Hybrid Reinforcement Learning for Eco-Driving of Vision-based Connected and Automated Vehicle under Signalized Intersection

Zhengwei Bai
School of Electronic and Information Engineering, Beijing Jiaotong University
No.3 Shangyuancun, Haidian district, Beijing, China, 100044
Tel: +86 188 1309 6440; Email: zwbai@bjtu.edu.cn

Peng Hao, Ph.D.
Center for Environmental Research & Technology, University of California, Riverside
1084 Columbia Avenue, Riverside, CA 92507, USA
Tel: +1 951 781 5777, Fax: +1 951 781 5790, Email: haop@cert.ucr.edu

Word Count: 6489 words + 4 tables (250 words per table) = 7489 words

Submitted: July 30, 2019

ABSTRACT

The autonomous driving strategy is of great importance because it plays a significant role in managing the driving performance, such as travel time and energy consumption. In recent decades, various state-of-art driving strategy models are conducted from the time, energy or safety perspectives. Specifically, for energy efficiency, most of the conventional eco-driving strategies are rule-based or model-based. Although these traditional model can generate precise control trajectories, their capabilities will be greatly constrained in the complex real-world traffic environment, like a signalized intersection with mixed traffic. In this study, we proposed a hybrid reinforcement learning (HRL) framework which combines the manual-designed rule strategy and the deep reinforcement learning (deep RL). In addition, we implemented vision-perceptive methods to traditional vehicle-to-infrastructure (V2I) based connected and automated vehicle (CAV), which endows the CAV with the ability to interact with mixed traffic. The HRL framework is composed of three main parts: a rule-based driving manager that manages the collaboration between rule strategy and RL policy; a multi-stream neural network that extracts the hidden features of vision and V2I information; and a deep RL-based policy network that generate both longitudinal and latitudinal eco-driving actions. In order to evaluate the method, we developed a Unity-based simulator and designed an intersection scenario which includes mixed human-driven vehicles. Moreover, several baselines are implemented to compare with the designed method and numerical experiments are conducted to test the performance of the HRL model. The experiments show that the HRL method can save 1.2%-6.9% travel time and reduce 12.25%-47.5% energy consumption comparing with several baselines.

Keywords: Hybrid reinforcement learning, Connected and automated vehicle, Eco-driving, Signalized Intersection

INTRODUCTION AND MOTIVATION

In recent decades, the transportation system has developed tremendously, which leads people to a higher standard of life. Due to the development of vehicle technology and economy, vehicle ownership is increasing drastically. It is estimated that the ownership of the motor vehicle is over 1 billion in the world and will be doubled in recent decades (*1*). The intensive transportation activity, however, has brought various serious problems. For instance, from the environment perspective, it is reported that transportation sector accounts for nearly 27% of total greenhouse gas (GHG) emission in the U.S where motor vehicles play a domain role (*2*). From the traffic efficiency perspective, the hour lost in traffic congestion in Los Angeles, for instance, is over 120 hours per year and the cost of congestion (per driver) is near 1800 dollars (*3*).

With the rapid development of vehicle communication technology, connected vehicles (CVs) has the capacity to communicate with other CVs or infrastructures via vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication. Therefore, CV can interact or cooperate with other traffic agents and improve the performance in traffic efficiency or energy efficiency. On the other hand, owing to the rapid development of artificial intelligent (AI) technology, autonomous vehicles (AVs) which equipped with multi-source on-board sensors (like various kinds of camera, Lidar, Radar and etc.) can cognize the environment and then take proper actions. Thus, for AVs, they can drive under some specific conditions without any human control. Connected and automated vehicle (CAV), which includes both of the vehicle-to-everything (V2X) ability and the self-driving capability, takes advantage of both CVs and AVs (i.e. connectivity and automation). CAV can receive information from other CVs and V2X-based infrastructure to optimize the traffic. Therefore, CAV is now regarded as one of the transformative solutions for addressing the aforementioned issues.

In order to improve traffic efficiency and reduce energy consumption, a series of studies have been conducted for the application of CAVs under specific traffic environment, such as intersection driving. A survey conducted by Rios-Torres et al. summarized the relative studies and the research trend of CAVs on the connectivity-based intersection driving optimization (*4*). Lee et al. proposed a cooperative vehicle intersection control (CVIC) system, which enables the interaction between vehicles and infrastructure to optimize the efficient intersection operation (*5*). Lin et al. proposed an autonomous vehicle-intersection coordination method, which divided the road network into three logical sections (*6*). From the eco-driving perspective, Peng et al. proposed an Eco-Approach and Departure (EAD) model and evaluated the EAD application in real-world traffic, which takes the advantage of signal phase and timing (SPaT) and Geometric Intersection Description (GID) information broadcasted by the traffic signals (*7, 8*). The aforementioned approaches are, to most extent, rule-based or model-based algorithms. These traditional rule-based and model-based driving control strategy can generate precise control trajectory and is easy to define and cognize. Nevertheless, most of these conventional algorithms are on the basis of some ideal assumptions. For instance, some study assumed that the penetration rate of CV is 100% (i.e. all vehicles are fully connected), which is, at least in recent decades, not realistic in real-world traffic. In addition, some intersection-cooperative driving methods assumed that the environment is traffic-free (i.e. there is no other vehicle in the current intersection), which will not work well in real traffic condition. The main constraint of the traditional model-based or rule-based methods comes from the complexity and uncertainty of real-world traffic. Thus, those methods can work well only under very specific traffic conditions.

Recently, with the AlphaGo (*9*) defeated the best human player, reinforcement learning

(RL) demonstrated its significant power in dealing with policy learning tasks (*10*). Reinforcement learning can learn the optimal policy by itself without predefined human rules or models, which endows the RL model the significant power to deal with a complex environment. Therefore, researchers in the transportation fields are also trying to implement RL to address the aforementioned limitations. A deep RL-based autonomous driving framework is proposed by Sallab et al., which enables the automatic lane-keeping with interaction with simple traffic (*11*). Desjardins et al. proposed an RL-based cooperative adaptive cruise control (CACC) method by utilizing V2V information, which can result in efficient behavior in CACC (*12*). Shalev-Shwartz proposed an RL-based safe driving model, which enables multi-agents to merge smoothly in a double-merge scenario (*13*). For signalized intersection scenario, Chen et al. proposed a hierarchical RL-based driving behavior control model, which enables the vehicle to interact with the traffic signal (*14*). Admittedly, the above RL-based algorithms can enhance the autonomous vehicle capability in various perspective, like lane-keeping, CACC, merging and traffic-signal interaction. However, so far, to the authors' knowledge, few RL algorithms are used in intersection-based eco-driving strategy which is a crucial issue in the transportation system. One reason may be that RL algorithms are good at solving single logical task while the intersection-based eco-driving has at least three different logical tasks: (1) Energy efficient, which requires the vehicle to drive through the intersection with less energy consumption. (2) Intersection interaction, which requires the vehicle to interact properly with the traffic signal. (3) Traffic interaction, which requires the vehicle to interact with traffic without colliding with each other. In this study, in order to further explore the capacity of the eco-driving strategy of CAV under realistic mixed traffic in a signalized intersection, we proposed a hybrid reinforcement learning (HRL) framework to learn the long-term driving strategy. The main contributions of this research are:

- Proposed a HRL framework for logically complex task, like eco-driving under signalized intersection.
- Proposed a long-short term reward algorithm, which endows the RL model with the ability to learn long-term driving strategy in complex driving task.
- The traffic environment is considered as mixed traffic where the environment vehicles are human-driven (i.e. without connectivity) and have different dynamic models.
- In order to make the mixed traffic more realistic, intelligent driver model (IDM) is applied in building the mixed traffic.
- Proposed a vision-based RL-network which enables the ego-vehicle to interacte properly with the mixed traffic.

The following section provides details of problem formulation, which includes the design of traffic environment and ego-vehicle, along with the HRL-based eco-driving framework, which includes the system architecture, algorithms design of decision manager, preprocessing, eco-driving RL algorithm and the definition of network structure. The methodology section is followed by the experiment section which introduces the simulator development and the procedures of training and testing. Next, comprehensive numerical experiments are conducted to address the energy efficiency benefit of the proposed method. The last section concludes the paper with the discussion on future works.

METHODOLOGY

Problem Formulation

The main purpose of this study is to design an RL-based framework to conduct an eco-driving strategy for vision-based CAV under mixed traffic in the signalized intersection. This topic is substantially an optimal policy learning task. There are three main goals of the policy: (1) to reduce energy consumption, (2) to improve the travel efficiency, and (3) to safely interact with traffic and signal. RL has five key compositions and the connections between these compositions and the traffic situation in this paper are shown as follows:

- Agent: the ego-vehicle which can perceive the environment via front camera and V2V-based SPaT information;
- Environment: traffic environment which includes various kinds of vehicles and signalized intersection;
- Policy: the proposed eco-driving policy;
- Action reward: the short-term benefit (i.e. speed reward, energy consumption) of taking action right at this moment and the long-term benefit (i.e. travel time, total energy consumption) of the journey;
- Action-value function: the function to determine which action is the best choice at the next moment to achieve a long-term optimal result.

These connections illustrate that the issue in this research can be well interpreted by the RL framework. RL framework is established based on Markov Decision Process (MDP) which is a mathematical framework for decision making via the interaction between a learning agent and its environment in terms of state, actions and rewards. In this research, the ego-vehicle (i.e. agent) interacts with the environment (i.e. mixed traffic and signalized intersection). To be specific, the traffic environment, agent observation, agent actions are discussed below.

