2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

Autonomous Driving using Safe Reinforcement Learning by Incorporating a Regret-based Human Lane-Changing Decision Model

Dong Chen*[1], Longsheng Jiang*[2], Yue Wang[2], Zhaojian Li[1]

Abstract— **It is expected that human-driven vehicles and autonomous vehicles (AVs) will coexist in a mixed traffic for a long time. To enable AVs to safely and efficiently maneuver in this mixed traffic, it is critical that the AVs can understand how humans cope with risks and make driving-related decisions. In this work, we incorporate a human decision-making model in reinforcement learning to control AVs for safe and efficient operations. Specifically, we adapt regret theory to describe a human driver's lane-changing behavior and fit the personalized models to individual drivers for predicting their lane-changing decisions. The predicted decisions are incorporated in the safety regulations for reinforcement learning in training and in implementation. We then use an extended version of double deep Q-network (DDQN) to train our AV controller within the safety set. By doing so, the number of collisions in training and testing is reduced to zero, while the training accuracy is not impinged.**

Index Terms— **Safe Reinforcement Learning, Human Lane-changing Decisions, Regret Theory, DDQN**

I. INTRODUCTION

Autonomous driving has attracted significant research interest in the past two decades as it offers the potential to release drivers from exhausting driving and mitigate traffic congestion [1]. However, high-level decision-making for autonomous vehicles remains a big challenge due to the involvement of complex, cluttered environments and the dynamic, uncertain behaviors of other traffic users. Among numerous methods, reinforcement learning (RL) has been extensively explored with promising results [2]. RL-based methods can learn decision-making and driving behaviors that are hard for traditional rule-based designs, often with less human effort.

However, it is reported in [3] that when using RL-based methods lots of collisions happen before the agent is adequately trained to start to behave properly. Although these collisions are not prohibitive in simulations, all RL-based driving algorithms must be tested and retrained in real world traffic where collisions can cause disastrous consequences. Furthermore, autonomous vehicles (AVs) using trained RL algorithms may not behave safely in unseen driving environments; the trained models may choose unsafe actions due to function approximation used in most RL algorithms [4].

To provide safety guarantees in RL, the idea of safe reinforcement learning (SafeRL) [5] has been proposed where safety supervisors are deployed to ensure safe exploration and exploitation for the RL agents. Nageshrao et. al. [3] incorporate a short-horizon safety check in the RL-based method. The supervisor replaced identified risky actions with safe ones during training and implementation. Collisions were significantly reduced. Wang et al. [6] developed a rule-based decision-making framework for lane-changing. The framework examines the trajectories prescribed by the controller and changes the actions resulting in collisions. In [5], a dynamics-enabled safe RL framework is developed to train a fuel-efficient adaptive cruise control policy without collisions. However, when supervising the learning process, oversimplified, non-interactive environment vehicles are used in these studies.

To enable AVs safely interacting with manual-driven vehicles (MVs), it is crucial to understand and characterize how human drivers make driving-related decisions when they interact with other road users. Extant models of human driving behaviors are either data-driven [7] or motivational [8]. Data-driven methods typically lack explaniability [9], whereas motivational models ignore the usefulness of data and fail in generating testable predictions [10].

Risks in driving have two dimensions: harm (costs) and probability. Human drivers most times manage these risks well by making rational decisions. A regret decision model developed in our prior work [11] is a good candidate for modeling human decision-making under risks. It is based on regret theory [12] in behavioral economics which emphasizes the regret effect. It also acknowledges the probability weighting effect [13] and the range effect [14] in human decision-making. To model decision-making in driving, the abstract harm and probability are described in physical terms like speed and distance. Similar to motivational models, the regret decision model is explainable. Also, after its parameters are estimated using drivers' data, the model can predict a driver's decisions, hence the movement of the MV.

In this paper, we integrate the regret decision-making model into a safety supervisor in a SafeRL framework. The safety supervisor identifies any unsafe actions based on the predictions from the regret model. We design a hierarchical learning structure that includes a RL-based decision-making agent with an extended double Q-network (DDQN) [15]. We exploit both safe (exploited) and unsafe (virtual) experiences for training to improve the learning efficiency. The efficacy of the proposed framework is demonstrated in simulation on CARLA [16].

The remainder of the paper is organized as follows. We formulate the research problem in Section II. Section III

[1]Dong Chen and Zhaojian Li are with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA. Email: {chendon9,lizhaoj1}@egr.msu.edu.

[2]Longsheng Jiang and Yue Wang are with the Department of Mechanical Engineering, Clemson University, Clemson, SC 29634, USA. Email: {longshj,yue6}@g.clemson.edu.

*Both authors contributed equally to this work.

978-1-5386-8266-1/$31.00 ©2020 AACC

shows how to build the regret-based human lane-changing model. Our proposed SafeRL algorithm is described in Section IV. Experiments, results, and discussions are shown in Section V. We conclude the paper with future directions in Section VI.

