# **Behavior Planning For Connected Autonomous Vehicles Using Feedback Deep Reinforcement Learning**

## Songyang Han, Fei Miao<sup>1\*</sup>

<sup>1</sup>University of Connecticut 371 Fairfield Way, Department of Computer Science and Engineering Storrs, Connecticut 06269 songyang.han@uconn.edu, fei.miao@uconn.edu

#### **Abstract**

With the development of communication technologies, connected autonomous vehicles (CAVs) can share information with each other. We propose a novel behavior planning method for CAVs to decide actions such as whether to change lane or keep lane based on the observation and shared information from neighbors, and to make sure that there exist corresponding control maneuvers such as acceleration and steering angle to guarantee the safety of each individual autonomous vehicle. We formulate this problem as a hybrid partially observable Markov decision process (HPOMDP) to consider objectives such as improving traffic flow efficiency and driving comfort and safety requirements. The discrete state transition is determined by the proposed feedback deep Q-learning algorithm using the feedback action from an underlying controller based on control barrier functions. The feedback deep Q-learning algorithm we design aims to solve the critical challenge of reinforcement learning (RL) in a physical system: guaranteeing the safety of the system while the RL is exploring the action space to increase the reward. We prove that our method renders a forward invariant safe set for the continuous state physical dynamic model of the system while the RL agent is learning. In experiments, our behavior planning method can increase traffic flow and driving comfort compared with the intelligent driving model (IDM). We also validate that our method maintains safety during the learning process.

## Introduction

The development of the Dedicated Short-Range Communication (DSRC) technology (Tangade, Manvi, and Lorenz 2018) and 5G technology enables Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. The U.S. Department of Transportation (DOT) estimated that DSRC based V2V communication could potentially address up to 82% of crashes in the United States and prevent thousands of automobile crashes every year (Kenney 2011). When basic safety message (BSM), including current velocities and positions, is communicated for connected autonomous vehicles (CAVs), control approaches for scenarios

such as cross intersections or lane-merging (Rios-Torres and Malikopoulos 2017; Lee and Park 2012; Ort, Paull, and Rus 2018) have shown the advantage of information sharing.

When a lane-changing decision is already made, continuous state space controllers are designed for vehicles to merge into or leave the platoon (Li et al. 2017). However, existing control frameworks for CAVs, such as platooning (Liang, Mårtensson, and Johansson 2016), adaptive cruise control (ACC) (Nilsson et al. 2016), and cooperative adaptive cruise control (CACC) (Gritschneder, Graichen, and Dietmayer 2018) mainly focus on the controller design when a decision about lane changing or keeping has been provided (Guanetti, Kim, and Borrelli 2018). End-to-end neural network-based approaches to learn control signals for steering angle and acceleration directly, based on camera images has been designed (Codevilla et al. 2018; Bansal, Krizhevsky, and Ogale 2018; Amini et al. 2020). RL is also used based on the predicted trajectory (Henaff, LeCun, and Canziani 2019), to learn traffic control policies of a group of autonomous vehicles (Jang et al. 2019). However, these deep learning-based approaches only consider lane-keeping scenarios without lane-changing cases or safety guarantees.

To solve autonomous vehicle's challenges, one paramount concern raised by RL is how to ensure the safety of the learned driving policy. It has been proved that merely assigning a negative reward for accident trajectories cannot ensure driving safety-it enlarges the variance of the expected reward, which in turn prohibits the convergence of stochastic gradient descent algorithm for policy learning (Shalev-Shwartz, Shammah, and Shashua 2016). Instead, researches have used deep learning to learn a policy to mimic the control law of model predictive control (MPC) (Chen et al. 2018) or used MPC to generate samples for training in the guided policy search (Zhang et al. 2016). Paper (Cheng et al. 2019) adds control barrier functions (CBF) to the model-free RL to ensure end-to-end safe training. The main difference between this work and ours is: (1). We consider a hybrid system where the CBF and RL have different action and state space, while in previous work they share the same action and state space; (2). We have stronger motivation to apply model-free RL algorithms since the state transition is completely unknown, while in their work they have a rough system dynamic model at the beginning and the model is get-

<sup>\*</sup>This work was supported by NSF S&AS-1849246, NSF CPS-1932250 grants.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ting more accurate along with training. The other popular approach is to decompose the learning and control phases. Learning-based methods can make a high-level decision, including lane-changing or lane-keeping (Pan et al. 2017). However, splitting learning and control steps may cause inconsistency in control-based and learning-based decisions. Is there a better way to integrate the strength of both? Moreover, when a CAV gets extra knowledge about the environment via V2X communication, how to make tactical decisions such as whether to change lane or keep lane to improve traffic efficiency, to best utilize the shared information is still unsolved challenges for hybrid systems.

The existing multi-agent reinforcement learning literature (MARL) (Zhang, Yang, and Başar 2019) has not fully solved the CAV challenges yet-either how communication among agents will improve systems' performance or safe policy learning is not analyzed clearly. The recent advances in MARL include Multi-Agent Deep Deterministic Policy Gradient (MADDPG) and League training. MADDPG uses a centralized Q-function to alleviate the problem caused by the non-stationary environment (Lowe et al. 2017). League training is used in AlphaStar to avoid the cycles that appear in self-play and increase the diversity of the strategies (Vinyals et al. 2019). In this work, we focus on safe policy learning of each individual autonomous vehicle in the CAV system based on shared information, while the other vehicles (CAVs and non-CAVs) are considered as the environment that the ego vehicle can observe. MARL is out of the scope of this paper and will be considered as future work.

