [19] R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*. Springer, 2004, vol. 133.
- [20] F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," *Stochastic Analysis and Applications*, vol. 25, no. 2, pp. 417–443, 2007. [Online]. Available: <https://doi.org/10.1080/07362990601139628>
- [21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: <http://arxiv.org/abs/1707.06347>
- [22] E. Leurent, "An environment for autonomous driving decision-making," <https://github.com/eleurent/highway-env>, 2018.
- [23] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, J. Gonzalez, K. Goldberg, and I. Stoica, "Ray rllib: A composable and scalable reinforcement learning library," *CoRR*, vol. abs/1712.09381, 2017. [Online]. Available: <http://arxiv.org/abs/1712.09381>