Autonomous Vehicle Decision-Making at Unsignalized Intersection via Deep Reinforcement Learning

Junran Xie
Southeast University
Qingling Wang (qlwang@seu.edu.cn)
Southeast University

Research Article

Keywords: Autonomous vehicle, Reinforcement learning, Proximal policy optimization, Social-attention, Action mask

Posted Date: June 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1757806/v1

Autonomous Vehicle Decision-Making at Unsignalized Intersection via Deep Reinforcement Learning

Junran Xie[1,2] and Qingling Wang[1,2*]

[1]School of Automation, Southeast University, Sipailou, Nanjing, 210096, Jiangsu, China.
[2]Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Sipailou, Nanjing, 210096, Jiangsu, China.

*Corresponding author(s). E-mail(s): qlwang@seu.edu.cn;
Contributing authors: jrxie@seu.edu.cn;

Abstract

This paper is concerned with the performance of different state representation on the vehicle decision-making problem at unsignalized intersection based on deep reinforcement learning. A hybrid state representation architecture based on attention mechanism and collision time prediction is designed, which can effectively improve the autonomous decision-making ability of vehicles. Compared with the traditional state representation method, the feature of our method is that the fusion of features can improve the vehicle's obstacle avoidance ability, and the introduced action masking module can improve the vehicle's traffic efficiency. Finally, test results in the simulation environment verify the effectiveness of our proposed method.

Keywords: Autonomous vehicle, Reinforcement learning, Proximal policy optimization, Social-attention, Action mask

1 Introduction

Over the past decade, autonomous driving has received attention from various universities and institutes. Because autonomous driving technology has the potential to solve the problem of alleviating traffic congestion, especially at intersections where traffic congestion occurs most frequently. From the perspective of practical application, how to safely and effectively apply autonomous driving technology to the intersection environment is another issue of concern[1, 2].

The intersection is a complex road traffic environment due to the inclusion of pedestrians, non-motor vehicles, motor vehicles, traffic lights and other traffic elements. The classic decision-making method is to construct a decision-making framework based on the current intersection environment to guide the autonomous vehicle to make the correct decision-making choice. Commonly used assessment methods are divided into three directions: motion prediction, threat assessment and decision estimation[3]. Motion prediction is mainly based on physical modeling and analysis of vehicle motion, and is divided into two types: physics-based prediction methods[4] and maneuver-based prediction methods[5]. Threat assessment is an evaluation method based on expert experience, which mainly evaluates the state of the vehicle based on the collision time[6]. Time-to-Collision(TTC) is a classic metric used to evaluate the safety of autonomous vehicle(AV). The knowledge-based behavioral decisions of AV are realized by calculating the collision time of other vehicles on the road. The decision estimation method considers that the intersection environment is completely composed of autonomous vehicles and autonomous communication between vehicles can be achieved. The goal of decision estimation is to improve the traffic efficiency of the entire intersection, not just for a single vehicle[7–9].

The intersection is a complex traffic system that is difficult to make decisions with prior knowledge. In recent years, artificial intelligence technology has begun to be applied to decision-making problem at intersection, especially reinforcement learning algorithms. In intersection scenarios, reinforcement learning algorithms have achieved satisfactory results in behavior planning[10], continuous action decision-making[11], and driving strategies[12]. Reinforcement learning-based methods also enable autonomous vehicles to simultaneously handle multiple task objectives for navigation tasks[13]. But the state input of most current reinforcement learning algorithms is the list of features of all vehicles, which we call *list of features*[14, 15]. The advantages are that this state representation method has few parameters, fast training convergence, and simple state representation. However, the disadvantage of this representation method is that it is difficult to learn an effective policy in a scene with strong randomness such as an intersection.

In response to this problem, a framework based on reinforcement learning and supervised learning is proposed in [16]. This framework combines the graph neural network to model the connection between vehicles, which effectively improves the success rate of vehicles at the T-intersection. The attention mechanism can make the neural network discover the interdependence in the

input state, so that the neural network can assign different attention according to distinctive feature attributes to obtain better training results. In the intersection scenario, the list features can be processed by the attention mechanism to achieve better traffic efficiency[17]. The same attention mechanism in lane-changing decisions on highways can also help autonomous vehicles make better decisions[18, 19]. However, the attention mechanism pays more attention to the vehicles that may collide at the current moment, so it is difficult to predict the occurrence of collision in advance. Especially due to complex traffic scenarios such as intersections, the inability to predict potential collisions often leads to unsafe factors. Therefore, adding collision prediction features (such as TTC features) to the existing list features can effectively improve the vehicle's environmental perception and autonomous decision-making capabilities. In addition, for dangerous situations that may arise at any time, action masking is applied to ensure that the vehicle changes its state when encountering a danger, rather than maintaining its current state. Finally, feature extraction and action masking are integrated into the PPO algorithm framework, and a hybrid state representation framework for unsignaled intersections is proposed. The main contributions of the paper can be summarized as follows:

1) We propose a new hybrid state representation framework for autonomous decision-making in unsignaled intersections for autonomous vehicles. This hybrid state representation method can effectively improve the effectiveness and safety of autonomous vehicle decision-making.

2) Compared with the state representation method in the literature [17] which only uses the attention mechanism, in this paper we introduce the TTC feature based on threat assessment to enable the agent to have the ability to judge unknown threats. In addition, in a highly random environment, the vehicle can learn a cautious policy, which reduces the efficiency of vehicle traffic. This paper introduces an action masking module to solve the problem that the agent learns a strategy that is too cautious in a random environment.

The rest of this paper is organized as follows. In Section II, the basics of reinforcement learning are briefly introduced. The proposed reinforcement learning-based control algorithm is developed in Section III. In Section IV, the implementation details, results, and discussions of the experiments are given. The conclusion and future work is drawn in Section V.

2 Background and Problem Formulation

2.1 Reinforcement Learning and Policy Gradient

Reinforcement learning is a general framework for solving randomness problems. The randomness problem is define as a five-tuple Markov decision process(MDP) (S, A, P, R, γ), with state set S, action set A, transition probability matrix P , reward function R, and discount factor γ. The objective is to find a policy π that maximizes the cumulative reward. Formally, the value function $V_\pi(s)$ and the action value function $Q_\pi(s, a)$ are define as:

$$
\begin{aligned}
V_{\pi}(s) &\doteq \mathbb{E}[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} \mid s_t = s] \\
Q_{\pi}(s,a) &\doteq \mathbb{E}[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} \mid s_t = s, a_t = a]
\end{aligned}
\tag{1}
$$

By solving the value function and the action value function, the optimal policy for solving the randomness problem can be obtained[20]. The policy gradient algorithm is a policy-based reinforcement learning algorithm that approximates the policy as a function containing the parameter $\pi_{\theta}(s,a)$, and it uses the gradient descent optimization method to find the optimal policy. The principle of the algorithm is given in [21]:

$$
\bigtriangledown_{\theta} J(\theta) = \mathbb{E}_{\pi\theta}[\bigtriangledown_{\theta} log_{\pi\theta}(s,a) G_{\tau}] \tag{2}
$$

where $G_{\tau} = \sum_{k=0}^{\infty} \gamma^{k} r_{t+k}$ is the discounted return following time t.

Theoretically, the optimal policy can be found through the gradient descent method, but the policy gradient algorithm is not stable and effective when dealing with nonlinear problems. In recent years, many new policy gradient algorithms have been proposed to improve the stability and effectiveness, among which the PPO is the most representative algorithm. The PPO algorithm performs well in many challenging environments, so it is used as a baseline policy by Deepmind[22]. The core of the PPO is to propose a clipping surrogate objective:

$$
L^{PPO}(\theta) = \hat{\mathbb{E}}[min(r(\theta)\hat{A}, clip(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A})] \tag{3}
$$

where $r(\theta) = \frac{\pi_{\theta}(a|s)}{\pi_{\theta'}(a|s)}$ is the proportional coefficient between the new policy and the old policy, $\hat{A}$ is the advantage function and ϵ is the clipping range.

The clip method can ensure that the policy π does not decrease monotonically during the update, so that the algorithm obtains a stable performance improvement during the training process.

2.2 Attention Mechanism and Action Masking

The attention mechanism is introduced to enable the neural network to filter out the information that is more important to the current target task from among the input information[23]. Attention mechanisms are used in this work to handle variable-state inputs and achieve vehicle-to-vehicle attention. The attention calculation formulation for each head can be written as Eq.4:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

where Q is the attention function, K is the keys matrix and V is the value matrix. The output of the function represents the dot product of values and key-similarity, independent of sequence size and ordering.