II. Problem Formulation

A. Traffic Scenario

In a two-lane highway scenario, as shown in Fig. 1, the ego vehicle (blue, AV) is surrounded by human-driven vehicles (green). Vehicles (red) far away from the ego vehicle are not considered in the decision making. All vehicles want to run at their, possibly different, desired speeds. To achieve that goal, they need to change lanes whenever necessary while maintaining safe distances with neighboring vehicles. The AV is controlled by a RL-based intelligent agent. The agent learns how to drive, including longitudinal speed control, lane-changing strategy, etc, from interacting with the environment vehicles. We make the following assumptions:

- The AV is equipped with sensing capabilities and can measure the relative distances to the neighboring vehicles (in green).
- There is no communication between any vehicles.
- Each MV makes its driving decisions according to the driver's decision-making model (see Section III).

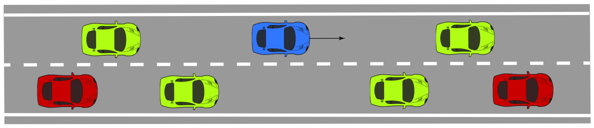

Fig. 1. Two-lane traffic scenario

B. Training with Conventional RL

When using conventional RL for training an AV, state s_t characterizes the traffic scenario, including the velocities and locations of the vehicles, and the agent (AV) applies action a_t to navigate. Given a reward function r_{s_t,a_t}, the optimal policy $\pi^*(s_t)$ is to maximize the expected cumulative future rewards:

$$R \triangleq \mathbf{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{s_t,a_t} \right] \tag{1}$$

where scalar γ is the discount factor.

Q-learning [17] is a model-free method which works well on discrete action space. It uses a function approximator $Q_\phi(s_t, a_t)$ to approximate the Q function, where the optimal Q value is defined by $Q^*(s_t, a_t)$, and ϕ are the parameters of the approximator. The action is chosen by $a_t = \max_{a'_t} Q_\phi(s_t, a'_t)$.

Deep Q network (DQN) [18] uses the deep neural networks to approximate the Q function. It stores the explored experiences into a replay buffer and samples K experiences each time to update the parameters,

$$\phi \leftarrow \phi + \alpha \big(Y^Q_{\text{target}} - Q_\phi(s_t, a_t)\big) \nabla_\phi Q_\phi(s_t, a_t) \tag{2}$$

where α is the learning rate. The target is defined as

$$Y^Q_{\text{target}} \triangleq r_{t+1} + \gamma \max_{a'} Q_{\phi_N}(s_{t+1}, a') \tag{3}$$

The parameters ϕ_N of the target network are updated only every N steps by $\phi_N \leftarrow \phi$ and kept fixed at other steps.

C. The Drawback of Conventional RL

As we tested the deep Q-network on autonomous driving in two-lane traffic (Section V), we found that about 14.5% training epochs ended with collisions. Even after the policy converged, still 3.46% of the trials caused collisions. Collisions mainly came from the fact that the AV cannot estimate the intentions of MVs. Especially for scenarios involving lane-changing, collisions can happen either because the ego vehicle changes its lane but collides with the vehicles already in that lane or an environment vehicle suddenly changes to the ego vehicle's lane so the ego vehicle cannot react in time.

Collisions cause unstable training as the RL algorithm needs to reset the simulation whenever collisions happen. Real deployment of AVs will also not tolerate any collision.

To address the problem, we present a framework for RL to incorporate a safety supervisor which uses a human lane-changing decision model for making predictions. The architecture is shown in Fig. 2.

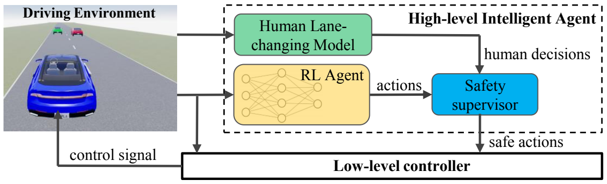

Fig. 2. Framework of SafeRL system.

Here we adopt a hierarchical structure. A high-level intelligent agent in the ego vehicle receives states of the traffic in the driving environment. Inside the agent, based on the current traffic scenario, the human-lane changing model (Section III) estimates the lane-changing decisions that will be taken by the MVs. In parallel, the RL agent determines optimal actions based on the states (Section IV). The actions from the RL agent are supervised by the safety supervisor: the supervisor uses the predicted human decisions to check if an action is safe. The unsafe actions are replaced by the safe actions. The safe actions are then applied to the low-level controller for navigating the ego vehicle in the driving environment.

III. Regret-based human lane-changing model

In this section, we present how to build a human lane-changing decision model. One of the most safety-critical decisions in the two-lane traffic (Fig. 1) is whether a driver intends to change lanes. When a driver can drive at the desired speed, or faster than vehicles in the neighboring lane, the decision is straightforward: staying in the current lane. When traffic in the current lane is slower, the decision-making becomes hard to predict. We will focus on the later situation.