Hence, we design a learning algorithm that uses shared information to enhance behavior planing performance for future CAVs with feedback from controllers and safety guarantees. We show the benefit of V2V communication in improving both system efficiency and safety. We focus on the behavior planning challenge for CAVs, where we want to ensure the safety for both discrete and continuous dynamics when learning a good policy for a hybrid system. We model the problem as a hybrid partially observable Markov decision process (HPOMDP) with information sharing. The discrete state transition policy is explored by a RL algorithm and the continuous state is controlled under a CBF-based quadratic programming (QP). We add a feedback process to guarantee the action to be explored is safe and feasible. We also prove the CBF-QP can be used to render a forward invariant safe set even when there is a bounded noise of the continuous dynamics. From experiment results, the feedback DRL based behavior planning improves the system efficiency in terms of the traffic flow and driving comfort while ensuring safety during the entire learning process.

The main contributions of this work are:

- We propose a feedback deep Q-learning algorithm for a hybrid system to explore a policy for discrete state transitions. This algorithm has a feedback process to ensure the safety of both the discrete and continuous dynamics of the proposed HPOMDP.
- We prove that the quadratic programming with control barrier functions renders a forward invariant safe set for the continuous dynamics of the physical system. We also

- prove that the forward invariant set still exists when the continuous dynamics has a bounded noise.
- We utilize the proposed feedback deep Q-learning method to solve the behavior planning problem for CAVs given shared information. We show in experiments that the safety is guaranteed and the system efficiency is improved in terms of traffic flow and driving comfort.

## **Problem Formulation**

The V2V and V2I communications extend the information gathered by a single-vehicle further beyond its sensing system. This work addresses how to utilize shared information to make better behavior decisions such as when to change/keep lane for CAVs, considering safety guarantees of CAVs and the transportation system efficiency. The vehicles can only observe the state of the world by its own sensors (observation) or use the information shared by others (shared information). As shown in Fig. 1, we formulate our problem as a HPOMDP defined as follows.

**Definition 1** (Hybrid Partially Observable Markov Decision Process (HPOMDP)). A HPOMDP is a collection  $H = (S, A, \mathcal{O}, \mathcal{M}, \mathcal{R}, \pi_{\theta}, \mathcal{T}, \mathbf{o}, \mathbf{r}, \mathcal{X}, \mathcal{U}, \text{Init}, \mathcal{F}, \mathcal{C})$  where

- S is a set of states describing the world.
- A is a discrete set of actions including all behaviors. The action determines the discrete state of this hybrid system.
- O is a set of observations.
- $\mathcal{M}$  is a set of shared information.
- $\mathcal{R}$  is a set of reward.
- $\pi_{\theta}: \mathcal{O} \times \mathcal{M} \times \mathcal{A} \to [0,1]$  is a policy parameterized by  $\theta$ .
- $\mathcal{T}: \mathcal{S} \times \mathcal{A} \to \mathcal{S}$  is a state transition function.
- o: S → O is a private observation that the agent receives correlated with the state.
- $\mathbf{r}: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$  is reward assigned for each state-action pair
- $\mathcal{X} \subseteq \mathbb{R}^n$  is a compact set of continuous states of the underlying controller.
- $\mathcal{U}\subseteq\mathbb{R}^m$  is a compact set of the control inputs of the underlying controller.
- Init  $\subseteq \mathcal{A} \times \mathcal{X}$  is the set of initial behavior and underlying states.
- $\mathcal{F}: \mathcal{A} \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$  assigns to each  $a \in \mathcal{A}$  a continuous vector field  $\mathcal{F}(\cdot, a)$ , a function from  $\mathcal{X} \times \mathcal{U}$  to  $\mathcal{X}$ .
- C is a safe set that defines safe state  $x \in \mathcal{X}$ .

We consider the vector field  $\mathcal{F}(\cdot, a)$  in control affine form:

$$x_{t+1} = f(x_t) + g(x_t)u_t,$$
 (1)

with f and g locally Lipschitz,  $x \in \mathcal{X} \subset \mathbb{R}^n$  is the state and  $u \in \mathcal{U} \subset \mathbb{R}^m$  is the control input. Here we use a discretization of the continuous time state space system, because we consider a controller with 100 Hz frequency in our problem. We define forward invariant within a safe set and the control barrier function for guaranteeing safety of this system as follows.

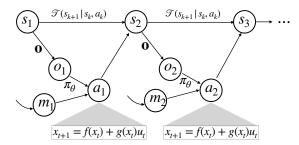


Figure 1: The hybrid partially observable Markov decision process.

**Definition 2** (Forward invariant (Ames et al. 2019)). Denote the safe set  $\mathcal{C}$  by the super-level set of a continuously differential function  $h: \mathbb{R}^n \to \mathbb{R}$ ,

$$C: \{x \in \mathcal{X} \in \mathbb{R}^n : h(x) \ge 0\}. \tag{2}$$

The set  $\mathcal{C}$  is forward invariant if  $\forall x_0 \in \mathcal{C}$ ,  $x_t \in \mathcal{C} \ \forall t \geq 0$ . The system is safe with respect to the set  $\mathcal{C}$  if the set  $\mathcal{C}$  is forward invariant, i.e., the system state x always remains within the safe set  $\mathcal{C}$ .

