IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 May 2024, accepted 26 May 2024, date of publication 3 June 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3408223

RESEARCH ARTICLE

Cooperative Merging Control Based on Reinforcement Learning With Dynamic Waypoint

XIAO YANG[1], HONGFEI LIU[1], MIAO XU[2], AND JINTAO WAN[1]
[1]School of Transportation, Jilin University, Changchun 130022, China
[2]School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China

Corresponding author: Hongfei Liu (hongfeiliu@jlu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4300300, and in part by the National Natural Science Foundation of China under Grant 52372297.

ABSTRACT Reinforcement learning algorithms can cooperate with trajectory planning idea to improve the training efficiency in the field of autonomous driving for the fixed geometric constraints of the road and limited dynamics. In this study, we propose a Dynamic Waypoint Proximal Policy Optimization (DW-PPO) framework for the merging into a platoon scenario, in which the target location is constantly changing as the platoon travels. Specifically, we set up a waypoint generator based on Bezier curve to aid in the composition of the state space and reward calculation. Moreover, we refine the waypoint tracking reward in terms of both distance and direction and add an additional merging reward to complete the merging task. We test our model on three dimensions: learning performance, control performance, and generalization performance and compare with baseline model. Experimental results show that our proposed method has better training efficiency, control stability and generalization ability.

INDEX TERMS Autonomous driving, deep reinforcement learning, merging control, cooperative driving, proximal policy optimization.

I. INTRODUCTION

With the rapid development of autonomous driving and intelligent transportation system (ITS), multi-vehicle cooperative driving has received widespread attention [1], [2]. Multi-vehicle cooperative driving is an advanced technology which uses autonomous driving technologies and intelligent and connected technologies to coordinate the trajectory and motion control of vehicles, making the vehicles run more smoothly and quickly. Platoon is one of the ways of cooperative driving, which can take full advantage of multi-vehicle formation in road efficiency, energy consumption and pollution.

One of the difficulties of cooperative driving is how to take the surrounding traffic flow into account, and scholars have made numerous explorations. An and Talebpour [3] proposed a vehicle platooning algorithm to minimize the disruption from a lane-changing maneuver, which implemented both adaptive cruise control and model predictive control (MPC). The result showed the approach can generate an additional gap for the lane-changing vehicle. Zhang et al. [4] developed a platoon control to coordinate the trajectories of a connected autonomous vehicle (CAV) platoon under a platoon-centered platooning control to accommodate the CAV lane-change requests from its adjacent lane. Liu et al. [5] extended a model which combined a detailed velocity planning strategy and considered more complete driving environment information. These methods are more likely to optimize the planning and control modules individually, and less likely to consider the overall optimization.

Thanks to the developing of end-to-end driving, considerable scholars are increasingly favoring the use of reinforcement learning (RL) to achieve cooperative driving. Different from the field of robotics, cooperative driving has road geometry constraints and limited dynamics, which make it special in accelerating convergence. Yurtsever et al. [6]

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

firstly used the combination of the A* algorithm and deep reinforcement learning (DRL) to achieve path generation and tracking between fixed two points, and the learning speed of the driving task was greatly improved. Since then, there were many neural networks designed to input path information into the network. Chen et al. [7] fed global path and the images from font-facing cameras into the state encoded network to provide the directional navigation information. The route way-points from mission planner are also send to the synthesized primary program to determine which reinforcement learning agent to trigger in the paper [8]. Russo et al. [9] also used the trajectory planner to endow the agent with a capability of tracking a specific trajectory in pedestrian collision avoidance scenarios.

We focus on the DRL approach that incorporates trajectory planning idea in merging task. A waypoint is sequentially chosen to be the objective that the vehicle must reach. Once it is reached, then a next waypoint is recovered (from the path sequence) to make it be the next objective for the vehicle and so on. The destination of merging task is changing with the platoon, which makes it more difficult to apply this method. For one thing, in the navigation task, the path can be determined when the map and destination are given. The waypoint usually is selected at a certain distance according to the path sequence. In the merging task, the destination is constantly changing as the platoon moves. Accordingly the path to destination changes over time and the waypoint is not selected from a fixed sequence. In other words, the update of the waypoint in merging task does not depend on whether the last waypoint is reached, but depends on the change of destination and vehicle position. The update frequency of waypoint is consistent with the environment and is faster than navigation task, which puts forward higher requirements on the ability of agents to learn to track waypoint.

For another, it is easy for vehicle to collide with platoon members when joining the platoon. This leads to the model converging to the suboptimal solution, which is reflected in the simulation that the vehicle will accompany the platoon but not merge into the platoon, which makes the design of reward function harder.

To solve the aforementioned problems, we propose a deep reinforcement learning method combining dynamic waypoint (DW-PPO), which is formed by a dynamic waypoint generator and PPO algorithm of Actor-Critic structure. This training framework can improve the training speed of merging task, and also provide ideas for the reward design of driving tasks with changing destination. The main contributions of this paper are as follows:

1) We combine the ideas of trajectory planning and deep reinforcement learning in an innovative application of a dynamic waypoint generator to the merging task. The generator is responsible for planning a merging path and giving the neural network the waypoint. Different from previous studies, the update frequency of waypoint participating in the state space is faster than that of the navigation task waypoint.

2) Waypoint information is incorporated into the state space in the DW-PPO framework to help with the state space composition and reward computation. We design the waypoint tracking reward more finely to adapt to the high frequency update of waypoint, which is composed of distance reward and direction reward. In navigation task, only distance reward is often used.

3) Aiming at the problem of the agent accompanying the platoon in merging task, we designed an exponentially increasing merging reward item to encourage agents to jump out of the suboptimal solution. In addition, we conduct baseline model comparison to evaluate the superior ability of the proposed model.

The remainder of the paper is organized as follows. Section II presents a comprehensive overview of related works. Section III defines the problem and specific modeling process is presented in section IV. The experiment setup and detailed analysis of results are presented in Section V. Section VI summarizes the study and looks forward to the prospects for future.

II. RELATED WORK

During the past decades, considerable efforts have been devoted to cooperative driving, such as lane changing, merging control, and platoon control. The main research works for these different types of cooperative driving are the same which can be categorized into two aspects: planning and control and end-to-end driving.

A. PLANNING AND CONTROL

The study of cooperative driving has been developing over the past few decades. In the beginning, researchers often used model-based approaches for cooperative driving including quintic polynomial and Bezier curve. The control layers used linear quadratic regulator, MPC and sliding model control [10], [11], [12]. These studies mostly assume that the state of surrounding vehicles remains constant throughout the lane change process, which is not consistent with the actual traffic.

Thanks to the advancement of autonomous driving technologies and ITS, more and more studies began to take the surrounding vehicles dynamic situation into consideration [13], [14], [15], [16]. Wang et al. [17] proposed a strategy to enable centralized decision making and active cooperation to achieve safety, comfort and higher traffic efficiency. Zhang et al. [4] developed a platoon controller, which can coordinate the trajectories of connected autonomous vehicle platoon to accommodate CAV lane change requests from adjacent lanes. Luo et al. [18] designed trajectory planner and sliding mode controller, which can effectively avoid potential collisions during lane changing. Zhang et al. [19] proposed a driving strategy for CAV merging into platoon, and systematically analyzed the stability conditions of multi-vehicle cooperative operation state.

However, the cooperative driving is still not separated from the modular process of planning and control, and it

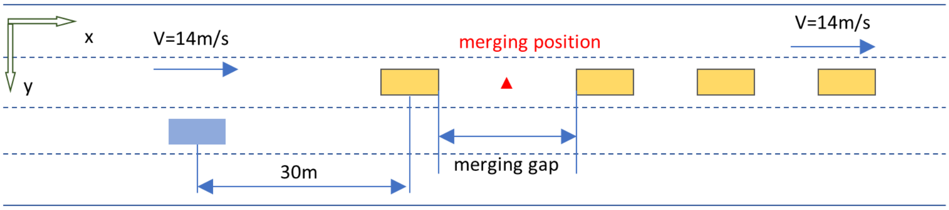


FIGURE 1. Merging into the platoon scenario.

is more difficult to solve the path for complex scenarios. Moreover, the planning and control modules are optimized in isolation, which is hard to guarantee the whole system optimality.