Traffic Environment

The traffic environment includes three main part: the ego-vehicle, environment vehicles (EVs) and a five-lane signalized intersection. In order to make the proposed environment more similar to the real traffic, we designed different kind of EVs and different start phase time of the traffic light. To be specific, the EVs are divided into five kind vehicles which have different dynamic model and behavior strategy. To make the EVs more realistic, we applied the intelligent driver model (IDM) to the longitudinal control method of EVs. The applied IDM is illustrated below:

$$a_{free}(t) = a(1-(\frac{v(t)}{v_{tar}})^2) \tag{1}$$

$$a_{int}(t) = a(1-(\frac{v}{v_{tar}})^{\delta}-(\frac{s_0+vT+\frac{v\Delta v}{2\sqrt{ab}}}{s})^2) \tag{2}$$

$$a(t) = \begin{cases} a_{free}(t), & no\ front\ vehicle \\ a_{int}(t), & otherwise \end{cases} \tag{3}$$

where the $a_{free}(t)$, $a_{int}(t)$ represent the acceleration when there is no front vehicle and there is front vehicle and the $a, b, v, v_{tar}, T, \Delta v, and s_0$ represent the maximum acceleration, maximum deceleration, current speed, target speed, safe time headway, speed difference with front vehicle, and minimum distance separately. For the latitudinal control, we designed different rates for EVs to change target lane. The detail of the description of EVs is shown in Table 1.

For the traffic light, every time the simulation start, the initial phase and time are randomly

TABLE 1: The description of the dynamic model of the vehicles.

Vehicle	a	b	s_0	T	v_{tar}	r_{lat}
EV1	$6.0m/s^2$	$6.0m/s^2$	$3m$	$1.5s$	$13.8m/s$	0.3
EV2	$5.0m/s^2$	$4.5m/s^2$	$3m$	$1.5s$	$12.5m/s$	0.2
EV3	$3.0m/s^2$	$5.0m/s^2$	$2m$	$1.2s$	$11.1m/s$	0.2
EV4	$3.0m/s^2$	$3.0m/s^2$	$3m$	$1.5s$	$9.72m/s$	0.1
EV5	$2.0m/s^2$	$1.5m/s^2$	$5m$	$1.5s$	$8.33m/s$	0.1

selected in the whole time period. Specifically, the greed time, yellow time, red time and all-red time are set as the 20s, 3s, 40s, and 1s separately.

Agent

In this paper, the ego-vehicle is selected as an electric CAV. The maximum acceleration and deceleration are set as $3m/s_2$ and $-5m/s_2$ separately. Furthermore, the observation input, energy consumption model (ECM) and action output are defined as follow.

For the input observation, in this study, the perception information comes from three main parts: (1) V2I communication-based signalized traffic light information which includes the current phase state and the duration time; (2) the on-board sensor which includes three radars (left distance d_l, right distance d_r and front distance d_f) and front camera (image size 320x160, 50fps); and (3) on-board diagnosis which include the ego-vehicle speed v_e and acceleration value a_e.

Due to the multiple-input data, in order to enhance the learning performance, we find an efficient way to decrease the complexity of the input observation without losing too much information. For the radar data, we define three variables which are the forward warning w_f, the left warning w_l and the right warning w_r. The definition of these variables are shown as follow:

$$w_d = \frac{3 + (v_e - V_f)^2}{2a_e} \tag{4}$$

$$w_f = \begin{cases} 0, & d_f > w_d \\ 1, & d_f <= w_d \end{cases} \tag{5}$$

$$w_l = \begin{cases} 0, & d_l > w_l \\ 1, & d_l <= w_l \end{cases} \tag{6}$$

$$w_r = \begin{cases} 0, & d_r > w_r \\ 1, & d_r <= w_r \end{cases} \tag{7}$$

Where w_d, v_f, w_l, w_r represent the forward warning threshold distance, forward vehicle velocity, left-warning threshold distance and right-warning threshold distance separately. Specifically, w_l, W_r are both set as 2m. Thus, the observation space O is composed of a 12-dimensional vector

$$O = \{t_g, t_y, t_r, w_l, w_r, w_c, d_r, d_f, v_f, v_e, a_e\} \tag{8}$$

where $t_g, t_y, t_r, d_r, w_c, d_f, v_f$ represent the duration time of green light, yellow light, red light, the remain distance, collision warning, forward distance, and the speed of forward vehicle separately.

For the energy consumption model, we applied one of our previous work in which an energy consumption model was proposed and calibrated by real-world driving data from a 2013

NISSAN LEAF (*15*). The original energy consumption model is shown below.

$$\begin{aligned} E(v,a,\alpha) = & -3.037 - 0.591v\cos\alpha - 1.047\times10^{-3}v^3 - 1.403v\alpha + 2.831\times10^{-2}v^2\cos\alpha \\ & - 7.980\times10^{-2}v^2a - 1.490v\alpha\sin\alpha + 3.535\times10^{-3}v^3a - 0.243va^2 \\ & - 1.279v\alpha\cos\alpha + 6.484\times10^{-4}v^3\alpha + 0.998va\alpha \end{aligned} \tag{9}$$

Where v, a, α represent the instant acceleration (m/s2), speed (m/s) and road grade (rad) separately. In this study, the road grade is set as zero. Besides, in the original model, the breaking will charge the battery, which may cause a negative influence on RL learning. Thus, we redefined the original model which is shown below where E_{energy} represent the energy consumption model applied in our framework.

$$E_{energy} = \begin{cases} E, & a >= 0 \\ 0, & a < 0 \end{cases} \tag{10}$$

For the output action, in this study, the main purpose is to learn an optimal strategy in both longitudinal and latitudinal driving maneuvers. Thus, the output actions are defined both on these two dimensions. For longitudinal maneuver, the action space A is

$$A = \{1.0a, 0.8a, 0.6a, 0.4a, 0.2a, 0.0, -0.2a, -0.4a, -0.6a, -0.8a, -1.0a\} \tag{11}$$

where a represent the maximum acceleration. For the latitudinal maneuver, the target lane action space is $\{-1, 0, 1\}$ where $-1, 0, 1$represent the target lane is the left lane, the current lane, and the right lane separately.

Furthermore, in order to enhance the learning performance, we find a way to decrease the dimension of action space. Generally, the action space will be a 33-dimensional vector. However, we define that the lane-changing maneuvers are only available when the longitudinal acceleration is zero, which means when the vehicle is accelerating or decelerating it should not change lane (this also fit with the real world traffic safety rules). Thus, in our paper, the action space is a 13-dimensional vector.

Hybrid RL-based Eco-Driving Framework

System Architecture

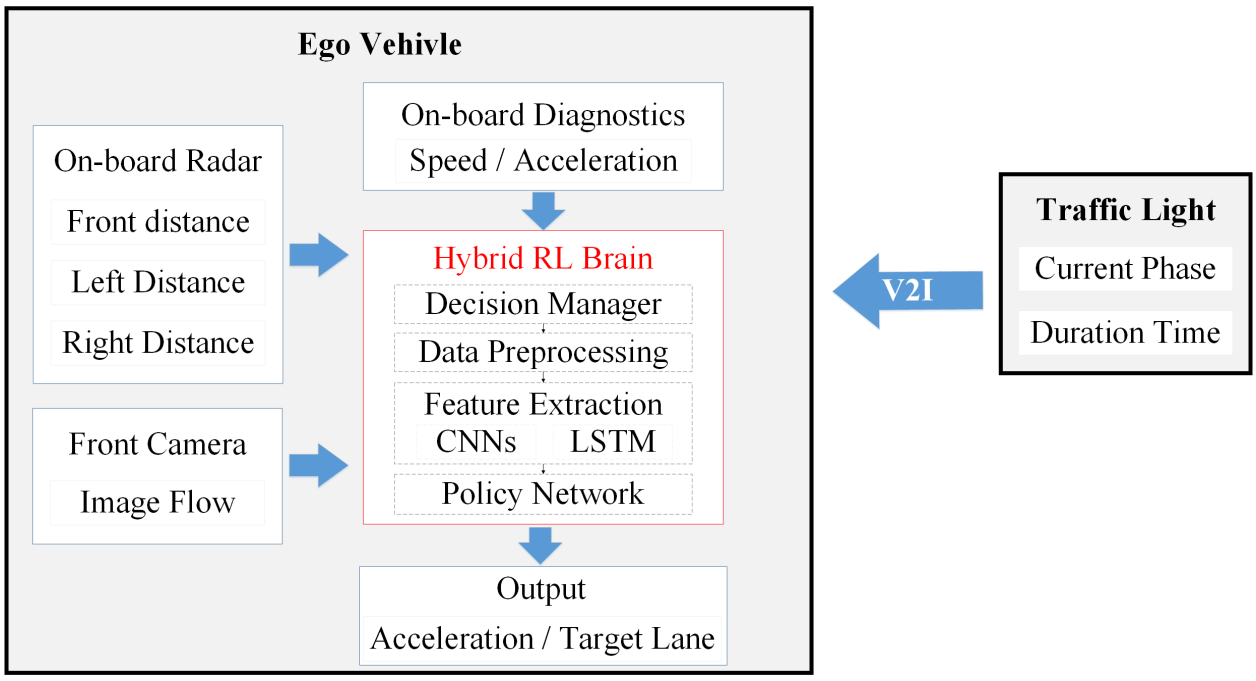
In this research, we proposed the hybrid RL-based CAV eco-driving framework and algorithms for electric passenger vehicles. **Figure 1(a)** illustrates the key components of the hybrid-RL based CAV eco-driving system which consists of several components as described briefly below. **Figure 1(b)** shows the structure of intersection traffic scenario.