**Definition 3** (Control Barrier Function (CBF) (Cheng et al. 2019)). Given a set  $\mathcal{C} \subset \mathcal{X} \subset \mathbb{R}^n$ , the continuously differentiable function  $h: \mathcal{X} \to \mathbb{R}$  is a discrete-time control barrier function for dynamic system (1) if there exists  $\eta \in [0,1]$  such that for all  $x_t \in \mathcal{X}$ ,

$$\sup_{u_t \in \mathcal{U}} \left\{ h(x_{t+1}) + (\eta - 1)h(x_t) \right\} \ge 0, \tag{3}$$

where  $\eta$  represents how strongly the barrier function "pushes" the state inwards within the safe set C. When  $\eta = 0$ , this condition is the Lyapunov condition.

For example, if the state x of a vehicle is always within the safe set  $\mathcal{C}$ , it means this vehicle always stay in some bounding box/range and will not collide with other obstacles. The CBF is a condition to add the forward invariant property of the safe set C. Once  $x_0 \in \mathcal{C}$ , then  $x_t \in \mathcal{C} \ \forall t \geq 0$ .

We want to find a policy in the HPOMDP to maximize its total expected return  $G_k = \sum_{i=k+1}^K \gamma^{i-k-1} r_i$  where  $\gamma$  is a discount factor and K is the time horizon. At the same time, there exists a controller that generates control inputs  $u \in \mathcal{U}$  to guarantee the underlying continuous state  $x \in \mathcal{C}$  and the safe set  $\mathcal{C}$  is forward invariant. To achieve these goals, we use a feedback deep Q-learning algorithm to learn the action value function Q(o, m, a) and a feedback process to generate feedback action a' for the underlying controller to implement. The policy to generate the feedback action  $\pi'(a'|o,m)$  can maximize the total expected return while guaranteeing a' is safe, i.e., the underlying state  $x \in \mathcal{C}$  during the learning process. We use t and t to distinguish the underlying controller time step and reinforcement learning time step.

## **Feedback Deep Q-Learning**

We consider the challenge that the state transition function  $\mathcal{T}$  is not available. Traditional sample-based RL is good at

exploiting historical data to find a good policy. However, it does not guarantee either the optimality or the safety property of the converged policy. Different from the traditional DRL, we design a feedback controller to the continuous vector field  $\mathcal{F}(\cdot,a)$  to introduce a safe set for the continuous sate x. In this section, we first introduce the feedback does not lower the data efficiency. Then we introduce the quadratic programming (QP) formulated based on the CBF and prove how it induces a forward invariant set. We call the final policy a feedback policy that generates actions implemented to the environment, and the feedback policy  $\pi'(a'|o,m)$  has the ability to utilize the learned action-value function and ensure the safety during the learning process.

#### **Algorithm**

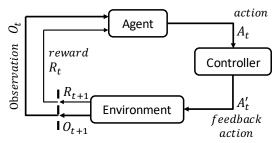


Figure 2: The agent-environment interaction in the feedback deep reinforcement learning.

The agent-environment interaction is shown in Fig. 2 for the feedback DRL. The DRL agent is like a smart toddler who is good at learning new knowledge but may do something dangerous, such as eating inedible things and playing sharp items. In this case, an adult (the feedback controller) needs to look after her and guide her as a guardian. For the hybrid system, when the RL algorithm is exploring different actions  $a \in \mathcal{A}$ , the underlying state may have  $x \in \mathcal{X} \land x \notin \mathcal{C}$ . For example, lane-changing may lead to collisions under some traffic scenarios. The underlying controller will monitor action a from the DRL agent and determine feedback action  $a' \in \mathcal{A}$  based on the current traffic condition.

The process to get the feedback action is shown in Fig. 3. Once the behavior policy (e.g., the  $\epsilon$ -greedy) selects an action, the underlying CBF-QP-based controller evaluates whether it is feasible to implement this action. This QP will be defined in detail in the following subsection. If the CBF-QP has a feasible solution, it means a is safe, then a'=a; if there is no solution, the controller will search other actions in  $\mathcal A$  in descending order according to their action-value Q and find a safe one. If all the actions in  $\mathcal A$  are not safe in the worst case, then the underlying controller will implement the Emergency Stop (ES) process.

Meanwhile, the transition experience is represented by (o, m, a', r, o', m') in our HPOMDP, where o and m is the current observation and shared information, a' is the feedback action, r is the reward, and o' and m' is the next observation and shared information. This algorithm is shown in Alg. 1. This algorithm learns an action-value function  $Q^*$  by minimizing the Bellman loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{o,m,a',r,o',m'}[(y - Q^*(o,m,a';\theta))^2], \quad (4)$$

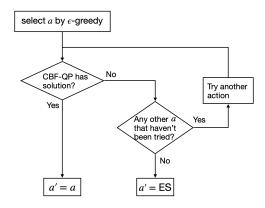


Figure 3: Get the feedback action.

where  $y = r + \gamma \cdot max_a \hat{Q}(o, m, a; \hat{\theta})$ . The corresponding policy can be derived as

$$\pi_{\theta}(a|o,m) = \underset{a}{\operatorname{argmax}} Q^{*}(o,m,a). \tag{5}$$

## Algorithm 1: Feedback deep Q-learning

- 1 Initialize replay memory D with a capacity N;
- 2 Initialize action-value function Q with random weights  $\theta$ ;
- 3 Initialize target action-value function  $\hat{Q}$  with  $\hat{\theta} = \theta$ ; 4 **for** episode = 1, E **do** 5 | Initialize  $o \in \mathcal{O} \land m \in \mathcal{M}$ ;

```
for k = 1, K do
6
           Choose a \in \mathcal{A} based on o, m and policy
7
             derived from Q-function (e.g., \epsilon-greedy);
            Give a to the feedback controller and get
8
             feedback action a':
            Take action a', get r, o', m';
9
           Store transition (o, m, a', r, o', m') in D;
10
           Sample random a minibatch from D;
11
            Set target y = r + \gamma \cdot max_a \hat{Q}(o, m, a; \hat{\theta});
12
           Do gradient descent on (y - Q(o, m, a'; \theta))^2
13
             with respect to \theta;
            Assign Q = Q for some time steps;
14
       end
15
16 end
```

Notably, the feedback does not lower the data efficiency for the RL. In traditional RL, the transition experience is represented by (o, a, r, o'). In feedback RL, the corresponding experience is represented by (o, a', r, o'), where a' is the feedback action executed by the feedback controller. Therefore, the feedback does not slow down the growth of the experience replay buffer.