TABLE 1. Abbreviations.

Abbreviation	Full title
DW-PPO	Dynamic Waypoint Proximal Policy Optimization Framework
ITS	Intelligent Transportation System
MPC	Model Predictive Control
CAV	Connected Autonomous Vehicle
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
PPO	Proximal Policy Optimization

B. END-TO-END DRIVING

End-to-end learning can integrate planning and control modules to achieve system optimization. Reinforcement learning (RL) is one of the important methods for end-to-end driving. There are some states where the absence of reward values makes training difficult in RL in the process of modeling real problems [20]. These tasks are referred to as sparse reward tasks, which is also faced in merging scenario. Nowadays, the main solutions to the sparse reward problem are reward shaping [21], [22], [23], Hindsight experience replay [23], and hierarchical reinforcement learning [8]. In this part, we will discuss the relates work on the reward shaping method.

The idea of reward shaping is to provide an additional reward which will improve the performance of the agent. This shaping reward does not come from the environment. It is extra information which is incorporated by the designer of the system and estimated on the basis of knowledge of the problem [24]. The studies related reward shaping can be categorized into single agent and multi-agent based on RL [25], [26], [27], [28], [29].

In terms of the single agent RL approach, Okudo and Yamada [25] extended potential-based reward shaping and propose a subgoal-based reward shaping, which divided a task into a sequence of subgoals and incorporate human subgoal knowledge into the RL algorithm. Yang et al. [26] proposed a reinforcement learning method with hybrid exploration by perfecting the rewards distribution in the state space of environment. Mo et al. [27] demonstrated that reward shaping may be done in a more systematic way by using a set of design principles together with the reward machine. They applied the method to train neural networks with RL that can perform block stacking and block lining up tasks with unspecified repetition.

In terms of the multi-agent RL approach, Huang and Jin [28] investigated the impact of reward shaping in the context of an "L -shape" assembly task that involves collision avoidance. Zhu et al. [29] used two-level agent organization structures and combined reward shaping and action shaping to coordination mechanisms impact learning algorithms.

Considering the sparse reward and the fact that autonomous driving is limited by the road, we use waypoint as non-environmental information and incorporate it into the reward of RL to accelerate the training.

III. PROBLEM DEFINITION

This section defines the problem we are trying to solve. On a one-way four-lane road, the ego-vehicle traveling at 14 m/s decides to join a platoon traveling at 14 m/s 30 m ahead of it. The platoon has left enough space (merging gap) for the ego-vehicle by speed adjustment. The merging position is the midpoint of the merging gap. The scenario is shown in Fig. 1. Our goal is to make the ego-vehicle smoothly join the platoon and enter the cruise state by taking effective acceleration, deceleration, and steering maneuvers. Lane keeping, collision avoidance and maintenance of cruise after entering the platoon are expected to do during the merging.

IV. METHODOLOGY

In this section, to begin with, we introduce the architecture of DW-PPO and then the design of the dynamic waypoint generator is presented. Finally, we show the details of reinforcement learning.

A. DW-PPO FRAMEWORK

The framework consists of three parts which are environment, dynamic waypoint generator and PPO module as shown in Fig. 2. Environment provides observation information about all vehicles. Dynamic waypoint generator outputs a waypoint, and the waypoint cooperates with observations

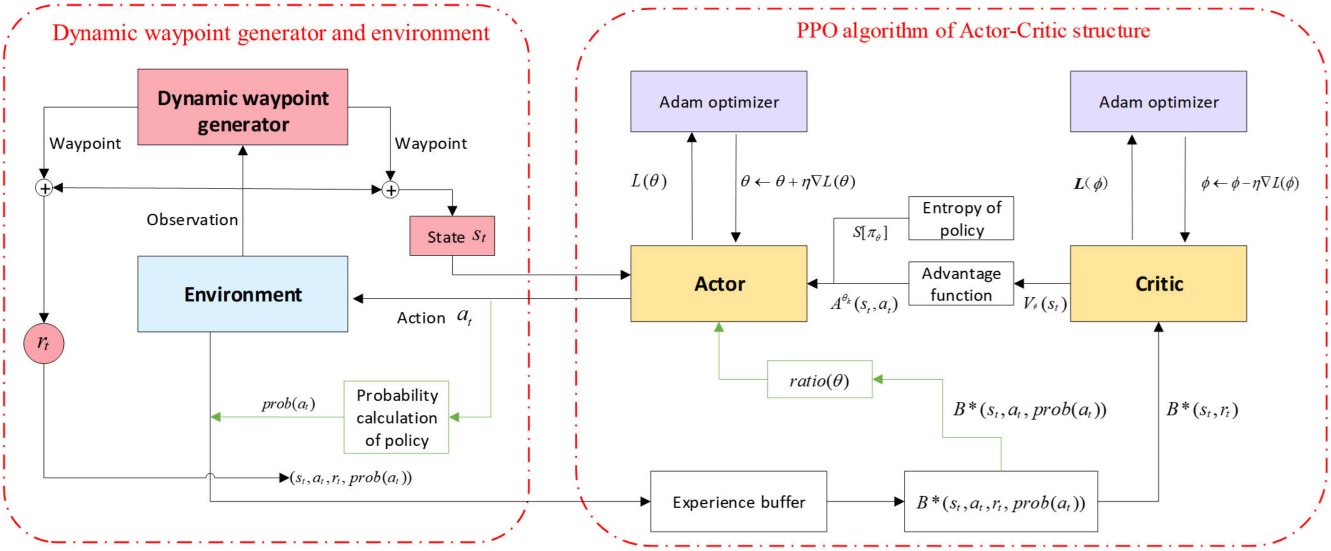

Dynamic waypoint generator and environment
Dynamic waypoint generator
Waypoint
Waypoint
Observation
State s_t
Environment
r_t
Action a_t
$prob(a_t)$
Probability calculation of policy
$(s_t, a_t, r_t, prob(a_t))$
PPO algorithm of Actor-Critic structure
Adam optimizer
Adam optimizer
$L(\theta)$
$\theta \leftarrow \theta + \eta \nabla L(\theta)$
$L(\phi)$
$\phi \leftarrow \phi - \eta \nabla L(\phi)$
Entropy of policy
$S[\pi_\theta]$
Actor
Advantage function
Critic
$A^{\theta_k}(s_t, a_t)$
$V_\phi(s_t)$
$ratio(\theta)$
$B*(s_t, a_t, prob(a_t))$
$B*(s_t, r_t)$
Experience buffer
$B*(s_t, a_t, r_t, prob(a_t))$

FIGURE 2. DW-PPO framework.

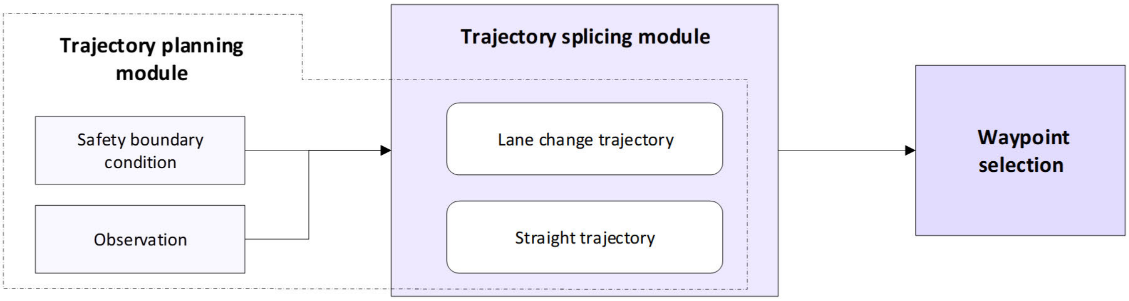

Trajectory planning module
Safety boundary condition
Observation
Trajectory splicing module
Lane change trajectory
Straight trajectory
Waypoint selection

FIGURE 3. Flow chart of waypoint generator.

to form RL state space. It is also an important basis for reward calculation. The RL module decides what action the agent takes. The PPO algorithm uses the off-policy structure to update the actor network, which can enable multiple utilization of data collected at one time by means of importance sampling. In the PPO module, the actor network is used to control the acceleration and steering of ego-vehicle and the critic network calculates the state value function to help the actor network update.