A. Human Decision-making in Two-lane Traffic

The considered two-lane scenario is shown in Fig. 3a, the green MV runs at speed v_c in the right lane. The driver has a best possible (desired) speed v_b in mind, but the current lane is blocked by an environment vehicle (red) driving at a speed $v_s \leq v_b$. In the other lane, there is a stream of traffic running at a faster speed $v_f \geq v_s$. The vehicle (blue), which approaches the MV longitudinally, has size V and is currently at speed v_f. The gap between the approaching vehicle and the MV currently is $d \geq 0$. The speeds v_s, v_c, v_f, the distance d, and the volume V can be observed by the MV and the approaching vehicle. Speed v_b can be inferred by the approaching vehicle, because it usually either is the speed limit or the speed of the fast lane traffic.

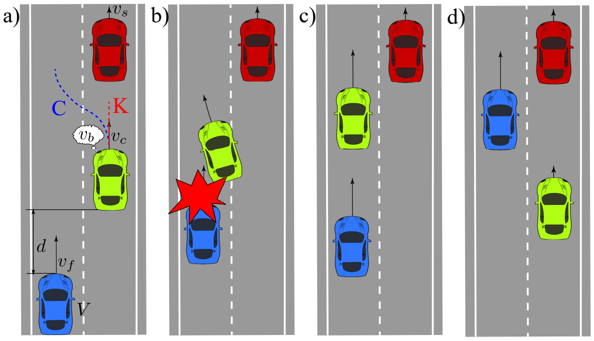

Fig. 3. The MV (green) in the traffic (a) can either change lanes or keep lanes. If it changes lanes, it may collide with the approaching vehicle (b) or safely merge in (c); otherwise, it has to slow down (d).

The driver in the MV has two options. He/she can either make a lane change (option C) or keep the current lane and yield to the approaching vehicle (option K). If the driver decides to change lanes, there may be two possible outcomes. The MV may rear collide with the approaching vehicle, as shown in Fig. 3b, or it may successfully merge in the new lane and run at its best speed v_b as shown in Fig. 3c. The blocking vehicle is assumed not to brake abruptly. If the driver chooses to yield, the MV has to slow down to match the speed of the blocking vehicle, letting the approaching vehicle pass (Fig. 3d). Which option will the driver take, and how is the decision made? This question can be answered by the human-decision model we describe next.

B. Regret Decision Model

The regret decision model deals with the two options (option C and K) which are formulated in terms of harms (costs) and probabilities. Choosing option C can either lead to no harm (Fig. 3c) with an objective probability p or lead to a cost of collision c_{collide} (Fig. 3b) with the probability $1-p$. Choosing option K is always safe. But deceleration generates some cost c_{slow} (Fig. 3d), with probability 1.

The regret decision model includes three components to address the range effect, the regret effect, and the probability weighting effect. The range effect claims that the utility u of a cost c is relevant to a reference cost $|\bar{c}|$ that defines the overall severity of the decision-making problem. In the lane-changing problem, a collision is the most critical event that may happen. Cost c_{collide} thus draws drivers' particular attention and is used as the reference to judge the situation. Hence, it is plausible that $|\bar{c}| \triangleq |c_{\text{collide}}|$. The utilities are defined as the scaled costs [14]: $u \triangleq c/|c_{\text{collide}}|$.

The anticipated emotion in the regret effect is triggered by the comparison of costs in different options. When using the regret decision model, it is convenient to use Table I.

TABLE I

OPTION C AND OPTION K REPRESENTED IN THE COMPARATIVE FORM

Events:		No collision	Collision
Joint probability:		p	$1-p$
Cost:	**Option C**	0	u_{collide}
	Option K	u_{slow}	u_{slow}

Each column is a comparison between utilities. The probability in a column is the joint probability for both utilities in the comparison to happen. The influence of regret emotion is nonlinear and the regret effect is depicted as [19],

$$q(\Delta u) \triangleq \sigma_1 \sinh(\sigma_2 \Delta u) + \sigma_3 \Delta u, \qquad (4)$$

where Δu is the difference between utilities, e.g., $u_{\text{collide}} - u_{\text{slow}}$ (column 4 in Table I). The parameters $\sigma_1, \sigma_2, \sigma_3 \geq 0$ are specific to individuals and will be determined (Section V). The linear part, i.e., $\sigma_3 \Delta u$, represents the objective (so-called rational) evaluation of utility differences; the nonlinear part, i.e., $\sigma_1 \sinh(\sigma_2 \Delta u)$, represents the influence of regret emotion triggered by cost differences.

The probability weighting effect is described by [20]:

$$w(p) \triangleq \exp(-\beta_1(-\log(p))^{\beta_2}), \qquad (5)$$

where parameters $\beta_1, \beta_2 \geq 0$ are specific to individuals and will be determined in Section V. Eqn. (5) either overweights or underweights probability p depending on its value.

Using $|\bar{c}|$, $q(\Delta u)$, and $w(p)$, the regret decision model calculates the net advantage of option C over option K:

$$e_{ck} \triangleq w(p)\, q(0 - u_{\text{slow}}) + (1 - w(p)) q(u_{\text{collide}} - u_{\text{slow}}). \quad (6)$$

When $e_{ck} > 0$, option C is chosen; otherwise option K is chosen.