#### **Controller And Safety**

We use standard basis vectors in  $\mathbb{R}^n$  to denote each action, where n is the number of total actions in the action space  $\mathcal{A}$ . The action  $a_n = (0, \dots, 0, 1, 0, \dots, 0)^\mathsf{T}$  is a vector equal to zero in every entry except for the nth entry, where it equals to one. In other words,  $(a_n)_i = 1$  when i = n and  $(a_n)_i = 0$ 

when  $i \neq n$ . The action  $a = \{a_n | a_n \in A\}$  is selected based on the policy  $\pi_{\theta}(a|o,m)$  learned by the Alg. 1.

Assemble the underlying controller's references for each action as a matrix as follows.

$$U = \begin{bmatrix} u_{11} & u_{21} & \cdots & u_{n1} \\ u_{12} & u_{22} & \cdots & u_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1m} & u_{2m} & \cdots & u_{nm} \end{bmatrix}, \tag{6}$$

where  $u_{n1}, u_{n2}, \dots, u_{nm}$  is the reference for action  $a_n$ . The reference of a controller is also called the setpoint. It is a target for the controller to track. The reference to track a trajectory for CAVs will be introduced in the next section.

In experiment, we consider an affine barrier function of form  $h = p^T x + q$ ,  $(p \in \mathbb{R}^n, q \in \mathbb{R})$ . This restriction means the set  $\mathcal{C}$  is a polytope constructed by the intersecting of half spaces, since the safe area for a vehicle is usually represented by a bounding box (Cesari et al. 2017). By applying a Frenét frame, the curve road can also be regarded as a straight road (Werling et al. 2010).

We formulate the following Quadratic Programming (QP) based on the CBF condition (3) that can be solved efficiently at each time step:

$$\underset{u_t}{\operatorname{argmin}} \quad \|u_t - U \cdot a\|^2$$
s.t. 
$$p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + q \ge (1 - \eta) h(x_t).$$
(7)

**Lemma 1.** If there exists  $\eta \in [0,1]$  such that there exists a solution to the QP (7) for the dynamical system (1) for all  $x_t \in \mathcal{C}$  (defined in (2)), then the controller derived from (7) renders set  $\mathcal{C}$  forward invariant.

*Proof.* For any  $x_t \in \mathcal{C}$ , we have  $h(x_t) \geq 0$  according to the definition (2). Therefore,

$$h(x_{t+1}) = p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + q$$
  
 
$$\geq (1 - \eta) h(x_t) \geq 0.$$
 (8)

Thus,  $x_{t+1} \in \mathcal{C}$  and  $\mathcal{C}$  is forward invariant.

In fact, we can relax the constraint in (7), and the forward invariant property holds for a larger set. Consider this QP:

$$\underset{u_t,\zeta}{\operatorname{argmin}} \quad \|u_t - U \cdot a\|^2 + M\zeta$$
s.t. 
$$p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + q \ge (1 - \eta) h(x_t) - \zeta$$

$$\zeta \ge 0,$$
(9)

where M is a large constant.

**Theorem 1.** If there exists  $\eta \in [0,1]$  and  $Z \geq 0$  such that there exists a solution to the QP (9) for the dynamical system (1) for all  $x_t \in \mathcal{C}$  with  $\zeta \leq Z$ , then the controller derived from (9) renders set  $\mathcal{C}' : \{x \in \mathbb{R}^n : h'(x) = h(x) + \frac{Z}{\eta} \geq 0\}$  forward invariant.

*Proof.* From  $Z \ge \zeta \ge 0 \ge -\eta h(x_t)$ , we have  $h(x_t) + \frac{Z}{\eta} \ge 0$ . Thus,  $x_t \in \mathcal{C}'$  with  $h'(x_t) \ge 0$ . Also, we have

$$h(x_{t+1}) = p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + q$$
  
 
$$\geq (1 - \eta) h(x_t) - \zeta \geq (1 - \eta) h(x_t) - Z,$$
 (10)

$$h(x_{t+1}) + \frac{Z}{\eta} \ge (1 - \eta) \left[ h(x_t) + \frac{Z}{\eta} \right],$$
 (11)  
 $h'(x_{t+1}) \ge (1 - \eta)h'(x_t) \ge 0.$ 

Thus,  $x_{t+1} \in \mathcal{C}'$  and  $\mathcal{C}'$  is forward invariant. Note that it simplifies to Lemma 1 with set C = C' when Z = 0.

The value of Z denotes how large the CBF condition (3) is violated for the original  $h(x_t)$ . In this case, the safety condition should be formulated according to the set C', which is forward invariant.