Action masking is used to avoid repetition of invalid actions in discrete action space. Action masking is usually described as an auxiliary detail, but ineffective action masking in discrete action spaces can effectively improve the convergence speed and data utilization efficiency of reinforcement learning algorithms. At present, action masking has been widely used in the training of large discrete action space, especially in the training process of OpenAI-Five and King of Glory game agents, which plays an important role in the convergence of the algorithm[24, 25]. Although action masking changes the action space distribution, it has been shown to be an effective gradient descent algorithm[26]. In this work, motion masking is utilized to induce the vehicle to make state changes in a possible collision scenario, rather than maintaining the current state.

2.3 Problem Formulation

In this paper, we consider the decision-making problem of autonomous vehicles in unsignalized intersections. A autonomous vehicle with a reinforcement learning controller makes a left turn at an unsignalized intersection, while the vehicle can obtain environmental perception data at the current moment to help the vehicle make a decision. The control objective of the reinforcement learning controller is to help the autonomous vehicle make the right decisions in the face of complex traffic scenarios to pass the unsignalized intersection safer and faster.

The kinematic model of the vehicle is a Kinematic Bicycle Model:

$$\begin{aligned}
\dot{x} &= vcos(\varphi + \beta) \\
\dot{y} &= vsin(\varphi + \beta) \\
\dot{\varphi} &= \frac{v}{\ell} sin(\beta) \\
\dot{v} &= a \\
\beta &= tan^{-1}(1/2tan\delta)
\end{aligned} \tag{5}$$

In addition, a hierarchical control framework is applied to vehicle motion control, the lateral control is to track the target route through a low-level steering controller. The longitudinal control of the AV is realized by the reinforcement learning control algorithm, and the longitudinal control of the surrounding vehicles is based on the Intelligent Driver Model[27].

3 Proposed Method

In this section, we first introduce the Markov decision process and its parameters. Then a reinforcement learning framework for solving the problem of autonomous vehicles passing through unsignaled intersections is introduced. Finally, the attention mechanism module and action masking module in the framework are introduced in detail.

3.1 Specification of the MDP

In this subsection the process of modeling the environment as a Markov decision process is introduced, mainly including the design of states, actions and rewards.

• States: the state contains two parts, the list feature state and the TTC feature state. The list of feature states contains descriptive information about the ego-vehicle and surrounding vehicles, including position, speed, and direction. So state information includes ego-vehicle state s_0 and surrounding vehicle state $(s_i)_{i\in[1,N+1]}$ can be expressed as:

$$\begin{aligned} s =& (s_0, s_1, ..., s_{N+1}) \\ where \quad s_i =& (s_x, s_y, v_x, v_y, cos_h, sin_h) \end{aligned} \tag{6}$$

where s_x and s_y represent vehicle coordinates, v_x and v_y represent vehicle lateral and longitudinal speeds and the value of last two elements cos_h and sin_h represent vehicle orientation. It should be noted that if the number of surrounding vehicles is less than N, it will be filled with 0.

Although $lists\ of\ feature$ can reflect the state of all vehicles on the road, it is difficult to estimate the probability of collision between the ego-vehicle and surrounding vehicles through neural networks. TTC state is a state representation for estimating the possibility of vehicle collision, which can be obtained by calculating whether the ego-vehicle will collide with other vehicles. The addition of TTC state can enable the agent to learn about possible conflict situations, thereby improving the decision-making ability of the agent. The TTC feature calculates possible collisions within five time steps for three different values of the ego vehicle speed and three lanes on the road around the current line, respectively. So the default dimension size of TTC feature is $[3, 3, 5]$.

• Actions: the action space of the vehicle includes acceleration and steering angle. In this paper, the agent selects appropriate acceleration from a discrete finite action space $A = \{Slow, Idle, Fast\}$. The steering control of the agent is automatically carried out by the low-level controller, making the reinforcement learning algorithm control focused on the decision-making ability of the agent.

• Rewards: appropriately setting the reward method is of great importance to the training results of reinforcement learning. The reward of the agent includes three parts: speed reward r_s, arrival reward r_a and collision reward r_c. When the speed of vehicle is between $7m/s$ and $9m/s$, the ego-vehicle gets a normalized speed reward r_a. When the ego-vehicle reaches the target point,

its reward r_a is 1, and the reward r_c is -5 if a collision occurs. The reward function is defined as:

$$r = \begin{cases} r_s + r_c & if\ not\ arrival \\ r_a & if\ arrival \end{cases} \tag{7}$$

3.2 Network Architecture

The multi-head self-attention mechanism is used to extract feature information from list features, and the fully connected layer is used to extract feature information from TTC features. The overall framework of the reinforcement learning algorithm based on the PPO and the hybrid state representation method is shown in Fig. 1.

