SafeDriveRL: Combining Non-cooperative Game Theory with Reinforcement Learning to Explore and Mitigate Human-based Uncertainty for Autonomous Vehicles

Kenneth H. Chan, Sol Zilberman, Nick Polanco, Joshua E. Siegel, and Betty H.C. Cheng
Michigan State University
East Lansing, Michigan, USA
{chanken1,zilberm4,polanco3,chengb,jsiegel}@msu.edu

ABSTRACT

Increasingly, artificial intelligence (AI) is being used to support automotive systems, including autonomous vehicles (AVs) with self-driving capabilities. The premise is that learning-enabled systems (LESs), those systems that have one or more AI components, use statistical models to make better informed adaptation decisions and mitigate potentially dangerous situations. These AI techniques largely focus on uncertainty factors that can be explicitly identified and defined (e.g., environmental conditions). However, the unexpected behavior of human actors is a source of uncertainty that is challenging to explicitly model and define. In order to train a learning-enabled AV, developers may use a combination of real-world monitored data and simulated external actor behaviors (e.g., human-driven vehicles, pedestrians, etc.), where participants follow defined sets of rules such as traffic laws. However, if uncertain human behaviors are not sufficiently captured during training, then the AV may not be able to safely handle unexpected behavior induced by human-operated vehicles (e.g., unexpected sudden lane changes). This work introduces a non-cooperative game theory and reinforcement learning-based (RL) framework to discover and assess an AV's ability to handle high-level uncertain behavior(s) induced by human-based rewards. The discovered synthetic data can then be used to reconfigure the AV to robustify onboard behaviors.

ACM Reference Format:
Kenneth H. Chan, Sol Zilberman, Nick Polanco, Joshua E. Siegel, and Betty H.C. Cheng. 2024. SafeDriveRL: Combining Non-cooperative Game Theory with Reinforcement Learning to Explore and Mitigate Human-based Uncertainty for Autonomous Vehicles. In *19th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '24), April 15–16, 2024, Lisbon, AA, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3643915.3644089

1 INTRODUCTION

Recently, significant and rapid advancements in deep learning (e.g., object detection algorithms for onboard cameras) have facilitated their use in autonomous vehicles (AVs) [1]. These learning-enabled systems (LESs) are safety critical [19], where their failure can result in loss of life, injuries, and financial damage.[1] LESs may not operate as expected if the training data does not sufficiently capture run-time contexts [22]. For example, an AV may be trained with external agents such as pedestrians or other vehicles, whose training behavior conforms to defined rules [39]. In contrast, human agents may exhibit unexpected behaviors (e.g., suddenly braking, unexpected lane changes, etc.), thereby introducing a new source of uncertainty for AVs. This paper uses the term "ego vehicle" to describe the AV under study and "non-ego vehicle" to describe other vehicles. We propose a non-cooperative game theory[2] [30] and reinforcement learning (RL) [37] framework to discover unexpected behavior exhibited by a trained (selfish) non-ego vehicle and the unexpected responses from a *naïve*[3] ego vehicle, which can be used to enable the robustification of the ego vehicle to prevent and/or mitigate unsafe situations.

In order to guarantee a high level of safety during operation, LESs must operate safely and correctly in various operating contexts (i.e., environmental conditions, pedestrians, obscured lane markings, etc.). Existing research has largely addressed well-defined uncertainties and their effects on LESs, including environmental uncertainties [22–24] and adversarial examples [10, 35]. In contrast, human behaviors provide a source of uncertainty that cannot be explicitly defined and modeled. Specifically, human behaviors that are often motivated by various personal factors, such as temperament, schedules, attention level, etc., lead to unpredictable actions from the human driver and unknown responses in the AV [17].

This paper introduces SafeDriveRL, an automated synthetic test data generation and analysis framework used to discover unexpected non-ego behavior, assess the undesirable response of the ego vehicle, and inform the reconfiguration of the ego LES to improve its robustness against the discovered behaviors. In order to achieve these goals, SafeDriveRL contributes three key insights. First, we use non-cooperative game theory to define and model the relationship and interactions between an ego AV and a non-ego human-driven vehicle on the road in order to capture real-world driving scenarios. Second, SafeDriveRL discovers uncertain and unexpected behaviors from non-ego vehicles whose objective is to optimize its human-based objective using RL. Specifically, a developer can declaratively specify different types of human-driver personas [5] to capture the non-ego vehicles' objectives (e.g., aggressive, distracted, overly-cautious drivers, etc.), then RL is used

SEAMS '24, April 15–16, 2024, Lisbon, AA, Portugal

ACM ISBN 979-8-4007-0585-4/24/04…$15.00
https://doi.org/10.1145/3643915.3644089

[1] This paper uses the term LESs to refer to AI-based systems whose behaviors are optimized based on training experiences (e.g., object detection algorithms).
[2] Non-cooperative game theory describes a game theory setup where players are only aware of and attempt to optimize their individual objectives [30].
[3] Basic driving behavior with no experience with uncertainty.

to discover how a non-ego player may learn unexpected maneuvers (i.e., sequence of atomic driving actions) that adversely affect an ego AV's performance, such as swerving, sudden braking, etc. Third, the learned non-ego behavior information can be subsequently used to assess the robustness of the ego vehicle. The discovered information enables the developer to improve or fine-tune the ego vehicle against the unexpected behavior(s) of the non-ego vehicle.