1. On-board computer: it houses the hybrid-RL brain which is the decision-making center of the whole system. The brain will receive perception data and process the data. Then it will return the longitudinal acceleration and target lane information to the vehicle control center.

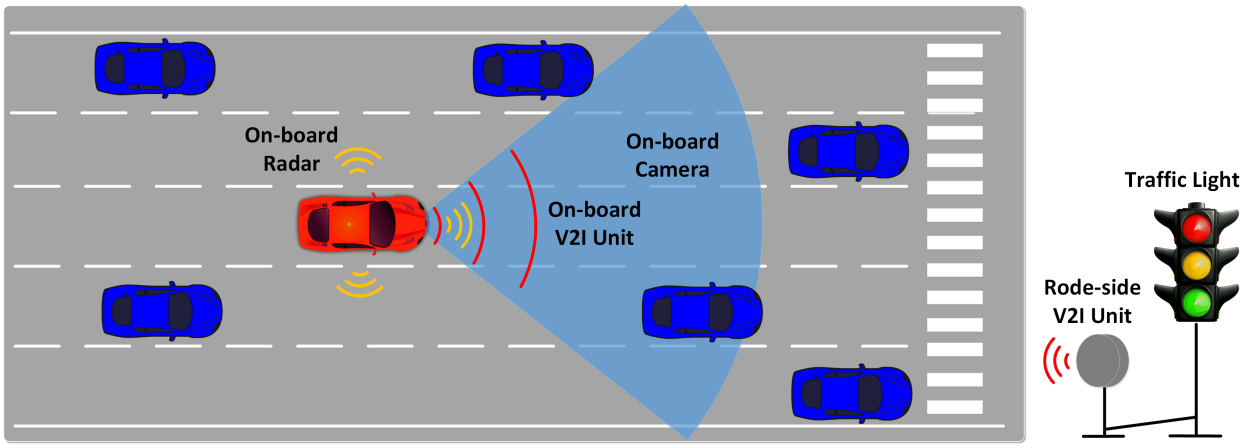
2. On-board radar: there are three radars applied in this research. One is installed in front of the vehicle, which is used to detect the front distance and front-vehicle velocity. The other two radars are installed at the left side and right side of the vehicle, which are used to detect the left distance and right distance. The radar information will be sent to the on-board computer as part of the agent observation.

3. On-board camera: it is installed in the front of the vehicle. The direction of the camera is the same as the vehicle‘s driving direction. The camera information will be sent to the on-board computer, which is the key part of the traffic perception.

4. On-board diagnostics (OBD): this component can get the instant speed and acceleration



(a) System architecture



(b) Traffic environment structure

FIGURE 1: **The HRL-based Eco-Driving system architecture.**

information and then the message will be sent to the on-board computer as part of the observation.

5. Traffic light and road-side unit: the traffic light will send the real-time traffic light information to the road-side unit. Then the road-side unit will send the information to any CAV within its communication coverage.

As the most crucial component in the system, the hybrid-RL brain aims to drive through an intersection with less time and energy consumption by generating appropriate instant longitudinal acceleration and target lane. As we have discussed above, driving through intersection traffic is actually a logically complex task. Because there are at least four sub-tasks: (1) cruise control that avoid collision with other vehicle; (2) lane-changing decision that interact with the traffic; (3)

stop-light reaction that stop the vehicle before the stop line when the light is red or yellow; and (4) go-light reaction that speed up when the light is now becoming green. So far, it is nearly impossible for one RL algorithm to handle such a multi-phase task. In this research, we proposed a hybrid RL brain which combines the manual-designed rules and RL algorithm. The hybrid-RL brain consists of several components which are illustrated as follows.

Decision Manager

The key component of the hybrid-RL framework is the decision manager because this part combines the manual-designed rules and the RL algorithm to endow the framework with the ability to handle the complex task. In the decision manager, the driving process is divided into different running situations according to the immediate situations of ego-vehicle and traffic light. The architecture of the decision manager is shown in **Figure 2(a)** where the $v_{ego}(t)$, $w_{light}(t)$, $w_{forward}(t)$, $d_{ego}(t)$, d_{light}, d_{total} represent the ego-vehicle speed, light warning, forward warning, ego-vehicle distance, the distance between the start point and the entry of intersection and the total distance of the environment separately. To be more specific, the definition of light warning is described as **Figure 2(b)**.

Figure2(b) illustrates the definition of light warning $w_{light}(t)$. When the vehicle enters the light-warning trigger area and the light is red or yellow, the $w_{light}(t)$ is True. On the other hand, if the vehicle does not enter the trigger area or the light is green, the $w_{light}(t)$ is False. The main purpose of light-warning trigger method is to build a stable vehicle-signal interaction. As we have discussed, it is hard for the RL to learn all the logically different tasks.

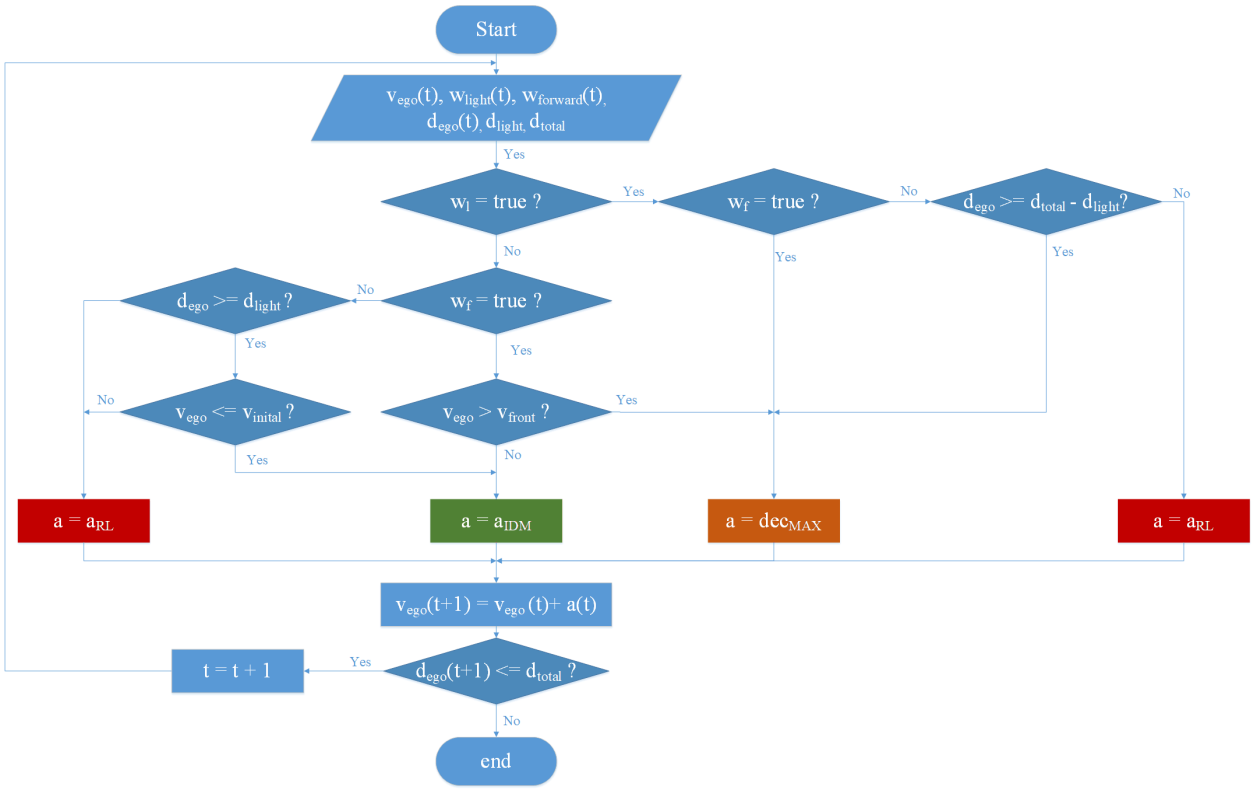
In addition, the IDM and emergency breaking model are also integrated into the decision manager. The IDM model is activated when the ego-vehicle need to start at an intersection when the light becomes green. The emergency braking model is defined as $v = vb$ and the emergency braking will be activated when the $w_{forward}(t)$ is True.