B. DYNAMIC WAYPOINT GENERATOR

In the merging task, the dynamic waypoint generator is composed of trajectory planning module, trajectory splicing module and waypoint selection module, as shown in Fig. 3. The basic process of waypoint determination is as follows:

Step1: Determine the lane change trajectory according to the safety boundary conditions.

Step2: Determine the straight trajectory according to the observation of environment, and splice straight trajectory and lane changing trajectory.

Step3: Select the waypoint according to specific rules and output it.

In step one, the lane change trajectory is obtained from third-order Bezier curve, which is widely used in lane change trajectory planning because of convenient curvature control, simple solution, and good fitting properties. By using it, the lane change trajectory can be transformed into the determination of the four control points as shown in Fig. 4. The Bezier curve can be expressed as (1).

$$P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \tag{1}$$

where P0,P1,P2,P3 represent the control points of the Bezier curve, and t is a proportional quantity in the formation process of the Bezier curve, ranging from 0 to 1. In order to simplify the calculation, we assume that the lane changing trajectory is centrosymmetric about the point o.

P3 is the merging position, whose coordinates are the midpoint of the platoon members before and after the merging gap. Combining with the c_1 and c_2, the coordinates of lane centerline, we can get the information of the control points,

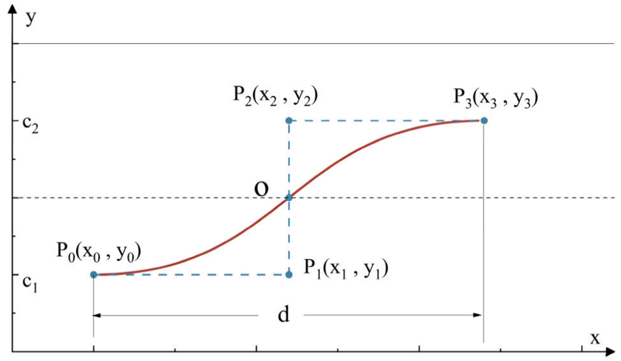


FIGURE 4. A third-order Bezier curve.

as shown in (2).

$$\begin{cases} x_0 = x_3 - d \\ x_1 = x_2 = x_3 - d/2 \\ y_0 = y_1 = c_1 \\ y_2 = y_3 = c_2 \end{cases} \tag{2}$$

where x_i and y_i represent the coordinates of the control point P_i, d is the longitudinal distance of lane change trajectory, c_1 denotes the y coordinate of the current lane where the ego-vehicle is located, c_2 denotes the y coordinate of the target lane.

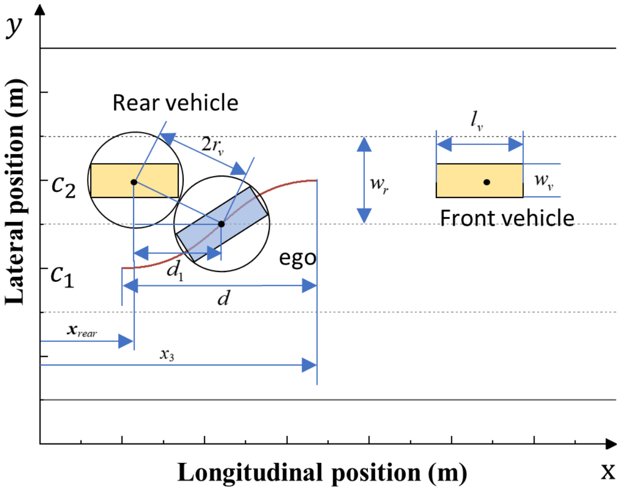


FIGURE 5. Safety boundary and related parameters. d_1 denotes the difference in longitudinal distance between the two vehicles that are likely to collide at the collision moment. x_{rear} denotes the x coordinate of the rear vehicle of the platoon. r_v denotes the radius of the collision circle of the vehicle. l_v denotes the length of the vehicle. w_v denotes the width of the vehicle. w_r denotes the width of the lane.

Then, the range of d can be determined by the safety boundary. We establish the boundary circle for the vehicles which may collide and consider that the collision occurs when the two circles are tangent. The radius of the boundary circle can be obtained from (3) and the meaning of related parameters is shown in Fig. 5.

$$\begin{cases} d_1^2 + \left(\frac{w_r}{2}\right)^2 = (2r_v)^2 \\ r_v^2 = \left(\frac{w_v}{2}\right)^2 + \left(\frac{l_v}{2}\right)^2 \end{cases} \tag{3}$$

As can be seen in Fig. 5, when the distance d_1 is bigger than $x_3 - d/2 - x_{rear}$, the two vehicles do not collide during the lane changing process. This means that when d is greater than $2(x_3 - x_{rear} - d_1)$, the ego-vehicle can safely merge into the platoon. Once d is determined, all control points coordinates can be calculated and so can the lane change trajectory. Note that the coordinates of rear vehicle and P3 are obtained at time t+1 when calculating safety boundaries.

In step two, a straight line is generated from the current position of the ego vehicle to the start of the lane change trajectory. Note that the straight line follows the centerline of the lane. Then, the two trajectories are joined together to formthe merging trajectory. Fig. 6 shows the merging trajectories at different initial positions.

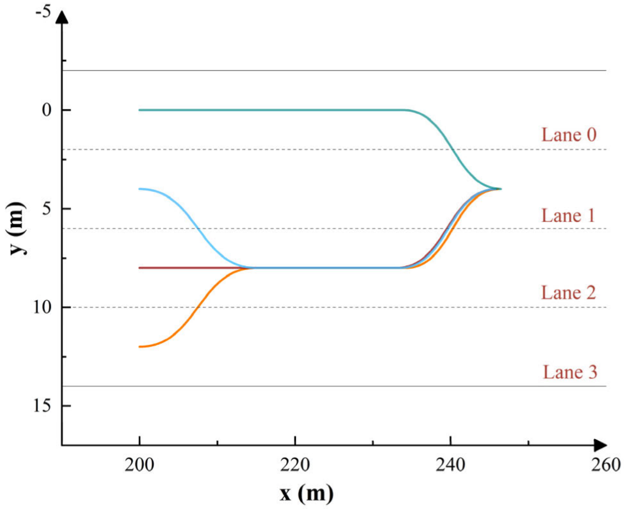


FIGURE 6. Merging trajectories for different initial lanes.

In step three, when the ego-vehicle is in the straight phase, the point 4 m in front of the vehicle on the trajectory is selected as the waypoint. When the vehicle is in the lane change phase, the waypoint 2 m in front of the vehicle is selected as the waypoint.

C. REINFORCEMENT LEARNING

1) PROXIMAL POLICY OPTIMIZATION

Proximal Policy Optimization (PPO) [30] is a model-free reinforcement learning algorithm based on policy gradient, which is faster, more efficient, and more robust than the Trust region policy optimization. The PPO cleverly improves the efficiency of the sample utilization by importance sampling and the addition of a clipping term to limit the difference between the old and the new policies. The alternative objective function is (4).

$$\mathcal{L}^{clip}(\theta) = \mathbb{E}\left[\min\left(\begin{array}{l} ratio(\theta)A^{\theta_k}(s_t,a_t),\\ clip\,(ratio(\theta),1-\varepsilon,1+\varepsilon)\,A^{\theta_k}(s_t,a_t)\end{array}\right)\right] \quad (4)$$

where $ratio(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ denotes the probability ratio between the old and new policy, which is used to compensate for the gap between the training data distribution and the current strategy state distribution. $A^{\theta_k}(s_t, a_t)$ is the advantage function, which represents the deviation of the action a_t in the state s_t relative to the action mean.

