IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 January 2023, accepted 7 February 2023, date of publication 15 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3245831

RESEARCH ARTICLE

Toward Learning Human-Like, Safe and Comfortable Car-Following Policies With a Novel Deep Reinforcement Learning Approach

M. UGUR YAVAS[1], TUFAN KUMBASAR[2], (Senior Member, IEEE), AND NAZIM KEMAL URE[3], (Member, IEEE)

[1]Department of Mechatronics Engineering, Istanbul Technical University, 34469 Istanbul, Turkey
[2]Department of Control and Automation Engineering, Istanbul Technical University, 34469 Istanbul, Turkey
[3]Artificial Intelligence and Data Science Research Center, Istanbul Technical University, 34469 Istanbul, Turkey

Corresponding author: M. Ugur Yavas (yavasm@itu.edu.tr)

T. Kumbasar was supported by the Turkish Academy of Sciences of Turkey (TÜBA) in part by a TÜBA Outstanding Young Scientist Award Programme (GEBİP).

ABSTRACT In this paper, we present an advanced adaptive cruise control (ACC) concept powered by Deep Reinforcement Learning (DRL) that generates safe, human-like, and comfortable car-following policies. Unlike the current trend in developing DRL-based ACC systems, we propose defining the action space of the DRL agent with discrete actions rather than continuous ones, since human drivers never set the throttle/brake pedal level to be actuated, but rather the required change of the current pedal levels. Through this human-like throttle-brake manipulation representation, we also define explicit actions for holding (keeping the last action) and coasting (no action), which are usually omitted as actions in ACC systems. Moreover, based on the investigation of a real-world driving dataset, we cast a novel reward function that is easy to interpret and personalized. The proposed reward enforces the agent to learn stable and safe actions, while also encouraging the holding and coasting actions, just like a human driver would. The proposed discrete action DRL agent is trained with action masking, and the reward terms are completely derived from the real-world dataset collected from a human driver. We present exhaustive comparative results to show the advantages of the proposed DRL approach in both simulation and scenarios extracted from real-world driving. We clearly show that the proposed policy imitates human driving significantly better and handles complex driving situations, such as cut-ins and cut-outs, implicitly, in comparison with a DRL agent trained with a widely-used reward function proposed for ACC, a model predictive control structure, and traditional car-following approaches.

INDEX TERMS Adaptive cruise control, reinforcement learning, deep learning, naturalistic driving, advanced driving assistance systems.

I. INTRODUCTION

The main responsibility of the Adaptive Cruise Control (ACC) system is to determine the safe spacing policy and required actions (e.g., acceleration, wheel torque, throttle/brake pedal, etc.), which would be handled with two different control loops: one for spacing and velocity determination, and the other for actuation [1]. Constant time headway, also known as time gap t_{gap}, is one of the most popular spacing policies both in the literature and industrial applications. By using t_{gap}, the target following distance and speed can be calculated via methods such as the Intelligent Driver Model (IDM) [2] and Model Predictive Control (MPC) [3]. Recent studies show that current ACC products are sensitive to the t_{gap} configuration, which may lead to abrupt braking due to the amplification of the lead vehicle behavior [4]. It has also been shown in [5] that, through a comparison with naturalistic driving data, it lacks anticipation. This conclusion coincides

The associate editor coordinating the review of this manuscript and approving it for publication was Chuan Li.

with the survey results conducted on ACC users, who were unsatisfied with the cut-in response [1]. Moreover, human drivers tend to drive with a variety of t_{gap} values depending on the traffic flow and personality [6], rather than a fixed and constant one. Considering the aforementioned information, we define the key research areas regarding ACC systems and car-following policies as follows:

- The ability to handle dynamic driving conditions, such as cut-in and cut-out situations, which impacts both safety and comfort.
- The capability to reflect the individual car-following style, which will be crucial for end-user satisfaction.

A. RELATED WORK

Numerous notable works in the literature tackle the previously mentioned research areas of ACCs, especially by using MPC, supervised learning, or their combination. For instance, in [7], a classifier is learned to label the driver's style, and then the MPC alongside t_{gap} is updated to match the controller behavior with the detected driving style. The efficiency of this hybrid approach is validated in simulation with simple cut-in scenarios. As an enhancement to the constant t_{gap} spacing policy, a classifier is trained to predict the target t_{gap} as well as the high-level driver action, which are processed by the MPC to calculate the desired acceleration [8]. Authors in [9] trained Neural Network (NN) based car-following policy to capture discrete driving behavior of human drivers (eg. pauses between different acceleration levels) from the NGSIM dataset [10]. It is worth underlining that the NGSIM dataset does not provide sufficient data to learn the personalized driving behavior of a single driver but rather provides the opportunity to learn an average human driver as it consists of short driving segments (less than a minute) in a multi-lane highway from hundreds of different drivers. To sum up, the major disadvantage of the usage of supervised learning approaches is the requirement for a large, comprehensive training dataset that includes must be labeled such that it includes all possible scenarios. This may be impractical or even impossible to achieve.

Various Deep Reinforcement Learning (DRL) based ACC or car-following modes have been proposed with the deep learning breakthrough in reinforcement learning [11] and its extension to the continuous control domain - Deep Deterministic Policy Gradient (DDPG) [12]. In [13], a DDPG-based car-following policy is trained using a real-world driving dataset to model human driving more accurately than the IDM and supervised NN policy. The same authors further extended the work on the DDPG-based car-following policy by parameterizing the reward function that consists of the t_{gap} term for efficiency, the Time-To-Collision (TTC) term for safety, and the jerk (J) term for comfort, using the NGSIM dataset [14]. The reported results show that the trained policy outperforms both human drivers and the MPC algorithm in terms of safety and comfort. Another work [15] utilized the NGSIM data to characterize the speed-acceleration distribution of human drivers to improve exploration through action constraints and reported better results than human drivers and MPC-based car-following. Moreover, by using a similar reward function and state representation as the mentioned works, the string stability of the DRL-based driver model was shown to be better than the IDM, while noting the training instabilities of the DDPG algorithm [16]. The drawbacks of all the aforementioned DRL-based approaches are:

- Assuming a single lead vehicle rather than multiple ones, which is not feasible or realistic for real-world cut-in/cut-out situations [13], [14], [15], [16].
- Overparameterization of the reward functions, i.e., the order of magnitude difference in the weights of the reward function terms [15], [16].
- Conducting training and testing in the same dataset, which may lead to overfitting [13], [14], [15], [16].
- Oversimplified comparison with human drivers. Jerk is mainly used as the sole comfort metric, even though jerk values cannot be accurately estimated from the NGSIM dataset [17].

To tackle these challenges, in our previous work, we first analyzed the influence of considering multiple lead vehicles in a single-lane simulation environment [18]. According to our preliminary findings, knowing what is ahead of the nearest lead vehicle or most of the important object (MIO) increased the comfort and efficiency of the DRL policy, which is defined with a continuous action space, and the resulting approach outperformed the one in [14]. Yet, we concluded that further scaling up the study to the multi-lane environment requires significantly larger training space and a more complex training environment. One of the methods to handle complex environments through DRL is shaping the action space by using techniques such as discretization and masking [19]. In a recent example of action discretization [20], the authors approximated longitudinal actions with five distinct acceleration levels by analyzing the NGSIM dataset and combined them with lateral lane-change actions to predict driver behavior.

B. CONTRIBUTION

In this work, we propose a novel DRL approach to generate safe, human-like, and comfortable car-following policies in a multi-lane dynamic driving environment with multiple lead vehicles. To accomplish such a goal, we raised and answered two main research questions within this paper which are "*How should the action space of the DRL policy be defined ?*" and "*How should the reward function be constructed and parameterized by real-world data without excessive tuning*?".