First, list features and TTC features are fed into their respective encoders. Next, the input of the two encoders is fed into their respective feature processing modules, the list feature uses an attention mechanism, and the TTC feature uses a fully connected layer network. Finally, the outputs of the two feature processing modules are concatenated together as a feature vector, which then uses the actor-critic network to output actions. It is worth noting that the neural network of the actor-critic architecture is updated using a PPO-based approach.

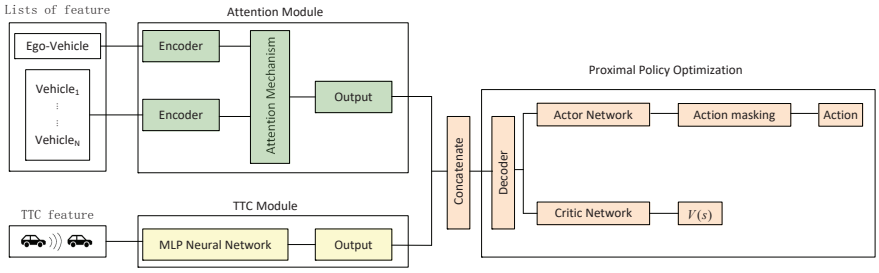


Fig. 1 The hybrid state representation method based on deep reinforcement learning

3.3 Attention Module and Action Masking Module

From the perspective of human driving habits, drivers need to pay more attention to vehicles that are close to or in conflict with the route. So the function of the attention module is to enable the neural network to discover which vehicle need to be paid attention to. The attention module is a variant of the traditional social attention, where only the ego state has query encoding[28]. The structure of this attention mechanism is shown in Fig. 2. The introduction of the attention mechanism helps the ego-vehicle to focus on the vehicles that may collide with itself, and the size of the attention depends on the speed, direction and position of the surrounding vehicles .

Algorithm 1 PPO Algorithm with Hybrid State Representation

Input: List of feature state s_f^0, TTC feature s_T^0
Output: the ego vehicle action a

for t ← 0 to *Iteration times* **do**
  Run policy π with the state s_f^t and s_T^t
  Get the discrete action distribution $p(a)$
  Through action masking module. $p'(a) \leftarrow p(a)$
  Get action a from $p'(a)$ and reward r^t
  Calculate $\pi_\theta(a_t \mid s_t)$ and V_π
  Store($s_f^t, s_T^t, a, \pi_\theta$ and V_π) into replay buffer
  if buffer length == Maximum buffer length **then**
    Update the neural network with PPO algorithm
    Clear replay buffer
  end if
end for

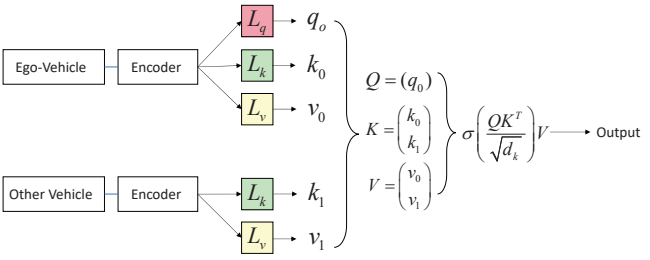


Fig. 2 The structure of the attention module. The blocks L_q, L_k, L_v represents the linear layer, the keys K and values V are obtained from all vehicle states, and the queue Q is generated only by the ego-vehicle.

The PPO algorithm is an actor-critic architecture reinforcement learning algorithm, where the output of the actor network represents the actions of the agent. For discrete action spaces, actor networks usually output a probability distribution of all actions, and the action is randomly selected from a set of actions with probability. The function of the action masking module is to change the action probability distribution output by the actor network, so that the probability of the action masked is infinitely small. In this paper, the idle action in the action set is masked when the ego vehicle is likely to collide with a road vehicle within 2 seconds. In other words, for a situation where the ego vehicle may collide, the ego vehicle should choose to accelerate or decelerate to prevent the collision.

Therefore, the hybrid state representation algorithm based on PPO is summarized in Algorithm 1.