When dynamic system has noise in the model in reality:

$$x_{t+1} = f(x_t) + g(x_t)u_t + w_t. (12)$$

We can also find a forward invariant set as long as the noise is bounded, i.e.,  $||w_t|| \leq W$ . Consider the following QP:

$$\underset{u_t,\zeta}{\operatorname{argmin}} \quad \|u_t - U \cdot a\|^2 + M\zeta$$
s.t. 
$$p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + q - p^{\mathsf{T}} W \cdot \mathbf{1} \qquad (13)$$

$$\geq (1 - \eta) h(x_t) - \zeta$$

$$\zeta > 0.$$

**Theorem 2.** If there exists  $\eta \in [0,1]$  and  $Z \geq 0$  such that there exists a solution to the QP (13) for the dynamical system (12) for all  $x_t \in \mathcal{C}$  with  $\zeta \leq Z$  and bounded noise  $||w_t|| \leq W$ , then the controller derived from (13) renders set  $C': \{x \in \mathbb{R}^n : h'(x) = h(x) + \frac{Z}{n} \geq 0\}$  forward invari-

*Proof.* Similar with the Theorem 2,  $x_t \in \mathcal{C}'$  with  $h'(x_t) \geq$ 0. Also, we have

$$h(x_{t+1}) = p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + p^{\mathsf{T}} w_t + q$$

$$\geq p^{\mathsf{T}} f(x_t) + p^{\mathsf{T}} g(x_t) u_t + q - p^{\mathsf{T}} W \cdot \mathbf{1}$$

$$\geq (1 - \eta) h(x_t) - \zeta \geq (1 - \eta) h(x_t) - Z,$$
(14)

$$h(x_{t+1}) + \frac{Z}{\eta} \ge (1 - \eta) \left[ h(x_t) + \frac{Z}{\eta} \right].$$
 (15)

Thus,  $x_{t+1} \in \mathcal{C}'$  and  $\mathcal{C}'$  is forward invariant.

#### **Behavior Planning of CAVs**

In this section, we apply the feedback RL to the behavior planning problem of CAVs. We consider a scenario with autonomous vehicles running on a multi-lane freeway, as shown in Fig. 4.

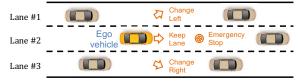


Figure 4: The scenario of a 3-lane freeway. The ego vehicle's behavior is determined based on observation and shared information.

The behavior planning process based on Alg.1 is shown in Fig. 5 for the CAV hybrid system. The discrete state transition policy  $\pi_{\theta}(a|o,m)$  is learned by a RL agent with 2 Hz;

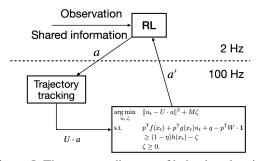


Figure 5: The concept diagram of behavior planning. the underlying continuous state is controlled by a CBF-QP

(13) with 100 Hz. After the RL selects an action a, the control reference will be calculated by trajectory tracking introduced in the following subsection. The CBF-QP will find possible feasible control inputs to track the trajectory. If a is safe, then the feedback action a' = a; if not, the feedback action will be searched in descending order according to their action-value. If  $\forall a \in \mathcal{A}$ , there is no feasible solution of the CBF-QP, then the vehicle will implement the Emergency Stop (ES) process.

## Physical Dynamic Model of an Ego Vehicle

The physical dynamics of an ego vehicle is described by a kinematic bicycle model that achieves a balance between accuracy and complexity (Brito et al. 2019). The discrete-time equations of this model can be obtained by applying an explicit Euler method with a sampling time  $T_s$  as follows:

$$x_{t+1} = \begin{bmatrix} x_1(t) + x_4(t)cos(x_3(t))T_s \\ x_2(t) + x_4(t)sin(x_3(t))T_s \\ x_3(t) \\ x_4(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{x_4(t)T_s}{L} & 0 \\ 0 & T_s \end{bmatrix} u_t,$$
(16)

where  $x = [p_x, p_y, \psi, v]^\mathsf{T} \in \mathcal{X}$  includes the two coordinates of the Center of Gravity (CoG), its orientation and its velocity at the CoG. The input  $u = [tan\delta, \dot{v}] \in \mathcal{U}$  are the tangent of the steering angle and its acceleration. The parameter L is the distance between the front and rear axle. Equation (16) is the vector field  $\mathcal{F}(\cdot, a)$ .

Trajectory Tracking. For each action, we can use stateof-the-art trajectory planning methods to generate trajectories  $\sigma = [\sigma_1, \sigma_2]^\mathsf{T} = [p_x, p_y]^\mathsf{T}$ , such as potential fields, cell decomposition, MPC (Dixit et al. 2018; Cesari et al. 2017). One can use endogenous transformation to compute the controller's references to track the trajectories for each action:

$$u_1 = \frac{\dot{\sigma}_1 \ddot{\sigma}_1 + \dot{\sigma}_2 \ddot{\sigma}_2}{\sqrt{\dot{\sigma}_1^2 + \dot{\sigma}_2^2}}, \quad u_2 = \frac{\dot{\sigma}_1 \ddot{\sigma}_2 - \dot{\sigma}_2 \ddot{\sigma}_1}{(\dot{\sigma}_1^2 + \dot{\sigma}_2^2)^{\frac{3}{2}}}.$$
 (17)

Plugging them in equation (6), the references are generated for the CBF-QP (13).

#### **RL Model**

**Action.** The set of actions A includes:

• Keep Lane (KL): The ego vehicle stays in the current lane with this action. It may either accelerate or decelerate.

- Change Left (CL): The ego vehicle changes to a neighbor lane on the left. The incentive to change lane is to achieve higher speed or to avoid obstacles/traffic jams.
- Change Right (CR): The ego vehicle changes to a neighbor lane on the right.

This action set is defined for the freeway scenario in Fig. 4. Our method can also apply for other scenarios by extending the action set, for example, adding turn-left and turn-right at intersections.