C. Human Drivers' Lane-changing Decision-making

The driver in the MV, however, cannot directly observe the values of objective probability p and the objective costs c_{collide} and c_{slow}. What he/she can observe are the physical terms that define the traffic, including the speeds v_s, v_b, v_c, v_f, the distance d, and the volume V of the approaching vehicle. We should develop a bridge between the physical terms and the costs and probabilities.

Not all drivers have experienced collisions, but all can perceive the threat of a potential collision. We speculate the driver perceives the threat of the approaching vehicle to be proportional to its kinematic energy. This is because the vehicles of larger size are perceived of greater threats, which grows more than linearly when vehicle speed increases. Hence, we define $c_{\text{collide}}(v_f, V) \triangleq \lambda \rho V v_f^2/2$, where parameter $\lambda < 0$ is the subjective threat factor.

We also speculate the cost of slow-down c_{slow} is the time loss and is defined as $c_{\text{slow}}(v_d, v_o) \triangleq \frac{d_a}{v_d} - \frac{d_a}{v_o} = \tau_a \left(1 - \frac{v_d}{v_o}\right)$. We explain the variables v_d, v_o, d_a, and τ_a as follows. The speed v_d is the desired speed of the driver and v_o is the resulting speed of choosing an option. Note v_d may not necessarily equal the best speed v_b. Over some subjectively anticipated distance d_a, the different speeds generate the time loss $c_{\text{slow}}(v_d, v_o)$. One example in daily life is that we may often feel pressed in a traffic jam, because stopping implies the time loss is infinitely large. The anticipated distance d_a is defined as $d_a \triangleq \tau_a v_d$, where parameter $\tau_a \geq 0$ is the subjective anticipated impediment-free time.

The arguments v_d and v_o in cost $c_{\text{slow}}(v_d, v_o)$ take different values when the driver is evaluating the options by mentally simulating four possible traffic conditions.

1) If option C was chosen and no collision could happen, the desired speed for the driver was $v_d = v_b$, and the MV could run at $v_o = v_b$. Cost $c_{\text{slow}} = 0$.
2) If no collision could happen by choosing option C but the the driver chose option K, the desired speed was $v_d = v_b$ and the MV had to match the speed of the blocking vehicle with $v_o = v_s$. The slow-down cost became $c_{\text{slow}} = \tau_a - \frac{\tau_a v_b}{v_s}$.
3) If collision could happen by choosing option C and the driver chose option K. The driver knew option K was the only feasible option. The desired speed for the driver became $v_d = v_s$ and the MV ran at $v_o = v_s$. The slow-down cost is $c_{\text{slow}} = 0$.
4) If collision could happen when option C was taken but the driver already chose option C, this decision would result in collision, generating cost $c_{\text{collide}}(v_f, V)$.

The objective probability p to make a successful lane change is also hidden from the driver; what can be observed are the distance d and the relative speed $v_f - v_c$. The driver must estimate a probability $\hat{p}$ directly from the observed physical variables. We speculate that probability $\hat{p}$ strongly relates to the time-to-collision with the approaching vehicle t_c. We define $t_c \triangleq \frac{d}{v_f - v_c}$ if $v_c < v_f$ but we let $t_c = \infty$ otherwise. We further speculate that t_c is compared against a subjective time constant τ_s which represents the duration for safely and comfortably changing lanes. We let $\hat{p} = t_c/\tau_s$ if $0 \leq t_c \leq \tau_s$ and otherwise $\hat{p} = 1$.

TABLE II
OPTION C AND OPTION K IN TERMS OF PHYSICAL VARIABLES

Events:		No collision	Collision
Joint probability:		$\hat{p}$	$1 - \hat{p}$
Cost:	**Option C**	0	-1
	Option K	$\eta_1 \left(\frac{1}{v_f^2} - \frac{v_b}{v_s v_f^2}\right)$	0

The definitions of costs and probabilities and the definition of utilities, $u = c/|c_{\text{collide}}|$, express Table I in terms of the observable physical variables. If we further assume all the vehicles in the traffic are of the same size, as in Fig. 3, we can treat volume V as a constant parameter. To reduce the total amount of parameters, we define $\eta_1 \triangleq -\frac{2\tau_b}{\lambda \rho V} \geq 0$. The two options faced by the driver are in Table II. Based on 6, the decision of the driver is

$$e_{ck} = w(\hat{p})\, q\Big(\eta_1 \cdot \Big(\frac{v_b}{v_s v_f^2} - \frac{1}{v_f^2}\Big)\Big) + (1 - w(\hat{p}))q(-1), \quad (7)$$

where functions $w(\cdot)$ and $q(\cdot)$ are defined in Eqns. (5) and (4), respectively.

IV. Safe RL algorithm

In this section, we will focus on the development of a safe RL algorithm that integrates a safety supervisor. First, we define the underlying Markov decision process (MDP) by specifying the state representation, the action space, and the reward function for the reinforcement learning agent. Then, we demonstrate how to incorporate the human lane-changing model as safety supervisor into the RL algorithm.