In our study, PPO algorithm of Actor-Critic structure is adopted. In order to encourage policy exploration, entropy of policy $S[\pi_\theta]$ is added to the objective function. The objective function is expressed as (5).

$$L(\theta) = L^{clip}(\theta) + \lambda S[\pi_\theta] \quad (5)$$

$S[\pi_\theta]$ is used to measure the uncertainty of the policy. The greater the entropy of the policy, the more uniform the probability distribution of action in each state, and the more exploratory the policy is. The entropy of a policy can be calculated using (6).

$$S[\pi_\theta] = -\mathbb{E}_{a\sim\pi*\theta}[\log \pi_\theta(a|s)] \quad (6)$$

where $\pi_\theta(a|s)$ is the probability of taking an action a in the state s, $\log \pi_\theta(a|s)$ is the logarithmic probability.

The actor network are updated by the Adam optimizer performing gradient ascent as shown in (7).

$$\theta' \leftarrow \theta + \eta\nabla_\theta \mathcal{L}^{clip}(\theta) \quad (7)$$

where θ' is the parameter of the new neural network, η is the learning rate.

The network architecture of our method is shown in Fig. 7. In the actor network, all layers are comprised of ReLU activated neurons and the final layer of the actor network which determines acceleration and steering. The critic network is a three-layer neural network. All layers are also comprised of ReLU activated neurons. Critic network calculates the state value function to help the actor network update. The training process of DW-PPO is shown in Table 2, and the data flow of PPO algorithm is shown in the right part of Fig. 2.

2) STATE SPACE

In our study, the state space is defined as $S = [V_i, N]$, $i = 0, \ldots, n$, which is an $(n+1)$*6 dimensional vector. n is the number of vehicles in the environment.

V_i is the state of each vehicle in the environment. It can be expressed as $[x, y, v_x, v_y, \cos(h), \sin(h)]$, which contains the vehicle's longitudinal and lateral position, longitudinal velocity, lateral velocity, and two trigonometric values of the vehicle's heading angle, respectively. Note that the position and speed of platoon are the relative state with ego vehicle as the reference system. For example, the state of a vehicle in the platoon is $V_1 = [-30, 2, 3, 0, 1, 0]$, which indicates that the vehicle is 30 meters behind and 2 meters to the right of the ego-vehicle, the longitudinal speed is 3 m/s faster than

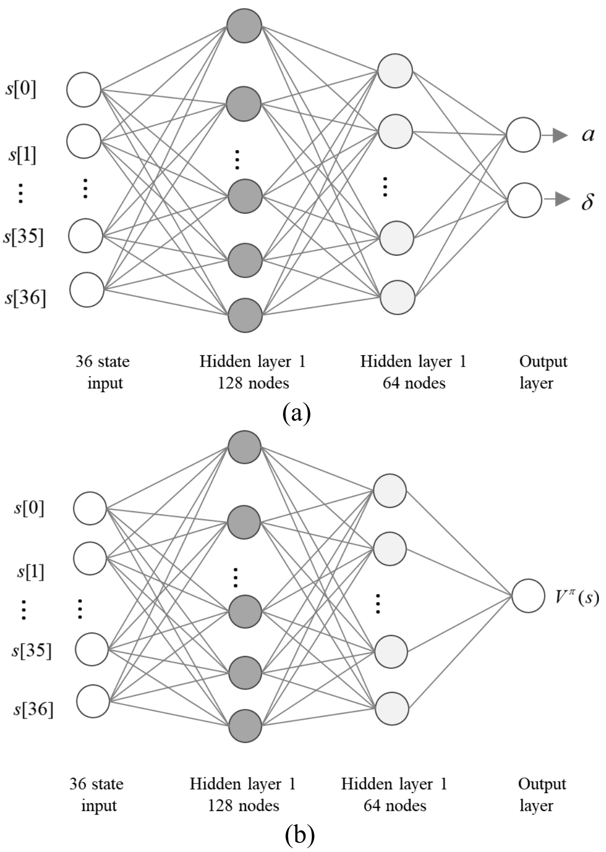
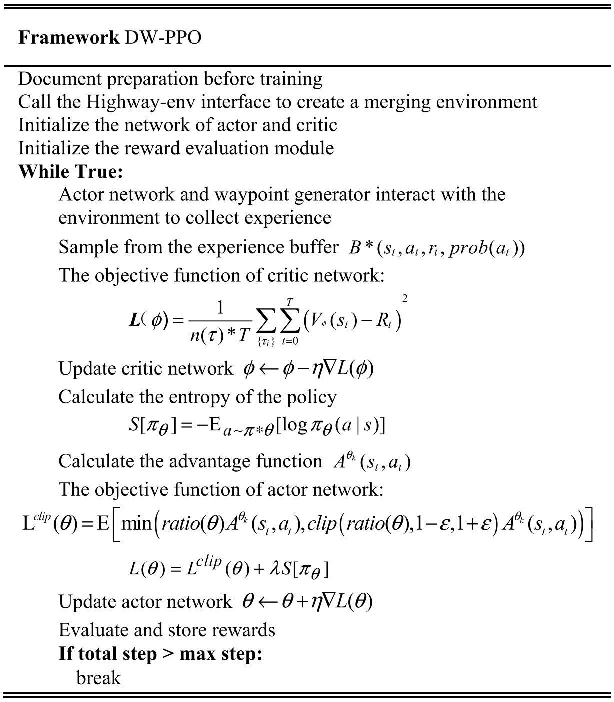


FIGURE 7. Network architecture of DW-PPO. (a) Actor network. (b) Critic network.

TABLE 2. The training process of DW-PPO.



```
Framework DW-PPO
Document preparation before training
Call the Highway-env interface to create a merging environment
Initialize the network of actor and critic
Initialize the reward evaluation module
While True:
    Actor network and waypoint generator interact with the
    environment to collect experience
    Sample from the experience buffer $B*(s_t, a_t, r_t, prob(a_t))$
    The objective function of critic network:
        $L(\phi) = \frac{1}{n(\tau)*T}\sum_{\{\tau_i\}}\sum_{t=0}^{T}\left(V_\phi(s_t) - R_t\right)^2$
    Update critic network $\phi \leftarrow \phi - \eta\nabla L(\phi)$
    Calculate the entropy of the policy
        $S[\pi_\theta] = -\mathrm{E}_{a\sim\pi*\theta}[\log \pi_\theta(a|s)]$
    Calculate the advantage function $A^{\theta_k}(s_t, a_t)$
    The objective function of actor network:
    $\mathrm{L}^{clip}(\theta) = \mathrm{E}\left[\min\left(ratio(\theta)A^{\theta_k}(s_t,a_t), clip\left(ratio(\theta),1-\varepsilon,1+\varepsilon\right)A^{\theta_k}(s_t,a_t)\right)\right]$
        $L(\theta) = L^{clip}(\theta) + \lambda S[\pi_\theta]$
    Update actor network $\theta \leftarrow \theta + \eta\nabla L(\theta)$
    Evaluate and store rewards
    If total step > max step:
        break
```

ego, the lateral speed is the same as ego, and the vehicle's heading angle is 0.

N is related to the state of waypoint, which is a six-dimensional vector. It can be expressed as $[\Delta p_x^{t-1}, \Delta p_y^{t-1}, \Delta p_x^t, \Delta p_y^t, \cos(p), \sin(p)]$, which contains the difference between the vehicle and the waypoint in the longitudinal and lateral directions at timestep t-1, the difference between the vehicle and the waypoint in longitudinal and lateral directions at timestep t, and thetrigonometric values of the expected vehicle heading angle at timestep t. The meanings of each parameter are shown in Fig. 8. The first two items in the expression represent the t-1 time waypoint information. The closer the agent is to the waypoint of t-1 time, the better the action agent takes at t-1 time. The waypoint at timestep t-1 can facilitate the critic network to learn the relationship between state and reward. The last four terms represent the distance information and direction information of the waypoint at time t, so that the agent can drive to the waypoint.