Inspired by the decision-making strategy of a human driver and previous success of the action discretization and masking, we formulate a discrete action DRL problem that outputs actions as the required increases or decreases over pedals (throttle, brake) alongside the explicit action definition of holding (keep the last action) and coasting (no action) rather

than defining a continuous action space as it is widely done in the literature such as [14] and [15]. Secondly, we have designed a novel reward function R_{pro} which is straightforward to parameterize by real-world data and does not require exhaustive weight tuning. Combined with the reward function, the discrete action DRL-policy (DRL-R_{pro}) addresses both research gaps we raised as keeping safe and comfortable driving policies under dynamic driving scenarios while making human-like strategic decisions such as the use of coasting to decelerate.

We conducted a comprehensive evaluation of the proposed DRL-R_{pro} approach in three stages. First, we compared the proposed reward function R_{pro} with the widely used reward function R_{ref} proposed in [14], to assess its impact on learning stability. Next, we analyzed the performance of the DRL-R_{pro} approach, in comparison with the reference algorithms DRL-R_{ref}, MPC, and IDM in a dynamic simulation environment with frequent cut-ins and cut-outs. Finally, all algorithms were tested in scenarios extracted from real-world driving data and compared to human driving. Our results demonstrate that the proposed DRL-R_{pro} approach significantly outperforms other benchmark car-following methods, including DRL-R_{ref}, in terms of both human likeness and comfort. It is worth stating that we have not only quantified the human likeness by only the average position error in real-world driving data but also by the ratio of conformance to R_{pro} rules (r_{hl}).

The main contribution of this paper can be summarized as:

- We propose a novel reward function R_{pro} that is easy to interpret and personalize with real-world data without the need for exhaustive parameterization.
- We propose a new discrete-action representation with action masking aiming at the direct throttle and brake pedal manipulation and explicit action definitions for holding and coasting, enabling tactical-decision making in dynamic car-following scenarios.
- Exhaustive results are presented not only in simulations (i.e. training environment) but also in scenarios generated from a real-world driving dataset where frequent cut-ins and cut-outs occur in a multi-lane environment.
- The comparative results focusing on dynamic driving conditions clearly show the superiority of the novel discrete action space representation and reward function, namely DRL-R_{pro} in comparison with DRL-R_{ref} and the reference ACC algorithms, namely MPC and IDM.

II. BACKGROUND

A. PROBLEM STATEMENT

Fig. 1a shows the block diagram of the classical and modular ACC system. Here, a camera and radar sensor are used to detect and track traffic objects in the surrounding area. This information is fused to create a list of objects with their relative positions and velocities according to the estimated ego-vehicle states (speed, acceleration). The MIO is selected from this list based on a set of rules and the spacing policy is calculated based on t_{gap} to maintain the desired distance between the ego-vehicle and MIO. Then, the required acceleration is calculated (in an "outer loop") which is then converted to throttle and brake inputs (in an "inner loop") based on the specific vehicle setup. This modular approach has the advantage of being easy to develop and test individual modules, but it also has some limitations, such as the need for tuning each module and lack of anticipation for dynamic driving conditions [18].

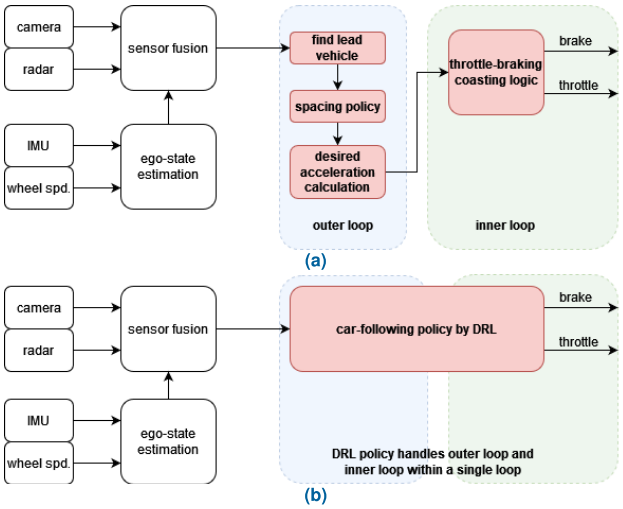


FIGURE 1. Block diagram of (a) the conventional ACC system, and (b) the proposed DRL approach which directly processes the inputs from the sensor fusion module.

B. REINFORCEMENT LEARNING: AN OVERVIEW

Reinforcement learning is a learning paradigm that relies on self-learning agents driven by a reward that is evaluated through interactions with the environment [21]. At every time step, a new observation S_t is received from the environment and an action A_t is selected by the agent, then another feedback S_{t+1} is received with a reward R_{t+1}. These units form a tuple $\langle S, A, T, R\rangle$ that is used to model the Markov Decision Process (MDP) [22]. In the model-free setup, the transition function $T(s, a, s') = P[S_{t+1} = s' \mid S_t = s, A_t = a]$ is not known, so the agent tries to find the best action set (policy: π) to maximize the reward without knowing the dynamics. The problem for a finite horizon H is defined as

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} R_t\left(S_t, A_t, S_{t+1}\right)|\pi\right]. \tag{1}$$

Q-learning is a value-based technique that estimates the optimal action-value function $Q^*(s, a)$ [11], which is

$$Q^*(s, a) = \mathbb{E}\left[r + \gamma \max_{a'} Q^*(s', a')|s, a\right]. \tag{2}$$

The Q function determines the value (expected future reward) of each action in a particular state. By calculating the Q value of each possible state, the action pair, and using the Bellman optimality equation, the optimal policy can be attained with a greedy policy of choosing actions with maximum Q values.

Yet, because of the computational burden of expressing each state-action pair, Q-learning cannot handle environments that have large, continuous state-action pairs. Starting with the Deep Q Network (DQN) [11], which approximates the Q function with deep NN and uses large experience-replay buffers to break any correlations in the training data. Ape-X DQN [23] decouples experience collection and learning with distributed experience replay to utilize many workers with different exploration configurations that send data to the shared replay buffer, resulting in state-of-the-art performance. In this study, we have customized and used the RLlib [24] implementation of the Ape-X DQN algorithm.

III. PROPOSED DRL-BASED ACC APPROACH

Fig. 1b illustrates the general block diagram of the proposed approach for ACC. Having input the object list of surrounding vehicles and ego-vehicle states, the DRL policy would determine what would be the best pedal strategy (throttle and brake level) to maximize the reward function that is parameterized to imitate human-like driving. The learning and evaluation steps of the proposed DRL-R_{pro} approach are summarized in Algorithm 1.

Algorithm 1 Policy Generation and Evaluation With DRL-R_{pro}

Require: Real-world dataset with desired behavior M, proposed reward function R_{pro}, simulation env. S with action mask F, DRL Policy P

1: Determine discrete action space by M ▷ (Sec. IV-C1)
2: Parameterize R_{pro} by M ▷ (Sec. IV-C2)
3: **while** iteration < max iteration **do**
4: P =Train(S, F, R_{pro}, P) ▷ (Sec. V-A)
5: Evaluate P in S ▷ (Sec. V-B)
6: Extract validation scenarios (V) from M
7: Evaluate P in V ▷ (Sec. V-C)

A. STATE REPRESENTATION

The state definition of the MDP is similar to our previous work [25] which has two parts. The first part includes the ego-vehicle information such as the speed, acceleration, and throttle-brake pedal levels whereas the second part includes the relative position and velocities of all vehicles ahead of the ego-vehicle. The representation of the states with their normalization formulas is presented in Table 1.

