# Progression Cognition Reinforcement Learning with Prioritized Experience for Multi-Vehicle Pursuit

Xinhang Li, *Graduate Student Member, IEEE*, Yiying Yang, Zheng Yuan, Zhe Wang, *Graduate Student Member, IEEE*, Qinwen Wang, Chen Xu, *Member, IEEE*, Lei Li, Jianhua He, *Senior Member, IEEE*, and Lin Zhang*, *Member, IEEE*

***Abstract*—Multi-vehicle pursuit (MVP) such as autonomous police vehicles pursuing suspects is important but very challenging due to its mission and safety-critical nature. While multi-agent reinforcement learning (MARL) algorithms have been proposed for MVP in structured grid-pattern roads, the existing algorithms use random training samples in centralized learning, which leads to homogeneous agents showing low collaboration performance. For the more challenging problem of pursuing multiple evaders, these algorithms typically select a fixed target evader for pursuers without considering dynamic traffic situation, which significantly reduces pursuing success rate. To address the above problems, this paper proposes a Progression Cognition Reinforcement Learning with Prioritized Experience for MVP (PEPCRL-MVP) in urban multi-intersection dynamic traffic scenes. PEPCRL-MVP uses a prioritization network to assess the transitions in the global experience replay buffer according to each MARL agent's parameters. With the personalized and prioritized experience set selected via the prioritization network, diversity is introduced to the MARL learning process, which can improve collaboration and task-related performance. Furthermore, PEPCRL-MVP employs an attention module to extract critical features from dynamic urban traffic environments. These features are used to develop a progression cognition method to adaptively group pursuing vehicles. Each group efficiently targets one evading vehicle. Extensive experiments conducted with a simulator over unstructured roads of an urban area show that PEPCRL-MVP is superior to other state-of-the-art methods. Specifically, PEPCRL-MVP improves pursuing efficiency by 3.95% over Twin Delayed Deep Deterministic policy gradient-Decentralized Multi-Agent Pursuit and its success rate is 34.78% higher than that of Multi-Agent Deep Deterministic Policy Gradient. Codes are open-sourced.**

***Index Terms*—autonomous driving, multi-agent reinforcement learning, multi-vehicle pursuit, prioritized experience**

## I. Introduction

EMPOWERED by the self-learning ability of reinforcement learning (RL) and significantly improved environment perception, autonomous driving (AD) [1]–[3] is growing with fast pace and great potentials to improve driving safety and traffic efficiency [4]–[6]. Multi-vehicle pursuit (MVP) is a specific application of AD technology, where multiple autonomous pursuing vehicles chase one or more moving vehicles. MVP problems have been attracting extensive research attention due to their increasing applications, including collision avoidance designs in intelligent transportation systems, sport/game strategies, the balance and game between generators and loads in smart grid dispatch, disaster relief strategies, autonomous police vehicles pursuing suspects, and similar confrontation scenarios [7], [8]. The MVP tasks are usually mission and safety-critical. Efficient multi-vehicle collaboration and comprehensive perception under complex and dynamic traffic environments are important to successfully complete the MVP tasks [9].

Cooperative multi-agent reinforcement learning (MARL) has been widely studied for multiple agents collaboration and connected-automated vehicles (CAVs), and could be applied to MVP applications [10]–[12]. Many MARL-based cooperative control schemes for CAVs have been proposed. Guan et al. presented a centralized coordination framework [13] for autonomous vehicles at intersections without traffic signals, which significantly improved the road efficiency. [14] and [15] studied distributed cooperation methods to realize the conflict-free control of CAVs. [16] and [17] implemented a multi-agent systems-based hierarchical controller to improve vertical and horizontal cooperation among the automated vehicles. It is noted that all the above MARL algorithms for CAVs were designed to improve driving safety. However, the MVP tasks have additional mission-critical requirements and require strong collaboration and adaptation to dynamic environments, which present significant new challenges to the design of MARL algorithms.

In the literature, a few game theory-based and other classical methods for MVP have been proposed [18]. Huang et al. presented a decentralized control scheme [19] based on the Voronoi partition of the game domain. Pan et al. designed a region-based relay pursuit scheme [7] for the pursuers to capture a single evader. A policy iteration method-based continuous-time Markov decision process [20] was proposed to optimize the pursuer strategy. [21] employed a graph-theoretic approach to study the interactions of the agents and obtain distributed control policies for pursuers. [9] and [22] introduced curriculum RL to train pursuers to approach

*Corresponding author

X. Li, Y. Yang, Z. Yuan, Q. Wang, C. Xu and L. Li are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: lixinhang, yyying, yuanzheng, wangqinwen, chen.xu, leili@bupt.edu.cn).

Z. Wang is with Centre for Telecommunications Research, King's College London, London WC2R 2LS, U.K. (e-mail: tylor.wang@kcl.ac.uk).

J. He is with School of Computer Science and Electronics Engineering, University of Essex, Colchester, U.K. (e-mail: j.he@essex.ac.uk).

L. Zhang is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Beijing Big Data Center, Beijing, China (e-mail: zhanglin@bupt.edu.cn).

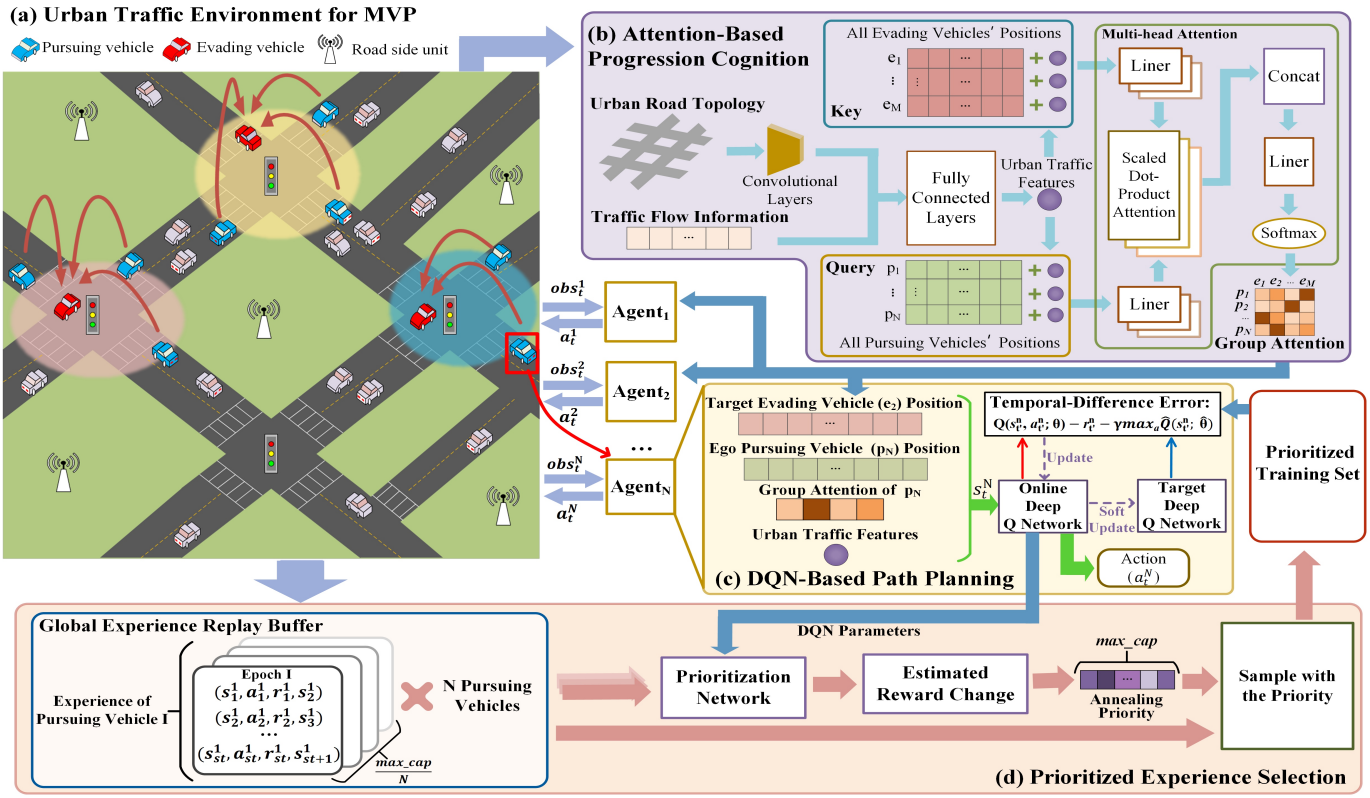


Fig. 1. Architecture of PEPCRL-MVP. Urban traffic environment for MVP (**a**) provides complex pursuit-evasion scenes and an interactive environment for MARL. Attention-based progression cognition module (**b**) provides accurate urban traffic information with critical features and group attention. The critical features and group attention are used to improve DQN-based path planning (**c**). Prioritization network and prioritized experience selection (**d**) are used to improve diversity and personalization of MARL.

the evader. In order to improve the pursuing efficiency, [23] weighted different evaders to encourage the pursuers to capture the close evader. In addition, [24] designed cooperative multi-agent schemes with a target prediction network. Yang et al. design a graded-Q RL framework [25] to enhance the coordination capacity of pursuing vehicles. [26] and [27] adopted MARL to accomplish collaborative pursuit tasks in simplified traffic scenes with structured grid-pattern roads. However, these above methods did not consider the dynamic urban pursuit-evasion environment and the fixed allocation of pursuit tasks greatly affects the efficiency of the pursuit.

In the existing MARL algorithms for MVP and AD, deep neural network parameters are shared among agents via centralized training with decentralized execution (CTDE), which significantly improves learning efficiency and experience utilization [28]–[30]. Many CTDE-based deep MARL methods achieve state-of-the-art performance on some tasks, such as group matching game and path finding [31]–[34]. Although CTDE can accelerate training [35], it has poor performance in complex and difficult tasks, such as Google Research Football [36] and MVP. These complex tasks typically require substantial exploration, diversified strategies, and efficient collaboration among agents [37]. But homogeneous agents tend to behave similarly because of parameter sharing, limiting efficient exploration and collaboration of MARL agents. Besides, prioritized experience replay has been the focus of several studies. Previous studies adopted the temporal-difference error as priorities of experience [38]–[41] without considering their variability among multiple agents. And they did not address the problem of multi-agent homogenization.