**Observation.** The ego vehicle observes the environment by its own sensors. The observation set  $\mathcal{O}$  includes observations from its own sensors, like radar, camera, and liar. Besides its sensors, we also add an observation for lane-changing frequency to count the total lane-changing times within a fixed time window. The motivation is passengers may feel uncomfortable if a vehicle changes lane frequently.

**Definition 4** (Observation for lane-changing frequency). The observation for lane-changing frequency is

$$O(k) = -\sum_{i=1}^{K_f} I_{change}(k-i),$$
 (18)

where k is the current time instant,  $K_f$  is a constant determining the window size of  $[k-K_f,k-1]$  and the indicator

$$I_{change}(i) = \begin{cases} 1, & \text{if lane-changing from time i;} \\ 0, & \text{otherwise.} \end{cases}$$
 (19)

**Shared Information.** For CAVs, the ego vehicle can also gather information from its peers by communication. According to the current development of BSM and DSRC security, the message can be both authenticated and encrypted (Tangade, Manvi, and Lorenz 2018). Hence, in this work, we assume that vehicular communication is true information that is not manipulated by attackers. We assume autonomous vehicles to share information with their  $\epsilon$ -neighbors defined as follows.

**Definition 5** ( $\epsilon$ -neighbors). One vehicle j is one  $\epsilon$ -neighbor of a vehicle i if  $\|p_t^i-p_t^j\|\leq \epsilon$ , where i and j are vehicles' indexes,  $p_t^i$  and  $p_t^j$  represent the positions of vehicle i and j at time t respectively, and  $\epsilon$  is a constant. The set that includes all the  $\epsilon$ -neighbors of the vehicle i except for i itself is denoted as  $\mathcal{N}_i(\epsilon)$ . This set is called vehicle i's  $\epsilon$ -neighbors.

In this application, we consider autonomous vehicles to share their current velocity  $v_t$  and lane number  $l_t$  (labeled as 1, 2, 3) with  $\mathcal{N}_i(\epsilon)$ . When the reinforcement learning is making decision at step k, the ego vehicle uses the most recently information received from other neighbors. The nearest information determine  $v_k$  and  $l_k$ . One problem is the total number of shared information increases proportionally to  $|\mathcal{N}_i(\epsilon)|$ . To reduce the curse of dimensionality, we define the shared information for the average velocity of lane #l as:

**Definition 6** (Shared information for the average velocity). The shared information for the average velocity of lane #l is

$$M(k,l) = \frac{1}{N} \sum_{i:i \in \mathcal{N}_l} v_k^i, \tag{20}$$

where  $\mathcal{N}_l = \{i | i \in \mathcal{N}_i(\epsilon), l_k^i = l\}, N = |\mathcal{N}_l|, l_k^i \in \{1, 2, 3\}, v_k^i$  is the velocity of the vehicle i at time k.

This function is the average velocity of CAVs on lane # l. It helps to avoid unnecessary lane-changing. For example, even though the close vehicles on the neighbor lane have a higher speed than the ego vehicle, there may be a traffic jam in front of them. The ego vehicle could not observe this traffic jam because it cannot see through its neighbors. In this case, there is no need to change lane.

Based on this definition, the shared information set  $\mathcal{M} = \{M(k, l-1), M(k, l), M(k, l+1)\}$ , where the lane #(l-1) and #(l+1) is the left and right neighbor lane respectively.

**Reward.** After taking an action, the ego vehicle gets a reward from the environment. We consider a behavior planning policy to improve two system-level evaluation criteria: traffic flow F and driving comfort C. When there is a central cloud center to collect information from all the agents, a global reward function can be used for the MARL (Wai et al. 2018). Therefore, the reward function is defined as a weighted sum of the above two criteria:

**Definition 7** (Reward function). The reward function is

$$\mathbf{r}(s,a) = w \cdot F + C,\tag{21}$$

where  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ , w is a trade-off weight.

• Traffic Flow: The traffic flow reflects the quality of the road throughout with respect to the traffic density. The traffic density  $\rho$  is the ratio between the total number of vehicles and the road length. It is calculated as

$$F = \rho \times \bar{v},\tag{22}$$

where  $\bar{v}$  is the average velocity of all the vehicles (Rios-Torres and Malikopoulos 2018).

 Driving Comfort: The driving comfort of a road segment is defined to be the average driving comfort of all the vehicles on this segment. The driving comfort of a single-vehicle is related to its acceleration and driving behavior. Define the driving comfort for the vehicle i's acceleration v<sub>t</sub> at time t and its action a ∈ A as follows:

$$C_{single}(\dot{v}_t^i, a) = \begin{cases} 3, & \text{if } \dot{v}_t^i < \Theta \text{ and } a = KL; \\ 2, & \text{if } \dot{v}_t^i \ge \Theta \text{ and } a = KL; \\ 1, & \text{if } a = CL/CR; \\ 0, & \text{if in } ES. \end{cases}$$
 (23)

where  $\Theta$  is a predefined acceleration threshold. Therefore, driving comfort (for the freeway) is calculated as

$$C = \frac{1}{N \cdot T} \sum_{i} \sum_{t} C_{single}(\dot{v}_{t}^{i}, a), \tag{24}$$

where N is the total number of all the vehicles.

#### **Experiment**

In experiments, vehicles are randomly scattered on different lanes of a 1000-long freeway as their initial positions. The total number of vehicles ranges from 100 to 900. All vehicles loop on this freeway. For different traffic densities,

experiments run for 4000 time steps. Traffic flow and driving comfort are evaluated based on the statistics in the last 1000 time steps. Also, the experiment runs 30 times under different initialization for each traffic density.