A. Reinforcement Learning Agent

1) States Representation: In this work, we use the affordance indicator method [21] to encode the world representation. For a two-lane road as shown in Fig.1, we use the following indicators to represent the world.

- In front of the ego vehicle, the relative distances and the relative velocities of the vehicle in the right lane are d_{fr} and v_{fr}; and those in the left are d_{fl} and v_{fl}.
- To the rear of the ego vehicle, the relative distances and the relative velocities of the vehicle in the right lane are d_{rr} and v_{rr}; and those in the left are d_{rl} and v_{rl}.

Besides the above 8 affordance indicators, we also include the lateral position y, longitudinal velocity v_x, steering angle θ, and throttle value of the ego vehicle. A total of 12 affordance indicators are considered as the input to the RL agent. For generalization consideration, we normalize all indicators to range $[-1, 1]$.

2) Action Space: Laterally, the ego vehicle can take two actions: turning left or right. We assume the ego vehicle uses constant lateral speed $v_y \in \{-\bar{v}_y, 0, \bar{v}_y\}$ for lane changing. Longitudinally, there are three actions: decelerating, cruising, or accelerating, i.e., $a_x \in \{-\bar{a}_x, 0, \bar{a}_x\}$. The RL agent chooses one of the above actions each time. If the action is safe, it is sent to the low-level controller for generating control signals.

3) Reward Function: RL algorithms rely on reward functions to guide the agent to learn the desired policy. A reward/cost function penalizes the agent for choosing dangerous actions and rewards actions that bring efficiency, safety, and comfort. Here we adopt a linear-weighted reward function as

$$r = w_s r_s + w_v r_v + w_c r_c + w_h r_h, \quad (8)$$

where w_s, w_v, w_c, and w_h are weighting parameters for collision evaluation r_s, stable-speed evaluation r_v, lane-centering evaluation r_c, and headway evaluation r_h, respectively. Safety is the most important criteria, so we choose $w_s \gg w_v, w_c, w_h$.

The various performance evaluations are defined as follows. The collision evaluation r_s is set to -1 if collision happens, otherwise $r_s = 0$.

We encourage the ego vehicle to run at a stable speed. Hence, the stable-speed reward is

$$r_v \triangleq \begin{cases} \frac{v_x - \bar{v}_{\min}}{\bar{v}_{\text{target}} - \bar{v}_{\min}}, & \bar{v}_{\min} < v_x \leq \bar{v}_{\text{target}}; \\ \frac{\bar{v}_{\max} - v_x}{\bar{v}_{\max} - \bar{v}_{\text{target}}}, & \bar{v}_{\text{target}} < v_x \leq \bar{v}_{\max}; \\ 0, & v_x \leq \bar{v}_{\min} \text{ or } v_x > \bar{v}_{\max}; \end{cases} \quad (9)$$

where v_x is the current longitudinal speed of the vehicle, constant $\bar{v}_{\min}$, $\bar{v}_{\text{target}}$, and $\bar{v}_{\max}$ are minimum, target, and maximum speeds, respectively. Any speed larger than the maximum speed or less than the minimum speed is suppressed. Speeds $\bar{v}_{\min}$ and $\bar{v}_{\max}$ can be changed according to different traffic conditions. We want the ego vehicle to stay at the center of the road. So, we set the lane-centering evaluation r_c to -1 if distance between the ego vehicle and the lateral location of the center of the current lane is larger than a constant distance threshold $\bar{d}_c$.

Lastly, the ego vehicle should keep a safe time headway $\bar{T}_{\min}$ and distance $\bar{d}_s$. The headway evaluation is set to -1 if the safety requirements are not met, otherwise it is set to 0.

B. Safety Supervisor & Low-level Controller

1) Safety Supervisor: To train the ego vehicle safely and avoid frequent resets due to collisions, the safety supervisor uses the human lane-changing model. Since the physical variables observed by the driver in an MV (v_s, v_c, v_f, v_b, d) can also be measured by the AV through its sensing system, the lane-changing decisions of the driver can be predicted by the AV through the human lane-changing model. While there are different types of drivers, in this work we assume all drivers are homogeneous; we save the task of modelling and identifying types of various drivers as our future work.

Using these predictions, the safety supervisor can evaluate the consequences of actions from the RL agent. Regarding inter-vehicle consequences, within a short prediction time horizon t_{pred}, the future locations of an MV is estimated based on the predicted lane-changing decision and the current velocity of the MV. Likewise, the future locations of the ego vehicle within t_{pred} is also estimated based on its current action and velocity. A collision is predicted if the distance between the MV and the ego vehicle is within a predefined threshold $\bar{d}_s$ at any moment according to the projected trajectories. The action is labelled as unsafe. The safety supervisor reselects and replaces the action as follows (Lines 6-11 in Alg. 1).