According to the above analyses, it can be observed that low adaptation to dynamic traffic environments and homogeneous agents severely limit the collaborative pursuing performance. To address these problems, this paper proposes a progression cognition reinforcement learning with prioritized experience (PEPCRL-MVP) for MVP in urban traffic scenes. A framework of the PEPCRL-MVP is shown in Fig. 1. There are two distinct modules in the new PEPCRL-MVP architecture. The first is a proposed prioritization network, which is used to select prioritized training set for each agent in MARL to adjust its deep neural network parameters. Optimizing agents with the personalized training set enables each agent to distinguish itself from others, thereby encouraging efficient collaboration. The proposed prioritization network can also be applied to a wide range of multi-agent systems to improve collaboration. In addition, an attention-based progression cognition module is designed to adaptively group multiple pursuing vehicles considering dynamic traffic awareness. With the above designs, the PEPCRL-MVP can address the problems of low adaptation and homogeneous agents in the existing MVP approaches and is expected to greatly improve pursuing performance.

The contributions of this paper can be summarized as follows.

- This paper proposes a multi-agent reinforcement learning

approach with prioritized experience for collaborative multi-vehicle pursuit. A novel prioritization network is proposed to diversify the optimization and strategies of MARL, encouraging more efficient collaboration and experience exploration.
- An attention-based progression cognition module is proposed to divide pursuing vehicles into improvisational groups according to dynamic urban traffic situations. Critical features are extracted from the sensor data to support the grouping process, which supports vehicles more effectively focusing on one target evading vehicle and greatly improves pursuing performance.
- This paper applies PEPCRL-MVP to the simulated urban large-scale roads with 46 junctions and sets different pursuing difficulty levels with variable numbers of pursuing vehicles and evading vehicles. In the three tested difficulty levels, PEPCRL-MVP improves pursuing efficiency by 3.95% on average compared with TD3-DMAP, and improves pursuing success rate by 34.78% on average compared with MADDPG. Codes are open sourced in *https://github.com/BUPT-ANTlab/PEPCRL-MVP*.

The rest of this paper is organized as follows. Section II describes multi-vehicle pursuit in an urban pursuit-evasion scene and models MVP problem based on partially-observable stochastic game (POSG). Section III presents MARL with prioritized experience and its training process. Section IV presents the reinforcement learning-based path planning algorithm with progression cognition. Section V gives the performance of the proposed method. Section VI draws conclusions.

## II. Multi-Vehicle Pursuit in Dynamic Urban Traffic

This section firstly illustrates the urban complex pursuit-evasion environment and the constraints, bridging the 'sim-to-real' gap. Section II-B formulates the MVP problem based on POSG and introduces MARL-based solution to POSGs.

### A. MVP in Large-Scale Urban Traffic

This paper focuses on the problem of multi-vehicle pursuit under the complex urban traffic. We consider a closed large-scale urban traffic scene with multi-intersection road structure [42]. The considered scene basically retains the settings of urban traffic, such as traffic lights and speed limits. Without loss of generality, we assume there are $N$ pursuing vehicles, $M$ evading vehicles ($N > M$), $B$ background vehicles, and $L$ lanes. The background vehicles and evading vehicles follow the randomly selected routes. For the MVP task, an evading vehicle is deemed captured if any pursuing vehicle is less than a pre-configured distance $d_{min}$ from its target evading vehicles. If all evading vehicles are captured with a given $st$ time steps, the pursuit task is successfully *Done*.

In this paper, we have a few constraints for the MVP task.
- All vehicles in the scene (pursuing, evading, and background vehicles) follow traffic rules, such as obeying traffic lights, and driving on the right lanes without collisions.
- All pursuing vehicles and evading vehicles are initialized at different diagonal points in the map with $0\ m/s$.
- The maximum speed $v_{\max}$, maximum acceleration $ac_{\max}$ and maximum deceleration $de_{\max}$ of all pursuing vehicles and evading vehicles are set to be the same.

### B. POSG-Based MVP Problem Formulation

In MVP, the decision-making process of a finite set of agents $\mathcal{I}$ deployed in pursuing vehicles with partial observability can be formalized as POSG, which can be defined as a tuple $\mathcal{M}_G := (\mathcal{I}, \mathcal{S}, [\mathcal{A}_n], [\mathcal{O}_n], Tr, [R_n])$ for $n = 1, ..., N$. In time step $t$, the pursuing vehicle $n$ receives a local observation $o_t^n : \mathcal{S} \to \mathcal{O}_n$ that is correlated with the underlying state of the environment $s_t \in \mathcal{S}$. $o_t^n$ is further processed to $s_t^n$ as the state of the pursuing vehicle $n$, that takes an action $a_t^n \in \mathcal{A}_n$ according to $s_t^n$. Consequently, the environment evolves to a new state $s_{t+1}$ with the transition probability $Tr = P(s_{t+1} \mid s_t, a_t) : \mathcal{S} \times \mathcal{A}_1 \times ... \times \mathcal{A}_N \to \mathcal{S}$ and then the agent receives a decentralized reward $r_t^n : \mathcal{S} \times \mathcal{A}_n \to \mathbb{R}$. $[R_n]$ is the rewards of all multiple agents. The probability distribution of actions at a given state is determined by the stochastic policy $\pi_n$. The goal of an optimal policy $\pi_n^*$ is to generate a distribution that maximizes the discounted sum of future rewards over an infinite time horizon, which can be expressed as

$$\pi_n^* := \arg\max_{\pi} \mathbb{E}(\sum_{t=0}^{\infty} \gamma^t R(s_t^n, \pi(s_t^n))), \tag{1}$$

in which, $\gamma \in [0, 1)$ is the discount factor, indicating the impact of future earnings on current expectation value. The optimal policy maximizes the state-action value function, i.e., $\pi_n^*(s_t^n) = \arg\max_a Q_n^{\pi}(s_t^n, a)$.

According to Bellman optimality equation, the optimal state-action value function can then be derived as

$$Q_n^{\pi}(s_t^n, a_t^n) = \mathbb{E}_{s_{t+1} \sim P(. \mid s_t, a_t)}(r_t^n + \gamma \operatorname{argmax}_a Q_n^{\pi}(s_{t+1}^n, a)). \tag{2}$$

As an emerging AI algorithm, MARL enables agents in POSGs to make optimal strategies without exact state transition probability $Tr$ and reward function $R$. It provides an excellent solution to MVP with the dynamic and complex environment. The agent $n$ in MARL updates the Q value function according to temporal difference error $\delta$ by off-policy learning,

$$Q_n^{\pi}(s_t^n, a_t^n) \leftarrow Q_n^{\pi}(s_t^n, a_t^n) + \alpha\delta, \tag{3}$$

in which,

$$\delta = r_t^n + \gamma \arg\max_{a_{t+1}} Q_n^{\pi}(s_{t+1}^n, a_{t+1}^n) - Q_n^{\pi}(s_t^n, a_t^n), \tag{4}$$

where $\alpha$ is the learning rate. For pursuing vehicle $n$, the function $Q_n^{\pi}(s_t^n, a_t^n)$ calculates the expectation values of turning left, turning right and going straight according to the current state $s_t^n$ to assist the vehicle to select the optimal route to pursue the evading vehicle.

## III. MARL WITH PRIORITIZED EXPERIENCE

To introduce diversity among collaborative agents, we design a prioritized experience boosting MARL equipped with a prioritization network. Subsection III-A describes the overall framework and Subsection III-B presents the prioritization network in detail. Finally, the training process is introduced.

### A. Prioritized Experience Boosting MARL Framework

Emerging MARL algorithms, such as Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [28] and QMIX [43], adopt centralized training with randomly sampling experience to improve the utilization of experience and deploy the same trained model to all the agents. Homogeneous learning may lead agents to behave similarly. For example, two pursuing vehicles choose to chase behind one evading vehicle simultaneously, rather than one pursuing vehicle chasing and the other intercepting. As a result, homogeneous learning policy hinders collaboration among agents [44]. Moreover, randomly sampling experience also affects the training efficiency. Therefore, this paper proposes a prioritized experience boosting MARL framework, as shown in Fig. 2, to introduce diversity among agents. It employs a prioritization network $PN$ to select personalized training set $\mathcal{E}_{per}^{n} \in \mathcal{G}$ by a central server from the global experience replay buffer $\mathcal{G} = \{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_{\max\_\text{cap}}\}$ for each agent. $max\_cap$ represents the maximum capacity of the global buffer, and $\mathcal{E}$ is the replay experience collected by agent $n'$ in one epoch,

$$\mathcal{E} = \{(s_1^{n'}, a_1^{n'}, r_1^{n'}, s_2^{n'}), ..., (s_{st}^{n'}, a_{st}^{n'}, r_{st}^{n'}, s_{st+1}^{n'})\}. \quad (5)$$

In the prioritized experience boosting MARL framework, all agents upload exploration experience to the central server. Every agent samples prioritized experience for training and updates parameters in the global experience buffer via the prioritization network. The prioritization network is trained to model the relationship between the training set features and the reward change $\Delta r$ after updating parameters, thus assisting the agents in selecting appropriate experience for efficient training and improving decision performance.

The prioritized experience boosting MARL essentially optimizes the gradient descent and parameter update process of RL-based agents. It differs from the conventional training way of simply replaying experiences at the same frequency, regardless of their importance. By prioritizing experiences based on their significance, the framework enables agents to train more efficiently with optimal replay transitions and fosters better collaboration among the agents.