SafeDriveRL uses RL to support non-cooperative gaming between the ego and non-ego vehicles to explore uncertain behavior(s) and discover appropriate mitigation strategies for the ego vehicle. We define a two-player non-cooperative game between the ego and non-ego vehicle, where the players compete to learn their respective optimal policies. The objective of the ego and non-ego vehicle is to complete a navigation task subject to operational and safety constraints. Additionally, the non-ego vehicle must also capture a given human driver persona and unique motivations. SafeDriveRL leverages RL to reduce the search space and solve the two-player game, approximating the best strategies for the players that maximize their respective rewards without changing strategies [30].

Preliminary results indicate that SafeDriveRL can successfully discover previously unknown behaviors that a non-ego vehicle may use to optimize their human-based objective, discover erroneous or undesirable behaviors in the naïve ego vehicle induced by the non-ego vehicle's unexpected maneuvers, and enable the developer to reconfigure or improve the ability of the ego vehicle to handle the uncertainty. We implemented a proof-of-concept prototype for SafeDriveRL and demonstrated our approach on a sample use case scenario using different driver personas. The remainder of the paper is organized as follows. Section 2 reviews background information and relevant enabling technologies. Section 3 overviews the methodology used in the proposed approach. Section 4 describes the proof-of-concept implementation and demonstration of the SafeDriveRL framework. Section 5 discusses related work. Finally, Section 6 concludes the paper and discusses future directions.

2 BACKGROUND

This section describes background information and related enabling techniques used in SafeDriveRL.

2.1 Uncertainty Challenges for AVs

Increased reliance on machine learning algorithms and other black-box approaches has introduced new challenges concerning the ability of AVs to behave safely in the face of uncertainties. In order to optimize safety and mitigate liabilities, AVs must demonstrate safe behavior during deployment and avoid accidents in addition to satisfying safety constraints. As such, some AVs tend to err on the side of caution and choose conservative actions when applicable [4, 34]. However, self-serving drivers may exploit the safety properties of the AVs for personal motivations. For example, an aggressive driver who learns that AVs will yield to avoid a collision during a lane merge may aggressively cut off an AV to merge into the AV's lane with minimal distance and time. While all road users should abide by traffic laws, human drivers learn to drive defensively as they accumulate road experience and safely mitigate similar situations (e.g., increasing the minimum trailing distance) [12]. In contrast, AVs exhibit semi-deterministic safe behaviors that can be exploited by human drivers (i.e., AVs demonstrate similar behavior(s) when encountering similar scenarios).

2.2 Non-cooperative Game Theory

In game theory, a number of players P interact with each other and the environment to maximize their individual objective(s) or a global objective [31, 32]. A number of existing approaches have explored the use of game theory for AVs [8, 40, 42], but they often assume a cooperative setting, where individual players may form coalitions with other players in the environment in order to maximize their collective reward(s) [9, 36]. A cooperative game setting is applicable to traffic scenarios where every vehicle can communicate with each other, nearby infrastructure, and share a common goal to optimize the traffic flow. However, vehicles on the road (currently) do not always share goals nor collaborate with each other. In contrast, players in a non-cooperative game setting act independently without collaboration and only consider their own interests [30]. A non-cooperative game theory setting closely resembles an AV's environment when deployed in a real-world setting, as individual vehicles (i.e., players) are not aware of other vehicles' intentions and cannot explicitly share messages or intentions. Furthermore, each vehicle on the road pursues their individual goals of reaching the destination and preventing collisions, which may be augmented by individual performance and motivations (e.g., speeding tendencies, sudden lane change movements, etc.).

2.3 Deep Reinforcement Learning

In RL, a *player* learns a set of *actions* in an interactive *environment* to maximize cumulative *rewards*. A goal-seeking player interacts with its environment to learn the optimal actions for any state through trial-and-error. In deep reinforcement learning (DRL), the learned policy is represented by a deep neural network (DNN). For example, a DRL algorithm that trains an AV to park automatically might reward the player based on the distance and orientation to the desired parking spot, incur a penalty for each timestep (to encourage fast convergence), and heavily penalize the player for collisions [43]. The AV learns a policy to maximize rewards based on past training experiences, known as *episodes*.

3 METHODOLOGY

This section introduces the SafeDriveRL framework. Figure 1 shows a data flow diagram of SafeDriveRL, where circles denote computational steps, parallel lines represent persistent data stores, and arrows show the data flow. Although SafeDriveRL can be applied to a variety of traffic scenarios, we use a *merge* scenario as the running example to illustrate the key elements of our framework.