- If the unsafe action is to change lanes, then the replacing action is to stay in the current lane instead.
- If the unsafe action is to speed up, then the replacing action is to slow down or remain current speed to avoid collisions.

Sometimes, the consequences involve only the ego vehicle, e.g., the ego vehicle chooses an action that pulls it off-road. In such cases, the safety supervisor predicts the trajectory and determines the ego vehicle will be off-track. Then, it provisions a default safe action, for instance, lane keeping.

The actions which are admitted by the safety supervisor are labelled as safe actions. They are sent to the lower-level controller, which controls the AV to interact with the environment and generate rewards according to Eqn. (8). A safe action, the states before and after the action, and the corresponding reward are considered a safe experience.

To fully utilize the experiences, an unsafe action and the associated experience is not simply discarded; we keep unsafe experiences by attaching appropriate penalties and recording the associated states. We store the unsafe experiences along with the safe experiences into the experience replay buffer. Every time we sample a mini-batch of experiences from the replay buffer, we use them to update our policy (Lines 12-25 in Alg. 1).

2) Low-level Controller: Once an action has been received from the high-level agent, the low-level controller controls the vehicle directly. This hierarchical design greatly reduces the training time compared to methods using agents to output control signals directly. For a low-level controller, classical feedback control methods, for instance, PID and MPC, are good choices. In this work, we use PIDs for both lateral (steering angle) and longitudinal controls (throttle).

C. The SafeRL Algorithm

Our SafeRL algorithm is shown in Alg. 1. We use an extended version of DQN called Double Deep Q-Network (DDQN), which mitigates the over-estimation problem of DQN [15]. Though we used the DDQN in our experiment, the proposed framework is suitable for other RL algorithms.

As parameters, M is the total number of training epochs, T is the total training time in each epoch, and K is the size of sampled experiences at each time. After initialization, line 5 shows the action selection using $\epsilon - greedy$ method [17]. Lines 6–11 illustrate how the safety supervisor works: every time after the RL agent chooses an action a_t, the supervisor checks whether this action is safe or not. It is replaced by a safe action $a_t{}'$ if it is determined unsafe. Line 8 stores the unsafe experience $(s_t, a_t, *, r_{col})$ to the replay buffer $\mathcal{D}$, where s_t is the previous state, $*$ means no next state because of collision, and r_{col} is the penalty of collision. After the agent takes the safe actions, lines 12–17 save the corresponding experiences with different rewards, r_{col} or r_{t+1}, to the replay buffer. Lines 18–25 update the Q-network. Line 18 samples a mini-batch of experiences from the replay buffer. Lines 19–22 estimate the value of the policy by the target network either as r_{t+1} or as

$$Y_{\text{target}}^{\text{DDQN}} \triangleq r_{t+1} + \gamma Q_{\phi_N}(s_{t+1}, \text{argmax}_{a'} Q_\phi(s_{t+1}, a')) \quad (10)$$

Line 24 calculates the gradients with respect to ϕ (Eqn. (2)) and updates the Q-network. Line 25 updates the target network every N steps and keeps it fixed at other steps.

V. Experiment, Results, and Discussion

In this section, we present the experiments conducted for identifying the parameters of the proposed human lane-changing decision model, as well as for evaluating our SafeRL methods. The experiments are performed on an open-source driving simulation platform CARLA [16].

A. Variables & Parameters in Human Lane-changing Model

To exploit the human drivers' regret decision model in AV learning, values of the objective variables, v_s, v_c, v_f, v_b, d, and the parameters, σ_1, σ_2, σ_3, η_1, β_1, β_2, τ_s, must be obtained. We assume human drivers are well informed about the driving-related objective variables, which are inputs to the decision-making model in Eqn. (7). On the other hand, the parameters are driver-specific constants that can be estimated through well-designed experiments.

Due to the space limit, here we only sketch the procedure of parameter estimation through experiments in this work. As a pilot study, one subject was invited to drive in a two-lane traffic scenario. The environment vehicles were set up with different speeds and distances to test the subject's lane-changing decision-making. The objective variables and the lane-changing decisions (option C or option K) were collected to construct a labelled data set. We then used logistic regression to fit the parameters to the data set. The accuracy of the fitted model on the data set is 83.33%. The parameters estimated are $\sigma_1 = 10.1795$, $\sigma_2 = 0.1130$, $\sigma_3 = 0.5108$, $\eta_1 = 152.5796\text{m}^2/\text{s}^2$, $\beta_1 = 9.9170$, $\beta_2 = 2.3812$, and $\tau_s = 3.5193$.

```
Algorithm 1 SafeRL for autonomous driving
Parameters: $M, T, K$
Initialize the Q-network, $Q_\phi$; the corresponding target network $Q_{\phi_N} \leftarrow Q_\phi$; and the safe replay buffer $\mathcal{D} \leftarrow \emptyset$
for $j = 0$ to $M - 1$ do
    Initialize $t \leftarrow 0$ and initial state $s(0) \leftarrow s_0$
    while $t < T$ do
        Select a random action with probability $\epsilon$, otherwise select action $a_t \leftarrow \text{argmax}_{a'} Q_\phi(s_t, a')$
        if $a_t$ is unsafe then
            Replace it with a safe action $a_t{}'$
            Store ($s_t$, $a_t$, *, $r_{col}$) to $\mathcal{D}$
        else
            $a_t{}' \leftarrow a_t$
        end if
        Perform $a_t{}'$ and observe $s_{t+1}$, $r_{t+1}$
        if termination then
            Store ($s_t$, $a_t$, *, $r_{col}$) to $\mathcal{D}$
        else
            Store ($s_t$, $a_t$, $s_{t+1}$, $r_{t+1}$) to $\mathcal{D}$
        end if
        Sample a mini-batch of size $K$ from $\mathcal{D}$
        if termination then
            $Y^{\text{DDQN}}_{\text{target}} \leftarrow r_{t+1}$
        else
            Update $Y^{\text{DDQN}}_{\text{target}}$ according to Eqn. (10)
        end if
        Update $Q_\phi$ according to Eqn. (2)
        Update $Q_{\phi_N} \leftarrow Q_\phi$ every $N$ steps
    end while
end for
```