### B. Prioritization Network and Annealing Priority

The prioritization network is responsible for assessing the importance of experience replay transition $\mathcal{E}_i$ for all agents. For a given agent $n$ with parameters $\theta_n$, the prioritization network determines the priority and the sampling probability $P_n(i)$ of experience replay transition $\mathcal{E}_i$. Thus, the prioritization network is designed to estimate the performance gain of the agent $n$ after training with $\mathcal{E}_i$. And, as for RL, the rewards directly reflect the performance of an agent. Therefore, we use the prioritization network to fit the reward change $\Delta r_n^i$ after training the agent $n$ with experience replay transition $\mathcal{E}_i$,

$$\hat{\Delta r_n^i} = PN(\mathcal{E}_i, \theta_n; \vartheta), \quad (6)$$

where $\vartheta$ is the parameters of the prioritization network $PN$. Meanwhile, in order to improve the stability of the algorithm, we use the average reward over $k$ historical epochs as the base reward to calculate the reward change $\Delta r_n^i$. We adopt gradient back-propagation to update $\vartheta$. The loss of $PN$ is defined as,

$$J(\vartheta) = \frac{1}{K} \sum_{k} [PN(\mathcal{E}_i, \theta_n; \vartheta) - \Delta r_n^i]^2, \quad (7)$$

where $K$ is the batch size.

The prioritization network output $\hat{\Delta r_n^i}$ is used for stochastic sampling. The prioritization network computes the gain of each replay transition in the global experience replay buffer according to the current parameters of agent $n$, which is defined as $\{\hat{\Delta r_n^i} \,|i = 1, ..., max\_cap\}$. And then the maximum-minimum normalization is performed on the sequence,

$$q_n^i = \frac{\hat{\Delta r_n^i} - \min_j(\hat{\Delta r_n^j})}{\max_j(\hat{\Delta r_n^j}) - \min_j(\hat{\Delta r_n^j})} + \zeta, \quad (8)$$

where $\zeta$ is a small positive constant that prevents the sampling probability becoming zero. However, reward prioritization sampling may focus on a small subset of the experience that makes the agent prone to over-fitting. Therefore, the annealing priority is proposed to calculate sampling probabilities,

$$P_n(i) = \frac{(q_n^i)^\beta}{\sum_j (q_n^j)^\beta}, \quad (9)$$

where the exponent $\beta \in (0, 1]$ determines how much prioritization is introduced. $\beta = 0$ corresponds to the uniform sampling. In the early stage of training, uniform sampling is expected to facilitate agents learning and the prioritization network convergence. As training proceeds, the $PN$ can gradually and reliably compute the value of an experience replay transition and guide the agents gradient decreasing. In practice, we linearly anneal $\beta$ from $\beta_0$ to 1 to ensure stable MARL update and continuous performance improvement.

### C. Training Process of MARL with Prioritized Experience

Typical reinforcement learning utilizes random experience replay to estimate the distribution of policy and states via Monte Carlo. However, prioritized replay inevitably changes the distribution and introduces bias [38]. And the optimal solution that the estimates converge to is influenced by the bias. Therefore, this paper adopts Importance Sampling (IS) to abate the impact of the bias,

$$\omega_n^{i\,\prime} = (\max\_\text{cap} \times P_n(i))^{-\lambda}, \quad (10)$$

that fully compensates for the non-uniform probabilities $P_n(i)$ if $\lambda = 1$. And the weights is normalized by $1/\max_j \omega_n^{j\,\prime}$ for

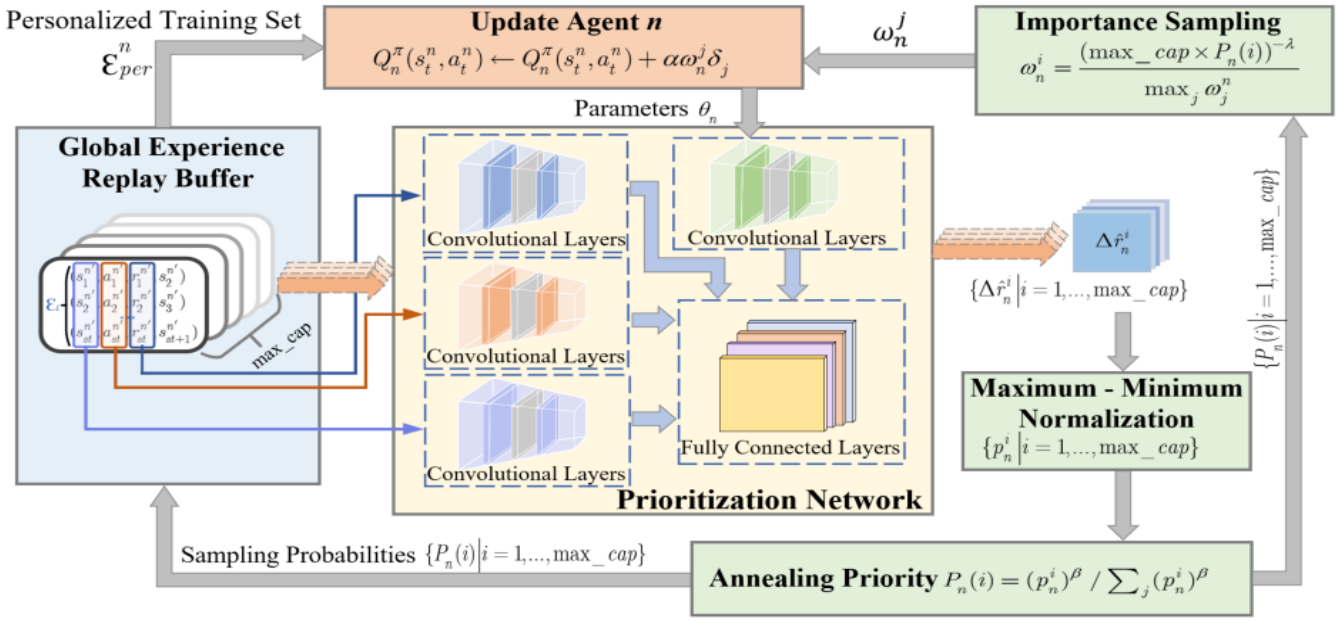


Fig. 2. Architecture of prioritized experience boosting MARL.

stability to get $\omega_n^i$. It is worth noting that the choice of hyper-parameters $\lambda$ interacts with $\beta$ in annealing priority. Increasing them both simultaneously encourages more aggressive priority sampling. Considering IS, Eq. (3) can be reformulated as,

$$Q_n^\pi(s_t^n, a_t^n) \leftarrow Q_n^\pi(s_t^n, a_t^n) + \alpha\, \omega_n^i\, \delta. \tag{11}$$

Introducing IS into the training process has another benefit. IS can reduce the step size of non-linear function approximation, e.g. deep neural networks. In prioritized experience boosting MARL, experience replay transitions favored by the prioritization network may be revisited many times, and the IS correction reduces the gradient magnitude to ensure that the agents converge to the globally optimal policy.

Distributed training is used in the proposed prioritized experience boosting MARL. The overall training process is shown in Algorithm 1. For every RL-based agent, the prioritization network firstly evaluates the priority of each experience transition according to the agent's parameters. And the annealing priority is obtained to help select personalized training set $\mathcal{E}_{per}^n$. Then the agent's parameters are updated with $\mathcal{E}_{per}^n$ via Eq. (11). After the parameters of all the agents have been updated, the prioritized experience boosting MARL is tested, and the gain of each agent's reward is calculated. Finally, the prioritization network is trained via Eq. (7). With this training process, the prioritization network can accurately compute the value of experience replay transitions for each agent. Moreover, by training with personalized and optimal experience, diverse multiple agents can largely improve collaboration and convergence efficiency.

**Algorithm 1:** Training Process of Prioritized Experience Boosting MARL

**Input:** $N$ agents, prioritization network $PN$, average reward of each agent in $k$ historical test epochs $[\bar{r}_1, \bar{r}_2, ..., \bar{r}_N]$, and global experience replay buffer $\mathcal{G}$

```
for n=1:N do
    for i=1:max_cap do
        Get the priority of E_i in G by PN considering
        parameters θ_n of agent n;
    end
    Calculate the annealing priority P_n for each replay
    transition via Eq. (8) and Eq. (9);
    Sample personalized training set E^n_per for agent n
    according to P_n;
    Update parameters θ_n of agent n via Eq. (11);
end
Test the updated MARL and get distributed rewards
[r_1, r_2, ..., r_N] ;
Calculate the change of rewards [Δr_1, Δr_2, ..., Δr_N] ;
Calculate the loss of PN via Eq. (7) and update PN;
```

## IV. Progression Cognition DQN-Based Cooperative Path Planning

This section will introduce a progression cognition DQN-based cooperative path planning for pursuing vehicles. Firstly, the attention-based progression cognition module is presented in Section IV-A. DQN-based path planning, as the core decision making algorithm, is then described in Section IV-B. Finally, Section IV-C introduces the decision-making and training process of the proposed PEPCRL-MVP.

### A. Attention-Based Progression Cognition Module

In complex urban traffic environments pursuing vehicles need real-time and accurate sensing of driving environments and the status of evading vehicles. We propose attention-based progression cognition module to extract critical traffic

features and assist pursuing vehicles to select suitable evading vehicle as the target. It helps each pursuing vehicle focus on only one evading vehicle and work with other pursuing vehicles in a group to improve pursuit performance. Moreover, the allocation of pursuing tasks with progression cognition enhance collaboration among pursuing vehicles.

The locations of the pursuing and evading vehicles are very important for collaboration and decision making of the pursuing vehicles. In this paper, the location of vehicle $i$ is denoted by $loc_t^i = \left\{C_l, pos_t^{i,l}\right\}$. $C_l$ denotes the binary code of lane $l$ where vehicle $i$ is located, and $pos_t^{i,l}$ denotes the distance between vehicle $n$ and the starting of lane $l$ at time $t$. And the length of $loc_t^i$ is denoted by $len_{loc}$. The positions of pursuing and evading vehicles can be represented respectively as $\mathcal{LOC}\_\mathcal{P}_t = \{loc_t^1, loc_t^2, ..., loc_t^N\}$ and $\mathcal{LOC}\_\mathcal{E}_t = \{loc_t^1, loc_t^2, ..., loc_t^M\}$. Moreover, the adjacency matrix $RT$ is used to represent the topology of roads. Assume that there are $L$ lanes in the pursuit-evasion environment. The size of matrix $RT$ is $L \times L$. An element $e_{i,j}$ in row $i$ and column $j$ of $RT$ indicates whether the vehicles can drive directly from lane $i$ to lane $j$.