TABLE 1. Ego-centering, Normalized Cartesian State Representation.

s_1	Normalized ego-vehicle speed $2v_0/v_{\max} - 1$
s_2	Normalized ego-vehicle lane position $2y_0/y_{\max} - 1$
s_3	Normalized ego-vehicle acceleration $2a/a_{max} - 1$
s_4	Ego-vehicle throttle - break pedal value
s_{4i+1}	Normalized relative position of vehicle i, $(x_i - x_0)/x_{\text{sensor}}$
s_{4i+2}	Normalized relative position of vehicle i, $(y_i - y_0)/y_{\max}$
s_{4i+3}	Normalized relative velocity of vehicle i, $\frac{v_{x,i} - v_{x,0}}{v_{\text{set}}^{\max} - v_{\text{set}}^{\min}}$
s_{4i+4}	Normalized relative delta vel. of vehicle i, $\Delta v_i/\Delta v_{max}$

B. ACTION SPACE REPRESENTATION AND MASKING

As pointed out in [19], discretization, masking, and reducing the number of actions by using domain-specific knowledge are key to more efficient exploration and better results. Therefore, we propose combining the throttle and brake output in the same discrete action space with distinct increment/decrement levels for both, as shown in Table 2. These levels can be determined considering actuator dynamics, pedal level measurement accuracy, or completely data-driven. Obviously, more levels may approximate the continuous control problem better, but it also increases the risk of poor exploration to larger action space which is a clear trade-off to experiment. In this study, we propose to determine increment/decrement levels from a real-world driving dataset.

TABLE 2. Discrete action space of the DRL-R_{pro} agent.

$a_{1,2}$,	level 0 increment/decrement throttle, t_{l0}
$a_{i,i+1}$,	level i increment/decrement throttle, t_{li}
$a_{i+2,i+3}$,	level 0 increment/decrement brake b_{l0}
$a_{i+k,i+k+1}$,	level k increment/decrement brake b_{lk}
a_{i+k+2},	hold the current pedal
a_{i+k+3},	coasting (**apply no action**)

Algorithm 2 shows the conversion of the current action of the DRL agent a_n to the individual pedal levels after action masking. An action mask is generated at every time step by considering the infeasible and available actions, based on the actuation of pedals in the real world. This action mask is designed as a single-state machine: while braking, only coasting and brake manipulation actions are allowed, and while throttling, only actions other than braking are allowed. In addition, while coasting, only braking and throttling increments are allowed.

Algorithm 2 Conversion of DRL Actions to the Pedal Levels

Require: Current action a_n of DRL from Table 2, throttle T_n and brake B_n value

$T_{avail} = [a_1, a_2, \ldots, a_i, a_{i+1}]$
$B_{avail} = [a_{i+2}, a_{i+3}, \ldots, a_{i+k}, a_{i+k+1}]$
$\Delta = [t_{l0}, -t_{l0}, \ldots, t_{li}, -t_{li}, b_{l0}, -b_{l0}, \ldots, b_{lk}, -b_{lk}]$
if a_n is in T_{avail} **then** ▷ Adjust throttle pedal
  $T_{n+1} = T_n + \Delta[a_n]$
  $B_{n+1} = 0$
else if a_n is in B_{avail} **then** ▷ Adjust brake pedal
  $B_{n+1} = B_n + \Delta[a_n]$
  $T_{n+1} = 0$
else if a_n is a_{i+k+2} **then** ▷ Holding action
  $T_{n+1} = T_n$, $B_{n+1} = B_n$
else ▷ Coasting action
  $T_{n+1} = 0, B_{n+1} = 0$
return T_{n+1}, B_{n+1}

C. REWARD FUNCTION

In the previous DRL-based ACC approaches [13], [14], [15], [16], carefully engineered and differentiable multi-objective

reward functions considering the safety (R_s), efficiency (R_e), and comfort (R_c) terms are defined in the following generic structure:

$$R = w_1 R_s(TTC) + w_2 R_e(t_{gap}) + w_3 R_c(J) \tag{3}$$

where each reward term has a weight (w_i, $i = 1, 2, 3$) that is tuned for controlling the influence of individual reward terms. Yet, the weight tuning might turn into excessive parameter search activity, and also the alignment of reward function with actual human driving would be difficult to assess.

In this work, we propose a novel reward function that consists of logical driving rules which can be parameterized to the individual driver. The violation of each rule results in an additive negative reward.

The following reward components are designed after a detailed inspection of a real-world driving dataset:

- Drive stable: Key comfort component as the data analysis shows holding the previous actuator level of more than 80% during a ride of a human driver. Thus, we define:

$$R_{stb} = \begin{cases} -P_{stb} & \text{if } a_{n+1} - a_n > a_{stb} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

- Drive safe: We punish low TTC situations that would trigger forward collision warning (fcw)

$$R_{fcw} = \begin{cases} -P_{fcw} & \text{if } TTC < TTC_{fcw} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

- Follow the MIO: To avoid high t_{gap} situations, we include:

$$R_{flw} = \begin{cases} -P_{flw} & \text{if } t_{gap} > t_{flw} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

- Cut-in comfort: We comprise R_{cin} to not brake for the faster lead vehicles during close cut-ins.

$$R_{cin} = \begin{cases} -P_{cin} & \text{if } t_{gap} > t_{cin} \text{ and } v_{mio} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

- Prefer coasting over braking while approaching slower vehicles: Data analysis shows 7% more coasting than braking. Thus, we define:

$$R_{co} = \begin{cases} -P_{co} & \text{if } TTC > TTC_{co} \text{ and } t_{gap} > t_{co} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

- Tailgating comfort: To end up with a DRL policy that does not accelerate for the slower and too close MIO, we encompass:

$$R_{gth} = \begin{cases} -P_{gth} & \text{if } t_{gap} < t_{gth} \text{ and } v_{mio} < -0.5 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The proposed total reward R_{pro} is then defined as follows:

$$R_{pro} = R_{stb} + R_{fcw} + R_{flw} + R_{cin} + R_{co} + R_{gth} \tag{10}$$

In the proposed DRL approach, if the driving policy is conforming with each of the components of R_{pro}, then a positive reward (P_{pos}) is given as follows:

$$R = \begin{cases} P_{pos} & \text{if } R_{pro} == 0 \\ R_{pro} & \text{otherwise} \end{cases} \tag{11}$$

When we compare the proposed reward (10)-(11) with the widely employed one defined in (3), the individual reward values (P_x) in R_{pro} are used to balance the reward function instead of weights. Moreover, each component of R_{pro} can be matched with the reward terms of (3) as $R_s = R_{fcw}$, $R_e = R_{flw}$, and $R_c = R_{stb} + R_{gth} + R_{cin} + R_{co}$. On the other hand, as tabulated in Table 3, there are various thresholds to be set, yet they can be extracted from real-world driving data by examining the residency of the human driver in critical driving states in terms of TTC, t_{gap}, and actuation histograms. Thus, the components of the proposed total reward R_{pro} can be parameterized for human-like and also safe driving, as shown in the succeeding section, with minimized tuning efforts for the reward values (P_x).

TABLE 3. Description of the reward parameters.

a_{stb}	Stable driving actuation threshold
TTC_{fcw}	Critical safety threshold
TTC_{co}	TTC threshold for encouraging coasting
t_{flw}	Upper threshold of t_{gap} for MIO following
t_{cin}	Minimum t_{gap} for faster MIO
t_{co}	t_{gap} threshold for encouraging coasting
t_{gth}	Minimum t_{gap} for slower MIO
P_x	Reward values, $x \in \{co, gth, cin, flw, fcw, stb, pos\}$

IV. IMPLEMENTATION

In this section, we present all the details on how we parameterized the proposed DRL-based ACC approach alongside descriptions of the real-world dataset and simulation environment. We also provide information about the compared ACC approaches.

A. OVERVIEW OF THE REAL-WORLD DRIVING DATASET

In this study, to parameterize R_{pro} and also to evaluate the trained policy, we have collected 200 km of driving data (M) with the instrumented test vehicle shown in Fig. 2 from an

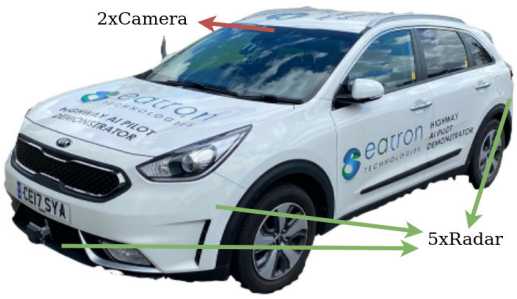


FIGURE 2. Target vehicle platform with multiple cameras and radar sensors.