B. Experimentation of SafeRL

We created a 400-meter standard two-lane road scenario. The driving scenario setup is shown in Fig. 3, where the blue one is the AV (ego) while the green and the red one are two MVs. The current speed v_c of the red MV is 5.56 m/s and its best speed v_b is also 5.56 m/s. On the other hand, the speed v_c of the green MV is 5.56 m/s while its v_b is 12.5 m/s. Since the green MV is behind the red, it may want to change lanes. The distance between the blue ego vehicle and the green MV, d, is 10 m. The MVs are assumed to use the regret-based human lane-changing model identified above for high-level decision-making. The ego vehicle was controlled by the SafeRL agent.

The trajectories of both the AV and MVs are predicted using the Euler's method:

$$v_x(t+1) = v_x(t) + a_x(t)\Delta t, \tag{11a}$$

$$x(t+1) = x(t) + v_x(t)\Delta t, \tag{11b}$$

$$y(t+1) = v_y(t) + v_y(t)\Delta t. \tag{11c}$$

For the AV, the longitudinal acceleration a_x and lateral velocity v_y are from the RL agent as discussed in Sec. IV-A.2. The SafeRL agent was trained on CARLA simulations for 1500 epochs. At the beginning of each training epoch, all the vehicles started from prespecified positions. An epoch will stop when a collision happens or the ego vehicle reaches the end of the road.

The Q-network Q_ϕ and the target network Q_{ϕ_N} are neural networks of two fully-connected layers and each layer has 64 nodes followed by ReLU activation. The networks are trained by Adam optimizer with learning rate $\alpha = 1e-4$. The exploration rate is continuously annealed from 1 to 0.05 over the first 1000 epochs and then kept constant for the remaining epochs. Actions are updated every two steps [21]. Parameters for the reward function, network training, and Euler's equation are as follows: $w_s = 2000$, $w_v = 10$, $w_c = 3$, $w_h = 15$, $\bar{d}_c = 0.5$ m, $\bar{d}_s = 18$ m, $\bar{T}_{\min} = 2$ s, $\bar{v}_{\text{target}} = 12.5$ m/s, $\bar{v}_{\min} = 5.56$ m/s, $\bar{v}_{\max} = 16.67$ m/s, $t_{\text{pred}} = 0.7$ s, $M = 1200$, $K = 256$, $\gamma = 0.99$, $\Delta t = 0.1$ s, $\bar{a}_x = 2$ m/s^2, and $\bar{v}_y = 1.8$ m/s.

C. Results and Discussion

We compare our proposed SafeRL algorithm against the conventional DDQN without safety supervision (we call it ConvRL thereafter). Fig. 4 shows the learning curves of the SafeRL and the ConvRL.

It is clear that SafeRL quickly leads to a reasonably good control policy with much larger rewards than ConvRL in the first 200 epochs. This is because the SafeRL is able to avoid collisions and offers more training experience without early epoch termination. Our experiments show there were no collisions during training when using SafeRL, whereas

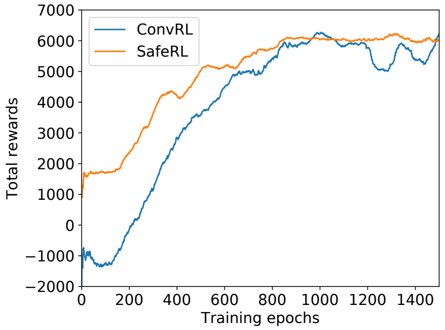

Fig. 4. Learning curves of SafeRL and ConvRL

using the ConvRL 8.6% epochs ended up in collision. Safety is guaranteed during training when using the SafeRL.

We should note that the safety supervisor in the SafeRL decreases the exploration space to avoid collisions. As a result, the available explorations of the SafeRL are less than the ConvRL. However, in Fig. 4, as the training progresses, the two curves converge almost at the same level around the 800th epoch, despite the fact the SafeRL has a smaller exploration space. This is reasonable because the optimal policy for choosing actions should be within the constraints set by the safety supervisor. The action space that SafeRL cannot explore is the part the AV should indeed avoid. Conversely, the unconstrained exploration by the ConvRL leads to slightly degraded evaluation performance.