To choose an optimal pursuit route, the information of the number of background vehicles in each lane is also utilized by the progression cognition module. The number of background vehicles in each lane forms a vector of size $1 \times L$, defined as $BV_t$. We use convolutional neural networks (CNNs) in the module to extract key traffic features from $RT$ and $BV_t$. Specifically, $RT$ is fed into the convolutional layers. Then its output combined with $BV_t$ is input to the fully connected layers, and finally the urban traffic feature $F$ is obtained.

As the core of progression cognition module, multi-head attention is used to help pursuing vehicles focus on evading vehicles. It simultaneously takes into account urban traffic features $F$. All pursuing vehicles share their locations. And $F$ is attached to the vehicle position vectors $loc_t^i$, which is a word embedding. Therefore, the query $q$ of multi-head attention is represented as $[[loc_t^1, F], ..., [loc_t^N, F]]$ and the key $K$ is represented as $[[loc_t^1, F], ..., [loc_t^M, F]]$. Group attention weights $W_g$ can be derived as

$$W_g = \psi^{linear}(Concact(W_1, W_2, ..., W_h)), \tag{12}$$

*in which,*

$$W_i = Softmax(\frac{q_i\ K_i^T}{\sqrt{d_K}}). \tag{13}$$

Here, $h$ is the number of heads and $d_K$ is the dimension of $K$'s features. The size of group attention weights $W_g$ finally obtained is $N \times M$. The $n$th row of $W_g$ represents the attention weight of pursuing vehicle $n$ on all evading vehicles, where a larger value means more attention. Every pursuing vehicle selects an evading vehicle with the maximum attention weight as its target vehicle. Due to the high dynamic of the urban traffic, the target evading vehicle selected by pursuing vehicle $n$ may vary from time steps. Therefore, the progression cognition divides the pursuing vehicles adaptively to collaborative groups according to the traffic situations.

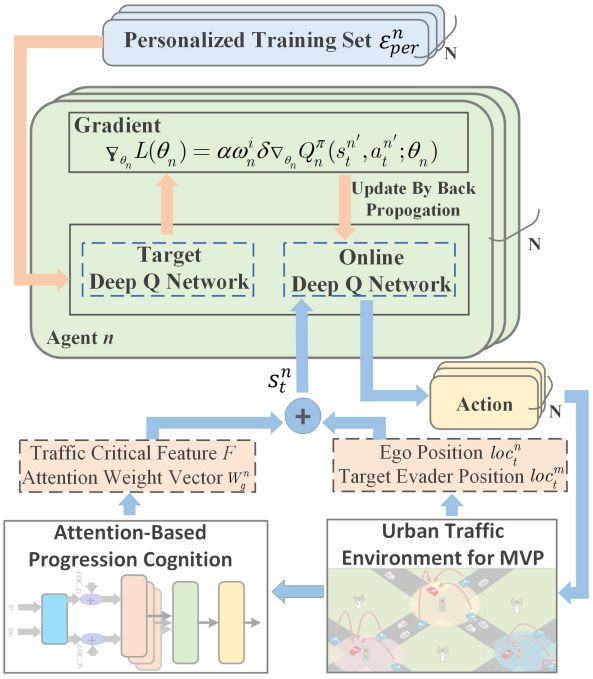


Fig. 3. Architecture of DQN-based multi-vehicle pursuit path planning.

### B. Multi-Vehicle Pursuit Path Planning

Deep Q-Network (DQN) is a popular reinforcement learning algorithm and has been applied for decision-making in various scenarios. In DQN, artificial neural networks (ANNs) are used to approximate $Q_n^\pi(s_t^n, a_t^n)$. DQN adopts a dual network framework, consisting of $online$ network and $target$ network, which have the same structure. The two ANNs are parameterized by $\theta_n$ and $\theta_n{}'$. In MVP path planning, DQN is utilized to evaluate the value $Q$ of the pursuing vehicle's each action according to its real-time state $s_t^n$, denoted as $Q_n^\pi(s_t^n\,|\theta_n)$. $Q_n^\pi(s_t^n\,|\theta_n)$ is used as the policy for agent $n$ to select the appropriate action. The architecture of multi-vehicle pursuit path planning algorithms is shown in Fig. 3.

In this paper, the action space of each pursuing vehicle includes three actions, turning left, turning right, and going straight. For agent $n$, the state $s_t^n$ consists of four parts, including its own position $loc_t^n$, the position of the target evading vehicle being focused on $loc_t^m$, the traffic critical feature $F$ and its attention weight vector $W_g^n$. Here, $W_g^n$ represents the attention weight of agent $n$ on all evading vehicles obtained via attention-based progression cognition module, which is the $n$th row of $W_g$.

To motivate the capture of pursuing vehicles and incentivize efficient training, a carefully designed reward $r_t^n$ consists of *sparse reward* and *dense reward*. ***Sparse Reward***: Only when a collaborative group successfully captures its target, all members in the group obtain a positive reward $V$. ***Dense Reward***: Sparsity of reward hinders the exploration of optimal policy by agents. Therefore, a *dense reward* is set for each pursuing vehicle to address this problem. The dense reward contains the following two components, 1) The pursuing vehicles which do not capture the target vehicle are given a negative reward of $c$ at each time step; 2) A distance-sensitive reward is set

to improve the pursuing efficiency. When a pursuing vehicle reduces the distance from its current target compared to that at the last time step, it will obtain a positive reward, and conversely, it will be punished with a negative reward.

Therefore, the formulation of $r_t^n$ is expressed as

$$r_t^n = \begin{cases} V & \text{if successful pursuit} \\ c \times t + \sigma \left( d_t^{n,m} - d_{t-1}^{n,m} \right) & \text{else} \end{cases}, \tag{14}$$

where $\sigma$ is a negative reward factor, and $d_t^{n,m}$ denotes the distance of the pursuing vehicle $n$ from its target evading vehicle $m$ at time step $t$.

Adopting prioritized experience boosting MARL, we can compute the following gradient by differentiating the loss function with respect to the weights,

$$\nabla_{\theta_n} L(\theta_n) = \alpha \, \omega_n^i \, \delta \, \nabla_{\theta_n} Q_n^{\pi}(s_t^{n'}, a_t^{n'}; \theta_n), \tag{15}$$

*in which,*

$$\delta = r_t^{n'} + \gamma \max_{a_{t+1}^{n'}} Q_n^{\pi'}(s_{t+1}^{n'}, a_{t+1}^{n'}; {\theta_n}') - Q_n^{\pi}(s_t^{n'}, a_t^{n'}; \theta_n), \tag{16}$$

$$(s_t^{n'}, a_t^{n'}, r_t^{n'}, s_{t+1}^{n'}) \in \mathcal{E}_{per}^n, \tag{17}$$

$\theta_n$ is updated via stochastic gradient descent and Eq. (15). And $target$ network performs soft update at each training step,

$${\theta_n}' \leftarrow \tau \theta_n + (1 - \tau) {\theta_n}'. \tag{18}$$

### C. PEPCRL-MVP Decision-Making and Training Process

The decision-making and training process of the proposed PEPCRL-MVP is shown in Algorithm 2. $N$ DQN-based path planning agents and the prioritization network are initialized firstly. At the beginning of each epoch, $N$ agents are trained distributedly with personalized and prioritized experience via Algorithm 1, if the number of transitions in the global experience replay buffer $\mathcal{G}$ reaches $max\_cap$. Then the pursuit-evasion environment is initialized to test PEPCRL-MVP. In each time step of pursuit, attention-based progression cognition processing is invoked to get $F$ and $W_g$. Then each agent uses the partial observation to obtain optimal path planning and the strategy of all agents is performed. At the end of the epoch, experience of all pursuing vehicles is stored to $\mathcal{G}$. Finally, change of rewards is calculated and used to update the prioritization network $PN$ via Eq. (7).

In the decision-making and training process of PEPCRL-MVP, collaboration among agents is performed in the following three aspects. ***1) Information Sharing:*** During the pursuit process, the pursuing vehicles upload their own positions and observation information to the central server for traffic feature extraction and task allocation; ***2) Task Allocation:*** The proposed attention-based progression cognition module dynamically calculates group attention to adaptively group pursuing vehicles; ***3) Experience Sharing:*** To increase the utilization of experience, all agents upload their experience to the global experience buffer. During the training process, every agent selects prioritized experience from the global experience buffer via the prioritization network for training and updates its parameters.

**Algorithm 2:** PEPCRL-MVP Decision-making and Online Training Algorithm

```
Initialize N DQN-based path planning agents and the
  prioritization network PN;
Initialize global experience replay buffer G ;
for e=1:max_epoch do
    if len(G) = max_cap then
        Train N agents via Algorithm 1, Eq. (15) ;
    end
    Initialize an urban pursuit-evasion environment and
      obtain S_1 = {LOC_P_1, LOC_E_1, BV_1};
    Get F and W_g by progression cognition;
    for t=1:st do
        for n=1:N do
            s_t^n ← [loc_t^n, loc_t^m, F, W_g^n];
            Obtain a_t^n by Q_n^π(s_t^n |θ_n);
        end
        Perform the strategy a_t and observe S_{t+1};
        Get F and W_g by progression cognition;
        Append (s_t^n, a_t^n, r_t^n, s_{t+1}^n) to E_n, n = 1, ..., N;
        S_t ← S_{t+1} ;
        if Done then
            break;
        end
    end
    Store E_n to G, n = 1, ..., N;
    Get per step average rewards [r_1, r_2, ..., r_N];
    if len(G) = max_cap then
        Calculate change of rewards and update PN;
    end
    Update mean rewards of k epochs [r̄_1, r̄_2, ..., r̄_N];
end
```

## V. Experiments and Results

### A. The Simulator and Settings

As a MARL algorithm, PEPCRL-MVP collects training data and updates parameters by interacting with the simulated urban traffic environment. To comprehensively evaluate the proposed PEPCRL-MVP, we build three urban traffic road scenes with bidirectional two lanes based on SUMO [42], including $3 \times 3$ and $4 \times 5$ grid-pattern urban roads, and real map-based urban roads. The real map-based urban roads simulate those in an area inside the second ring road of Beijing, bridging the 'sim-to-real' gap. And the real urban road map is obtained from an open-sourced map website.[1] During the simulation process, the number of background vehicles remains constant, and the background vehicles follow randomly selected routes. Moreover, to evaluate the robustness of PEPCRL-MVP, we design three different difficulty levels of MVP tasks with variable numbers of pursuing vehicles $N$ and evading vehicles $M$, respectively, 6 pursuing vehicles chasing 3 evading vehicles (denoted by P6-E3), 7 pursuing vehicles chasing 4 evading vehicles (denoted by P7-E4) and 8 pursuing vehicles chasing 5 evading vehicles (denoted by P8-E5). All evading vehicles

[1] https://www.openstreetmap.org/

randomly select escape routes. The simulation parameters are shown in TABLE I and the PEPCRL-MVP parameters are shown in TABLE II. The internal structure of DQN is shown in TABLE III.