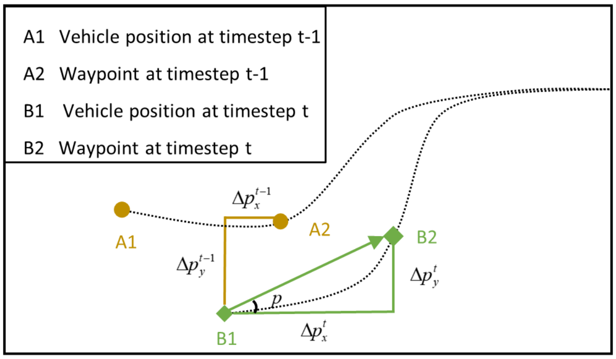

A1 Vehicle position at timestep t-1
A2 Waypoint at timestep t-1
B1 Vehicle position at timestep t
B2 Waypoint at timestep t
Δp_x^{t-1}
A1
A2
B2
Δp_y^{t-1}
Δp_y^t
p
B1
Δp_x^t

FIGURE 8. State parameters related to the waypoint.

3) ACTION SPACE

The action space is defined as $A = [a, \delta]$, which is a two-dimensional vector. It consists of acceleration a and steering wheel angle δ and both normalized to $[-1,1]$. For acceleration, positive numbers represent acceleration, negative numbers represent deceleration, and the acceleration range is $[-2, 2]$ m/s^2. Similarly, negative steering wheel angle represents left turn and positive numbers represents right turn, and the steering wheel angle is limited to [-pi/36, pi/36] [31].

4) REWARD

Reward is one of the most significant elements of RL, which provides a guide for the training process. In the DW-PPO framework, the reward is divided into four parts, as shown in (8).

$$r_{total} = c_1 r_{track} + c_2 r_{speed} + c_3 r_{center} + c_4 r_{merge} \tag{8}$$

where r_{track} is the waypoint tracking reward, r_{speed} is the speed reward, r_{center} is the lane center reward and r_{merge} is the merging reward. $c1, c2, c3, c4$ is the weight coefficient of each reward, the values of which are 1,1,0.5, 1 respectively.

Waypoint tracking reward r_{track} is designed from two aspects: distance reward r_{dis} and direction reward $r_{direction}$ as shown in (9).

$$\begin{cases} r_{track} = r_{dis} + r_{direction} \\ r_{dis} = D/D_0 \\ r_{direction} = -(arc\cos(\vec{a} \bullet \vec{b} /|\vec{a}||\vec{b}|)/\alpha \end{cases} \tag{9}$$

where D means the actual displacement of the agent, D_o is the planned displacement, $\vec{a}$ is the expected direction of travel at timestep t-1, $\vec{b}$ is the actual driving direction of the ego-vehicle, the meaning is shown in Fig. 9. α is taken as pi/2, which represents the maximum angle between $\vec{a}$ and $\vec{b}$.

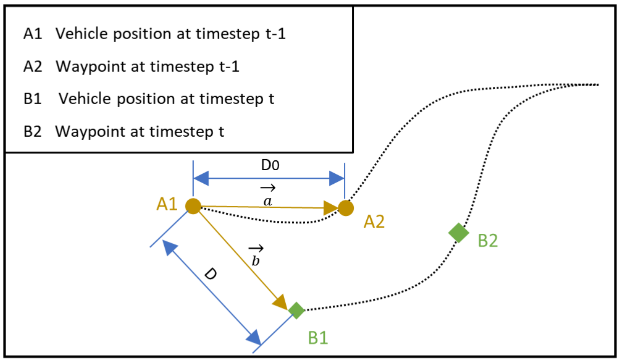

A1 Vehicle position at timestep t-1
A2 Waypoint at timestep t-1
B1 Vehicle position at timestep t
B2 Waypoint at timestep t
D0
$\vec{a}$
A1
A2
$\vec{b}$
D
B2
B1

FIGURE 9. The parameters of the direction reward.

The distance reward r_{dis} indicates that the closer the ego-vehicle is to the waypoint at timestep $t-1$, the larger the distance reward is. The direction reward $r_{direction}$ indicates that the greater the deviation between the vehicle's actual driving direction and the expected driving direction, the smaller the direction reward is.

The purpose of the speed reward r_{speed} is to allow the agent to travel faster than the platoon and stay within the lane speed limit. In addition, another purpose is to encourage the agent to accelerate to explore more possible actions. The ego-vehicle always receives a small positive distance reward as long as it does not go backwards, which keeps the agent from actively accelerating. The expression of r_{speed} is shown in (10).

$$r_{speed} = \begin{cases} -10, & v < 5 \text{ or } v > 20 \\ v/v_{platoon} - 1, & \text{else} \end{cases} \tag{10}$$

Lane center reward r_{center} is designed to keep the vehicle from crossing lane line into the adjacent lane while the agent is accelerating to catch up with the platoon. For merging task where the target point is always changing, tracking rewards alone cannot guarantee that the RL training can converge, so lane keeping rewards play an irreplaceable role in the initial guided exploration. The expression of lane keeping reward is shown in (11).

$$r_{center} = \frac{1}{2}(m/m_0 + w/w_0)^2 \tag{11}$$

where w_0 denotes is half the width of the lane, w is the minimum distance between the center of mass of the vehicle and the lane line on both sides, m is the nearest distance of vehicle from the lane line, which can be obtained from (12) and m_0 is the maximum value that can be achieved in the lane.

$$\begin{cases} \tan(\theta_2) = w_{vehicle}/l_{vehicle} \\ \theta_1 + \theta_2 + \gamma = \pi/2 \\ x' = x + \sin(\gamma)\sqrt{w_{vehicle}^2 + l_{vehicle}^2} \\ y' = y + \cos(\gamma)\sqrt{w_{vehicle}^2 + l_{vehicle}^2} \\ m = y_{lane\_right} - y' \end{cases} \tag{12}$$

where (x, y) represents the coordinates of the center of mass of the vehicle, (x', y') represents the coordinates of the vehicle corner point where the vehicle may exceed the boundary line, θ_1 represents the heading angle, θ_2 is determined by the geometric dimensions of the vehicle, and γ is used to help calculate the coordinates. Fig. 10 shows the meanings of these parameters.

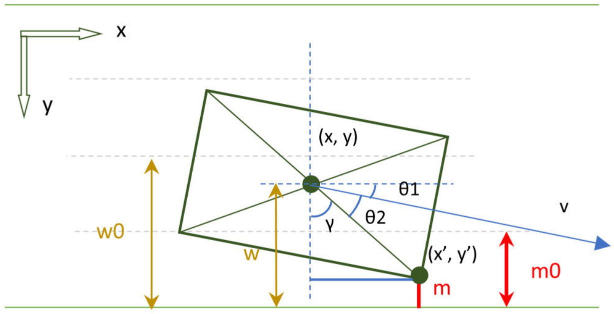


FIGURE 10. The parameters of the lane center reward.

The merging reward r_{merge} is utilized to encourage vehicles to enter the platoon. The merging reward will be 5 times larger than the tracking reward, when the distance between the agent and the destination is less than 1.5 times the width of the road. After entering the platoon, the ego-vehicle will easily be occurred collision with the front or rear vehicle due to improper action (acceleration or deceleration). So the ego-vehicle may choose to accompany the platoon to get a smaller reward instead of joining the platoon. That's why it's necessary to have merging reward to encourage the ego-vehicle to enter the platoon. The expression is shown in (13).

$$r_{merge} = \begin{cases} 5 \times 1.1^{-\Delta y} & \Delta \leq 1.5 w_r \\ 0, \end{cases} \tag{13}$$

Δ denotes the distance of the vehicle from the destination and the lateral distance of the vehicle from the destination is Δy.

V. EXPERIMENTS AND RESULTS

This section describes the experimental part and reports on the performance of our approach in Highway-Env [32], which is an open environment for autonomous driving.

A. TRAINING SETTINGS

These experiments were performed on a computer equipped with an Intel(R) Core i7-12700F CPU (12 cores 2.10 GHz), 32GB RAM and a NVIDIA GeForce GTX 1660 SUPER GPU and all code was written in the pytorch framework. The resource utilization of the gpu fluctuates between 10% and 90% because the data transfer to the gpu is a one-time transmission. The four million steps took about 15 hours. The size of the experience buffer is 512 and each experience is reused 8 times during training.