TABLE 4. Overview of Real-world Dataset (*M*).

Number of cut-ins:	30
Number of cut-outs:	22
Min.–max. ego-vehicle speed:	10–110 km/h

experienced driver. Note that driver support systems, such as ACC, were not active during the ride. The dataset contains many dynamic driving situations and a wide speed range as presented given in Table 4.

B. HIGHWAY SIMULATION DETAILS

The simulation environment from our previous work [25] is modified to create a rich set of car-following scenarios and simulate real-world driving case studies using real-world logged data. Moreover, to increase the fidelity of the simulation, the motion model of the ego vehicle is defined with the physics model [26] given in (12), and its parameters are estimated from the real-world testing data.

$$m\ddot{a} = F_{xf} + F_{xr} + F_{aero} - R_{xf} - R_{xr} - mg\sin\theta \quad (12)$$

We have assumed that only the front wheels have traction and that the road has no slope. Fig. 3 shows an example result from the model and actual vehicle data.

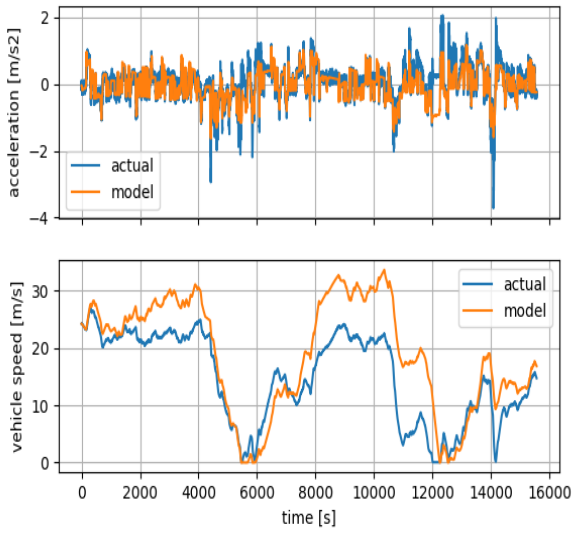


FIGURE 3. Example data showing how the force-based vehicle model captures the trends in real-world data.

In the simulations, the other vehicles, excluding the ego-vehicle, were defined with a constant acceleration model, and their requested acceleration is calculated by the standard car-following model, namely the IDM [2] which is defined as:

$$\frac{dv}{dt} = a = a_{max}\left(1 - \left(\frac{v}{v_d}\right)^{\delta} - \left(\frac{d^{\star}(v,\Delta v)}{d}\right)^{2}\right) \quad (13)$$

$$d^{\star}(v,\Delta v) = d_0 + vT_{set} + \frac{v\Delta v}{2\sqrt{ba_{max}}} \quad (14)$$

The lane change behavior of the other vehicles is modeled by the MOBIL algorithm [27], which uses the IDM to predict the acceleration of the other vehicles for potential lane changes. The safety criterion of MOBIL is defined as:

$$\tilde{a}_f > b_{safe}, \quad (15)$$

i.e., the new acceleration $\tilde{a}_f$ of the follower must be larger than the safe braking threshold b_{safe}. If the potential lane change is safe, then the incentive criterion is calculated using the current acceleration a_{ego} and the acceleration $\tilde{a}_{ego}$ in the new lane, which is defined as

$$\tilde{a}_{ego} - a_{ego} > a_{th}, \quad (16)$$

where a_{th} is the threshold. If (15) and (16) are simultaneously satisfied, the lane change maneuver is executed.

The simulation environment has a scenario generation algorithm that randomly spawns vehicles with different initial, desired speeds, and driving behavior according to the configuration values given in Table 5.

TABLE 5. Highway Simulation Parameters.

Number of lanes, n	4
Number of vehicles, m	14
Max. initial vehicle spread	200 m
Min. inter-vehicle distance	50 m
Initial speed range (rear vehicle), $[v_{min}^{rear}, v_{max}^{rear}]$	$[15, 25]$ m/s
Initial speed range (front vehicle), $[v_{min}^{front}, v_{max}^{front}]$	$[10, 25]$ m/s
Initial speed range (ego vehicle), $[v_{min}^{ego}, v_{max}^{ego}]$	$[10, 20]$ m/s
Desired speed range (rear vehicle), $[v_{min}^{d}, v_{max}^{d}]$	$[20, 30]$ m/s
Desired speed range (front vehicle), $[v_{min}^{d}, v_{max}^{d}]$	$[5, 30]$ m/s
Desired speed (ego vehicle), v_{ego}^{d}	30 m/s
Termination step	200
Sampling time, ts	0.1 s
Detection range, x_{max}	150 m

C. PARAMETERIZATION OF THE DRL-R_{pro} APPROACH

In this section, by visualizing the data statistics with histograms, we present all the details on how we parameterized the action space and reward function parameters of the proposed DRL-R_{pro} approach via the real-world driving dataset M.

1) PARAMETERIZATION OF THE ACTION SPACE

In order to define the levels of the discrete action space, we analyzed the difference of throttle and brake pedal values (starting with the action levels $(a_{l0,i}, b_{l0,k})$) with 100-ms intervals excluding the no-change instance observed during coasting and cruising with the same pedal input. Through the histograms of the throttle and brake pedal values presented in Fig. 4, we determined three distinct levels of increment/decrement values for both the throttle a_{li} $(i = 3)$ and the brake b_{lk} $(k = 3)$ that are defined with the same set of $\{0.01, 0.05, 0.1\}$ after extensive experimentation.

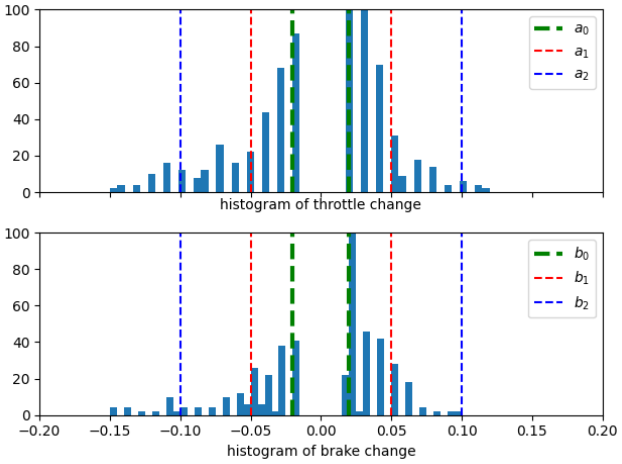


FIGURE 4. During actuation, pedal levels follow discrete steps. We select pedal increment/decrement values according to the shown $a_{0,1,2} - b_{0-1-2}$ lines.

2) PARAMETERIZATION OF THE REWARD FUNCTION

In this section, we describe how each R_{pro} component presented in (5)-(10) is parameterized using the real-world driving dataset. In this context, we analyzed and visualized the real-world driving dataset as presented in Fig. 5. The thresholds defined within R_{pro} are extracted as follows:

- The driving stability threshold a_{stb} defined in (4) is determined from Fig. 4 to be 0.01 that aligns with a_0 and b_0. Thus, any decision besides coasting and holding would trigger a negative reward of P_{stb}.
- The minimum observed time gap for a faster MIO, t_{cin}, is 0.5 seconds in all actuation conditions, which is caused by close cut-ins with faster lead vehicles, as shown in Fig. 5a.
- The minimum observed time gap for a slower MIO, t_{gth}, is 0.8 seconds when the driver is actuating the throttle pedal, according to the distribution of t_{gap} and throttle level in Fig. 5b.
- The upper threshold of the time gap for following an MIO, t_{flw}, is 2.2 seconds, indicated by the red dashed lines in Fig. 5b, considering that beyond this value, the driver actuates the throttle to reduce the gap.
- The TTC_{fcw} threshold, which regulates safe approaching limits to slower vehicles, is selected from the minimum value observed in the TTC distribution, as shown in Fig. 5c.
- The threshold for encouraging coasting while approaching slower vehicles, TTC_{co} and t_{co}, are calculated from Fig. 5d according to the red dashed lines as 10 and 2.2 seconds, respectively.