3.1 Preliminaries

SafeDriveRL uses a non-cooperative game-theoretic formulation of vehicles and their interactions to discover uncertain human behaviors in a simulation environment. We consider a two-player game between an ego vehicle and a non-ego vehicle. This game comprises a player set I, each player's strategy space S_i, and a payoff function π^i, where the term i corresponds to player i [13, 14]. This game-theoretic formulation is realized as an RL system. Each player i uses a DNN, $f^i : \texttt{state} \rightarrow \texttt{action}$, for decision-making,

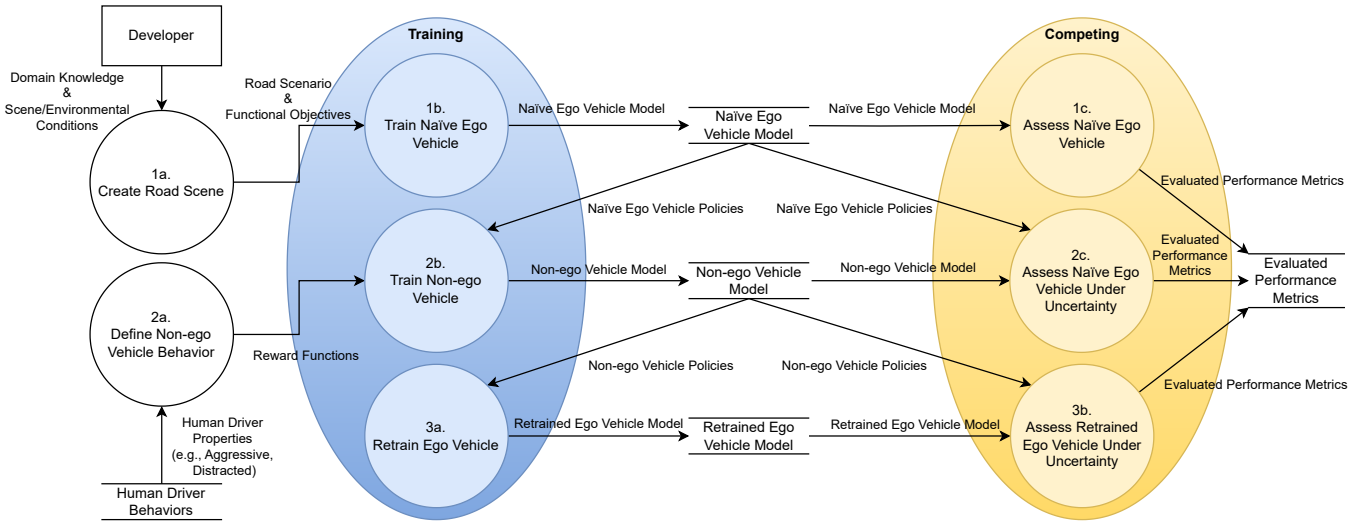


Figure 1: A high-level data flow diagram for the SafeDriveRL framework.

where f^i is a real-valued function that maps the current state (i.e., internal state such as position, velocity, and environment state such as external agents, roadways etc.) of the game to an action (i.e., actuator inputs such as acceleration, braking, and steering). Specifically, we consider the learned function f^i to represent the strategy s_i of player i [21]. Each player i is only aware of their own objective and the state of the current environment. Thus, player i in our non-cooperative game seeks to find a "best" strategy s_i^* that maximizes *individual* payoff $\pi^i(s_i)$. The payoff function π^i is constructed based on the weighted summation of a given player i's safety constraints (e.g., traffic rule violations, distance to other vehicles), operational constraints (e.g., offset from optimal navigation path), and human-based constraints (e.g., distracted driver behaviors) (Expression 1), aggregated over each timestep over an entire game [14].

$$\pi^i(s_i) = \alpha_0 \mathcal{R}_{\text{safety}}(i) + \alpha_1 \mathcal{R}_{\text{operational}}(i) + \alpha_2 \mathcal{R}_{\text{human}}(i) \quad (1)$$

The functions $\mathcal{R}$ are real-valued functions that map a player i's state to an RL reward value based on behavioral constraints (e.g., behaviors of an aggressive driver). The coefficients $\{\alpha_k : 0 \leq k \leq 2\}$ can be tuned by developers or generated from analyzing real-world driving data. After training both players, we consider the resulting players' DNNs to have approximated the best strategy selection s^*, where $\forall i \in I, \forall s_i \in S_i$:

$$\pi^i(s^*) \geq \pi^i(s_i, s^*_{-i}) \quad [13] \quad (2)$$

(s_i, s^*_{-i}) denotes a strategy selection where player i is the only player that does not use a strategy from s^* (i.e., any change in strategy for player i, assuming all other players use s^*, will result in a lower payoff) [13].

3.2 Step 1: Initialization

SafeDriveRL assesses the ability of a naïve ego vehicle to handle human-induced uncertainty and discovers undesirable behaviors. As such, **Step 1** involves training or creating a base player. **Step 1** is analogous to a novice driver who has just completed their driver's education course. This driver has learned basic driving maneuvers and can successfully drive from one location to another designated location with minimal failures (e.g., crashes). However, the novice driver has not been exposed to a variety of behaviors from other road users, including uncertain or dangerous behaviors. In SafeDriveRL, a developer can use different sources for the naïve ego vehicle, such as traditional software implementation of Advanced Driver-Assistance System features, RL models, or a digital twin that reflects the behavior of an AV under study [11, 18, 38].