Fig. 5 shows the policy performances evaluated every 50 epochs during training. The total rewards of an epoch of the SafeRL are always better. Its learning was smoother as evidenced by the constant improving rate. The initial setbacks of the ConvRL were due to the many collisions.

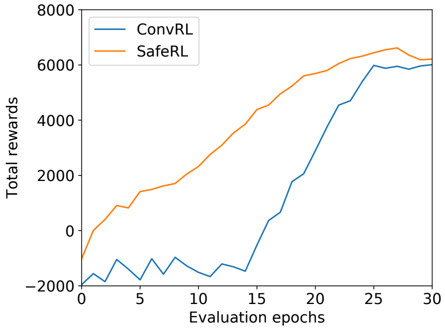

Fig. 5. Evaluating curve of SafeRL and ConvRL.

After both models converge after the 800th epoch in Fig. 4, the SafeRL is more stable. The ConvRL still has large fluctuations even after convergence. This is because collisions still happen. So even the completed trained ConvRL model still cannot be guaranteed to be collision-free. The SafeRL, nevertheless, ensures no collision for the trained model.

VI. Conclusions

We presented a framework for SafeRL to incorporate a safety supervisor which integrates a human lane-changing decision model. We developed a regret-based human lane-changing model and conducted pilot testing to show its validity. The model facilitates SafeRL learn its driving policy safely and stably. Experimental results showed our proposed SafeRL can reduce collisions to zero during training and implementation, without impinge the training performance.

In future work, we will make our method more robust to different traffic scenarios and further improve the learning efficiency for a faster learning rate.

References

[1] S. Le Vine, A. Zolfaghari, and J. Polak, "Autonomous cars: The tension between occupant experience and intersection capacity," *Transportation Research Part C: Emerging Technologies*, vol. 52, pp. 1–14, 2015.

[2] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1379–1384, IEEE, 2018.

[3] S. Nageshrao, E. Tseng, and D. P. Filev, "Autonomous highway driving using deep reinforcement learning," *CoRR*, vol. abs/1904.00035, 2019.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[5] Z. Li, U. Kalabić, and T. Chu, "Safe reinforcement learning: Learning with supervision using a constraint-admissible set," in *2018 Annual American Control Conference (ACC)*, pp. 6390–6395, IEEE, 2018.

[6] J. Wang, Q. Zhang, D. Zhao, and Y. Chen, "Lane change decision-making through deep reinforcement learning with rule-based constraints," *arXiv preprint arXiv:1904.00231*, 2019.

[7] C. C. MacAdam and G. E. Johnson, "Application of elementary neural networks and preview sensors for representing driver steering control behaviour," *Vehicle System Dynamics*, vol. 25, no. 1, pp. 3–30, 1996.

[8] J. A. Michon, "A critical view of driver behavior models: what do we know, what should we do?," in *Human behavior and traffic safety*, pp. 485–524, Springer, 1985.

[9] C. C. Macadam, "Understanding and modeling the human driver," *Vehicle system dynamics*, vol. 40, no. 1-3, pp. 101–134, 2003.

[10] T. A. Ranney, "Models of driving behavior: a review of their evolution," *Accident Analysis & Prevention*, vol. 26, no. 6, pp. 733–750, 1994.

[11] L. Jiang and Y. Wang, "A human-computer interface design for quantitative measure of regret theory," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 15–20, 2019.

[12] G. Loomes and R. Sugden, "Regret theory: An alternative theory of rational choice under uncertainty," *The economic journal*, vol. 92, no. 368, pp. 805–824, 1982.

[13] M. Hsu, I. Krajbich, C. Zhao, and C. F. Camerer, "Neural response to reward anticipation under risk is nonlinear in probabilities," *Journal of Neuroscience*, vol. 29, no. 7, pp. 2231–2237, 2009.

[14] K. Kontek and M. Lewandowski, "Range-dependent utility," *Management Science*, vol. 64, no. 6, pp. 2812–2832, 2017.

[15] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[17] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[18] M. Volodymyr, K. Koray, S. David, A. R. Andrei, and V. Joel, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[19] Z. Liao, L. Jiang, and Y. Wang, "A quantitative measure of regret in decision-making for human-robot collaborative search tasks," in *2017 American Control Conference (ACC)*, pp. 1524–1529, IEEE, 2017.

[20] D. Prelec *et al.*, "The probability weighting function," *ECONOMETRICA-EVANSTON ILL-*, vol. 66, pp. 497–528, 1998.

[21] N. Li, H. Chen, I. Kolmanovsky, and A. Girard, "An explicit decision tree approach for automated driving," in *ASME 2017 Dynamic Systems and Control Conference*, pp. V001T45A003–V001T45A003, American Society of Mechanical Engineers, 2017.