Data Preprocessing

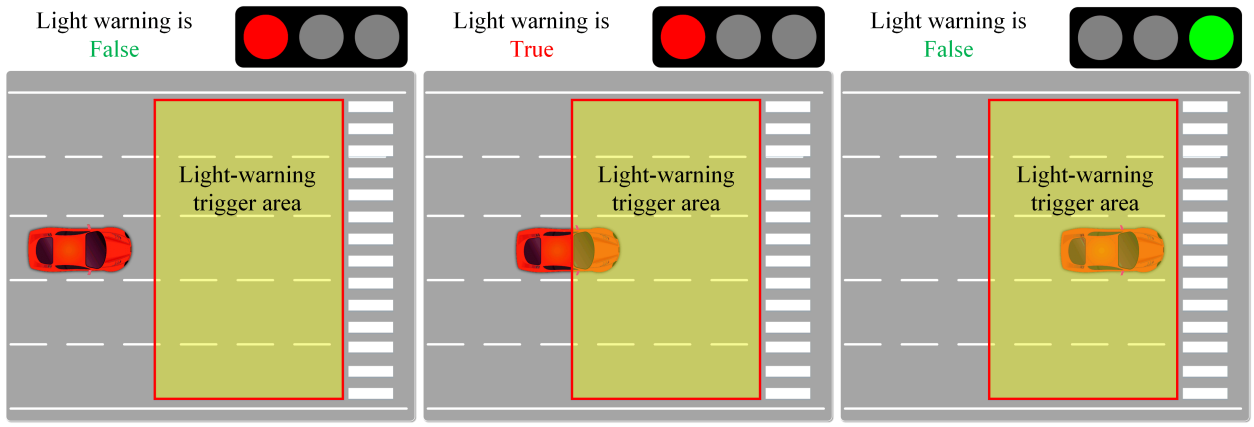
In this research, observation input of ego-vehicle consists of (1) a 50fps raw image flow and (2) a 13-dimensional vector. Generally, the raw image data cannot be fed into the training network directly because the size of the raw image is not compatible with the convolutional neural networks. Besides, the purpose of this study is to explore the ability of the hybrid RL to generate the eco-driving strategy for CAV. Hence, considering the driving strategy relies on both spatial and temporal information, instead of resizing the images, we also need to transform multiple single-frame images into a multi-frame spatiotemporal data format. To be more specific, in this study, in order to include more information without making the input data too heavy (having too many frames in one format). We proposed a select-stack preprocess method. The whole preprocess includes four steps as below:

1. Recording the raw image flow based on the time sequence;
2. Resizing every frame of the raw image flow into an $80 \times 80 \times 3$ format.
3. Selecting one frame out of N_{select} frames;
4. Stacking N_{stack} frame of images into a higher dimensional data format: $80 \times 80 \times 3 \times N_{stack}$.

Specifically, in this study, the Nselect and Nstack are both set as 4. Through the above preprocessing method, the training network could get spatiotemporal observation data, which is of considerable significance on the driving strategy learning procedure.



(a) Decision Manager



(b) Definition of Traffic-light Warning

FIGURE 2: (a) The visualization of the structure of Decision Manager and (b) the detail of the definition of one key variable in Decision Manager.

Deep RL for Eco-Driving

According to the above discussion, the eco-driving approach of going through an intersection can be formulated as an MDP in which the agent interacts with the environment. Furthermore, due to the discretion of action space in this research, we applied Dueling Deep Q Network (Dueling DQN) as our basic RL framework. The Dueling DQN is developed from Deep Q Network (DQN), which is briefly introduced below.

Deep Q Network is a typical deep RL algorithm that uses a deep neural network to predict the value function of each discrete action. DQN performs in discrete action spaces and aims to choose the action with maximum value output. Specifically, the input of DQN is observation state o_t, and the output is the evaluation value $Q(s_t, a_t)$ corresponding to each action a_t in state A. Then, according to the $\varepsilon - greedy$ algorithm, an action is selected from the action space. After the execution of action a_t, a reward r_t and an observation state o_{t+1} can be get from the environment. In addition, prioritized experience replay (*16*) algorithm is used to solve the problem of correlation and non-static distribution. Experience state $e_t(s_t, a_t, r_t, s_{t+1})$ will be stored in the experience pool $E_t = (e_1, e_2, ..., e_t)$. During the training process, a mini-batch of data will be selected randomly from the experience pool so that the training process can avoid correlation problem.

$$L(\theta) = (R_t + \gamma max_{A_{t+1}} Q(O_t, A_t; \theta))^2 \tag{12}$$

where γ is a discount factor, θ is parameters of neural network and θ^- is the parameters of target network.

In many visual perception-based DRL tasks, the value functions of different state actions are disparate, but in some states, the size of the value function is independent of the action. Thus, Wang (*17*) proposed a dueling network based DQN model named Dueling DQN. Dueling DQN is constructed with two streams which separately estimate (scalar) state-value and the advantages of each action and shows significant performance improvement than DQN. The equation (shown as follow) for calculating Q-value of Dueling DQN is designed to aggregate the states-value and action advantages.

$$Q(S_t, A_t; \theta, \alpha, \beta) = V(S_t; \theta.\beta) + A(S_t, A_t; \theta, \alpha) - \frac{1}{|A|} \sum_{A_t} A(S_t, A_t; \theta, \alpha) \tag{13}$$

where α represents the parameters of A (the advantage function). Besides, β represents the parameters of V (the state-value function) and θ is parameters of neural network. Thus, due to the performance of long-term reward learning and vision-based RL task, Dueling DQN is applied in this research as the basic RL algorithm of the hybrid RL framework.

However, in this research, the biggest challenge is that we want to find a method that can decrease energy consumption without spending more travel time (even save time) under mixed traffic condition. As is well-known, when driving on a freeway, the long-term travel time depends, to most extent, on the short-term reward such as the instant speed or lane changing. Nevertheless, if driving through an intersection, the long-term travel time and energy consumption depend more on the interaction between current vehicle status and traffic signal status. Thus, in order to figure out the optimal eco-driving solution in a realistic signalized intersection traffic situation, the most significant work is to build an RL model that can learn more from the long-term driving reward.

Although Dueling DQN framework is powerful in learning vision-based long-term policy, it is difficult to design an appropriate reward function for the eco-driving RL model. The main reason is the reward function is actually an instant value. On the contrary, the travel time and the total energy consumption can only be received when the journey is finished (i.e. ego-vehicle cannot know how many the total consumption is until it reaches the destination). Thus, the RL model will not work well or even don't work at all if the reward function is designed straight forward as usual (more information at the Experiments section).

In this study, we proposed a long-short term reward (LSTR) function, which not only considers the instance variables such as speed, lane change, and instant energy consumption but also

includes some long-term based indicators. The two conflicting factors in this issue are the short-term reward (instant speed, energy consumption) and the long-term reward (total travel time and total energy consumption). We designed some instant reward principles which include indications for long-term benefit, which are shown below.

- When the current phase is red or yellow and ego-vehicle cannot pass the intersection with its current speed, then it shouldn't accelerate.
- When the current phase is red or yellow and ego-vehicle may pass the intersection with current speed or driving faster, then try to accelerate.
- When the current phase is green and the ego-vehicle cannot pass the intersection with its current speed, then it shouldn't accelerate.
- When the current phase is green and the ego-vehicle may pass the intersection with current speed or driving faster, then try to accelerate.

Basing on these principles, the LSTR function is designed as **Algorithm 1** in which the definition of R_{light} is further explained by **Figure 3**.

Algorithm 1 Reward Function Descriptions.

Input:

The current speed $v(t)$, acceleration $a(t)$ and current position d(t) of ego-vehicle and the curent phase P_c and the duration time t_d of the traffic light;

Output:

The reward value $R(t)$;

Initialization: $R = R_{velocity} = R_{lanechange} = R_{danger} = R_{energy} = R_{time} = R_{light} = 0$;
$d_{pridction} = v(t)t(d)$; $d_{toLight} = d_{total} - d(t)$
$R_{velocity}(t) = (v(t) - v_{min})/(v_{max} - v_{min})$;
if $a(t) > 0$; **then**
    $R_{energy}(t) = (-3.037 - 0.591v(t) - 0.001047v(t)^3 - 1.403v(t)a(t) = 0.02831v(t)^2 - 0.0798v(t)^2a(t) + 0.003535v(t)^3a(t) - 0.243v(t)a(t)^2)/130$;
else
    $R_{energy}(t) = 0$;
end if
$R_{time}(t) = R_{time}(t-1) + \delta t$
if Lane-change action happens; **then**
    $R_{lanechange}(t) = -0.1$;
end if
if Dangerous action happens; **then**
    $R_{danger}(t) = -0.5$;
end if
$R(t) = R_{velocity}(t) + R_{energy}(t) + R_{time}(t) + R_{lanechange}(t) + R_{danger}(t) + R_{light}(t)$
return $R(t)$;

In the LSTR function, the $R_{velocity}(t)$, $R_{energy}(t)$, $R_{lanechange}(t)$, and $R_{danger}(t)$ are short-term benefit indicators, whicle the $R_{time}(t)$, $R_{light}(t)$ are long-term benefit indicators.