Step 1a. *Create Road Scene.* Developers must first specify the road scene and simulation environment for a given scenario. Namely, a developer should specify the road network configuration (e.g., road edges), initial vehicle configurations (e.g., locations, velocity, orientation), and environment parameters (e.g., speed limit, time delta). Figure 2 shows an example road scene setup for a lane changing merge scenario, where a non-ego vehicle (denoted in orange) is driving alongside an ego vehicle (denoted in blue) on a two-lane road that converges to a single-lane, thereby forcing the non-ego vehicle to merge into the ego vehicle's lane.

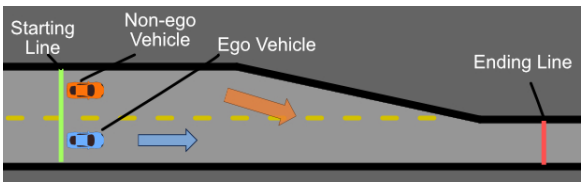


Figure 2: Graphical depictions of a merge road scenario.

Step 1b. *Develop Naïve Ego Vehicle.* The naïve ego-vehicle is developed (e.g., using rule-based behavior or is trained with RL) with the objective of completing the basic scenario. For example, an RL-trained ego vehicle is trained to complete the navigation task while optimizing a safety-oriented reward function. The ego vehicle

seeks to maintain a safe distance from other vehicles in the environment, avoid collisions with other objects and road boundaries, and navigate to the desired location.

***Step 1c**. Assess Naïve Ego Vehicle Performance.* The final initialization step assesses the naïve ego vehicle's performance using a set of performance metrics defined by the developers. For example, a developer may assess the average speed of the naïve ego vehicle, average trailing distance, number of crashes, etc. These values provide a basis for performance evaluation in the subsequent steps.

3.3 Step 2: Training the Non-ego Vehicle

The goal of this step is to identify two types of uncertain behaviors: unexpected maneuvers that a non-ego vehicle may exhibit on the road and unexpected responses from the naïve ego vehicle. This step captures Round 1 of the gaming setup, including both the training phase (of non-ego vehicle) and the competing phase (between ego and non-ego vehicles) to identify respective unexpected behaviors.

***Step 2a**. Define Non-ego Vehicle Behavior.* In order to model and simulate human driver maneuvers, the developers define an RL reward function for the non-ego vehicle based on existing human driver behavior models. For example, the developer may be interested in the AV's behavior in the presence of fast or distracted drivers. As such, a developer may define a reward function that promotes the non-ego vehicle to merge into the naïve ego vehicle's lane, while minimizing the time required to perform the maneuver. In contrast to existing non-cooperative game theory approaches where the malicious intent of the non-ego vehicle is to minimize the naïve ego vehicle's reward [16], the non-ego vehicle in SafeDriveRL only seeks to maximize its own reward. This approach closely captures the relationship between road users in practice, as drivers seek to prioritize their self-serving goals but do not typically maliciously sabotage the objectives of other vehicles.

***Step 2b**. Train Non-ego Vehicle.* Next, the non-ego vehicle is trained using RL to discover potential behaviors (e.g., maneuvers) that it may employ to maximize the reward objectives. Specifically, RL simulates and enables a non-ego vehicle to explore and test a variety of maneuvers over many episodes (i.e., instances of the road scenario). During each episode, the non-ego vehicle explores a number of states, or road scenario configurations (e.g., positions, speeds of the vehicle, etc.) at a given timestep, and approximates the best next action. By disregarding maneuvers that result in low rewards and learning maneuvers that result in high rewards, RL discovers an optimal policy for the non-ego vehicle to maximize the given human reward function based on the defined driver persona. The policy learned by the non-ego vehicle represents a mechanism for selecting optimal actions based on the state of the environment at a given timestep. For example, a non-ego vehicle may learn over time to quickly (and aggressively) merge in front of the naïve ego vehicle in order to complete its navigation task in minimal time.

***Step 2c**. Assess Naïve Ego Vehicle Under Uncertainty.* This step assesses the performance of the naïve ego vehicle as it operates in the context of the non-ego vehicle. We use the vehicle evaluation metrics defined in **Step 1** to evaluate both the naïve ego and non-ego vehicles. By analyzing how the behavior of the naïve ego vehicle changes/reacts in response to the presence of the non-ego vehicle, a developer can discover and assess the effects of maneuvers used by the non-ego vehicle that impact the safety or performance of the naïve ego vehicle. For example, the aggressive behavior learned by the non-ego vehicle may trigger a previously unknown response from the naïve ego vehicle (e.g., abruptly braking or swerving).

3.4 Step 3: Retraining the Ego Vehicle

The final step of SafeDriveRL addresses the ability of the retrained ego vehicle to operate safely in the presence of uncertain human behaviors exhibited by the non-ego vehicle. Analogously, the novice driver gained diverse driving experiences and exposure to different types of uncertain driver behaviors. As the novice driver gains on-road experience, they learn to drive defensively in the presence of other road users, including maneuvers that can reduce the chance of collisions in response to the uncertainties. **Step 3** represents Round 2 of the gaming process, where the ego vehicle is retrained with the (fixed) non-ego vehicle and then the retrained ego competes with the non-ego vehicle.