Finally, the individual reward values P_x are tuned manually as presented Table 6.

D. COMPARED ACC ALGORITHMS

The performance of the proposed DRL-R_{pro} approach is compared with the DRL-based ACC implementation presented in [14] and two widely employed baseline methods, which are the IDM and MPC-based ACC.

TABLE 6. Parameters of R_{pro}.

Parameter	Value	Parameter	Value
P_{stb}	0.5	a_{stb}	0.01
P_{fcw}	5	TTC_{fcw}	4
P_{flow}	2	t_{flw}	2.2
P_{cin}	2	t_{cin}	0.8
P_{co}	2	TTC_{co}	10
P_{pos}	0.5	t_{co}	2.2

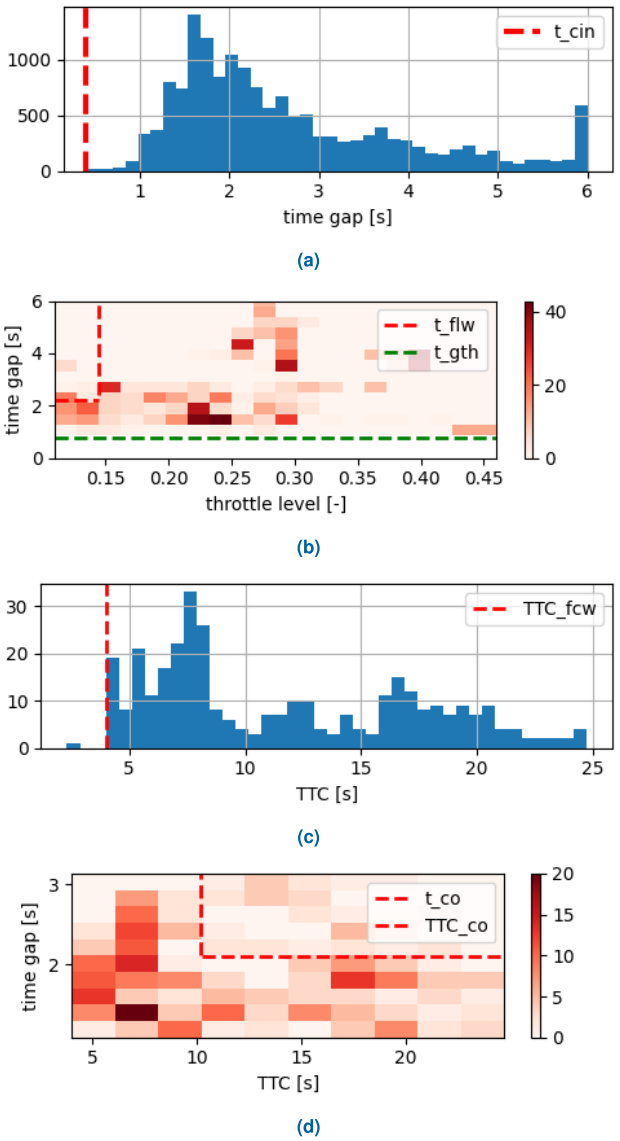


FIGURE 5. Parameterization of the reward function by the collected data. (a) t_{gap} histogram (b) t_{gap} - throttle histogram (c) TTC histogram (d) t_{gap} - TTC histogram.

1) REFERENCE DRL-BASED ACC METHOD: DRL-R_{ref}

To evaluate the effectiveness of the proposed reward function R_{pro}, we have also trained a DRL policy with the reward function defined in [14], namely R_{ref} (DRL-R_{ref}). The reward

R_{ref} and its variants are widely used in DRL-based ACC approaches [13], [14], [15], [16] since they are considered successful examples of balanced and smooth reward functions. R_{ref} is defined with the following terms that depend on TTC, t_{gap} (with $\mu = 0.4226, \sigma = 0.4365$), and J [14]:

$$R_{TTC} = \begin{cases} \log(TTC/4) & 0 \leq TTC \leq 4 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$R_{hw} = f_{\text{lognorm}}(t_{gap}), \quad R_J = \frac{J^2}{3600} \tag{18}$$

The total reward R_{ref} is then defined as the sum of these terms as follows:

$$R_{ref} = R_{TTC} + R_{hw} + R_J \tag{19}$$

2) BASELINE METHODS: IDM AND MPC

The IDM method is defined as given in (13) and (14). The IDM parameters can be chosen to reflect different driver behaviors as stated in [28]. In this work, we use aggressive (IDM-Agr) and normal (IDM-Norm) driver profiles as given in [28] to enrich our experiment evaluation.

The MPC-based ACC implementation is inspired by [14] which uses the kinematic point mass model as a motion model. The objective function consists of the following distance error $(x_n - \hat{x}_n)$, relative velocity difference with the lead vehicle (Δv_n), and ego-vehicle jerk (J). The following distance is to be tracked by the MPC $(\hat{x}_n)$ is defined via the target headway time (t_{gap}). The following finite-horizon quadratic cost function is being solved at every time step with respect to the constraints:

$$\begin{aligned} &\min_{\alpha} \sum_{t=0}^{N-1} [(\frac{x_n(t) - \hat{x}_n(t)}{x_{max}})^2 + (\frac{\Delta v_n}{\Delta v_{max}})^2 + (\frac{J(t)}{J_{max}})^2] \\ &\hat{x}_n(t) = v_n t_{gap} \\ &\text{s.t.:} \quad x_n > 0, v_n > 0 \\ &\qquad -3 < a_n < 3 \end{aligned} \tag{20}$$

where the prediction horizon, $(N = 10)$, normalization constants for the position, velocity, and jerk tracking $(S_{max} = 15, \Delta V_{max} = 8, J_{max} = 60)$ and target headway time $(t_{gap} = 1.2)$ values are set as in [14].

V. RESULTS

In this section, we first assess the training stability of the DRL-R_{pro} and DRL-R_{ref} policies. Second, we evaluate the trained policies in simulation and compare them against the baseline ACC algorithms, namely MPC and IDM. Finally, we further quantify the performance of the proposed and baseline car-following algorithms in two real-world driving scenarios. See the video provided as the **Supplementary Material** for these analyzes.

A. DRL TRAINING PERFORMANCE ANALYSIS

Training stability is one of the most important aspects of DRL-based solutions. As reported in [16], the policy may not converge in different seeds or may not give the desired performance when the reward function is slightly changed. We have trained a discrete DRL policy with five different seeds and showed the average results in Fig. 6 when R_{ref} and R_{pro} are deployed. Both reward functions result in stable training convergence. However, the reward R_{ref} generates more frequent, continuous values that result in faster learning but a less safe policy, as shown in the average accident ratios.

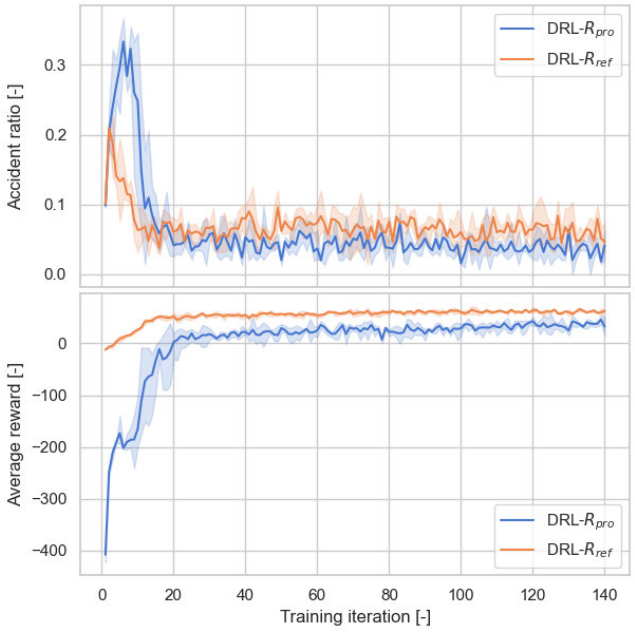


FIGURE 6. Average training dynamics for 5 different seeds.