On a one-way four-lane road from west to east, the ego-vehicle traveling at 14 m/s decided to join the platoon traveling at 14 m/s 30 m ahead of it, which has got with a merging gap. All lanes had a speed limit of 5-20 m/s with width of 4 m. All vehicles are 5 m long and 2 m wide. During the training process, the maximum number of steps in a round was set to 250 to prevent the agent from falling into a local optimum. If the vehicle went out of road or collided with a platoon member, the round ended. The hyperparameters of DW-PPO method are set as shown in Table 3. The parameters of the two models are different because the DW-PPO model cannot converge when the same hyperparameters are used, so some of the hyperparameters are adjusted according to experience. The reward evaluation process is shown in Table 4.

TABLE 3. Hyperparameters for DW-PPO and baseline.

Parameters	DW-PPO	baseline
Optimizer	Adam	Adam
Learning rate	6e-6	6e-4
Discount factor	0.99	0.99
Repeat times	8	8
Horizon length	512	1024
Ratio clip	0.1	0.1
Lambda entropy	0.03	0.007
GAE advantage	0.98	0.98

TABLE 4. Reward evaluation procedure.

```
Reward evaluation process
While True:
    Collect experiences and get the experience buffer
    Sampled from the experience buffer $B*(s_t, a_t, r_t, prob(a_t))$
    Update critic network and actor network
    If update step> evaluate steps:
        Load the latest actor network
        for k = 0, 1, 2:
            Store accumulated rewards per round
        Average the three rewards to get the round cumulative reward
    If total step > max step:
        break
```

B. BASELINE MODEL SETTINGS

The baseline model was training in stages. In the first stage, the vehicle learned longitudinal speed control to ensure that the ego-vehicle explores around the platoon. In the second stage, the vehicle learned the logitudinal and lateral

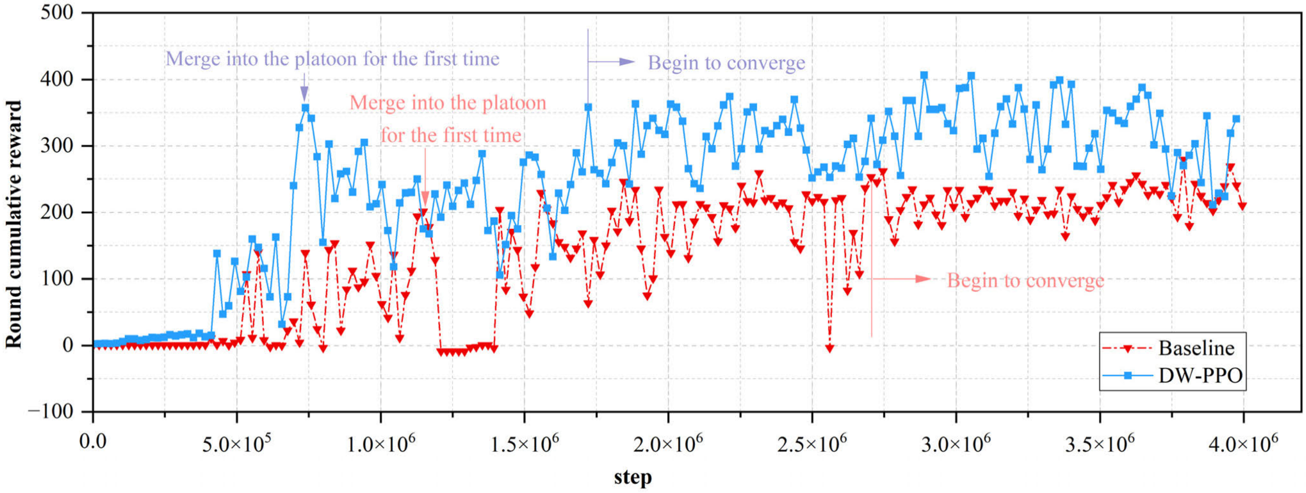


FIGURE 11. Cumulative reward curve of the training process.

control to complete the merging task. The hyperparameters of the baseline model are shown in Table 3. There is no waypoint guidance information in the baseline model state space and reward function.

In order to complete the merging task, the reward of the baseline model is designed from four aspects, and the reward function is (14).

$$r = r_{speed} + r_{dis} + r_{same} + r_{penalty} \tag{14}$$

where r_{speed} is the speed reward, r_{dis} is the distance reward and r_{same} is the state similarity reward, $r_{penalty}$ is penalty term.

The speed reward r_{speed} encourages the speed of ego-vehicle to be maintained at 13-15 m/s, as shown in (15).

$$r_{speed} = \begin{cases} 10, & \Delta v \leq 2 \\ -10/3 \times \Delta v + 50/3, 2 < \Delta v < 5 \\ -50, & else \end{cases} \tag{15}$$

where Δv represents speed difference between the vehicle and the platoon.

The distance reward r_{dis} allows the ego-vehicle to approach the merging position, as shown in (16).

$$r_{dis} = \begin{cases} -2.3d + 69, & 6 < d < 30 \\ 100 \times 1.1^{-d}, & 0 < d \leq 6 \\ -10, & else \end{cases} \tag{16}$$

where d is distance between the vehicle and the merging position.The state similarity reward r_{same} is set to assist the distance reward to keep the agent in a cruise state. as shown in (17).

$$r_{same} = \sqrt{(S_{ego} - S_{desire}) \bullet w} \tag{17}$$

where S_{ego} is the state of the ego-vehicle, S_{desire} the expected state of the vehicle after merging into the platoon, S can be expressed as $[x, y, v_x, v_y, \cos(h), \sin(h)]$. w is the weight vector, the value is [1, 3, 1, 1, 1, 1].

$r_{penalty}$ means that when the vehicle is involved in a collision or drives out of the road, the vehicle will receive a penalty, as shown in (18).

$$r_{penalty} = \begin{cases} -10, & \text{if collision or out of road} \\ 0, & \text{if else} \end{cases} \tag{18}$$

C. LEARNING PERFORMANCE

In this study, we first test the learning performance of the proposed the method and compare it with the baseline method to demonstrate the advantage of DW-PPO framework. Since the reward is a common criterion to evaluate the learning performance of reinforcement learning methods, the curve of the reward with respect to the iteration step (step-reward curve), as shown in Fig. 11. We also use the time to complete the merging task for the first time as another criterion.

The curve in Fig. 11 shows that DW-PPO framework has better learning performance than the baseline model. Since the agent is in the exploratory phase before the first 400,000 steps, the rewards based on the two methods do not increase, both agents performe poorly, turning left or right and then out of bounds. The reward of the DW-PPO method increases significantly starting at 410,000 steps and completing the merging task for the first time after exploring 300,000 steps. The reward of the baseline model increases significantly starting at 530,000 steps and completing the merging task for the first time after exploring 610,000 steps. In terms of the time to complete the merging task for the first time, DW-PPO is about 400,000 steps faster than the baseline model, indicating that DW-PPO learns faster.

In addition, since there is no quantitative index to judge the convergence of RL, it is generally believed that RL when the policy level is no longer significantly improving. In the cumulative reward curve, this means that the reward no longer changes significantly. We use this criterion and empirically determine the timepoint of convergence in Fig. 10. We can find that DW-PPO method converges earlier than baseline model in stages, which also helps to prove that the learning performance is better.

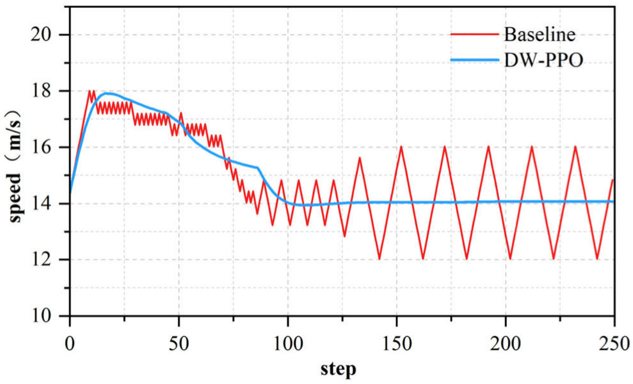


FIGURE 12. Speed control performance.

D. CONTROL PERFORMANCE

We focus on the model performance to control speed and trajectory.