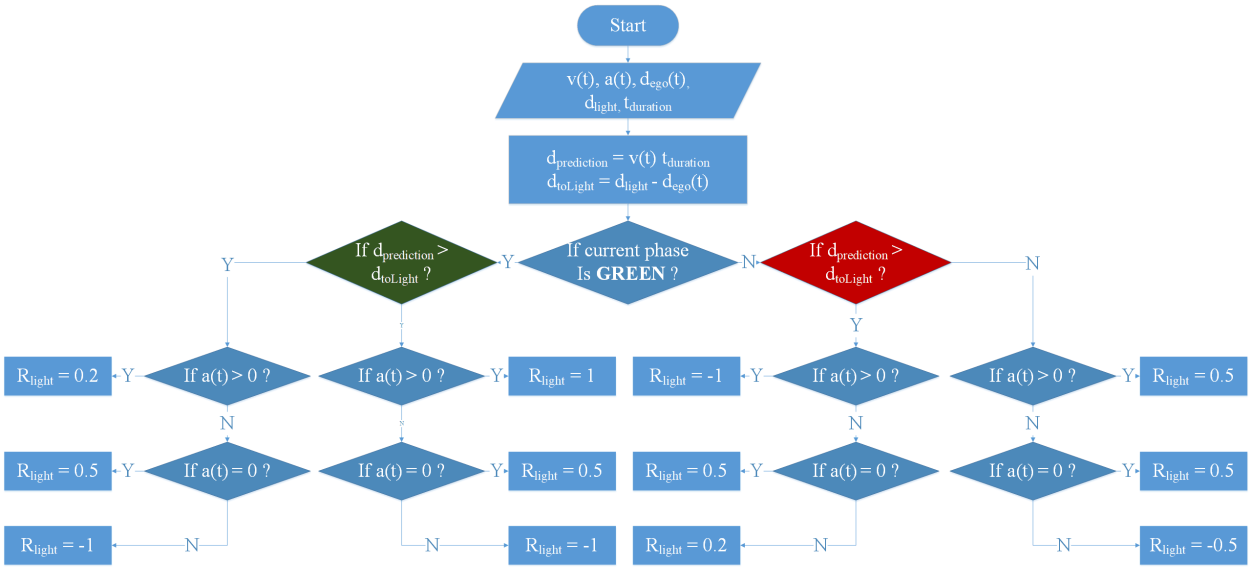


FIGURE 3: **The definition of R_{light} is based on the aforementioned four principles, which use manual-designed long-term indicators.**

Network Architecture

The main purpose of the deep neural network is mapping the current observation state O to the best action value A in action space. Thus, the main network can be divided into two components: (1) the hidden feature extraction network and (2) the policy network which applies Dueling DQN. Figure 3 illustrates the architecture of the proposed deep RL network.

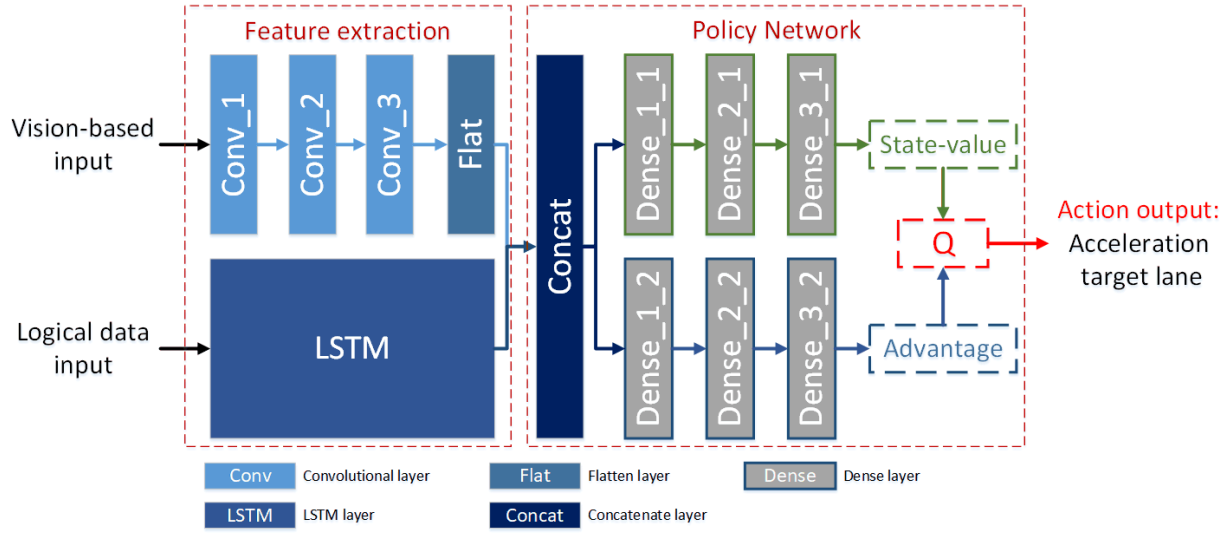


FIGURE 4: **The architecture of the neural network.**

For the hidden feature extraction network, its main purpose is to extract hidden features from the preprocessed high-dimensional spatiotemporal data. As discussed above, the network input consists of (1) image flow and (2) high dimensional vector. In order to extract features from these different type of data, we designed a multi-channel feature extraction network, which consists

of different kinds of parallel neural networks.

For the image flow input, the most popular method to extract image data is convolutional neural network (CNN), because the image is actually a spatial data format and the CNN has a tremendous capability of solving high-dimensional spatial data. Thus, in this study, a CNN stream is designed to extract the vision-based input data.

On the other hand, the Radar, OBD, and V2I information (named as logical data) are integrated into a 13-dimension vector. After the selected-stack preprocess, the logical data are transformed into a time-series based temporal data. Long-short term memory (LSTM) network is famous for its powerful ability in dealing with temporal data series. Thus, for the logical-data feature extraction, LSTM is applied in our deep RL net.

After the CNN and LSTM, the two-stream features are combined through a concatenate layer and then a dense layer-based policy network is designed by applying Dueling DQN. Fig illustrates that there are two streams of the dense layer. The upper stream is used to extract state value while the lower stream is used to extract advantage value. Finally, the action with maximum Q-value will be selected as the action at this moment. **Table 2** shows the details of the configuration of the neural network.

TABLE 2: **Parameters for deep neural network**

Layer	Actuation	Patch size	Stride	Filter	Unit
Conv_1	ReLU	8×8	4	32	-
Conv_2	ReLU	4×4	2	64	-
Conv_3	ReLU	3×3	1	64	-
LSTM	-	-	-	-	1024
Dense_1_1/2	ReLU	-	-	-	1024
Dense_2_1/2	ReLU	-	-	-	256
Dense_3_1/2	ReLU	-	-	-	128

Network Updata and Hyperparameters

There are four steps of the network updating process, which are:

1. Considering the current state as st and predicting the Qt value of different actions through the evaluation network.
2. Choosing the action ai,t with the largest Q value by utilizing e-greedy policy.
3. Generating the Q values at time t + 1: Qt+1 through the target network.
4. Calculating the loss function and then updating the evaluation network.

In addition, at each learning step, the weight coefficients of the proposed network were updated using the adaptive learning rate trick Adam (*18*) in order to minimize the loss function. For the adopted hyperparameters, the learning rate α, discount factor γ, batch size, steps used for observation, replay memory size, steps for target network update, training steps, and test steps are set as 0.00025, 0.99, 64, 10000, 50000, 10000, 2million, 10000 separately.

NUMERICAL EXPERIMENTS

Simulator Design and Development

In order to train the proposed model and evaluate its performance, a simulator is constructed in this paper by utilizing Unity 3D and Unity Machine-learning Agents (Unity ML-Agents). The intersec-

tion simulator is developed basing on our previous work and Min's work (*19, 20*). The main reason to use Unity to build the simulator is that Unity can provide a virtual reality environment in which the simulated camera can be applied. In addition, Unity ML-Agents provides a machine learning development platform in which the machine learning algorithms can be constructed readily. For the external RL brain, the deep RL algorithm and deep neural network are developed basing on Tensorflow (*21*) by Python.

As is discussed in the front section, the scenario in this study is built as a one-direction intersection with 5 lanes. There are five different kinds of human-driven vehicles. The length of the research area is 550 meters: from 500 meters upstream of the intersection to 50 meters after the stop line. The vehicle speed limit is set to 50 kilometers per hour (kph). For the traffic light, the time for the green phase is 20 seconds, the time for the yellow phase is 2 seconds and the time for the red phase is 41s.

Model Training and Numerical Test

For the training procedure, the $\varepsilon - greedy$ policy is implemented as the exploration policy. The ε is decreased linearly from 1 to 0.00001 over 2 million steps. When the training start, the initial phase and time of the traffic light will be randomly selected, which not only avoids the overfitting of the algorithm but also makes the training more realistic. In addition, the simulator is equipped with Intel(R) Core(TM) i7-7700k CPU @ 4.20GHz, 64 GB RAM, and an NVIDIA GTX 1080 GPU. The total training time is around 36 hours.

For the test procedure, there are three baselines implemented to compare with the proposed HRL framework, which are briefly introduced below.