B. EVALUATION IN SIMULATION

We evaluated the algorithms' performances with 200 validation episodes with a different seed than the training. In all runs, we set the initial conditions with the same values for all benchmark car-following algorithms for a fair comparison. To quantify the resulting performances objectively, in addition to the proposed reward function (R_{pro}), the average TTC below 4 seconds (avg. TTC), the total jerk, the average t_{gap} (avg. t_{gap}), and the average velocity (avg. v) is calculated.

Table 7 shows the normalized results (with respect to IDM-Agr) of each policy: IDM-Agr, IDM-Norm, MPC, DRL-R_{ref} and DRL-R_{pro}. IDM-Norm, MPC, and DRL-R_{ref} policies involve a few accidents in the validation runs. Note that the metrics exclude episodes with accidents to generate fair comparison besides DRL-R_{pro}. MPC and DRL-R_{ref} fill the gaps caused by cut-outs rather aggressively indicated by the lower t_{gap} and higher avg. v which leads to abrupt braking in the case of close cut-ins. IDM-Norm follows the lead vehicle with the smallest avg. t_{gap} which allows more cut-ins from the adjacent lanes and causes an increase in the total jerk. DRL-R_{pro} is the best-performing policy independent of the highest reward score, it brings the comfort benefit (lowest total jerk) with only a small loss over average velocity when compared

TABLE 7. Average statistics obtained over 200 validation episodes.

Method	avg. TTC	total jerk	avg. t_{gap}	avg. $\mathbf{v}$	R_{pro}
IDM-Agr	3.73	1	1.95	1	1
IDM-Norm	3.42	1.02	2.47	0.9	0.85
MPC	3.26	1.12	**1.78**	**1.05**	0.92
DRL-R_{ref}	3.62	1.6	1.85	1.01	0.9
DRL-R_{pro}	**3.76**	**0.97**	2.05	0.99	**1.12**

with IDM-Agr. Furthermore, the stability component of the reward function (R_{stb}) is the key factor in increasing comfort, as none of the reward components include a jerk component, unlike other reward functions in the literature.

To show the stability and success of the proposed approach, we have plotted an example validation episode in Fig. 7, where the ego-vehicle is in a rapidly changing driving environment. The plot shows the lead vehicle position, speed of the lead and ego-vehicles, throttle-brake output, and t_{gap} in comparison to DRL-R_{pro} and MPC. In the middle of the episode, there is a cut-out ahead of the lead vehicle, which causes a speed increase. DRL-R_{pro} anticipates the near future and uses coasting to slow down during the lead vehicle's acceleration, unlike MPC, which closely tracks the lead vehicle's speed with the inability to predict future speed.

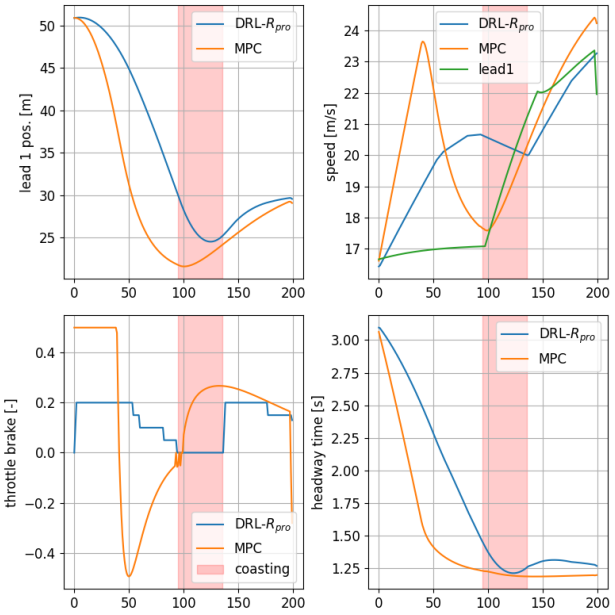


FIGURE 7. Example evaluation episode in simulation, DRL-R_{pro} uses coasting to slow down.

C. EVALUATIONS WITH REAL-WORLD DRIVING SCENARIOS

In this section, we share the results of the proposed approach and benchmark algorithms in scenarios extracted from the real-world driving dataset. Testing the DRL policy trained in the simulation environment with real-world scenarios is critical to evaluate its generalization capability. Moreover, evaluating DRL policies with metrics beyond the reward function is crucial since reward maximization may not translate into human likeness [29]. Therefore, we introduce three more metrics to monitor stability, comfort, and human likeness:

- V_{sta}: the number of time steps that the stability rule presented in (4) is violated, which is negatively correlated with comfort.
- r_{hl}: the ratio of the number of time steps with $R_{pro} = 0$ to the total number of time steps, which indicates how each policy conforms with R_{pro}.
- x_{rmse}: the mean square difference of the position between the human drive and all other policies, which is the classical metric to quantify human-like driving.

1) EVALUATION IN SCENARIO I

The first scenario is a typical low-speed commute where acceleration and deceleration waves are tracked by the preceding vehicles. The evaluation results are presented in Table 8. According to the results, the human driver rarely violates the stability rule (minimum V_{sta}) and gets the highest r_{hl}. This validates the reward thresholds found in section IV-C2 and the overall formulation of the R_{pro}. Moreover, the proposed approach DRL-R_{pro} performs closest to the human driver in terms of V_{sta}, avg. t_{gap}, r_{hl} and x_{rmse}. The normal configuration of IDM scores the second-best result after the DRL-R_{pro}.

The lead position, ego-velocity, and t_{gap} plots are given in Fig. 8 alongside the ones of the human driver. The results clearly show the difficulty of tracking a single t_{gap} under dynamic driving conditions and the variation of t_{gap} in the human-driving. One of the key findings of the analysis, trying to maintain a single t_{gap} value in dynamic conditions results in over-actuation reflected by the high V_{sta} which is not observed for the human driver and successfully imitated by the proposed approach. Another key observation is that the DRL-R_{pro} agent does not amplify the speed waves of the lead vehicles, which would help to improve the string stability.

TABLE 8. Evaluation Results in Scenario I - 603 total steps.

Method	V_{sta}	avg. TTC	avg. t_{gap}	r_{hl}	x_{rmse}
Human	32	7.09	1.57	0.92	-
IDM-Agr	163	3.63	1.29	0.72	3.71
IDM-Norm	155	4.4	1.49	0.74	3.12
MPC	78	3.5	1.24	0.70	3.8
DRL-R_{ref}	175	6.18	1.13	0.65	3.62
DRL-R_{pro}	**43**	6.52	1.72	**0.91**	**2.46**

2) EVALUATION IN SCENARIO II

The second scenario is much more challenging, as it involves a high-speed close cut-in followed by a cut-out. Table 9 tabulates the resulting performance measures. In this scenario, the human driver once again demonstrates a strong correlation with the r_{hl} metric. The DRL-R_{pro} approach performs best among the benchmark algorithms, as judged by V_{sta}, r_{hl}, and x_{rmse}. Unlike Scenario I, the aggressive configuration of IDM is the second-best-performing algorithm. This shows that slow-speed Scenario I and high-speed Scenario II require

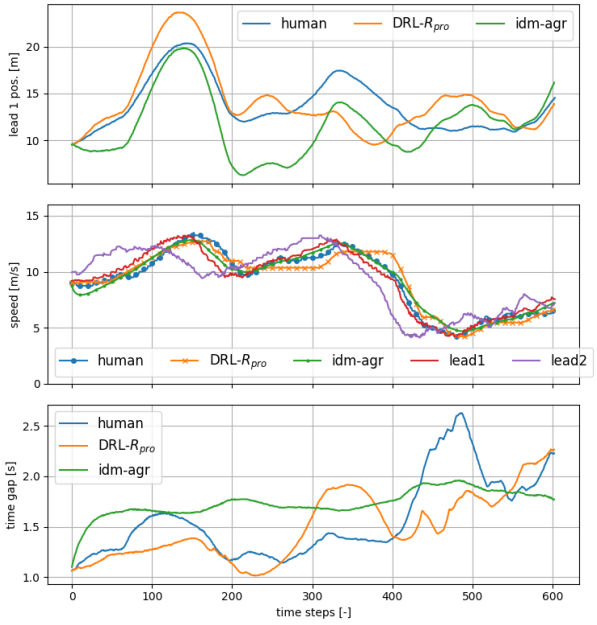


FIGURE 8. Low-speed tailgating in Scenario I.