***Step 3a**. Retrain Ego Vehicle.* This step retrains the ego vehicle in an environment with the non-ego vehicle from **Step 2**. Specifically, the *retraining scenario* involves a retrained ego vehicle learning to complete the intended navigation tasks (e.g., driving from the starting line to the ending line) in the presence of uncertainty posed by the non-ego vehicle. The objective of the non-ego vehicle in this step is to complete the navigation task using the policy learned in **Step 2**, while the objective of the retrained ego vehicle is to robustify its policy by learning new actions associated with states that result from the non-ego vehicle's unexpected behaviors. With RL, the retrained ego vehicle learns new defensive maneuvers, such as preemptively reducing their speed, to improve the performance and safety of its navigation objectives.

***Step 3b**. Assess Retrained Ego Vehicle Under Uncertainty.* Finally, the retrained ego vehicle is re-assessed to compare the performance of the naïve ego vehicle versus the retrained ego vehicle in the presence of the non-ego vehicle. This metric shows whether the retrained ego vehicle is better at safely handling uncertain maneuvers exhibited by the non-ego vehicle.

4 DEMONSTRATIONS

This section describes a proof-of-concept use case for SafeDriveRL. We overview our simulation environment and experimental setup. We discuss the implementation details for the driving scenarios, RL setup, and the results of the experiments.

4.1 Simulation Environment

SafeDriveRL requires a simulation environment to model the vehicles and their interactions in order to discover undesirable behavior using non-cooperative game theory and RL. While SafeDriveRL can be applied to any simulation environment that accepts player actions and returns environment states, we developed TinyRoad, a modular 2D simulation environment that models the behavior of vehicle interactions with an emphasis on computational efficiency to demonstrate the feasibility of our approach. TinyRoad takes a

configuration file for each scenario. The configuration file comprises all relevant information for instantiating a given scenario, such as the road environment setup (i.e., road edge specification).

4.2 Experimental Setup

To evaluate the ability of SafeDriveRL to discover erroneous behavior in the naïve ego vehicle induced by human behaviors, we applied SafeDriveRL with TinyRoad to demonstrate our example use cases. In order to discover the learned policies for both vehicles, we implement a standard DRL environment using OpenAI Gym [7]. Specifically, players use an actor-critic-based DNN for decision-making in the environment [2, 29, 33]. The position and velocity of both vehicles are randomly initialized for repeated episodes. While SafeDriveRL accepts any configuration of driver persona as defined by the developer, this work shows two different personas for demonstration purposes. The selected personas are based on the most common types of dangerous driving behaviors from a report by the AAA Foundation for Traffic Safety [15]. The parameters used for RL training are shown in Table 1. All experiments were conducted using Python on a computer running Ubuntu 20.04, with 32GB of RAM, Intel I7 CPU, and an NVIDIA GTX 3060 GPU.

For each use case, we address the following research questions (RQs) and define our null (i.e., H_0) and alternate (i.e., H_1) hypotheses as follows. To answer our RQs, we describe our numerical results and demonstrate the statistical significance using the paired-samples one-tailed t-test [23]. Finally, a demonstration package with trained models is provided for validation of our results. [4]

Research Questions

RQ1: Can SafeDriveRL train a non-ego vehicle model based on human objectives that induce failures in the ego vehicle compared to the baseline scenario?

$H_0(\mu_{\text{diff}} = 0)$: There is no difference in ego performance.
$H_1(\mu_{\text{diff}} > 0)$: There is a difference in ego performance.

RQ2: Can we use the non-ego vehicle model trained by SafeDriveRL to improve the ability of the ego vehicle to operate in the face of the discovered uncertainty?

$H_0(\mu_{\text{diff}} = 0)$: There is no difference between a reconfigured ego vehicle and the naïve ego vehicle.
$H_1(\mu_{\text{diff}} > 0)$: There is a difference between a reconfigured ego vehicle and the naïve ego vehicle

Table 1: Overview of RL training parameters.

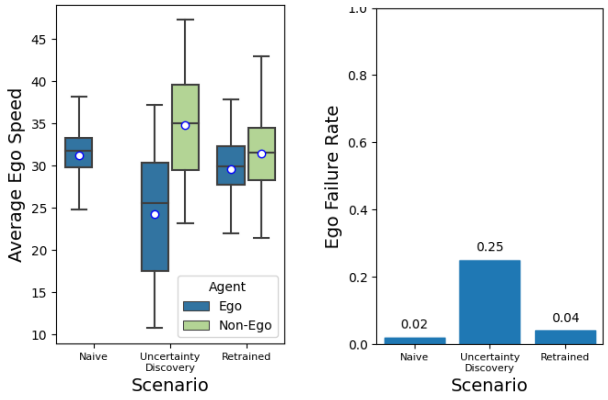
Hyperparameters	Value	Hyperparameters	Value
RL Training Steps	3×10^6	Clip ϵ	0.1
Early convergence	True	Entropy Coefficient	0.02
Optimizer	Adam	Learning Rate	1×10^{-4}
Discount Factor γ	0.99	Replay Buffer Size	2048

4.3 Use case study: Road Merge

Figure 2 shows a graphical example of the merge scenario in the use case, where a two-lane road *converges* into a single lane. As the orange non-ego vehicle approaches the end of its lane, it seeks to merge into the blue ego vehicle's lane. In this scenario, a naïve ego vehicle is trained to drive safely from the green starting line to the red ending line using RL. Next, a non-ego vehicle is trained to merge into the naïve ego vehicle's lane with a given human behavior model. The following subsections describe two non-ego vehicle behavior models: aggressive/speedy and distracted. Finally, the retrained ego vehicles assess whether SafeDriveRL can improve the robustness of the naïve ego vehicle against the newly discovered uncertainties.