## Freeway Scenario

We compare the traffic flow and driving comfort under different traffic densities between the feedback RL agent using Alg. 1 and an intelligent driving model (IDM). In this IDM, the vehicle's acceleration is a function of its current speed, current and desired spacing, and the leading and following vehicles' speed (Talebpour and Mahmassani 2016). Building on top of these IDM agents, we add lane-changing functionality using the gap acceptance method by considering the positions of the leading vehicles in the current and target lane and the following vehicle on the target lane (Butakov and Ioannou 2014).

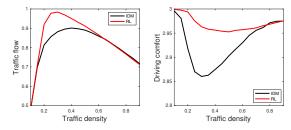


Figure 6: The comparison between feedback RL and IDM.

As shown in Fig. 6, the feedback RL agent gets both larger traffic flow and better driving comfort when traffic density  $\rho$  is low. When  $\rho$  grows, the result of the RL agent gets worse, but it is still comparable with the IDM. When the road is saturated, lane-changing tends to downgrade passengers' comfort but cannot bring higher speed. Consequently, a better choice is to keep a lane when  $\rho$  is high, and there is no significant difference between the RL and the IDM.

Table 1 shows the number of collisions avoided by the feedback process. It is also the number of actions altered by the feedback where a'!=a. This number is averaged over the last 10 episodes, which has a maximum epoch of 2000. Without the feedback process, each episode terminates very early in the first few epochs because of a collision. For example, when  $\rho=0.1$ , one episode can be lucky to run for 18 epochs after 2000 episodes. With the feedback process, each episode can run safely for the maximum epoch.

Table 1: Total number of collisions avoided by the feedback process.

$\overline{\rho}$	0.1	0.2	0.3	0.4	0.5
#	4416	9510	21897	43491	61689
ρ	0.6	0.7	0.8	0.9	-
#	91154	133135	191404	242976	-

The average headway of all the vehicles is shown in the first 500 epochs of one episode when  $\rho=0.6$  in Fig. 7. We see that the headway of vehicles is always greater than 0 in the feedback RL with the minimum value of 3.33. Nevertheless, the vanilla RL without the feedback process is likely to have a negative headway, which will cause collisions in

reality.

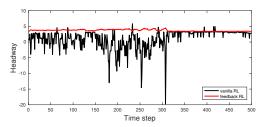


Figure 7: The comparison between feedback RL and vanilla RL without feedback.

#### **Road Closure Scenario**



Figure 8: The road closure scenario with two closures on lane #1.

This experiment shows a case study where there are two road closures on one lane. As shown in Fig. 8, the road closures may be caused by accident, or they are under construction. These road closures both have a length of 200. We want to test whether the learning algorithm can work in a more complicated scenario. CAVs are assumed to know the position of the road closures when it reaches the  $\epsilon$ -neighbor range of these segments. They can share this information with its  $\epsilon$ -neighbors (add as additional information to be shared in this case) or gather this information through V2I communication from infrastructure.

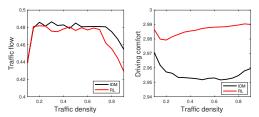


Figure 9: Feedback RL in the road closure scenario.

The result is shown in Fig. 9. The feedback RL agent gets a comparable traffic flow while improving driving comfort a lot. Though the feedback RL agent does not use the lane with closures to overtake and get higher speed, it avoids massive acceleration to provide better driving comfort.

#### Conclusion

This paper focuses on the behavior planning challenge for CAVs, and designs a novel policy learning algorithm that considers feedback from controllers and safety requirements with shared information to enhance operational performance. We model a hybrid partially observable Markov decision process (HPOMDP) for this hybrid system and design a feedback DRL framework to explore a safe behavior planning policy for both discrete and continuous dynamics. Our method combines the advantages of both DRL and CBF

and guarantees a safe learning process for this physical system. Though our method for the HPOMDP is proposed for the CAVs, it is applicable to other hybrid systems where the discrete transition policy is explored by the RL and the underlying controller renders a forward invariant safe set. More applications using this method can be investigated in the future. The recent advances in multi-agent RL can be used to further improve the result, like MADDPG and league training.

## References

- Ames, A. D.; Coogan, S.; Egerstedt, M.; Notomista, G.; Sreenath, K.; and Tabuada, P. 2019. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, 3420–3431. IEEE.
- Amini, A.; Gilitschenski, I.; Phillips, J.; Moseyko, J.; Banerjee, R.; Karaman, S.; and Rus, D. 2020. Learning Robust Control Policies for End-to-End Autonomous Driving From Data-Driven Simulation. *IEEE Robotics and Automation Letters* 5: 1–1. doi:10.1109/LRA.2020.2966414.
- Bansal, M.; Krizhevsky, A.; and Ogale, A. S. 2018. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. *CoRR* abs/1812.03079. URL http://arxiv.org/abs/1812.03079.
- Brito, B.; Floor, B.; Ferranti, L.; and Alonso-Mora, J. 2019. Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments. *IEEE Robotics and Automation Letters* 4(4): 4459–4466.
- Butakov, V. A.; and Ioannou, P. 2014. Personalized driver/vehicle lane change models for ADAS. *IEEE Transactions on Vehicular Technology* 64(10): 4422–4431.
- Cesari, G.; Schildbach, G.; Carvalho, A.; and Borrelli, F. 2017. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent Transportation Systems Magazine* 9(3): 23–35.
- Chen, S.; Saulnier, K.; Atanasov, N.; Lee, D. D.; Kumar, V.; Pappas, G. J.; and Morari, M. 2018. Approximating explicit model predictive control using constrained neural networks. In 2018 Annual American control conference (ACC), 1520–1527. IEEE.
- Cheng, R.; Orosz, G.; Murray, R. M.; and Burdick, J. W. 2019. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3387–3395.
- Codevilla, F.; Miiller, M.; López, A.; Koltun, V.; and Dosovitskiy, A. 2018. End-to-end driving via conditional imitation learning. In 2018 IEEE International Conference on Robotics and Automation (ICRA), 1–9. IEEE.
- Dixit, S.; Fallah, S.; Montanaro, U.; Dianati, M.; Stevens, A.; Mccullough, F.; and Mouzakitis, A. 2018. Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects. *Annual Reviews in Control* 45: 76–86.