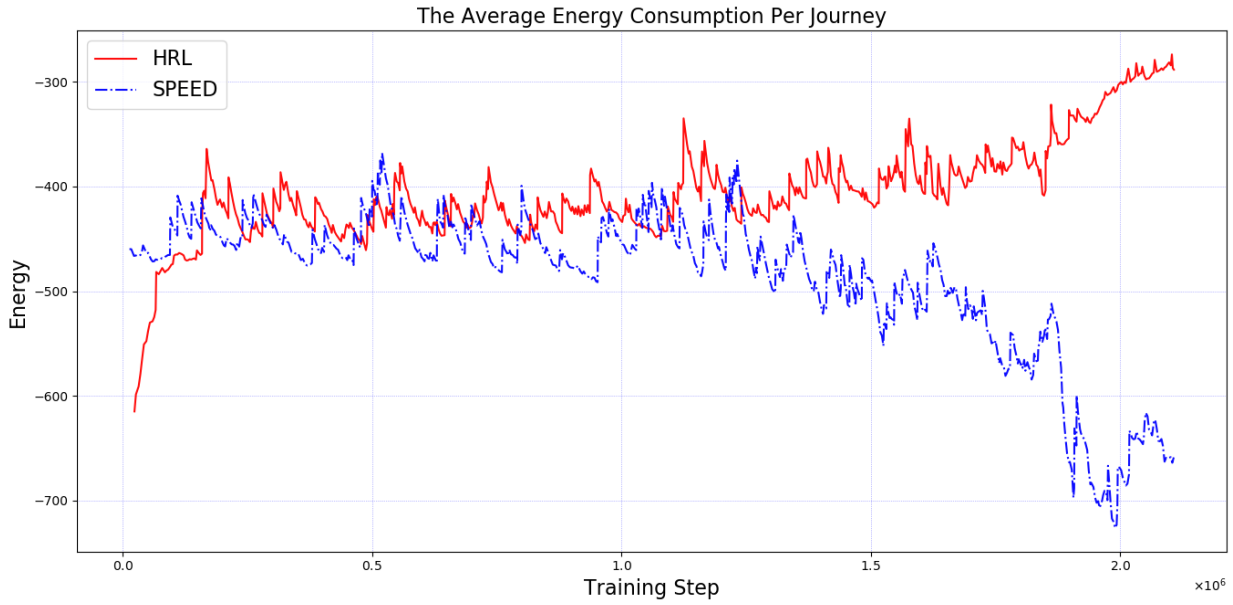
1. IDM method: the IDM-based control model, which means the ego-vehicle is totally controlled by IDM (i.e. like the environment vehicles);
2. INSTANT method: only considering short-term benefit in the reward function, i.e. without R_{light} and R_{time};
3. SPEED method: only considering speed efficiency in the reward function, i.e. without R_{light}, R_{time} and R_{energy}.

The proposed framework and each of the aforementioned baselines are tested through numerical experiments. As is illustrated in Fig, six different entry time in a cycle are tested: the 0th second of the cycle (C0), the 10th second of the cycle (C10), the 20th second of the cycle (C20), the 30th second of the cycle (C30), the 40th second of the cycle (C40) and the 50th second of the cycle (C50). Furthermore, different initial speeds from 10 kph to 50 kph with 10 kph as the increment (S10, S20, S30, S40, S50) are also tested. The training and numerical test results are discussed in the next section.

RESULTS

Training Results

Figure 5 illustrate the training results of HRL method and SPEED method, including the average energy consumption, traffic-light interaction reward, average speed and average lane-changing number in a single journey. Figure 5(a) shows that the energy consumption of HRL method is decreasing via the iteration of training while the energy consumption of SPEED is increasing, which is because the SPEED method is trying to get higher instant speed without considering the energy consumption. In addition, the training results of the INSTANT method is not shown below, because during the iteration of training the ego-vehicle will drive more and slower and finally stop in



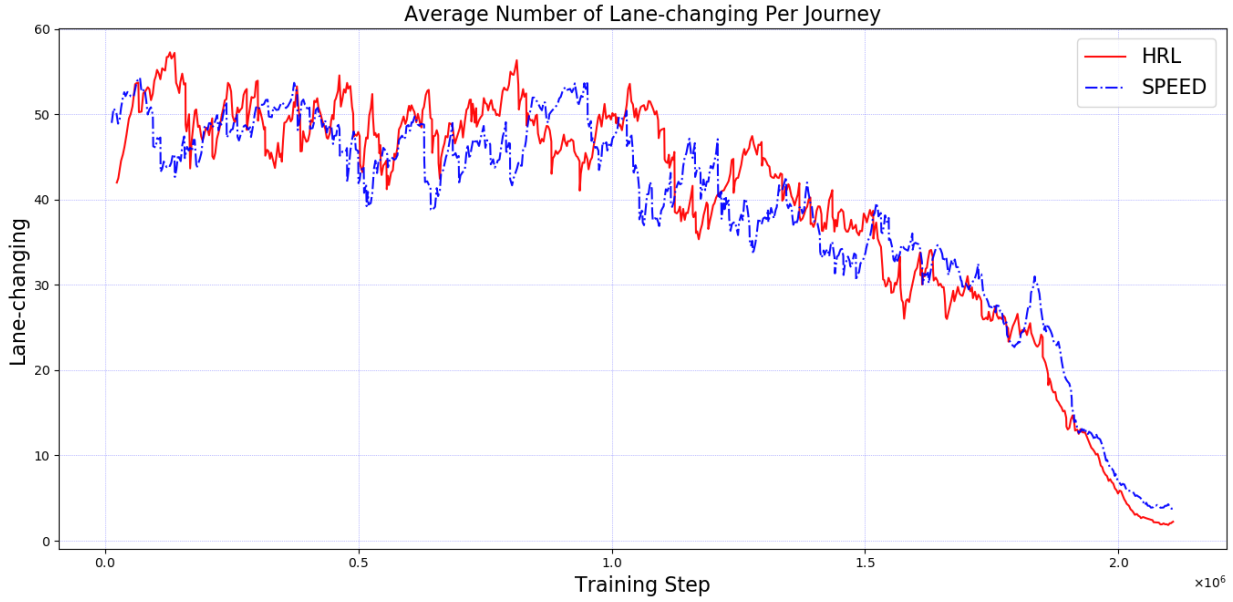
(a) The average energy consumption per journey during the training for HRL and SPEED model.

(b) The traffic-light interaction reward per journey during the training for HRL and SPEED model.

the road, which makes the training have to stop. We realize that this may be because the conflicting of two short-term rewards pushes the learning to fall into one side of the conflict factors (i.e. in this case, the short-term energy factor is stronger than speed, the ego-vehicle will not accelerate any more).

Basing on the above discussion, we realize that the Figure 5(b) can help to further explain why the HRL method can successfully learn an optimal policy in such a dilemma. Figure 5(b) shows that after the approximately half way of training steps, the traffic-light interaction reward is increasing obviously, which means that the ego-vehicle is learning more about how to interact with

(c) The average speed per journey during the training for HRL and SPEED model.



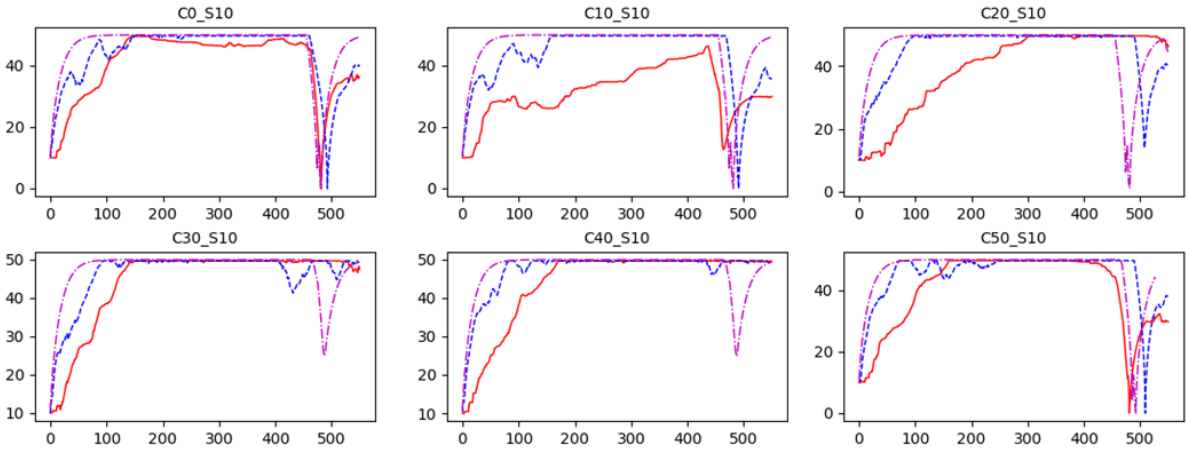
(d) The average number of lane changing per journey during the training for HRL and SPEED model.

FIGURE 5: **The results of the training including (a) the average energy consumption, (b) the traffic-light interaction reward, (c) the average speed and (d) the average number of lane changing.**

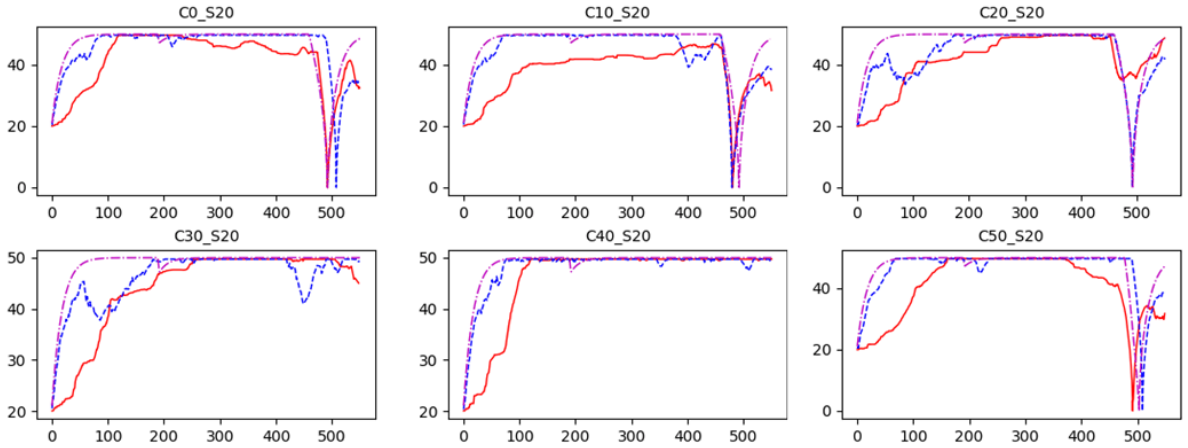
the traffic light intelligently (i.e. balancing the speed and energy consumption). On the other hand, it is also obvious that pursuing a speed-first driving strategy will actually cause a negative effect on the cooperation between vehicles and intersection. Figure 5(c) and 5(d) shows that the HRL method can also learn how to drive faster with less unnecessary lane-changing behaviors, which

represents a smoothly, time-efficient driving strategy.