4.3.1 Road Merge: Aggressive/Speedy Driver. This experiment explores the interaction between a speedy driver and the naïve ego vehicle. Traffic data (i.e., fatal accident causes, driver behavior) reported by the NHTSA [27] and the AAA Foundation for Traffic Safety [15] were used to inform the reward structure. For example, as a speedy driver often does not follow the local speed limit, thus we introduced a fuzzy logic function to promote the speedy non-ego vehicle to exceed the speed limit appropriately. To address our RQs, we evaluate the ability of SafeDriveRL to train a non-ego vehicle model with human-based behaviors and assess if the data can be used to improve the robustness of the ego vehicle.

Figure 3 shows a graphical comparison of the average speed (3a) and failure rate (3b) over 100 episodes. In the presence of the speedy driver, the naïve ego vehicle experiences a high failure rate and shows a slower average speed. Specifically, the presence of the non-ego vehicle induces the naïve ego vehicle to crash 25% of the episodes. We found strong evidence using the paired-samples one-tailed t-test (with $p < 0.01$) to reject H_0 and support H_1 for *RQ1*. In contrast, the retrained ego vehicle demonstrates improved performance, returning to a similar level of speed and failure rate as the scenario without the non-ego vehicle. The results indicate that SafeDriveRL can discover human-induced maneuvers for the non-ego vehicle that degrade the naïve ego vehicle's performance. The maneuvers discovered by SafeDriveRL informed a developer reconfiguration of the naïve ego vehicle that was then able to successfully mitigate the maneuvers of the non-ego vehicle. We found strong evidence ($p < 0.01$) to reject H_0 and support H_1 for *RQ2*.



(a) Average speed. **(b) Average failure rate.**

Figure 3: Comparison of the naïve ego vehicle and retrained vehicle in the presence of the speedy non-ego vehicle.

4.3.2 Road Merge: Distracted Driver. The second player explored in this work is a *distracted driver persona*. In 2021, NHTSA reported that 3,522 accidents were due to distracted driving [26]. A non-ego

[4] An anonymized demonstration package is available at https://anonymous.4open.science/r/safedriver-demo-5688/

vehicle is trained using the same rewards provided to the naïve ego vehicle. To capture distracted driving behavior, the non-ego vehicle enters a distracted state and omits atomic actions from the RL model during random intervals, based on NHTSA claims that indicate viewing a text message takes about 5 seconds [26]. Figure 4 shows a sample episode of the distracted merge scenario, where a non-ego driver enters a distracted state as their vehicle is drifting sideways towards the naïve ego vehicle. As the naïve ego vehicle seeks to maintain a large distance between itself and nearby objects, it swerves away and collides with the guard rail.

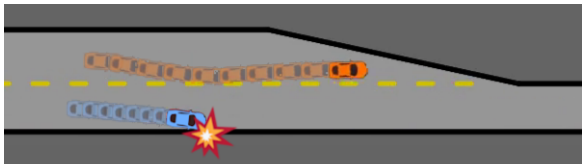
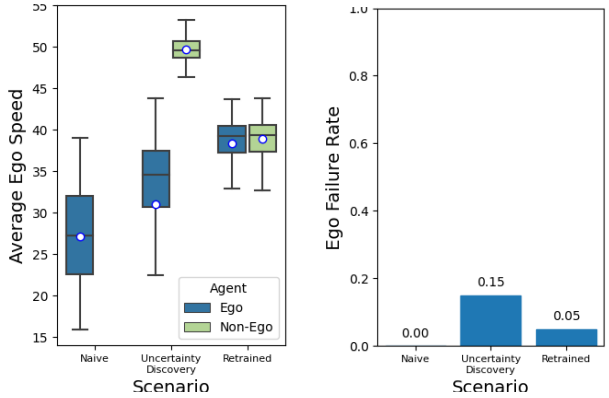


Figure 4: Example of discovered *uncertain* maneuver.

Figure 5 shows a graphical comparison of the average speed (5a) and failure rate (5b) over 100 episodes. In the presence of the distracted driver, the naïve ego vehicle's failure rate increases from zero to 15% of episodes. We found strong evidence ($p < 0.01$) to reject H_0 and support H_1 for *RQ1* of the distracted driver merge scenario. After the ego vehicle is retrained with the distracted non-ego vehicle, the retrained ego vehicle is capable of reducing the failure rate to 5%. We also found strong evidence ($p < 0.01$) to reject H_0 and support H_1 for *RQ2*. We found that the retrained scenarios include scenarios where the distracted driver drives directly into the ego vehicle, causing a failure. This experiment shows that a retrained ego vehicle is capable of learning defensive maneuvers to mitigate the uncertain maneuvers of the distracted non-ego vehicle.



(a) Average speed. **(b) Average failure rate.**

Figure 5: Comparison of the naïve ego vehicle and retrained ego vehicle in the presence of the distracted non-ego vehicle.