TABLE I
SIMULATOR SETTINGS

| Parameters | | Values |
|---|---|---|
| maximum time steps $st$ | | 800 |
| capture distance $d_{min}$ | | 5 $m$ |
| maximum speed $v_{\max}$ | | 20 $m/s$ |
| maximum acceleration $ac_{\max}$ | | 0.5 $m/s^2$ |
| maximum deceleration $de_{\max}$ | | 4.5 $m/s^2$ |
| 3×3 urban roads | number of lanes $L$ | 48 |
| | length of location code $len_{loc}$ | 7 |
| | number of junctions | 16 |
| | length of each lane | 500 $m$ |
| | number of background vehicles | 240 |
| 4×5 urban roads | number of lanes $L$ | 98 |
| | length of location code $len_{loc}$ | 8 |
| | number of junctions | 30 |
| | length of each lane | 400 $m$ |
| | number of background vehicles | 500 |
| real map-based urban roads | number of lanes $L$ | 106 |
| | length of location code $len_{loc}$ | 8 |
| | number of junctions | 46 |
| | number of background vehicles | 300 |

TABLE II
PARAMETER SETTINGS

| Parameters | Values | Parameters | Values |
|---|---|---|---|
| $\alpha$ | $10^{-4}$ | $V$ | 400 |
| $\gamma$ | 0.9 | $c$ | 0.02 |
| $\beta_0$ | 0.01 | $\sigma$ | 5 |
| $\lambda$ | 0.5 | $\tau$ | 0.001 |

TABLE III
STRUCTURE OF THE DEEP Q NETWORK

| Layers | Deep Q Network |
|---|---|
| Input | (batch size, $2 \times len_{loc} + M + 1$) |
| Dense Layer 1 | ($2 \times len_{loc} + M + 1$, 32 ) |
| Activation Function | Relu |
| Dense Layer 2 | (32,48) |
| Activation Function | Relu |
| Dense Layer 3 | (48,32) |
| Activation Function | Relu |
| Dense Layer 4 | (32,16) |
| Activation Function | Relu |
| Dense Layer 5 | (16,3) |
| Activation Function | SoftMax |
| Output | (batch size, 3) |

### *B. Ablation Experiments*

We conduct 100 tests on every model and measure the pursuit performance in terms of five metrics, which are average reward (AR), the standard deviation of reward (SDR), average time steps (ATS), the standard deviation of time steps (SDTS), and the pursuing success rate (SR). Ablation experiments are designed to investigate the effect of the proposed prioritized experience selection and progression cognition modules in PEPCRL-MVP. The ablation experiment results are shown in columns 4 to 6 in TABLE IV. The results of A-MVP correspond to a DQN-based path planning with progression cognition without prioritized experience selection, and the results of B-MVP correspond to a DQN-based path planning equipped with prioritized experience selection without attention-based progression cognition.

PEPCRL-MVP shows the best performance in scenes with different difficulty levels under the same urban traffic road structure. In the real map-based urban road, compared with A-MVP, the AR of PEPCRL-MVP increases $52.53\%$, $47.46\%$, and $20.96\%$, respectively, and the ATS of PEPCRL-MVP decreases $2.82\%$, $3.31\%$, and $3.24\%$ in P6-E3, P7-E4, and P8-E5 difficulty levels, respectively. Furthermore, the SDR and SDTS of PEPCRL-MVP are comparable to those of A-MVP. These results reveal that prioritized experience selection can effectively promote cooperation among pursuing vehicles and improve pursuing performance.

Given the pursuing difficulty level, PEPCRL-MVP also substantially outperforms the other methods under different urban traffic road scenes. Taking the P6-E3 as an example, the SDTS of PEPCRL-MVP is $7.55\%$, $2.80\%$ and $1.35\%$ lower than that of B-MVP under $3 \times 3$, $4 \times 5$, and real map-based scenes, respectively. The results show the proposed method has excellent robustness. The SDR is $10.72\%$ lower than that of B-MVP on average in the P7-E4 difficulty level. These results can be explained by the fact that attention-based progression cognition can considerably enhance the stability of pursuing vehicles' performance. Moreover, the proposed PEPCRL-MVP has a better generalization and pursuing validity than A-MVP and B-MVP. It is evident that in different scenes, whether the urban road structure or the pursuing difficulty level is different, PEPCRL-MVP has the greatest SR. Concretely, the SR of PEPCRL-MVP is $19.67\%$, $11.92\%$ higher than that of A-MVP and B-MVP on average, respectively.

In addition, to investigate the impact of the prioritized experience selection on the PEPCRL-MVP's training convergence, we present the average reward with the P6-E3 setting in Fig. 4. Fig. 4 (a) shows the average reward with training epochs under the $3 \times 3$ grid pattern. It can be noticed that compared with A-MVP, PEPCRL-MVP has a smaller fluctuation and a more stable convergence in the late stage of training. For the $4 \times 5$ grid pattern, as shown in Fig. 4 (b), although the average reward of A-MVP grows fast in the early training stage, PEPCRL-MVP has a faster convergence speed and higher convergence trend in the late training stage. In Fig. 4 (c), under the real map-based urban road, PEPCRL-MVP shows an obviously stronger convergence. According to the results in Fig 4, it can be observed that the prioritized experience selection greatly facilitates the convergence of agent learning, and helps improve the pursuing performance.

The above results obtained from the ablation experiments demonstrate that the prioritized experience selection network can select appropriate training set for each agent, which overcomes the problem of agent differentiation in the existing MVP approaches. It can effectively promote the convergence of agent learning, and enhance cooperation among agents. Furthermore, the progression cognition module can decide appropriate targets for each pursuing vehicle according to the real-time traffic situation and MVP task, consequently