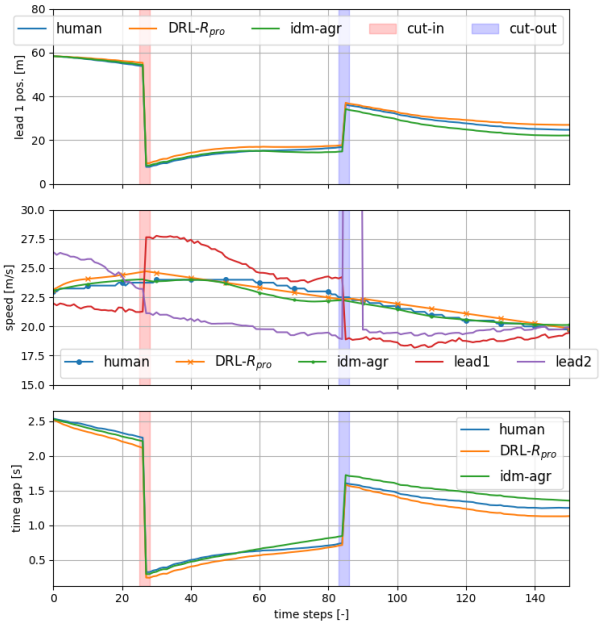


FIGURE 9. Ego vehicle is subject to the close cut-in followed by a cut-out in Scenario II.

completely different IDM parameters to accurately capture the same driver. In contrast, the DRL-R_{pro} approach consistently performs close to the human driver in both scenarios.

Fig. 9 shows the lead vehicle position, ego-speed, and t_{gap} for the human driver, IDM-Norm, and DRL-R_{pro}. Before the cut-in event, all algorithms close the gap with the lead vehicle as the human driver does. However, after the cut-in event, with access to information about the lead2 vehicle (the vehicle ahead of lead1 - MIO), DRL-R_{pro} smoothly decelerates until the end of the episode, resulting in the highest TTC, which indicates safer driving. In contrast, both the human driver and IDM-Agr continue to accelerate with the faster new lead1 vehicle and only start decelerating shortly after the lead1 vehicle before the cut-out. Therefore, Scenario II demonstrates the tactical decision-making capabilities of DRL when considering multiple dynamic traffic participants.

TABLE 9. Evaluation Results in Scenario II - 300 total steps.

Method	V_{sta}	avg. TTC	avg. t_{gap}	r_{hl}	x_{rmse}
Human	9	5.82	1.34	0.96	0
IDM-Agr	74	8.22	1.26	0.68	3.16
IDM-Norm	80	11.12	1.21	0.62	6.5
MPC	110	6.2	1.24	0.62	5.2
DRL-R_{ref}	102	11.63	1.7	0.64	4.02
DRL-R_{pro}	**29**	14.72	1.28	**0.85**	**2.42**

VI. DISCUSSION

One of the key contributions of our work is the development of the novel reward function R_{pro}, which is based on logical driving rules and can be fully parameterized using example driving data. To demonstrate the significance of this new reward function, we have trained the same DRL algorithm using the well-known reference reward function R_{ref} [14] for comparison. We have shown that R_{pro} performs exceptionally well in dynamic driving conditions. In our reward design, based on our analysis of a real-world driver, we have identified typical driving boundaries in terms of TTC and t_{gap}, and penalize deviations from these boundaries. This design is well-suited to the MDP assumption and Q-learning, where the learning process determines advantageous states that maximize reward. In contrast, the R_{ref} has terms that conflict with each other. For example, after a close cut-out, a new lead vehicle might trigger a negative reward from the TTC component while also triggering a positive reward from the t_{gap} component. Note that, in this work, we did not apply any exhaustive search over R_{ref} weights, which would have led to better performance but potentially over-parameterized reward weights as in other works [15], [16]. Another significance of R_{pro} is to generate car-following policies with minimized jerk without adding a jerk-related term to the reward function but aiming for stable actuation which would be easier to parameterize and better align with human driving.

Our second contribution is the design of the car-following problem as a discrete action tactical decision-making problem, rather than a continuous one. The results in Section V showed that, with the right number of discrete actions, there is no drawback in terms of comfort or smoothness for the discrete action design. We demonstrated, in both simulation episodes and real-world scenarios featuring dynamic behavior and varying velocity set-points, as well as frequent lane changes, that the discrete action policy outperformed classical car-following algorithms by being safer and more comfortable. Additionally, the success of the discrete action policy was supported by the inclusion of all surrounding vehicles as input, without the need for an explicit MIO selection

algorithm. A specific example from the simulation validation episodes, shown in Fig. 6, illustrates the importance of tactical decisions. In this instance, the DRL-R_{pro} agent anticipates the need to slow down for the entire episode and uses the coasting action to do so, unlike classical t_{gap}-following policies that simply try to follow the current speed of the lead vehicle without anticipating future events.

It is worth noting that a major limitation of the proposed approach is its ability to generalize beyond the training environment, yet this is a common issue with any DRL-based approach. For instance, the human driver's reaction to Scenario II is similar to the throttle-brake response shown in Fig. 10, as indicated by the low V_{sta} in Table 9. However, DRL-R_{pro} is not able to maintain the same level of stability and oscillates more. To address the differences between simulation dynamics and real-world dynamics, domain adaptation techniques [30] could be applied to improve performance. Another potential generalization issue would be the extension to urban networks, which are not as well structured as highway environments. Generating safe, comfortable, and human-like car-following policies in such an environment would require significant modification of the DRL states and R_{pro}, considering also non-motorized traffic.

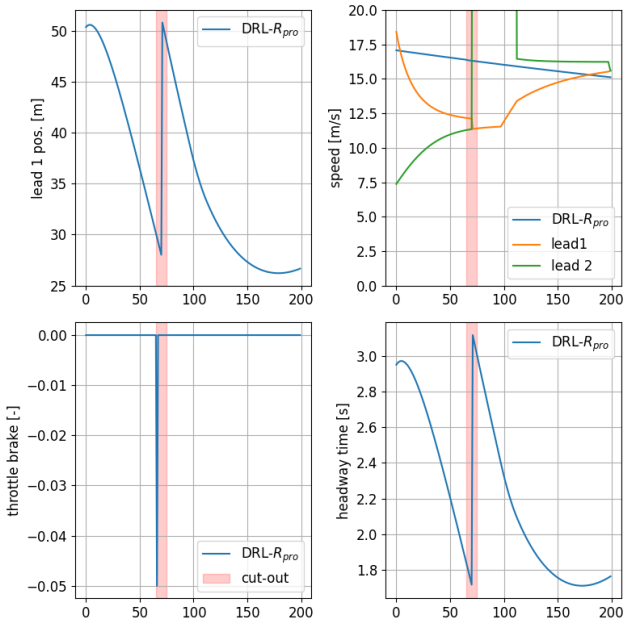


FIGURE 10. Full coasting episode.