4.4 Threats to validity

This paper shows that RL and non-cooperative game theory can be synergistically integrated to identify undesirable behaviors in an ego vehicle induced by human uncertainties. The results of the experiments may vary with repeated experiments, as the training of RL rely on a trial-and-error approach using non-determinism, and the decisions of RL are based on a probabilistic estimate to maximize the reward function. Additionally, the learned behavior of the driver may not fully capture human driver behavior. Changes in operational context may require restructuring the reward function. Finally, we acknowledge the possible discrepancies between behaviors observed in simulation and reality (i.e., "reality gap").

5 RELATED WORK

This section overviews related work for RL and applications of game theory related to AVs. To the best of our knowledge, SafeDriveRL is the first to address uncertain human driver behavior using non-cooperative game theory and RL for AVs.

In order to study the strategic interactions between AVs and other road users, a number of researchers have modeled their behavior using game theory. Banjanovic-Mehmedovic et al. [3], Michieli and Badia [28], and Wei et al. [41] proposed several game theory modeling approaches for AVs and surrounding objects (e.g., pedestrians, road users, other AVs, etc.). Xue et al. [42] proposed a game theory approach to address the leader-follower problem for multiple AVs. Bui et al. [8] and Wang et al. [40] proposed a cooperative game theory approach to improve traffic flows for intersections. Liniger et al. [25] defined a non-cooperative game theory approach for autonomous racing games, but both players share the same reward function. Our approach focuses on the interaction between a variety of human driver behaviors (modeled as a player) and the AV (modeled as the other player), where the two players do not share identical objectives to discover potential strategies that human drivers may exploit against the AV.

Finally, several researchers have explored the use of RL to train or improve the performance of AVs. Ko et al. [20] introduced an RL approach to learn how a vehicle system can share driving authority between a human and a vehicle. Bouton et al. [6] proposed using RL to learn how AVs can navigate dense merging scenarios with various "levels of cooperation" from other drivers. Gupta et al. [16] introduced the PAIN framework to train two dueling RL players, but do not leverage a game theory approach. Our approach uses RL in order to generate road scenario edge cases to test and improve the ego model's behavior using non-cooperative game theory.

6 CONCLUSION

This paper introduced SafeDriveRL, a non-cooperative game theory and RL approach to discover and mitigate unexpected AV behaviors induced by *uncertain and unexpected* human behaviors. We demonstrated that a non-ego vehicle optimizing their reward functions can learn policies that may leverage or exploit an ego AV's safety properties. Preliminary results show that SafeDriveRL can create more robust systems capable of handling a variety of representative human driving behaviors. Future work will explore how SafeDriveRL can be extended to assess and improve the robustness of AVs with other external actors (e.g., pedestrians, cyclist, obstacles, etc.), environmental factors, and infrastructures. Additional studies may combine existing technologies for procedurally-generated road scenes in order to automatically test an AV system (e.g., a digital twin [11, 18, 38]) under a variety of road scenarios.

REFERENCES

[1] 2023. Ford's BlueCruise Ousts GM's Super Cruise as CR's Top-Rated Active Driving Assistance System. https://www.consumerreports.org/cars/car-safety/active-driving-assistance-systems-review-a2103632203/.

[2] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* 34, 6 (2017), 26–38.

[3] Lejla Banjanovic-Mehmedovic, Edin Halilovic, Ivan Bosankic, Mehmed Kantardzic, and Suad Kasapovic. 2016. Autonomous vehicle-to-vehicle (v2v) decision making in roundabout using game theory. *International journal of advanced computer science and applications* 7, 8 (2016).

[4] Brian Bell. 2022. Autonomous vehicles can be tricked into dangerous driving behavior. https://www.universityofcalifornia.edu/news/autonomous-vehicles-can-be-tricked-dangerous-driving-behavior.

[5] Stefan Blomkvist. 2002. Persona–an overview. *Retrieved November* 22 (2002), 2004.

[6] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J Kochenderfer. 2019. Cooperation-aware reinforcement learning for merging in dense traffic. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 3441–3447.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[8] Khac-Hoai Nam Bui and Jason J Jung. 2018. Cooperative game-theoretic approach to traffic flow optimization for multiple intersections. *Computers & Electrical Engineering* 71 (2018), 1012–1024.

[9] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. 2012. Cooperative game theory: Basic concepts and computational challenges. *IEEE Intelligent Systems* 27, 3 (2012), 86–90.

[10] Kenneth Chan and B.H.C Cheng. 2022. EvoAttack: An Evolutionary Search-Based Adversarial Attack for Object Detection Models. In *Search-Based Software Engineering: 14th International Symposium, SSBSE 2022, Singapore, November 17–18, 2022, Proceedings*. Springer, 83–97.

[11] B.H.C. Cheng, Robert Jared Clark, Jonathon Emil Fleck, Michael Austin Langford, and Philip K McKinley. 2020. AC-ROS: assurance case driven adaptation for the robot operating system. In *Proceedings of the 23rd acm/ieee international conference on model driven engineering languages and systems*. 102–113.

[12] David W Eby. 1995. *An analysis of crash likelihood: age versus driving experience*. Technical Report.