4 Experiments and Result

4.1 Simulation Environment

In this paper, we use highway-env[29] as the simulation environment to simulate the driving behavior and decision-making choices of ego-vehicle in unsignaled intersections. The basic architecture of the simulation environment is shown in Fig. 3. As shown in Fig. 3, the green car represents the controlled ego vehicle and the blue car represents the surrounding vehicles. In the simulation environment, the ego-vehicle is randomly generated in the south of the intersection, and the mission goal is to pass the intersection and turn left. Moreover, the properties of other vehicles on the road including position, speed and target are randomly initialized. In particular, simple road priorities are place at prevent surrounding vehicle collisions at intersections. There are four levels of road priority, and the priority order from high to low is horizontal straight and right turn, vertical straight and right turn, horizontal left turn and vertical left turn.

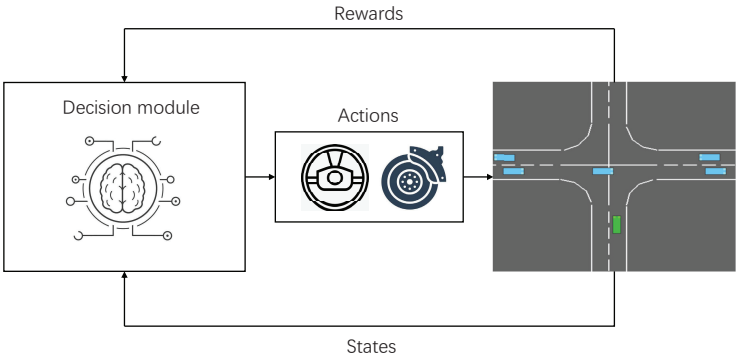


Fig. 3 The simulation environment framework for reinforcement learning

The default setting of the unsignaled intersection is determined as follows: the number of surrounding vehicles N is 15, the simulation time t is up to 20s and all vehicles execute the policy every second. When a surrounding vehicle collides, the colliding vehicle will be cleared if it does not collide with the ego-vehicle.

4.2 Implementation details

We evaluate five different state representations, namely list feature state, table-state[30], list feature with attention, list feature with attention and action masking, and hybrid features with attention and action masking. The feature of each state is as following:

• List feature state(LIST): The neural network structure of this model is a simple fully connected network, which only uses list features as input without using attention mechanism and action masking. Since the neural network

requires a fixed size input, zero-padding is used to pad the size of the list features up to the feature size when the number of vehicles is 15.

• Table-state(GRID): The table-state implicitly expresses the relationship between vehicles in a two-dimensional space, and the size of the state tensor is determined by the area covered and the size of a single grid. The table-state uses CNN to extract information.

• List feature with attention(LIST-A): Improvements are made based on the list feature state, replacing the fully connected network with the attention mechanism module mentioned in Section III. The advantage of this is that the attention mechanism module does not need to be zero-padding.

• List feature with attention and action masking(LIST-AA): Introduced the action masking module. The difference between this state representation method and the proposed hybrid representation method is that there is no TTC feature input. Through comparative experiments, it can be proved that both the action masking module and the TTC feature input module have a positive effect on the training results.

• Hybrid features with attention and action masking(HLIST-AA): Our proposed hybrid state input method, the neural network structure of this method is shown in Fig. 2.

Table 1 The neural network parameters

Module	FCN(LIST)	GRID	attention module	TTC module
Input sizes	[15, 7]	[32, 32, 7]	[· , 7]	[45, 1]
Layer sizes	[256, 256]	Conv layers:3 Kernel Size:2 Stride:2	Encoder:[64, 64] 2 heads, $d_k = 32$ Decoder:[64,64]	16

The neural network parameters of all five state representation methods are shown in Table 1. Additionally, all experiments were trained on the same number of epochs in the same environment, with the same hyperparameters and random seeds for the reinforcement learning algorithm. The parameters of the PPO algorithm are summarized in Table 2.

4.3 Result and Evaluation

The reinforcement learning algorithm was carried out on a computer with Ubuntu 18.04 system. Fig. 4 plots the evolution trend of the total reward of each state representation method during the training process. It should be noted that the curve of the training result is smoothed.

From the reward curve, HLIST-AA achieves the best results among all methods, proving that the agent learns better control policies from a mixture of TTC features and list features. The cumulative reward obtained by the LIST method is the lowest, which proves that it is difficult for the agent to learn the optimal policy by only using the fully connected network in a highly random

Table 2 Parameters of PPO Algorithm

Parameter	Value
Reward discount factor γ	0.95
Discount parameters for GAE λ	0.95
Learning rate for neural network	0.0005
Total training stride	1e6
Loop update operation	10
Batch size for updating	64
Buffer length	768
ϵ for the clipping surrogate objective	0.2
Maximum update gradient	0.5
Random seed	1234