TABLE IV
EVALUATION RESULTS

| Road Structure | N-M | Evaluate Metrics | PEMARL-MVP | A-MVP | B-MVP | C-MVP | DQN | DDPG | MADDPG | TD3-DMAP | PPO | T$^3$OMVP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3×3 | P6-E3 | AR | 3.164 | 2.463 | 2.558 | 3.041 | 2.154 | 1.461 | 1.552 | 2.252 | 2.247 | 0.698 |
| | | SDR | 6.927 | 7.656 | 8.069 | 6.833 | 7.179 | 6.549 | 6.831 | 6.919 | 7.831 | 5.701 |
| | | ATS | 642.87 | 684.77 | 655.40 | 643.21 | 677.35 | 700.75 | 693.26 | 676.9 | 687.30 | 717.61 |
| | | SDTS | 175.410 | 173.669 | 189.742 | 176.495 | 179.388 | 166.467 | 171.972 | 176.389 | 181.518 | 136.882 |
| | | SR | 0.62 | 0.48 | 0.51 | 0.58 | 0.47 | 0.36 | 0.42 | 0.48 | 0.49 | 0.37 |
| | P7-E4 | AR | 2.183 | 1.407 | 1.915 | 1.890 | 1.372 | 0.554 | 0.974 | 0.608 | 1.365 | 0.746 |
| | | SDR | 6.694 | 6.482 | 7.004 | 6.702 | 6.152 | 5.889 | 5.740 | 5.617 | 6.082 | 5.711 |
| | | ATS | 684.50 | 692.77 | 688.13 | 687.79 | 695.89 | 738.15 | 726.81 | 735.03 | 696.56 | 700.54 |
| | | SDTS | 150.082 | 149.366 | 159.150 | 152.516 | 153.201 | 119.352 | 132.714 | 120.75 | 152.391 | 145.638 |
| | | SR | 0.55 | 0.46 | 0.49 | 0.52 | 0.45 | 0.35 | 0.37 | 0.36 | 0.47 | 0.38 |
| | P8-E5 | AR | 1.186 | 0.864 | 0.997 | 0.927 | 0.672 | 0.138 | 0.294 | 0.084 | 0.589 | 0.484 |
| | | SDR | 4.537 | 4.814 | 5.030 | 4.484 | 4.769 | 3.472 | 3.825 | 2.046 | 4.779 | 4.268 |
| | | ATS | 690.51 | 701.13 | 696.55 | 695.95 | 723.58 | 749.65 | 741.73 | 764.18 | 699.74 | 733.95 |
| | | SDTS | 148.869 | 133.648 | 148.770 | 149.458 | 123.537 | 102.634 | 109.840 | 101.720 | 139.037 | 111.974 |
| | | SR | 0.52 | 0.46 | 0.48 | 0.49 | 0.44 | 0.35 | 0.35 | 0.30 | 0.46 | 0.37 |
| 4×5 | P6-E3 | AR | -0.176 | -0.874 | -0.628 | -0.591 | -1.249 | -2.179 | -1.979 | -1.395 | -1.903 | -2.302 |
| | | SDR | 3.769 | 4.573 | 2.301 | 4.046 | 5.329 | 1.484 | 3.722 | 5.428 | 4.121 | 4.381 |
| | | ATS | 726.55 | 738.63 | 734.33 | 728.18 | 742.07 | 748.62 | 745.99 | 739.82 | 743.47 | 738.68 |
| | | SDTS | 126.891 | 124.957 | 130.540 | 125.084 | 124.271 | 109.798 | 115.423 | 130.294 | 113.959 | 122.204 |
| | | SR | 0.39 | 0.35 | 0.38 | 0.37 | 0.33 | 0.27 | 0.27 | 0.31 | 0.29 | 0.27 |
| | P7-E4 | AR | -1.728 | -1.658 | -1.643 | -1.712 | -1.842 | -2.191 | -1.916 | -1.863 | -2.100 | -2.388 |
| | | SDR | 3.894 | 4.097 | 3.672 | 3.782 | 3.783 | 0.399 | 2.768 | 4.509 | 2.949 | 3.886 |
| | | ATS | 724.81 | 736.23 | 731.06 | 729.85 | 739.24 | 765.01 | 761.77 | 759.73 | 774.56 | 774.72 |
| | | SDTS | 130.391 | 118.947 | 114.458 | 128.493 | 127.289 | 111.900 | 124.328 | 132.343 | 74.837 | 78.892 |
| | | SR | 0.32 | 0.29 | 0.30 | 0.30 | 0.27 | 0.21 | 0.22 | 0.25 | 0.16 | 0.13 |
| | P8-E5 | AR | -1.892 | -2.285 | -1.972 | -2.016 | -2.314 | -2.785 | -2.735 | -2.187 | -2.026 | -3.725 |
| | | SDR | 3.075 | 2.819 | 3.012 | 3.140 | 3.396 | 2.722 | 2.431 | 3.059 | 3.229 | 2.646 |
| | | ATS | 758.66 | 760.63 | 765.71 | 762.57 | 772.47 | 781.62 | 781.54 | 779.42 | 773.13 | 784.37 |
| | | SDTS | 79.871 | 85.183 | 78.258 | 80.492 | 83.856 | 23.915 | 37.835 | 35.683 | 76.244 | 12.699 |
| | | SR | 0.18 | 0.16 | 0.16 | 0.17 | 0.15 | 0.11 | 0.11 | 0.12 | 0.15 | 0.09 |
| Real Map | P6-E3 | AR | -0.871 | -1.835 | -1.399 | -1.035 | -2.136 | -2.903 | -2.264 | -2.115 | -1.950 | -2.567 |
| | | SDR | 1.194 | 0.941 | 1.394 | 1.218 | 0.856 | 0.769 | 0.971 | 0.832 | 1.326 | 0.983 |
| | | ATS | 617.48 | 635.37 | 624.46 | 622.74 | 645.53 | 663.81 | 651.74 | 647.91 | 633.82 | 644.35 |
| | | SDTS | 197.582 | 206.494 | 200.281 | 196.934 | 191.161 | 188.034 | 192.899 | 200.556 | 198.539 | 204.794 |
| | | SR | 0.68 | 0.59 | 0.62 | 0.65 | 0.55 | 0.49 | 0.50 | 0.51 | 0.58 | 0.47 |
| | P7-E4 | AR | -1.497 | -2.849 | -2.068 | -2.107 | -3.097 | -3.773 | -3.318 | -3.019 | -2.971 | -3.416 |
| | | SDR | 0.835 | 0.954 | 1.261 | 1.192 | 0.751 | 0.662 | 0.892 | 0.767 | 0.946 | 0.596 |
| | | ATS | 635.90 | 657.62 | 643.58 | 642.19 | 665.49 | 674.75 | 660.97 | 668.32 | 677.15 | 685.13 |
| | | SDTS | 195.644 | 189.341 | 201.894 | 193.577 | 181.518 | 183.612 | 208.259 | 180.703 | 186.439 | 126.547 |
| | | SR | 0.62 | 0.48 | 0.53 | 0.55 | 0.45 | 0.41 | 0.47 | 0.43 | 0.39 | 0.30 |
| | P8-E5 | AR | -2.629 | -3.326 | -2.989 | -3.017 | -4.076 | -4.698 | -3.871 | -3.809 | -3.275 | -3.984 |
| | | SDR | 0.801 | 0.592 | 0.836 | 0.763 | 0.410 | 0.470 | 0.837 | 0.966 | 0.714 | 0.241 |
| | | ATS | 662.33 | 684.54 | 671.44 | 670.53 | 701.88 | 694.29 | 684.16 | 678.16 | 684.61 | 707.06 |
| | | SDTS | 153.71 | 175.043 | 160.515 | 151.375 | 150.537 | 139.478 | 179.114 | 178.559 | 180.397 | 155.917 |
| | | SR | 0.56 | 0.41 | 0.48 | 0.50 | 0.36 | 0.34 | 0.41 | 0.37 | 0.38 | 0.27 |

improving pursuing efficiency and system stability.

To verify the necessity of inputting group attention to DQN, we conducted an ablation experiment C-MVP, whose the DQN inputs are the ego pursuing vehicle position, the position of its target vehicle, and the urban traffic features without group attention. TABLE IV shows the experiment results. In the three urban traffic road scenes, the ATS of PEMARL-MVP is $4.38$ time steps less than that of C-MVP on average. With increasing difficulty, the advantage of PEMARL-MVP in pursuit efficiency becomes more and more apparent compared with C-MVP. For example, in the real map urban traffic road, the ATS of PEMARL-MVP decreases by $0.84\%$, $0.98\%$, and $1.2\%$ in P6-E3, P7-E4, and P8-E5 difficulty levels, respectively. Moreover, in all tested scenes, the SR of PEMARL-MVP improves by $7.5\%$ over that of C-MVP. The superiority of PEMARL-MVP over C-MVP illustrates that adding group attention to DQN can assist in the decision-making of pursuit vehicles and improve the pursuit efficiency.

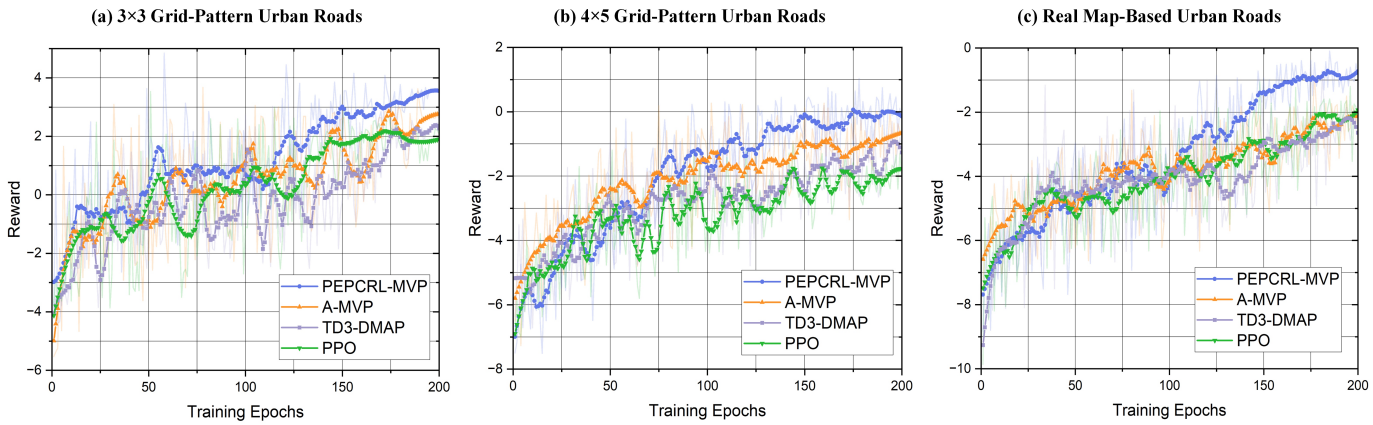


Fig. 4. Training reward of P6-E3 in different scenes.

### C. Comparison with Other Methods

We compare the PEPCRL-MVP to the other state-of-the-art RL approaches for MVP, including DQN, DDPG, MADDPG, Twin Delayed Deep Deterministic policy gradient-Decentralized Multi-Agent Pursuit (TD3-DMAP) [9], Proximal Policy Optimization (PPO), and Transformer-based Time and Team RL for Observation-constrained MVP ($T^3$OMVP) [26]. The comparison results are presented in the columns beginning from column 7 in TABLE IV.

According to the comparison results, for any given pursuing difficulty level, PEPCRL-MVP shows remarkable performance improvement under all the traffic road scenes. For the setting of P8-E5, the AR of PEPCRL-MVP is improved by $43.41\%$ and $42.5\%$ on average, respectively, compared with DQN and PPO under the $3\times3$, $4\times5$ and real map-based road structures, which are the top two performances of all comparison methods in general. And the ATS of PEPCRL-MVP decreases by $4.18\%$ on average compared with other methods in the P8-E5 difficulty level under the real map-based urban road. The results show that the proposed PEPCRL-MVP approach can highly improve pursuing efficiency and have excellent adaptability to different road scenes and traffic situations.

Under the same urban traffic road scenes, PEPCRL-MVP shows competitive robustness and pursuing effectiveness at different pursuing difficulty levels. Under the $3\times3$ grid pattern urban roads, as the pursuing difficulty level increases, the ATS of PEPCRL-MVP is $6.46\%$, $1.73\%$, and $1.32\%$ lower than that of PPO which has the second-best performance, in the P6-E3, P7-E4, and P8-E5 difficulty levels respectively. Comprehensively, we evaluate the SR metric for all methods to further compare PEPCRL-MVP to other methods. The SR of PEPCRL-MVP is $51.09\%$ higher than that of other methods on average under the $4\times5$ scene, and $47.53\%$ higher than that of other methods on average for all scenes. These results indicate that PEPCRL-MVP greatly improves pursuing efficiency and has stronger robustness.