[13] Drew Fudenberg and Jean Tirole. 1989. Noncooperative game theory for industrial organization: an introduction and overview. *Handbook of industrial Organization* 1 (1989), 259–327.

[14] Takako Fujiwara-Greve. 2015. Nash Equilibrium. In *Non-Cooperative Game Theory*, Takako Fujiwara-Greve (Ed.). Springer Japan, Tokyo, 23–55. https://doi.org/10.1007/978-4-431-55645-9_3

[15] Andrew Gross. 2023. Risky Business – More than Half of All Drivers Engage in Dangerous Behavior. https://newsroom.aaa.com/2023/11/risky-business-more-than-half-of-all-drivers-engage-in-dangerous-behavior/.

[16] Piyush Gupta, Demetris Coleman, and Joshua E. Siegel. 2023. Towards Physically Adversarial Intelligent Networks (PAINs) for Safer Self-Driving. *IEEE Control Systems Letters* 7 (2023), 1063–1068. https://doi.org/10.1109/LCSYS.2022.3230085

[17] Elizabeth M Hill, Lisa Thomson Ross, and Bobbi S Low. 1997. The role of future unpredictability in human risk-taking. *Human nature* 8 (1997), 287–325.

[18] David Jones, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. 2020. Characterising the Digital Twin: A systematic literature review. *CIRP journal of manufacturing science and technology* 29 (2020), 36–52.

[19] J. C. Knight. 2002. Safety critical systems: challenges and directions. In *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002.* 547–550.

[20] Sangjin Ko. 2021. Reinforcement Learning Based Decision Making for Self Driving & Shared Control Between Human Driver and Machine.

[21] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. 2017. A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3323fe11e9595c09af38fe67567a9394-Paper.pdf

[22] Michael Austin Langford, Kenneth H Chan, Jonathon Emil Fleck, Philip K McKinley, and B.H.C Cheng. 2023. MoDALAS: addressing assurance for learning-enabled autonomous systems in the face of uncertainty. *Software and Systems Modeling* (2023), 1–21.

[23] Michael Austin Langford and B.H.C Cheng. 2021. Enki: a diversity-driven approach to test and train robust learning-enabled systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 15, 2 (2021), 1–32.

[24] Michael Austin Langford and B.H.C Cheng. 2021. "Know What You Know": Predicting Behavior for Learning-Enabled Systems When Facing Uncertainty. In *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 78–89.

[25] Alexander Liniger and John Lygeros. 2019. A noncooperative game approach to autonomous racing. *IEEE Transactions on Control Systems Technology* 28, 3 (2019), 884–897.

[26] NHTSA Media. 2021. Distracted Driving. https://www.nhtsa.gov/risky-driving/distracted-driving

[27] NHTSA Media. 2021. Driving Behaviors Reported For Drivers And Motorcycle Operators Involved In Fatal Crashes. https://www.iii.org/fact-statistic/facts-statistics-aggressive-driving

[28] Umberto Michieli and Leonardo Badia. 2018. Game theoretic analysis of road user safety scenarios involving autonomous vehicles. In *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 1377–1381.

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[30] John Nash. 1951. Non-cooperative games. *Annals of mathematics* (1951), 286–295.

[31] Martin J Osborne et al. 2004. *An introduction to game theory*. Vol. 3. Oxford university press New York.

[32] Eric Rasmusen. 1989. *Games and information*. Vol. 13. Basil Blackwell Oxford.

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:arXiv:1707.06347

[34] Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. 2019. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences* 116, 50 (2019), 24972–24978.

[35] Prinkle Sharma, David Austin, and Hong Liu. 2019. Attacks on machine learning: Adversarial examples in connected and autonomous vehicles. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 1–7.

[36] John M Staatz. 1983. The cooperative as a coalition: a game-theoretic approach. *American Journal of Agricultural Economics* 65, 5 (1983), 1084–1089.

[37] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[38] Fei Tao, He Zhang, Ang Liu, and Andrew YC Nee. 2018. Digital twin in industry: State-of-the-art. *IEEE Transactions on industrial informatics* 15, 4 (2018), 2405–2415.

[39] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. 2008. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of field Robotics* 25, 8 (2008), 425–466.

[40] Hua Wang, Qiang Meng, Shukai Chen, and Xiaoning Zhang. 2021. Competitive and cooperative behaviour analysis of connected and autonomous vehicles across unsignalised intersections: A game-theoretic approach. *Transportation research part B: methodological* 149 (2021), 322–346.

[41] Haoran Wei, Lena Mashayekhy, and Jake Papineau. 2018. Intersection management for connected autonomous vehicles: A game theoretic framework. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 583–588.

[42] Lei Xue, Bei Ma, Jian Liu, Chaoxu Mu, and Donald C Wunsch. 2023. Extended Kalman Filter Based Resilient Formation Tracking Control of Multiple Unmanned Vehicles Via Game-Theoretical Reinforcement Learning. *IEEE Transactions on Intelligent Vehicles* (2023).

[43] Peizhi Zhang, Lu Xiong, Zhuoping Yu, Peiyuan Fang, Senwei Yan, Jie Yao, and Yi Zhou. 2019. Reinforcement learning-based end-to-end parking for automatic parking system. *Sensors* 19, 18 (2019), 3996.