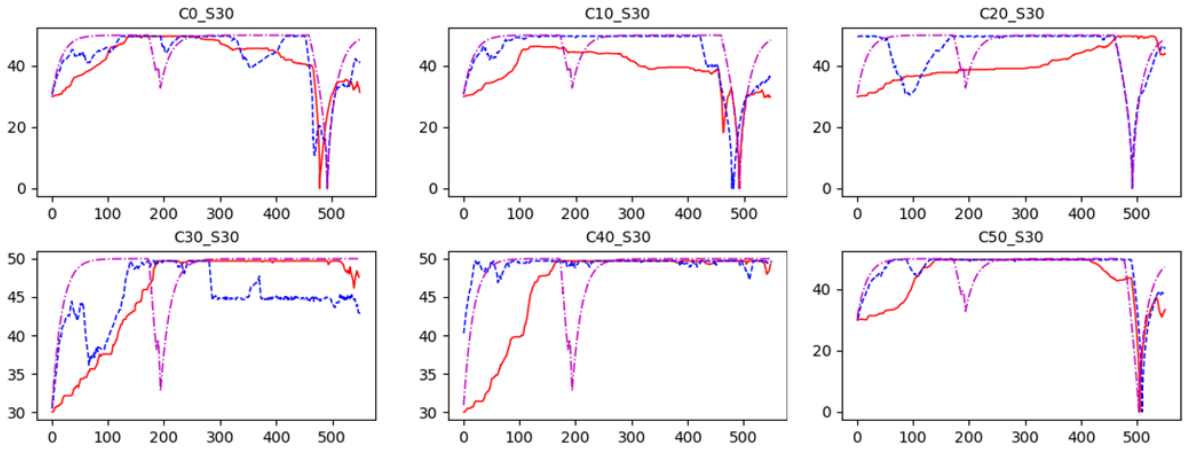
Numerical Testing Results



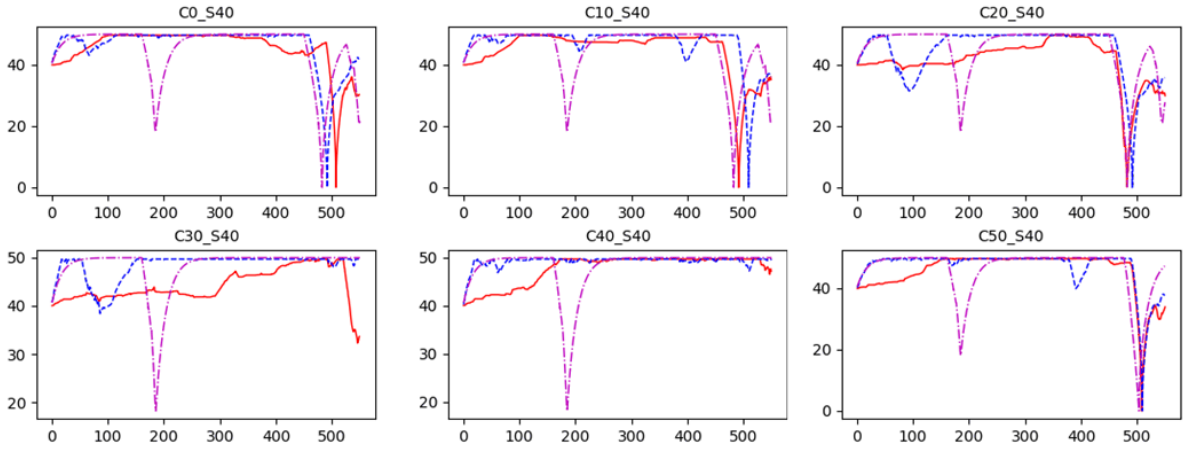
(a) The comparison of speed trajectories with 10 kph entry speed (S10).



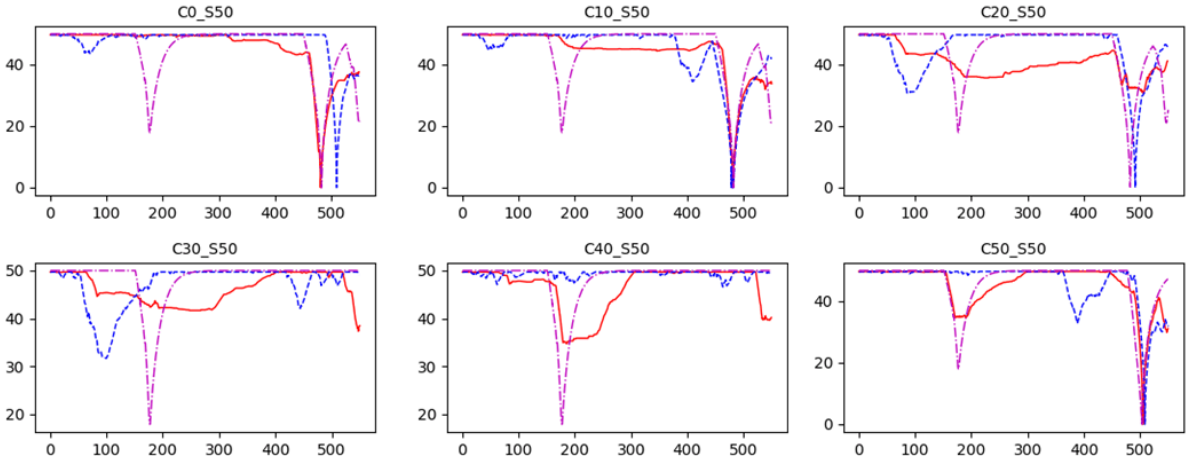
(b) The comparison of speed trajectories with 20 kph entry speed (S20).



(c) The comparison of speed trajectories with 30 kph entry speed (S30).



(d) The comparison of speed trajectories with 40 kph entry speed (S40).



(e) The comparison of speed trajectories with 50 kph entry speed (S50).

FIGURE 6: **The red line, purple line and the blue line represent the results of our HRL method, IDM method and SPEED method separately. The Y-axis and X-axis represent the instant speed (kph) and current distance (meter). The title of each subfigure represents the initial phase time and entry speed of ego-vehicle. For example, the C0_S10 represents the initial phase time is 0s in the signal cycle and the entry speed is 10 kph.**

Figure 6(a)(b)(c)(d)(e) illustrates the testing results of the comparison for speed trajectories of HRL, IDM and SPEED methods with different entry speed and initial phase time. According to the testing results, there are three points we want to discuss:

(1) The acceleration: it is evident that the acceleration value of HRL is obviously lower than either IDM or SPEED in almost all scenarios. Lower acceleration value is better for eco-friendly driving strategy, because, according to the energy consumption model, the energy consumption will go much higher if the acceleration increases. This demonstrates that the HRL model can drive in a more energy-efficient strategy.

(2) The target speed: for IDM and SPEED model, the target speed is always the maximum speed of the vehicle. However, for HRL model, it will not achieve the highest speed in some situations, such as C10_S10, C20_S10 and C20_S20, because the HRL model learns that in such

a situation, a lower speed can gain more benefit in the long run (i.e. waiting for the end of red light so that ego-vehicle can pass without stops, such as C20_S10 and C20_S30). This evidently shows that the HRL can drive in a higher travel-efficient way than IDM and SPEED method (i.e. better interact with the intersection).

(3) The interaction with the mixed traffic: for the IDM model, it can only longitudinally interacts with its front vehicle. For SPEED and HRL model, they can interact traffic from both longitudinal level and latitudinal level. Thus, in some situation, such as C40_S10, C0_S50, and C40_S40, the HRL and SPEED performed much better than IDM. In fact, due to the lower acceleration preference, HRL performed even better than SPEED, such as C30_S30, C20_S30, and C30_S50. This point illustrates that the HRL model can better interact with mixed traffic.

Table 3 shows the comparison for average travel time of HRL, IDM and SPEED method. According to the table, the HRL can achieve nearly the same performance to the SPEED method and better performance (1.2%) than IDM method. It is noticed that when the initial phase time is C20, the HRL has the relatively best time performance (6.9% better than SPEED).