- Gritschneder, F.; Graichen, K.; and Dietmayer, K. 2018. Fast Trajectory Planning for Automated Vehicles using Gradient-based Nonlinear Model Predictive Control. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 7369–7374. IEEE.
- Guanetti, J.; Kim, Y.; and Borrelli, F. 2018. Control of connected and automated vehicles: State of the art and future challenges. *Annual Reviews in Control* 45: 18 40. ISSN 1367-5788. doi:https://doi.org/10.1016/j.arcontrol.2018.04. 011. URL http://www.sciencedirect.com/science/article/pii/S1367578818300336.
- Henaff, M.; LeCun, Y.; and Canziani, A. 2019. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. In 7th International Conference on Learning Representations (ICLR).
- Jang, K.; Vinitsky, E.; Chalaki, B.; Remer, B.; Beaver, L.; Malikopoulos, A. A.; and Bayen, A. 2019. Simulation to Scaled City: Zero-Shot Policy Transfer for Traffic Control via Autonomous Vehicles. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '19, 291–300. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362856. doi:10.1145/3302509.3313784. URL https://doi.org/10.1145/3302509.3313784.
- Kenney, J. B. 2011. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE* 99(7): 1162–1182. ISSN 0018-9219. doi: 10.1109/JPROC.2011.2132790.
- Lee, J.; and Park, B. 2012. Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment. *IEEE Transactions on Intelligent Transportation Systems* 13(1): 81–90. ISSN 1524-9050. doi:10.1109/TITS.2011.2178836.
- Li, B.; Zhang, Y.; Ge, Y.; Shao, Z.; and Li, P. 2017. Optimal control-based online motion planning for cooperative lane changes of connected and automated vehicles. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3689–3694. IEEE.
- Liang, K. Y.; Mårtensson, J.; and Johansson, K. H. 2016. Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency. *IEEE Transactions on Intelligent Transportation Systems* 17(4): 1051–1061. ISSN 1524-9050. doi:10.1109/TITS. 2015.2492243.
- Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, 6379–6390.
- Nilsson, P.; Hussien, O.; Balkan, A.; Chen, Y.; Ames, A. D.; Grizzle, J. W.; Ozay, N.; Peng, H.; and Tabuada, P. 2016. Correct-by-Construction Adaptive Cruise Control: Two Approaches. *IEEE Transactions on Control Systems Technology* 24(4): 1294–1307. ISSN 1063-6536. doi:10.1109/TCST.2015.2501351.
- Ort, T.; Paull, L.; and Rus, D. 2018. Autonomous vehicle navigation in rural environments without detailed prior

- maps. In 2018 IEEE International Conference on Robotics and Automation (ICRA), 2040–2047. IEEE.
- Pan, X.; You, Y.; Wang, Z.; and Lu, C. 2017. Virtual to real reinforcement learning for autonomous driving. *arXiv* preprint arXiv:1704.03952.
- Rios-Torres, J.; and Malikopoulos, A. A. 2017. A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps. *IEEE Transactions on Intelligent Transportation Systems* 18(5): 1066–1077. ISSN 1524-9050. doi:10.1109/TITS.2016. 2600504.
- Rios-Torres, J.; and Malikopoulos, A. A. 2018. Impact of partial penetrations of connected and automated vehicles on fuel consumption and traffic flow. *IEEE Transactions on Intelligent Vehicles* 3(4): 453–462.
- Shalev-Shwartz, S.; Shammah, S.; and Shashua, A. 2016. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.
- Talebpour, A.; and Mahmassani, H. S. 2016. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies* 71: 143–163.
- Tangade, S.; Manvi, S. S.; and Lorenz, P. 2018. Decentralized and scalable privacy-preserving authentication scheme in VANETs. *IEEE Transactions on Vehicular Technology* 67(9): 8647–8655.
- Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in Star-Craft II using multi-agent reinforcement learning. *Nature* 575(7782): 350–354.
- Wai, H.-T.; Yang, Z.; Wang, Z.; and Hong, M. 2018. Multiagent reinforcement learning via double averaging primal-dual optimization. In *Advances in Neural Information Processing Systems*, 9649–9660.
- Werling, M.; Ziegler, J.; Kammel, S.; and Thrun, S. 2010. Optimal trajectory generation for dynamic street scenarios in a frenét frame. In 2010 IEEE International Conference on Robotics and Automation, 987–993. IEEE.
- Zhang, K.; Yang, Z.; and Başar, T. 2019. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *arXiv preprint arXiv:1911.10635*.
- Zhang, T.; Kahn, G.; Levine, S.; and Abbeel, P. 2016. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In 2016 IEEE International Conference on Robotics and Automation (ICRA), 528–535. ISSN null. doi:10.1109/ICRA.2016.7487175.