As can be seen from Fig. 12, the agents based on the two methods adopt the strategy of first accelerating to catch up and then slowing down to join the platoon. The speed curve of DW-PPO is smoother, the speed control stability is better. DW-PPO model had a smoother curve during deceleration and the whole process of speed fluctuation was about 0.4 m/s. In contrast, in process of deceleration speed stepped down and fluctuation amplified from 1.6 m/s to 4 m/s, which indicated the baseline model had poor ability to maintain cruise control after entering the platoon.

We divided the merging process into three phase of lane keeping, lane changing and cruise to evaluate the merging trajectory, as show in Fig. 13. The dividing standard is as follows: the trajectory between the initial lane center line and the target center line is defined as the lane changing trajectory, the left side of the lane changing trajectory is the lane keeping phase and the right side is the cruise phase.

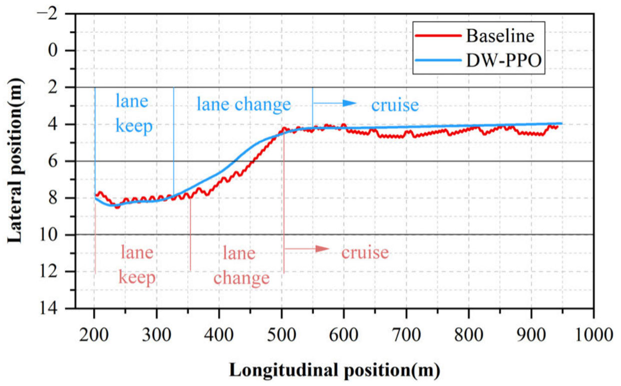


FIGURE 13. Trajectory control performance.

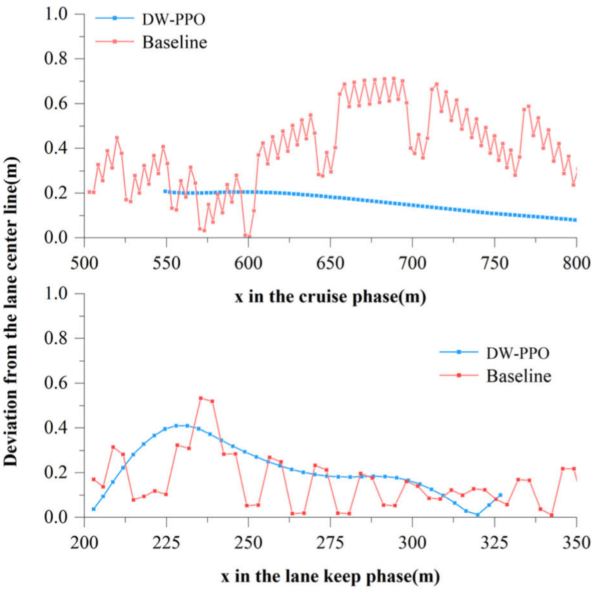


FIGURE 14. Distance error to center comparison.

The goal of the lane keeping stage and cruise stage is to drive along the lane center line, and the distance error of the lane center line is mainly compared. Fig.14 shows a comparison of distance errors between our model and the baseline model. We can see that the distance from the center line of the DW-PPO method is within 0.4 m in both the lane keeping stage and the cruise stage. However, the maximum deviation distance of the baseline model reached 0.7 m.

TABLE 5. Control performance comparison.

Phase		DW-PPO	Baseline
Lane keeping phase	Mean	0.213	**0.158**
	Std	**0.114**	0.122
Length of lane change trajectory		223	**153**
Lane keeping in cruise phase	Mean	**0.112**	0.369
	Std	**0.070**	0.175

We also compare the average distance error and standard deviation of the models for a quantitative analysis, which is shown in Table 5. The average distance error represents the difference between the trajectory and the control target, which can reflect the control accuracy of the model. The standard deviation reflects the stability of the control strategy. A low standard deviation implies stable driving behavior. From Table 5, we can find that DW-PPO has better control stability but needs to improve control accuracy in the lane keeping stage, with a higher mean value of 0.213 m and a lower variance of 0.114. In the lane change stage, the trajectory length based on DW-PPO method is 223 m, and the lane change efficiency is slightly lower than that of the baseline. In the cruise stage, we can find that our proposed method exhibits better control performance than baseline method, it has a lower average distance and standard deviation. The result shows that the mean distance to lane center line error is 0.112 m and the standard deviation is 0.070.

To sum up, the speed control and trajectory control stability of DW-PPO are better, and the accuracy of trajectory control in lane keeping stage needs to be further improved.

E. GENERALIZATION PERFORMANCE

We conduct a study on the generalization performance of the model. We forme a new environment by randomly initializing three parameters, namely, initial speed of the vehicle, platoon speed and distance from the rear car of the platoon to investigate the success rate of the two methods. The value range of each parameter is shown in Table 6. The merging task fails when the following situations occur:

TABLE 6. **The value table of environment parameters is randomly initialized.**

Parameter	Unit	Range of parameter values
Initial speed of the vehicle	m/s	[10, 20]
Platoon speed	m/s	[10, 20]
Distance from the rear car of the platoon	m	[10, 50]

a. When a collision occurs, the round is terminated abnormally.

b. The lane change phase is not completed within the specified 250 steps, resulting in the agent not entering the target lane.

c. In the lane change phase, the vehicle successfully enters the platoon, but in the cruise phase, it exits the target lane.

TABLE 7. **Generalization test results statistics.**

Number of test rounds		Success rate	success	failure
200	DW-PPO	**100%**	200	0
	Baseline	84.5%	169	31
500	DW-PPO	**98.8%**	494	6
	Baseline	83.0%	415	85

From Table 7, we can see that DW-PPO has higher model generalization ability regardless of whether the test round is 200 or 500. Observing the test process, the main reason for the failure of the baseline model is that the lane change stage is not completed within 250 steps when the platoon speed is greater than 18m/s.

VI. CONCLUSION

In this paper, we propose a DW-PPO framework that incorporates dynamic waypoint for improving the training speed of merging task. Specifically we design a waypoint generator for the scenario of merging into a platoon based on third-order Bessel curves, which generates a merging path based on the vehicle's location and provides the neural network with the waypoint at the next timestep. Moreover, we refine the waypoint tracking reward in terms of distance and direction and add an additional merging reward to complete the merging task. In order to evaluate the performance of the model, a baseline model comparison is used and the experimental results show that our method can improve the training efficiency and produce better control stability performance compared with the baseline. The accuracy of trajectory control still needs to be improved.

Since the scenario does not consider other free-flowing vehicles on the road, our future work will focus on introducing different densities of traffic flow that allow the agent to achieve obstacle avoidance during merging into a platoon.

ACKNOWLEDGMENT

The authors would like to thank all reviewers for their valuable comments and helpful suggestions.

REFERENCES

[1] R. Kianfar, B. Augusto, A. Ebadighajari, U. Hakeem, J. Nilsson, A. Raza, R. S. Tabar, N. V. Irukulapati, C. Englund, P. Falcone, S. Papanastasiou, L. Svensson, and H. Wymeersch, "Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 994–1007, Sep. 2012, doi: 10.1109/TITS.2012.2186513.

[2] J. Ploeg, E. Semsar-Kazerooni, A. I. M. Medina, J. F. C. M. de Jongh, J. van de Sluis, A. Voronov, C. Englund, R. J. Bril, H. Salunkhe, Á. Arrúe, A. Ruano, L. García-Sol, E. van Nunen, and N. van de Wouw, "Cooperative automated maneuvering at the 2016 grand cooperative driving challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1213–1226, Apr. 2018, doi: 10.1109/TITS.2017.2765669.

[3] G. An and A. Talebpour, "Vehicle platooning for merge coordination in a connected driving environment: A hybrid ACC-DMPC approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5239–5248, May 2023, doi: 10.1109/TITS.2023.3252567.

[4] H. Zhang, L. Du, and J. Shen, "Hybrid MPC system for platoon based cooperative lane change control using machine learning aided distributed optimization," *Transp. Res. B, Methodol.*, vol. 159, pp. 104–142, May 2022, doi: 10.1016/j.trb.2021.10.006.