TABLE 3: **The average time per travel for HRL method, SPEED method, and IDM method.**

Methods	S10/C0	S10/C10	S10/C20	S10/C30	S10/C40	S10/C50	Average
HRL	71.4	62.9	41.9	45.9	47.3	79.7	58.2
SPEED	69.7	62.4	44.8	44.2	43.2	80.2	57.4
IDM	70.7	62.2	51.2	45.2	44.9	81.5	59.3
Methods	S20/C0	S20/C10	S20/C20	S20/C30	S20/C40	S20/C50	Average
HRL	70.2	62.4	48.9	45.9	42.7	80.1	58.4
SPEED	67.9	63.1	52.6	43.1	41.4	79.6	58.0
IDM	69.6	61.5	49.7	42.5	40.72	80.3	57.4
Methods	S30/C0	S30/C10	S30/C20	S30/C30	S30/C40	S30/C50	Average
HRL	71.5	60.8	44.1	42.1	41.1	79.5	56.5
SPEED	70.2	61.6	51.1	44.2	40.3	79.3	57.8
IDM	71.1	60.2	50.1	44.7	41.9	79.4	57.9
Methods	S40/C0	S40/C10	S40/C20	S40/C30	S40/C40	S40/C50	Average
HRL	67.9	61.3	53.5	43.7	40.1	79.5	57.7
SPEED	72.2	58.18	53.1	40.5	40.2	79.3	57.2
IDM	72.3	61.5	52.1	42.4	41.9	79.1	58.2
Methods	S50/C0	S50/C10	S50/C20	S50/C30	S50/C40	S50/C50	Average
HRL	70.6	61.7	48	41.5	41.9	79.9	57.3
SPEED	67.5	61.8	52.4	42.5	39.6	79.8	57.3
IDM	72.1	61.5	52.7	42.2	42.7	79.9	58.5
HRL_Avg	70.3	61.8	47.3	43.8	42.6	79.7	57.6
SPEED_Avg	69.5	61.4	50.8	42.9	40.9	79.6	57.5
IDM_Avg	71.2	61.4	51.2	43.4	42.4	80.0	58.3

Table 4 shows the comparison of the average energy consumption of a single journey. It is obvious that the proposed HRL method can save energy in all the different situations. Due to the better performance in acceleration control, target speed control and interaction with mixed traffic,

the HRL method can save 12.2% energy comparing with IDM method and can save 47.1% energy comparing with SPEED method.

In addition, according to Table 4, when the initial phase time is C20 (i.e. the start of the yellow light), the HRL method has the relatively best performance. In this situation, the improvement is 24%, even comparing to the IDM method, which is a tremendous enhancement. On the other hand, when the initial phase time is near-zero or the entry speed is too fast, the improvement is only near 3% comparing to IDM method. According to this analysis, we realize that the initial phase time and entry speed will influence the performance of HRL method, which reminds us that there is an adjustment space of the HRL method. Different traffic situations have different adjustment space for ego-vehicle and if we can control the vehicle to enter the intersection with proper adjustment space, the HRL-based eco-driving approach will get its best performance.

TABLE 4: **The average energy consumption per travel for HRL method, SPEED method, and IDM method.**

methods	S10/C0	S10/C10	S10/C20	S10/C30	S10/C40	S10/C50	Average
HRL	43488	37870	38460	33576	33871	41295	38093
SPEED	77105	78726	88713	79917	76013	84277	80791
IDM	44790	41728	40106	36272	35949	43570	40402
Decrease	-2.91%	-9.25%	-4.10%	-7.43%	-5.78%	-5.22%	-5.72%
methods	S20/C0	S20/C10	S20/C20	S20/C30	S20/C40	S20/C50	Average
HRL	43307	36302	28202	27054	27095	42636	34099
SPEED	76061	82930	78222	69645	76947	80955	77460
IDM	45847	43190	41212	29100	29400	45297	39007
Decrease	-5.54%	-15.95%	-31.57%	-7.03%	-7.84%	-5.87%	-12.58%
Methods	S30/C0	S30/C10	S30/C20	S30/C30	S30/C40	S30/C50	Average
HRL	43769	35008	28618	27511	29733	43926	34760
SPEED	77599	83342	82494	59857	78600	85573	77910
IDM	46331	43714	42855	31841	30844	46812	40399
Decrease	-5.53%	-19.92%	-33.22%	-13.60%	-3.60%	-6.17%	-13.96%
methods	S40/C0	S40/C10	S40/C20	S40/C30	S40/C40	S40/C50	Average
HRL	39454	33745	30168	27570	33597	40522	34176
SPEED	79545	87708	85055	78179	77700	88732	82819
IDM	46565	45047	43671	30632	30548	46879	40557
Decrease	-15.27%	-25.09%	-30.92%	-10.00%	9.98%	-13.56%	-15.73%
methods	S50/C0	S50/C10	S50/C20	S50/C30	S50/C40	S50/C50	Average
HRL	39172	34786	31727	29344	32330	42539	34983
SPEED	85188	88754	84358	72231	66488	91066	81347
IDM	43008	41489	39621	36878	37163	43828	40331
Decrease	-8.92%	-16.16%	-19.92%	-20.43%	-13.00%	-2.94%	-13.26%
Average	-7.63%	-17.27%	-23.95%	-11.70%	-4.05%	-6.75%	-12.25%

CONCLUSION

In this paper, we proposed an eco-driving approach for CAV under mixed intersection traffic by designing a hybrid reinforcement learning (HRL) framework. The vehicle activity data from On-Board Diagnosis (OBD), front camera, on-board radar and the SPaT data from intersection are collected by the HRL framework. The output of HRL is longitudinal acceleration value and latitudinal target lane. The HRL framework combines a manual-designed rule-based strategy and a deep reinforcement learning-based policy learning algorithm. The predefined rules are designed on the basis of different driving conditions, which take over the low-level control in specific conditions to ensures the driving safety and efficiency. The eco-driving RL algorithm consists of a data preprocessor that generates spatiotemporal data, a feature extracting network that extracts hidden features, and a policy network. The HRL learns the optimal eco-driving strategy through Long-Short Term Reward (LSTR) algorithm, which considers both short-term and long-term benefits for passing the intersection. Numerical experiments are conducted at a signalized intersection with the mixed traffic situation.

According to the experiments, the proposed HRL method can reduce 12.25%-47.1% energy consumption comparing with IDM and SPEED method and can save 1.2%-6.9% time comparing with IDM. The HRL-based ego-vehicle can not only drive trough a signalized intersection with eco-driving strategy but can also interaction properly with mixed traffic conditions. The proposed framework can also be readily implemented to other types of vehicles by replacing the energy-reward function and vehicle dynamic model. For future work, the performance of the different type of vehicles (e.g. heavy-duty trucks) can be conducted and analyzed. Furthermore, more experiments including micro-simulation and field experiment can be conducted to analyze the performance in more complex situations.

ACKNOWLEDGMENTS

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: X. Author, Y. Author; data collection: Y. Author; analysis and interpretation of results: X. Author, Y. Author. Z. Author; draft manuscript preparation: Y. Author. Z. Author. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] Jia, D., K. Lu, J. Wang, X. Zhang, and X. Shen, A survey on platoon-based vehicular cyber-physical systems. *IEEE communications surveys & tutorials*, Vol. 18, No. 1, 2015, pp. 263–284.

[2] National Greenhouse Gas Emissions Data Report. *U.S. Environ. Protection Agency, Washington, DC*, 2013.

[3] INRIX, National Greenhouse Gas Emissions Data Report. *Los Angeles Tops INRIX Global Congestion Ranking, http://inrix.com/press-releases/scorecard-2017/, accessed July 10*, 2019.

[4] Rios-Torres, J. and A. A. Malikopoulos, A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, No. 5, 2016, pp. 1066–1077.

[5] Lee, J. and B. Park, Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 1, 2012, pp. 81–90.

[6] Lin, P., J. Liu, P. J. Jin, and B. Ran, Autonomous vehicle-intersection coordination method in a connected vehicle environment. *IEEE Intelligent Transportation Systems Magazine*, Vol. 9, No. 4, 2017, pp. 37–47.

[7] Hao, P., K. Boriboonsomsin, C. Wang, G. Wu, and M. Barth, *Connected eco-approach and departure (EAD) system for diesel trucks*, 2018.

[8] Hao, P., G. Wu, K. Boriboonsomsin, and M. J. Barth, Eco-approach and departure (EAD) application for actuated signals in real-world traffic. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 1, 2018, pp. 30–40.

[9] Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, and M. Lanctot, Mastering the game of Go with deep neural networks and tree search. *Nature*, Vol. 529, No. 7587, 2016, pp. 484–489.

[10] Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, 2015, p. 529.

[11] Sallab, A. E., M. Abdou, E. Perot, and S. Yogamani, Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, Vol. 2017, No. 19, 2017, pp. 70–76.

[12] Desjardins, C. and B. Chaib-Draa, Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Transactions on intelligent transportation systems*, Vol. 12, No. 4, 2011, pp. 1248–1260.

[13] Shalev-Shwartz, S., S. Shammah, and A. Shashua, Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

[14] Chen, J., Z. Wang, and M. Tomizuka, Deep hierarchical reinforcement learning for autonomous driving with distinct behaviors. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1239–1244.

[15] Ye, F., G. Wu, K. Boriboonsomsin, and M. J. Barth, A hybrid approach to estimating electric vehicle energy consumption for ecodriving applications. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, pp. 719–724.

[16] Schaul, T., J. Quan, I. Antonoglou, and D. Silver, Prioritized Experience Replay. *Computer Science*, 2015.

[17] Wang, Z., T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, Dueling network architectures for deep reinforcement learning, 2015, pp. 1995–2003.

[18] Kingma, D. and J. Ba, Adam: A Method for Stochastic Optimization. *Computer Science*, 2014.

[19] Bai, Z., B. Cai, W. Shangguan, and L. Chai, Deep Reinforcement Learning Based High-level Driving Behavior Decision-making Model in Heterogeneous Traffic. *arXiv preprint arXiv:1902.05772*, 2019.

[20] Min, K. and H. Kim, Deep Q Learning Based High Level Driving Policy Determination. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 226–231.

[21] Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, TensorFlow: a system for large-scale machine learning, 2016.