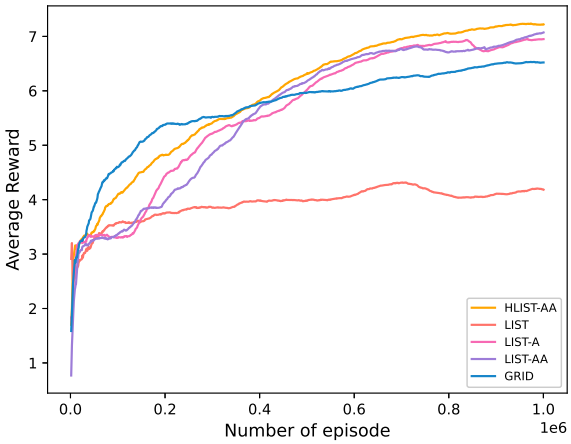
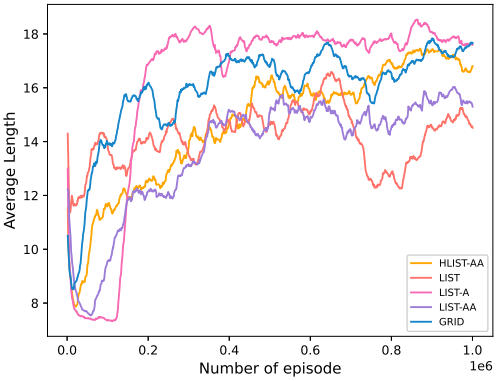


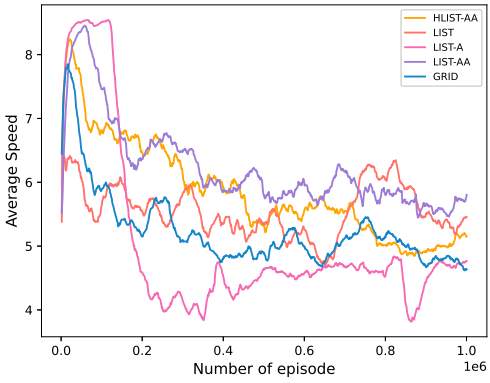
Fig. 4 The evolution of the reward curve during training

environment. The reward of LIST-AA is slightly higher than that of LIST-A, which also shows that the action masking module has achieved a positive effect in reinforcement learning training. The difference between the GRID method and the LIST-A method is the encoding of the vehicle position. From the training results, the two-dimensional matrix state is more difficult to extract the state encoding information of the vehicle than the attention mechanism.

In addition, the average length and average speed during training are also evaluated, as shown in Fig. 5. In general, the five methods that need to be evaluated can be divided into three categories based on average length and average speed. The first category is prudent policies including LIST-A and GRID, the second category is aggressive policies including LIST-AA and LIST, and the last category is our proposed HLIST-AA. The decision of an agent



(a) Average Length



(b) Average Speed

Fig. 5 The evolution of the average length and the average speed during training

trained with LIST-A and GRID to pass through the intersection is to wait until it is safe enough to pass through the intersection because the average length of the agent is longer and the average speed is small. Aggressive agents trained by LIST-AA and LIST tend to prefer to pass the intersection faster and obtain higher rewards when making decisions. Combined with the reward curve, the action masking module introduced in LIST-AA can well solve the problem that the agent behaves too cautiously in the intersection. Based on experience, prudent policies will lead to lower traffic efficiency, and aggressive policies will lead to reduced traffic safety. It is of positive significance to find a suitable compromise between prudent strategies and aggressive strategies. It can be seen from the curve evaluation results that the agents trained by the HLIST-AA method can have higher traffic efficiency than the first category of method, and can also have higher security than the second category of method.

In order to ensure the accuracy of the evaluation, 10,000 repeated experiments are performed on the model obtained by training, and the experimental results are shown in Table 3. Five metrics are used to evaluate the trained model, namely average step size, average reward, average speed, success rate and collision rate.