[5] X. Liu, J. Liang, and J. Fu, "A dynamic trajectory planning method for lane-changing maneuver of connected and automated vehicles," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 235, no. 7, pp. 1808–1824, Jun. 2021, doi: 10.1177/0954407020982712.

[6] E. Yurtsever, L. Capito, K. Redmill, and U. Ozguner, "Integrating deep reinforcement learning with model-based path planners for automated driving," 2020, *arXiv:2002.00434*.

[7] L. Chen, X. Hu, B. Tang, and Y. Cheng, "Conditional DQN-based motion planning with fuzzy logic for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 2966–2977, Apr. 2022, doi: 10.1109/TITS.2020.3025671.

[8] B. Gangopadhyay, H. Soora, and P. Dasgupta, "Hierarchical program-triggered reinforcement learning agents for automated driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10902–10911, Aug. 2022, doi: 10.1109/TITS.2021.3096998.

[9] L. Russo, M. Terlizzi, M. Tipaldi, and L. Glielmo, "A reinforcement learning approach for pedestrian collision avoidance and trajectory tracking in autonomous driving systems," in *Proc. 5th Int. Conf. Control Fault-Tolerant Syst. (SysTol)*, Saint-Raphael, France, Sep. 2021, pp. 44–49.

[10] I. Papadimitriou and M. Tomizuka, "Fast lane changing computations using polynomials," in *Proc. Amer. Control Conf.*, Denver, CO, USA, 2003, pp. 48–53.

[11] J. Chen, P. Zhao, T. Mei, and H. Liang, "Lane change path planning based on piecewise Bezier curve for autonomous vehicle," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Jul. 2013, pp. 17–22.

[12] P. Feng, H. Jin, L. Zhao, and M. Lu, "Active lane-changing control of intelligent vehicle on curved section of expressway," *Model. Simul. Eng.*, vol. 2022, pp. 1–12, May 2022, doi: 10.1155/2022/9374118.

[13] M. Xu, Y. Luo, G. Yang, W. Kong, and K. Li, "Dynamic cooperative automated lane-change maneuver based on minimum safety spacing model," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1537–1544.

[14] D. Yang, S. Zheng, C. Wen, P. J. Jin, and B. Ran, "A dynamic lane-changing trajectory planning model for automated vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 228–247, Oct. 2018, doi: 10.1016/j.trc.2018.06.007.

[15] Y. Ding, W. Zhuang, L. Wang, J. Liu, L. Guvenc, and Z. Li, "Safe and optimal lane-change path planning for automated driving," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 235, no. 4, pp. 1070–1083, Mar. 2021, doi: 10.1177/0954407020913735.

[16] S. Zhang, G. Deng, E. Yang, and J. Ou, "Optimal vehicle lane change trajectory planning in multi-vehicle traffic environments," *Appl. Sci.*, vol. 12, no. 19, p. 9662, Sep. 2022, doi: 10.3390/app12199662.

[17] D. Wang, M. Hu, Y. Wang, J. Wang, H. Qin, and Y. Bian, "Model predictive control–based cooperative lane change strategy for improving traffic flow," *Adv. Mech. Eng.*, vol. 8, no. 2, Feb. 2016, Art. no. 168781401663299, doi: 10.1177/1687814016632992.

[18] Y. Luo, Y. Xiang, K. Cao, and K. Li, "A dynamic automated lane change maneuver based on vehicle-to-vehicle communication," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 87–102, Jan. 2016, doi: 10.1016/j.trc.2015.11.011.

[19] R.-H. Zhang, F. You, and X.-N. Chu, "Lane change merging control method for unmanned vehicle under V2V cooperative environment," *China J. Highway Transp.*, vol. 31, no. 4, pp. 180–191, 2018, doi: 10.19721/j.cnki.1001-7372.2018.04.022.

[20] J.-W. Zhang, S. Lv, and Z.-H. Zhang, "Survey on deep reinforcement learning methods based on sample efficiency optimization," *J. Softw.*, vol. 33, no. 11, pp. 4217–4238, 2022, doi: 10.13328/j.cnki.jos.006391.

[21] J. Sang, Y. Wang, W. Ding, Z. Ahmadkhan, and L. Xu, "Reward shaping with hierarchical graph topology," *Pattern Recognit.*, vol. 143, Nov. 2023, Art. no. 109746, doi: 10.1016/j.patcog.2023.109746.

[22] L. Anzalone, P. Barra, and S. Barra, "An end-to-end curriculum learning approach for autonomous driving scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19817–19826, Oct. 2022, doi: 10.1109/TITS.2022.3160673.

[23] S. Lanka and T. Wu, "ARCHER: Aggressive rewards to counter bias in hindsight experience replay," 2018, *arXiv:1809.02070*.

[24] M. Grześ and D. Kudenko, "Online learning of shaping rewards in reinforcement learning," *Neural Netw.*, vol. 23, no. 4, pp. 541–550, May 2010, doi: 10.1016/j.neunet.2010.01.001.

[25] T. Okudo and S. Yamada, "Subgoal-based reward shaping to improve efficiency in reinforcement learning," *IEEE Access*, vol. 9, pp. 97557–97568, 2021, doi: 10.1109/ACCESS.2021.3090364.

[26] Y. Yang, W. Cao, L. Guo, C. Gan, and M. Wu, "Reinforcement learning with reward shaping and hybrid exploration in sparse reward scenes," in *Proc. IEEE 6th Int. Conf. Ind. Cyber-Phys. Syst. (ICPS)*, Wuhan, China, May 2023, pp. 1–6.

[27] Y.-W. Mo, C. Ho, and C.-T. King, "Managing shaping complexity in reinforcement learning with state machines–using robotic tasks with unspecified repetition as an example," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Guilin, China, Aug. 2022, pp. 544–550.

[28] B. Huang and Y. Jin, "Reward shaping in multiagent reinforcement learning for self-organizing systems in assembly tasks," *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101800, doi: 10.1016/j.aei.2022.101800.

[29] X. Zhu, C. Zhang, and V. Lesser, "Combining dynamic reward shaping and action shaping for coordinating multi-agent learning," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI) Intell. Agent Technol. (IAT)*, vol. 2, Nov. 2013, pp. 321–328.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[31] Y. Feng, S.-C. Jing, and F. Hui, "Deep reinforcement learning-based lane-changing trajectory planning method of intelligent and connected vehicles," *J Automot. Saf. Energy*, vol. 13, no. 4, pp. 705–717, 2022, doi: 10.3969/j.issn.1674-8484.2022.04.012.

[32] E. Leurent. (Apr. 2019). *A Collection of Environments for Autonomous Driving and Tactical Decision-Making Tasks*. [Online]. Available: https://github.com/eleurent/highway-en

XIAO YANG received the B.E. degree from Jilin University, Jilin, China, in 2021, where she is currently pursuing the master's degree with the School of Transportation. Her research interests include autonomous driving, deep reinforcement learning, and merging control.

HONGFEI LIU received the B.E. degree in machinery design and manufacture from Nanjing University of Science and Technology, Jiangsu, China, in 1996, and the M.A.Eng. and Ph.D. degrees in vehicle operating engineering from Jilin University, in 2002 and 2005, respectively. From 1996 to 1999, he was an Engineer in aeronautical facility research and development. From 2008 to 2009, he was a Visiting Scholar with Chiba Institute of Technology, Japan. Since 2006, he has been a Teacher with Jilin University, where he is currently a Professor. He has authored four books, more than 30 articles, and more than ten inventions. His research interests include traffic safety research and application, vehicle-road cooperative control, and vehicle reliability. He received the First Prize in the area of scientific and technological progress in Changchun.

MIAO XU received the Ph.D. degree in engineering from Jilin University, Changchun, China, in 2022. She is currently a Lecturer with the School of Automotive and Transportation Engineering, Jiangsu University. Her research interests include intelligent transportation systems, big data visualization analysis, and deep learning.

JINTAO WAN received the B.E. degree from Jilin University, Changchun, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Transportation. His research interests include autonomous vehicle technology, vehicle platoon motion control, and deep reinforcement learning.