It is noted that there is no significant SR and SDTS performance improvement by PEPCRL-MVP. This can be explained by the fact that we set the maximum time steps to 800, which leads to a higher standard deviation for the methods with better performance in the 100 tests. Although the SDR and SDTS do not obtain great results, there is a considerable increase in AS, ATS, and SR for the PEPCRL-MVP approach. Specifically, in the P7-E4 difficulty level under the real map-based simulation environment, even if the SDR and SDTS of PEPCRL-MVP are $11.16\%$ and $7.78\%$ higher than those of DQN, the AR of PEPCRL-MVP is greatly improved, increasing by $51.66\%$. The SDR and SDTS o f PEPCRL-MVP are respectively $11.54\%$ and $3.37\%$ lower than those of PPO, which has the second-best performance. At the same time, the AR increases by $40.81\%$, and the ATS decreases by $6.46\%$ in the P6-E3 difficulty level under the $3\times3$ road scene. These results demonstrate that PEPCRL-MVP can achieve great improvements both in algorithm stability and pursuing efficiency in simple scenes, and sacrifices some stability to obtain higher pursuing efficiency and better average performance in some complex scenes.

Fig. 4 describes the convergence curve of average reward with training epochs of the P6-E3 under different road structures. In Fig. 4 (a), both TD3-DMAP and PPO have large fluctuations in the late stage of training under the $3\times3$ road structure scene, while TD3-DMAP shows a superior growth trend and stable convergence. Fig. 4 (b) depicts the average reward under the $4\times5$ urban road scene, presenting that the PEPCRL-MVP has advantageous performances in both convergence rate and convergence stability compared with TD3-DMAP and PPO. For the real map-based urban road scene, as shown in Fig. 4 (c), compared with other methods, PERL has a better convergence trend and higher reward. In conclusion, Fig. 4 illustrates that compared with TD3-DMAP and PPO, which belong to the superior performance in all comparison methods, PEPCRL-MVP makes the competitive convergence trend and stability, demonstrating its superiority and effectiveness.

### D. Case Study

In this section, we analyze the PEPCRL-MVP pursuing processes in a case study with a real map-based scene in detail. Representative results are shown in Fig. 5. Fig. 5 (a) presents the distribution of background vehicles and the pursuing routes, where, the number marked next to the lane represents the average number of background vehicles per time

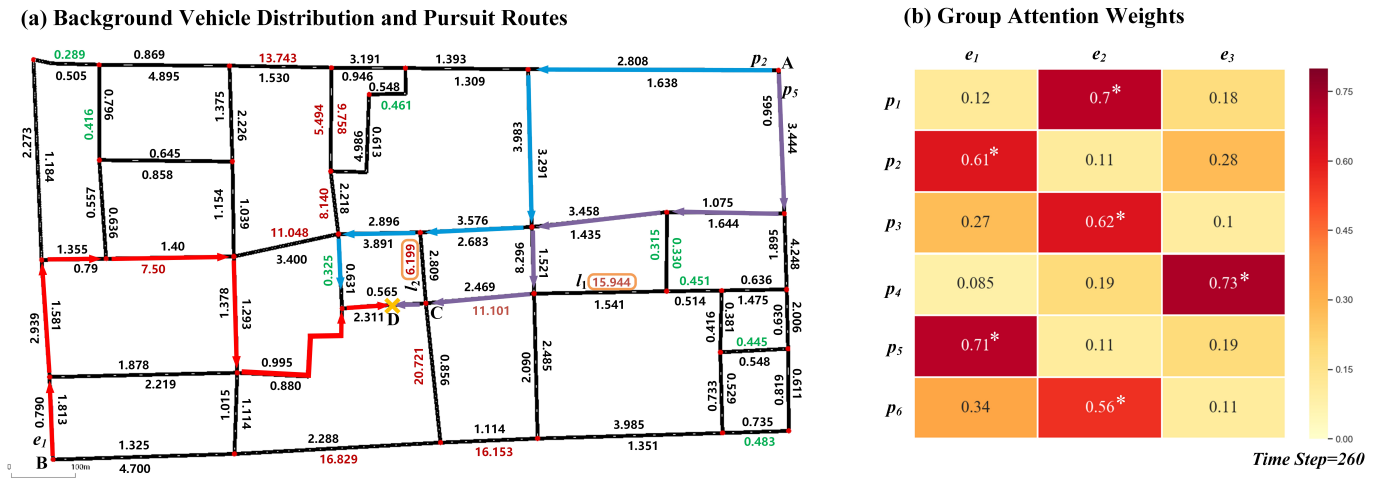


Fig. 5. Case study of P6-E3 in real map-based scene.

step in this lane during the pursuit. If the average number of vehicles in a lane is greater than 10, it means that the lane is congested. Also, Fig. 5 (a) shows the routes of pursuing vehicles $p_2$ and $p_5$. Following the group attention weights, $p_2$ and $p_5$ form a group to capture $e_1$ from $A$. From the routes of $p_2$ and $p_5$, it can be seen that they predict the trajectory of the target evading vehicle and collaboratively pursue and intercept $e_1$. It is worth noting that $p_5$ plans the path to $C$ while avoiding the congested lane $l_1$ and lane $l_2$. Finally, $p_5$ catches $e_1$ at $D$. This pursuit process shows the efficient cooperation of $p_2$ and $p_5$.

Fig. 5 (b) shows the group attention weights in 260 time steps during the pursuit. It can be observed that $p_2$ and $p_5$ both focus their attention on $e_1$. Moreover, each pursuing vehicle has its own target evading vehicle and each evading vehicle may be pursued by one or more pursuing vehicles. It indicates that the progression cognition module can select suitable targets for pursuing vehicles according to the traffic situation and the locations of evading vehicles. The PEPCRL-MVP can achieve efficient and effective collaborative multi-vehicle pursuit.

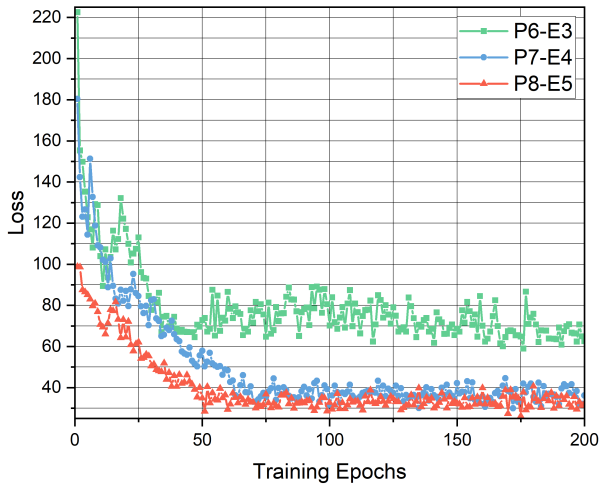


Fig. 6. Training loss of prioritization network in the real map-based scene.

To investigate the impact of the prioritization network, we show the training loss of the prioritization network in the real map-based scene in Fig. 6. The prioritization networks with different numbers of pursuing and evading vehicles all converge in 100 training epochs. The training of the prioritization network with a smaller number of agents converges more easily, but the mean square error between the network output and reward gain is larger. It is clear from Fig. 6 that as the number of agents increases, the loss of the prioritization network decreases. This suggests that extensive experience collection greatly contributes to prioritized network performance. It also demonstrates that the prioritization network can effectively evaluate the global experience pool, thus facilitating the learning and collaboration of multiple agents.

## VI. Conclusion

The emerging MARL technology is promising for multi-vehicle pursuit applications. However, the mission and safety-critical MVP tasks present great challenges, especially for the chasing of multiple target vehicles. While there are existing MARL algorithms proposed for MVP, they usually applied centralized training with randomly selected experience samples and did not adapt well to dynamically changing traffic situations. To address the problems in the existing MVP algorithm, in this paper we proposed a novel MVP approach (called PEPCRL-MVP) to improve MARL learning, collaboration, and MVP performance in dynamic urban traffic scenes. There are two major new components included in PEPCRL-MVP, a prioritization network and an attention-based progression cognition module. The prioritization network was introduced to effectively select training experience samples and increase diversity for the optimization and behavior of MARL, which improved agent collaboration and extensive exploration of experience. The progression cognition module was introduced to extract key traffic features from the sensor data and support the pursuing vehicles to adaptive adjust their target evading vehicles and path planning according to the real-time traffic situations. A simulator was developed for evaluation of the proposed PEPCRL-MVP approach and

comparison with existing ones. Extensive experiments were conducted over urban roads in an area inside the second ring road of Beijing on PEPCRL-MVP and several approaches. Experiment results demonstrate that PEPCRL-MVP significantly outperforms the other methods for all the investigated road scenes in terms of performance metrics including pursuing success rate and average rewards. The results also demonstrate the effectiveness of the proposed two components. Jointly they largely improve collaboration and traffic awareness, leading to improved MVP performance. In the future, we will investigate the impact of additional factors in MVP, such as pedestrians, social activities, and communication delay, on the design and analysis of MVP approaches. We will also design smarter MVP methods for more real scenes, such as evading vehicles not following traffic rules.

## ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 62176024 and 62071179), the National Key R&D Program of China (2022YFB2503202), Engineering Research Center of Information Networks, Ministry of Education, EPSRC with RC Grant reference EP/Y027787/1, UKRI under grant number EP/Y028317/1, the European Horizon 2020 MSCA programme under grant agreement No 824019 and 101022280, Horizon Europe MSCA programme under grant agreement No 101086228.



## REFERENCES

[1] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[2] Z. Zhu, N. Pivaro, S. Gupta, A. Gupta, and M. Canova, "Safe model-based off-policy reinforcement learning for eco-driving in connected and automated hybrid electric vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 387–398, 2022.

[3] A. Delarue, R. Anderson, and C. Tjandraatmadja, "Reinforcement learning with combinatorial actions: An application to vehicle routing," *Advances in Neural Information Processing Systems*, vol. 33, pp. 609–620, 2020.

[4] Z. Cao, S. Xu, X. Jiao, H. Peng, and D. Yang, "Trustworthy safety improvement for autonomous driving using reinforcement learning," *Transportation Research Part C-Emerging Technologies*, vol. 138, p. 103656, MAY 2022.

[5] Z. Li, J. Jiang, W.-H. Chen, and L. Sun, "Autonomous lateral maneuvers for self-driving vehicles in complex traffic environment," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2022.

[6] B. Xu, Y. Wang, Z. Wang, H. Jia, and Z. Lu, "Hierarchically and cooperatively learning traffic signal control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 669–677.