Table 3 Test results of different state representation methods

State types	Average step size	Average reward	Average speed(m/s)	Success rate	Collision rate
LIST	14.82	4.300	5.369	42.86%	28.48%
GRID	17.63	6.684	4.655	65.22%	6.96%
LIST-A	17.12	6.995	4.938	75.42%	6.40%
LIST-AA	**14.65**	7.299	**6.032**	80.15%	10.38%
HLIST-AA	16.08	**7.726**	5.472	**84.37**%	**3.88**%

From the test results, the proposed hybrid state representation method based on attention mechanism and TTC features achieves the best results in the two important indicators of collision rate and success rate. For the LIST method, the lowest success rate and the highest collision rate indicate that it is difficult to learn the relationship between vehicle actions and environmental states using fully connected layer neural networks. The average rewards of the LIST-A method and the GRID method are almost identical, however, the success rate of the LIST-A method is 10% higher than that of the GRID method. The test results show that the introduction of the attention mechanism can bring the agent a more accurate judgment based on the current state of the surrounding vehicles compared with the GRID state. As shown by the training curve, the agent trained by the LIST-AA method with the action masking module is more active in the intersection, which increases the success rate of the agent's passage by 5%. However, the more active and aggressive policy also brings more collision possibilities. The collision rate of the agent of the LIST-AA method is 4% higher than that of the LIST-A method. The main reason for the increase in the collision rate is that the attention mechanism only pays attention to the state of the surrounding vehicles at the current moment, but lacks the dynamic cognition of the environment. This is also the reason why the success rate of the HLIST-AA method is increased by 4% on the LIST-AA method after adding the TTC feature, and the collision rate is decreased by 3% compared with the LIST-A method. After introducing the TTC feature, the agent can learn to judge the situation of the intersection independently when passing through the intersection instead of adopting a more conservative passing policy or a more aggressive passing policy. Overall, the test results show that our proposed HLIST-AA method is an effective state representation method in decision-making passing strategies at unsignaled intersections. Compared with other state representation methods, it is optimal in terms of traffic rate and collision rate.

5 Conclusion and Future Work

In this paper, the decision-making performance of autonomous vehicles at unsignaled intersections with different state input is investigated. A hybrid state representation method based on an attention mechanism and TTC is proposed. The advantage of this method is that the hybrid state can enable the reinforcement learning algorithm to obtain more observation information for decision-making during training. Then, experiments are performed based on various state representation methods, and the results show that the decision model obtained by the hybrid state method has better performance. In the future, the application of this hybrid state method to different road traffic environments will be considered.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant Number 62111530149, 61973074, and 61921004.

References

[1] Shirazi MS, Morris BT (2017) Looking at Intersections: A Survey of Intersection Monitoring, Behavior and Safety Analysis of Recent Studies. IEEE Transactions on Intelligent Transportation Systems, 18(1):4-24. https://doi.org/ 10.1109/TITS.2016.2568920..

[2] Namazi E, Li J, Lu C (2019) Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic Literature Review. IEEE Access, 7:91946-91965. https://doi.org/10.1109/ACCESS.2019.2927412.

[3] Noh S (2019) Decision-Making Framework for Autonomous Driving at Road Intersections: Safeguarding Against Collision, Overly Conservative Behavior, and Violation Vehicles. IEEE Transactions on Industrial Electronics, 66(4):3275-3286. https://doi.org/10.1109/TIE.2018.2840530.

[4] Brännström M, Coelingh E, Sjöberg J (2010) Model-Based Threat Assessment for Avoiding Arbitrary Vehicle Collisions. IEEE Transactions on Intelligent Transportation Systems, 11(3):658-669. https://doi.org/10.1109/TITS.2010.2048314.

[5] Chen L, Hu X, Tian W, Wang H, Cao D, Wang F (2019) Parallel planning: a new motion planning framework for autonomous driving. IEEE/CAA Journal of Automatica Sinica, 6(1):236-246. https://doi.org/10.1109/JAS.2018.7511186.

[6] Minderhoud MM, Bovy PH (2001) Extended time-to-collision measures for road traffic safety assessment. Accident Analysis & Prevention, 33(1):89-97.

[7] Vaio MD, Falcone P, Hult R, Petrillo A, Salvi A, Santini S (2019) Design and Experimental Validation of a Distributed Interaction Protocol for Connected Autonomous Vehicles at a Road Intersection. IEEE Transactions on Vehicular Technology, 68(10):9451-9465. https://doi.org/10.1109/TVT.2019.2933690.

[8] Tian R, Li N, Kolmanovsky I, Yildiz Y, Girard AR (2020) Game-Theoretic Modeling of Traffic in Unsignalized Intersection Network for Autonomous Vehicle Control Verification and Validation. IEEE Transactions on Intelligent Transportation Systems. 23(3)3:2211-2226. https://doi.org/10.1109/TITS.2020.3035363.