VII. CONCLUSION AND FUTURE WORK

We have designed and validated a novel car-following policy powered by a discrete action DRL and a novel reward function, R_{pro}. Our detailed evaluations, conducted in simulations and scenarios extracted from real-world driving, show that the proposed algorithm not only outperforms the compared car-following algorithms in terms of safety, comfort but also aligns significantly better with human-like driving. In light of the results, we believe that the proposed DRL-R_{pro} policy generation method is one step forwards to achieving human-like tactical decision-making in the multi-lane dynamic driving environment as it generates discrete actions just like a human and R_{pro} reduces the efforts required for reward engineering and encourages human-like driving by promoting stable actuation and coasting.

In future work, we would proceed with real-world deployment of the DRL-R_{pro} agent. Considering the safety-critical nature of driving, our algorithm requires a safety monitoring algorithm to intervene when necessary. Moreover, instead of running the system completely end2end, we can deploy the DRL-R_{pro} agent as the high-level longitudinal policy generator and track the desired pedal values with MPC which would align better with functional safety. Another improvement of the proposed design would be considering the decision to change lanes, which would also require an update for the reward function. In terms of NN design, an attention layer after convolutions may increase overall performance and enable monitoring of the vehicles focused on by the DRL policy during the decision-making.

REFERENCES

[1] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle Syst. Dyn.*, vol. 48, no. 10, pp. 1167–1192, 2010.

[2] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 62, no. 2, pp. 1805–1824, Aug. 2000.

[3] M. Vajedi and N. L. Azad, "Ecological adaptive cruise controller for plug-in hybrid electric vehicles using nonlinear model predictive control," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 113–122, Jan. 2016.

[4] T. Li, D. Chen, H. Zhou, J. Laval, and Y. Xie, "Car-following behavior characteristics of adaptive cruise control vehicles based on empirical experiments," *Transp. Res. B, Methodol.*, vol. 147, pp. 67–91, May 2021.

[5] W. J. Schakel, C. M. Gorter, J. C. F. de Winter, and B. van Arem, "Driving characteristics and adaptive cruise control? A naturalistic driving study," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 17–24, Summer 2017.

[6] T. J. Ayres, L. Li, D. Schleuning, and D. Young, "Preferred time-headway of highway drivers," in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Feb. 2001, pp. 826–829.

[7] B. Gao, K. Cai, T. Qu, Y. Hu, and H. Chen, "Personalized adaptive cruise control based on online driving style recognition technology and model predictive control," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12482–12496, Nov. 2020.

[8] S. E. Baek, H. S. Kim, and M. Han, "Personalized speed planning algorithm using a statistical driver model in car-following situations," *Int. J. Automot. Technol.*, vol. 23, no. 3, pp. 829–840, Jun. 2022.

[9] X. Huang, J. Sun, and J. Sun, "A car-following model considering asymmetric driving behavior based on long short-term memory neural networks," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 346–362, Oct. 2018.

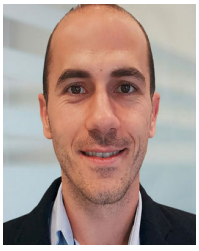
[10] *NGSIM—Next Generation Simulation*, U.S. Dept. Transp., Washington, DC, USA, 2009.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2019, *arXiv:1509.02971*.

[13] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 348–368, Dec. 2018.

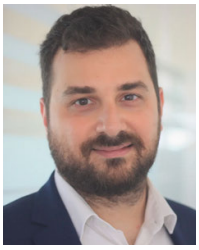
[14] M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, and R. Ke, "Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102662.
[15] W. Li, Y. Zhang, X. Shi, and F. Qiu, "A decision-making strategy for car following based on naturalist driving data via deep reinforcement learning," *Sensors*, vol. 22, no. 20, p. 8055, Oct. 2022.
[16] F. Hart, O. Okhrin, and M. Treiber, "Formulation and validation of a car-following model based on deep reinforcement learning," 2021, *arXiv:2109.14268*.
[17] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the next generation simulation (NGSIM) program data," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1243–1262, Dec. 2011.
[18] U. Yavas, T. Kumbasar, and N. K. Ure, "Model-based reinforcement learning for advanced adaptive cruise control: A hybrid car following policy," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 1466–1472.
[19] A. Kanervisto, C. Scheller, and V. Hautamäki, "Action space shaping in deep reinforcement learning," in *Proc. IEEE Conf. Games (CoG)*, Aug. 2020, pp. 479–486.
[20] B. M. Albaba and Y. Yildiz, "Driver modeling through deep reinforcement learning and behavioral game theory," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 2, pp. 885–892, Mar. 2022.
[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[22] L. Busoniu, R. Babuska, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL, USA: CRC Press, 2017.
[23] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver, "Distributed prioritized experience replay," 2018, *arXiv:1803.00933*.
[24] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "RLlib: Abstractions for distributed reinforcement learning," 2017, *arXiv:1712.09381*.
[25] U. Yavas, T. Kumbasar, and N. K. Ure, "A new approach for tactical decision making in lane changing: Sample efficient deep Q learning with a safety feedback reward," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1156–1161.
[26] R. Rajamani, *Vehicle Dynamics and Control* (Mechanical Engineering Series). New York, NY, USA: Springer, 2012.
[27] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec.*, vol. 1999, no. 1, pp. 86–94, 2007.
[28] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007.
[29] B. Matusch, J. Ba, and D. Hafner, "Evaluating agents without rewards," 2020, *arXiv:2012.11538*.
[30] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," 2020, *arXiv:2009.13303*.



M. UGUR YAVAS received the B.Sc. and M.Sc. degrees in control and automation engineering from Istanbul Technical University, where he is currently pursuing the Ph.D. degree. Since 2018, he has been with Eatron Technologies, Department of Advanced Driving Assistance Systems. He has also been working in the automotive software development industry for more than ten years and has contributed to numerous international research and development projects. His research interests include deep reinforcement learning, autonomous systems, machine learning, and advanced control concepts, such as model predictive and optimal control.

TUFAN KUMBASAR (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in control and automation engineering from Istanbul Technical University (ITU). He is currently an Associate Professor with the Control and Automation Engineering Department and the Director of the Artificial Intelligence and Intelligent Systems (AI2S) Laboratory, Faculty of Electrical and Electronics Engineering, ITU. His major research interests include computational intelligence, notably type-2 fuzzy logic, fuzzy control, neural networks, evolutionary algorithms, intelligent systems, robotics, machine learning, and intelligent control and their real-world applications.

He received the best paper awards from the IEEE International Conference on Fuzzy Systems, in 2015, and from the sixth International Conference on Control Engineering and Information Technology, in 2018. He was a recipient of the ODTÜ Mustafa N. Parlar Research and Education Foundation Research Incentive Award, in 2020, the Turkish Academy of Sciences Outstanding Young Scientists Award, in 2021, and the Istanbul Technical University Young Scientists Achievement Award, in 2022. He has served as the publication co-chair, the panel session co-chair, the special session co-chair, a PC, an IPC, and a TPC in various international and national conferences. He is also an Associate Editor for the IEEE TRANSACTIONS ON FUZZY SYSTEMS and an Area Editor for the *International Journal of Approximate Reasoning*.



NAZIM KEMAL URE (Member, IEEE) received the B.Sc. and M.Sc. degrees from Istanbul Technical University (ITU), in 2008 and 2010, respectively, and the Ph.D. degree in aerospace engineering from the Massachusetts Institute of Technology (MIT), in 2015. He is currently an Associate Professor with the Department of Aeronautical Engineering, ITU, where he also works as the Vice Dean of Research and the Vice Director of the ITU Artificial Intelligence and Data Science Applied Research Center (ITU AI). His research interests include applications of deep learning and deep reinforcement learning for autonomous systems, large scale optimization, and development of high performance guidance navigation and control algorithms.

He is also a Marie-Sklodowska Curie Fellow and has overseen several university-industry collaboration projects with companies, such as Boeing, NASA, General Electric, AVL, Turkish Airlines, Turkish Aerospace Industries, and Aselsan. He is also a member of IEEE Technical Committee on Intelligent Control. Besides his academic duties, he holds the positions of the Founder and the Director of AI with Eatron Technologies and Lisa AI.