[7] T. Pan and Y. Yuan, "A region-based relay pursuit scheme for a pursuit–evasion game with a single evader and multiple pursuers," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 3, pp. 1958–1969, 2023.

[8] *Patrol guide. section: Tactical operations. procedure no: 221-15*, New York City Police Department, 2016, available: https://www1.nyc.gov/assets/ccrb/downloads/pdf/investigations_pdf/pg221-15-vehicle-pursuits.pdf.

[9] C. De Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized multi-agent pursuit using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4552–4559, 2021.

[10] E. Candela, L. Parada, L. Marques, T.-A. Georgescu, Y. Demiris, and P. Angeloudis, "Transferring multi-agent reinforcement learning policies for autonomous driving using sim-to-real," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8814–8820.

[11] O. Natan and J. Miura, "End-to-end autonomous driving with semantic depth cloud mapping and multi-agent," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 557–571, 2023.

[12] Z. Hu, Y. Qiao, X. Li, J. Huang, Y. Jia, and Z. Zhong, "Design and experimental validation of event-triggered multi-vehicle cooperation in conflicting scenarios," *Frontiers of Information Technology & Electronic Engineering*, vol. 23, no. 11, pp. 1700–1713, NOV 2022.

[13] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.

[14] B. Xu, S. E. Li, Y. Bian, S. Li, X. J. Ban, J. Wang, and K. Li, "Distributed conflict-free cooperation for multiple connected vehicles at unsignalized intersections," *Transportation Research Part C-Emerging Technologies*, vol. 93, pp. 322–334, AUG 2018.

[15] C. Chen, Q. Xu, M. Cai, J. Wang, J. Wang, and K. Li, "Conflict-free cooperation method for connected and automated vehicles at unsignalized intersections: Graph-based modeling and optimality analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 897–21 914, 2022.

[16] J. Liang, Y. Li, G. Yin, L. Xu, Y. Lu, J. Feng, T. Shen, and G. Cai, "A mas-based hierarchical architecture for the cooperation control of connected and automated vehicles," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1559–1573, 2023.

[17] M. Jiang, T. Wu, Z. Wang, Y. Gong, L. Zhang, and R. P. Liu, "A multi-intersection vehicular cooperative control based on end-edge-cloud computing," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 2459–2471, 2022.

[18] E. Garcia, D. W. Casbeer, A. Von Moll, and M. Pachter, "Multiple pursuer multiple evader differential games," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2345–2350, 2020.

[19] H. Huang, W. Zhang, J. Ding, D. M. Stipanović, and C. J. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 4835–4840.

[20] S. Jia, X. Wang, and L. Shen, "A continuous-time markov decision process-based method with application in a pursuit-evasion example," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1215–1225, 2016.

[21] V. G. Lopez, F. L. Lewis, Y. Wan, E. N. Sanchez, and L. Fan, "Solutions for multiagent pursuit-evasion games on communication graphs: Finite-time capture and asymptotic behaviors," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 1911–1923, 2020.

[22] Q. Qi, X. Zhang, and X. Guo, "A deep reinforcement learning approach for the pursuit evasion game in the presence of obstacles," in *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2020, pp. 68–73.

[23] M. Z. Qadir, S. Piao, H. Jiang, and M. E. H. Souidi, "A novel approach for multi-agent cooperative pursuit to capture grouped evaders," *The Journal of Supercomputing*, vol. 76, no. 5, pp. 3416–3426, 2020.

[24] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of drones: Multi-uav pursuit-evasion game with online motion planning by deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7900–7909, 2023.

[25] Y. Yang, X. Li, Z. Yuan, Q. Wang, C. Xu, and L. Zhang, "Graded-q reinforcement learning with information-enhanced state encoder for hierarchical collaborative multi-vehicle pursuit," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, 2022, pp. 534–541.

[26] Z. Yuan, T. Wu, Q. Wang, Y. Yang, L. Li, and L. Zhang, "T3omvp: A transformer-based time and team reinforcement learning scheme for observation-constrained multi-vehicle pursuit in urban area," *Electronics*, vol. 11, no. 9, p. 1339, 2022.

[27] Q. Wang, X. Li, Z. Yuan, Y. Yang, C. Xu, and L. Zhang, "An opponent-aware reinforcement learning method for team-to-team multi-vehicle pursuit via maximizing mutual information indicator," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*, 2022, pp. 526–533.

[28] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6382–6393, 2017.

[29] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "Qplex: Duplex dueling multi-agent q-learning," in *International Conference on Learning Representations (ICLR)*, 2021.

[30] Y.-G. Hsieh, F. Iutzeler, J. Malick, and P. Mertikopoulos, "Multi-agent online optimization with delays: Asynchronicity, adaptivity, and

optimism," *Journal of Machine Learning Research*, vol. 23, no. 78, pp. 3377–3425, 2022.
[31] M. Kloock and B. Alrifaee, "Coordinated cooperative distributed decision-making using synchronization of local plans," *IEEE Transactions on Intelligent Vehicles*, pp. 1–15, 2023.
[32] K. Okumura, M. Machida, X. Défago, and Y. Tamura, "Priority inheritance with backtracking for iterative multi-agent path finding," *Artificial Intelligence*, vol. 310, p. 103752, 2022.
[33] S. Iqbal, C. A. S. De Witt, B. Peng, W. Böhmer, S. Whiteson, and F. Sha, "Randomized entity-wise factorization for multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4596–4606.
[34] Y. Liu, Z. Li, Z. Jiang, and Y. He, "Prospects for multi-agent collaboration and gaming: challenge, technology, and application," *Frontiers of Information Technology & Electronic Engineering*, vol. 23, no. 7, pp. 1002–1009, 2022.
[35] T. Wang, H. Dong, V. Lesser, and C. Zhang, "Roma: Multi-agent reinforcement learning with emergent roles," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 9876–9886.
[36] K. Kurach, A. Raichuk, P. Stanczyk, and M. Zajac, "Google research football: A novel reinforcement learning environment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4501–4510.
[37] H. Mao, W. Liu, J. Hao, J. Luo, D. Li, Z. Zhang, J. Wang, and Z. Xiao, "Neighborhood cognition consistent multi-agent reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7219–7226.
[38] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations (ICLR)*, 2016.
[39] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver, "Distributed prioritized experience replay," in *International Conference on Learning Representations (ICLR)*, 2018.
[40] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2021.
[41] R. Yang, D. Wang, and J. Qiao, "Policy gradient adaptive critic design with dynamic prioritized experience replay for wastewater treatment process control," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3150–3158, 2022.
[42] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 2575–2582.
[43] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 7234–7284, 2020.
[44] C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang, "Celebrating diversity in shared multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3991–4002, 2021.

**Xinhang Li** received the B.E. degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021. He is currently pursuing the Ph.D. degree in information and communication engineering from the School of Artificial Intelligence, BUPT. His research interests include deep reinforcement learning, intelligent information processing and intelligent transportation systems.

**Yiying Yang** received the B.E. degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021. She is currently pursuing an M.S. degree from the School of Artificial Intelligence, BUPT. Her research interests include cooperative connected vehicles decision-making and reinforcement learning.

**Zheng Yuan** received the B.E. degree in information engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2021. He is currently pursuing an M.S. degree from the School of Artificial Intelligence, BUPT. His research interests are reinforcement learning, autonomous driving and intelligent information processing.

**Zhe Wang** received the B.E. degree in Information Engineering from Xi'an University of Posts and Telecommunications, Xi'an, China, and the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2019 and 2022, respectively. He is currently working towards his Ph.D. in the Centre for Telecommunications at King's College London, London, UK. His research interests include cooperative intelligent transportation systems and the Metaverse.

**Qinwen Wang** received the B.E. degree in digital media technology from Communication University of China (CUC), Beijing, China, in 2020. She is currently pursuing an M.S. degree from the School of Artificial Intelligence, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. Her research interests include reinforcement learning and cooperative intelligent transportation systems.

**Chen Xu** (S'12-M'15) received the B.S. degree from Beijing University of Posts and Telecommunications in 2010, and the Ph.D. degree from Peking University, Beijing, in 2015. She is now an associate professor and Ph.D. supervisor in School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China. Her research interests mainly include wireless resource management, cooperative communication and computing, and intelligent network optimization. She served as a TPC member for IEEE Globecom 2016, IEEE ICC 2016, etc. She received the best paper award at the 2012 International Conference on Wireless Communications and Signal Processing (WCSP), IEEE Leonard G. Abraham Prize in 2016, WCSP 10-year Anniversary Excellent Paper Award in 2019, and the first prize of Natural Science of Chinas Ministry of Education in 2017. She is one of 2019 Beijing Nova of Science and Technology.

**Lei Li** is currently a Professor with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China. Her research interests include intelligent information processing, deep learning, machine learning, and natural language processing.

**Jianhua He** is a Professor at University of Essex, UK. He received a PhD degree from Nanyang Technological University, Singapore, in 2002. His research interests include mobile networking and computing, 5G/6G networks, Internet of Things, edge computing and intelligence, connected vehicles, autonomous driving, autonomous ship, machine learning, large language models and intelligent document understanding. Dr He has published over 150 research papers. Dr He is a member of the editorial board for international journals including IEEE Wireless Communication Letters and Computer Journal. He is the coordinator of EU Horizon 2020 projects COSAFE and VESAFE, and Horizon EU projects SECOM and COVER on cooperative connected autonomous vehicles. He was the workshop chair of MobiArch'20 and ICAV 2021, and a steering committee member of MobiArch.

**Lin Zhang** (Member, IEEE) received the B.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1996 and 2001, respectively. He is currently the Director of Beijing Bigdata Center and also a Professor of BUPT. He was a Postdoctoral Researcher with Information and Communications University, South Korea. He used to hold a Research Fellow position with Nanyang Technological University, Singapore. In 2004, he joined BUPT as a Lecturer, then an Associate Professor in 2005, and a Professor in 2011. He has authored more than 120 papers in referenced journals and international conferences. His research interests include intelligent information processing, deep learning, mobile cloud computing and Internet of Things.