[9] Sayin MO, Lin CW, Shiraishi S, Shen J, Başar T (2019) Information-Driven Autonomous Intersection Control via Incentive Compatible Mechanisms. IEEE Transactions on Intelligent Transportation Systems, 20(3):912-924. https://doi.org/10.1109/TITS.2018.2838049.

[10] Qiao Z, Schneider J, Dolan JM (2021) Behavior Planning at Urban Intersections through Hierarchical Reinforcement Learning. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2667-2673. https://doi.org/10.1109/ICRA48506.2021.9561095.

[11] Li G, Li S, Li S, Qu X (2021) Continuous decision-making for autonomous driving at intersections using deep deterministic policy gradient. IET Intelligent Transport Systems, 1-13. https://doi.org/10.1049/itr2.12107.

[12] Zhou M, Yu Y, Qu X (2020) Development of an Efficient Driving Strategy for Connected and Automated Vehicles at Signalized Intersections: A Reinforcement Learning Approach. IEEE Transactions on Intelligent Transportation Systems, 21(1):433-443. https://doi.org/10.1109/TITS.2019.2942014.

[13] Kai S, Wang B, Chen D, Hao J, Zhang H, Liu W (2020) A Multi-Task Reinforcement Learning Approach for Navigating Unsignalized Intersections. 2020 IEEE Intelligent Vehicles Symposium (IV),1583-1588. https://doi.org/10.1109/IV47402.2020.9304542.

[14] Cao Z, Yang D, Xu S et al (2021) Highway exiting planner for automated vehicles using reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 22(2):990-1000. https://doi.org/10.1109/TITS.2019.2961739.

[15] Tian Y, Cao X, Huang K, Fei C, Zheng Z, Ji X (2021) Learning to Drive Like Human Beings: A Method Based on Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2021.3055899.

[16] Ma X, Li J, Kochenderfer MJ, Isele D, Fujimura K (2021) Reinforcement Learning for Autonomous Driving with Latent State Inference and Spatial-Temporal Relationships. 2021 IEEE International Conference on Robotics and Automation (ICRA),6064-6071. https://doi.org/10.1109/ICRA48506.2021.9562006.

[17] Leurent E, Mercat J (2019) Social Attention for Autonomous Decision-Making in Dense Traffic. Advances in Neural Information Processing Systems 33 (NeurIPS 2020). http://arxiv.org/abs/1911.12250

[18] Zhang S, Wu Y, Ogai H, Inujima H, Tateno S (2021) Tactical Decision-Making for Autonomous Driving Using Dueling Double Deep Q Network With Double Attention. IEEE Access, 9:151983-191992. https://doi.org/10.1109/ACCESS.2021.3127105.

[19] Wang J, Zhang Q, Zhao D (2022) Highway Lane Change Decision-Making via Attention-Based Deep Reinforcement Learning. IEEE/CAA Journal of Automatica Sinica, 9(3):567-567. https:doi.org/10.1109/JAS.2021.1004395.

[20] Sutton, Richard S, Barto AG (2018) Reinforcement learning: An introduction. MIT press.

[21] Sutton RS, McAllester DA, Singh SP, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 1057–1063.

[22] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal Policy Optimization Algorithms. Advances in Neural Information Processing Systems(NIPS2017). http://arxiv.org/abs/1707.06347

[23] Vaswani A, Shazeer N, Parmar N et al. (2017) Attention is All you Need. Advances in Neural Information Processing Systems.

[24] Berner C, Brockman G, Chan B et al (2019) Dota 2 with large scale deep reinforcement learning. ArXiv. http://arxiv.org/abs/1912.06680, 2019.

[25] Ye D, Liu Z, Sun M et al (2020) Mastering Complex Control in MOBA Games with Deep Reinforcement Learning. Proceedings of the AAAI Conference on Artificial Intelligence(AAAI2020), 34(4):6672-6679.

[26] Huang S, Ontañón S (2020) A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. ArXiv. http://arxiv.org/abs/2006.14171

[27] Treiber M, Hennecke A, Helbing D (2020) Congested traffic states in empirical observations and microscopic simulations. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics,

62(2):1805–1824.

[28] Vemula A, Muelling K, Oh J (2018) Social attention: Modeling attention in human crowds. 2018 IEEE international Conf. Robotics and Automation (ICRA), 4601–4607.

[29] Edouard Leurent (2018) An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env.

[30] Fridman L, Terwilliger J, Jenik B (2018) Deeptraffic: Crowdsourced hyperparameter tuning of deep reinforcement learning systems for multi-agent dense traffic navigation. ArXiv. https://arxiv.org/abs/1801.02805