Li *et al. Intell Robot* 2023;3(3):436-52
DOI: 10.20517/ir.2023.25

Intelligence & Robotics

Research Article

Open Access

A distributed multi-vehicle pursuit scheme: generative multi-adversarial reinforcement learning

Xinhang Li[1], Yiying Yang[1], Qinwen Wang[1], Zheng Yuan[1], Chen Xu[1], Lei Li[1], Lin Zhang[1,2]

[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China.
[2]Beijing Big Data Center, Beijing 100101, China.

Correspondence to: Prof. Lin Zhang, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian District, Beijing 100876, China. E-mail: zhanglin@bupt.edu.cn

How to cite this article: Li X, Yang Y, Wang Q, Yuan Z, Xu C, Li L, Zhang L. A distributed multi-vehicle pursuit scheme: generative multi-adversarial reinforcement learning. *Intell Robot* 2023;3(3):436-52. http://dx.doi.org/10.20517/ir.2023.25

Received: 30 Jun 2023 **First Decision:** 15 Aug 2023 **Revised:** 28 Aug 2023 **Accepted:** 5 Sep 2023 **Published:** 13 Sep 2023

Academic Editor: Simon X. Yang **Copy Editor:** Yanbin Bai **Production Editor:** Yanbin Bai

Abstract

Multi-vehicle pursuit (MVP) is one of the most challenging problems for intelligent traffic management systems due to multi-source heterogeneous data and its mission nature. While many reinforcement learning (RL) algorithms have shown promising abilities for MVP in structured grid-pattern roads, their lack of dynamic and effective traffic awareness limits pursuing efficiency. The sparse reward of pursuing tasks still hinders the optimization of these RL algorithms. Therefore, this paper proposes a distributed generative multi-adversarial RL for MVP (DGMARL-MVP) in urban traffic scenes. In DGMARL-MVP, a generative multi-adversarial network is designed to improve the Bellman equation by generating the potential dense reward, thereby properly guiding strategy optimization of distributed multi-agent RL. Moreover, a graph neural network-based intersecting cognition is proposed to extract integrated features of traffic situations and relationships among agents from multi-source heterogeneous data. These integrated and comprehensive traffic features are used to assist RL decision-making and improve pursuing efficiency. Extensive experimental results show that the DGMARL-MVP can reduce the pursuit time by 5.47% compared with proximal policy optimization and improve the pursuing average success rate up to 85.67%. Codes are open-sourced in Github.

Keywords: Generative multi-adversarial reinforcement learning, graph neural network, intersecting cognition, multi-vehicle pursuit

1. INTRODUCTION

Enabled by novel sensing technology[1] and the self-learning ability of reinforcement learning (RL)[2], the intelligent traffic management system is enjoying a significant upgrade and showing great potential to solve various problems in intelligent transportation systems (ITS)[3]. As a complex special scene, multi-vehicle pursuit (MVP) describes the problem of multiple vehicles capturing several moving targets[4], represented by the New York City Police Department guideline on the pursuit of suspicious vehicles[5]. Moreover, various military intelligence combat scenes can also be modeled as MVP[6]. Effective reward guidance[7] and comprehensive perception[8] of complex and dynamic urban traffic environments are the keys to solving the MVP problem and are gradually becoming hot topics.

Aiming at the MVP problem, Garcia *et al.* extended classical differential game theory and devised saddle-point strategies[9] to address multi-player pursuit-evasion problems. Xu *et al.* considered greedy, lazy, and traitorous pursuers during the pursuit and rigorously re-analyzed Nash equilibrium[10]. A graph-theoretic approach[11] was employed to study the interactions of the agents and obtain distributed control policies for pursuers. A region-based relay pursuit scheme[12] was designed for the pursuers to capture one evader. Jia *et al.* proposed a policy iteration method-based continuous-time Markov decision process (MDP)[13] to optimize the pursuer strategy. However, these classical methods for MVP are not competent for complex traffic scenes with more constraints due to poor robustness. De Souza *et al.* introduced distributed multi-agent RL and curriculum learning to MVP problems[14]. To improve pursuing efficiency, Zhang *et al.* constructed a multi-agent coronal bidirectionally coordinated with a target prediction network[15] based on the multi-agent deep deterministic policy gradient. For efficient cooperation among pursuers, Yang *et al.* designed a hierarchical collaborative framework[16]. Zheng *et al.* extended multi-to-multi competition to air combat among unmanned aerial vehicles[17]. However, due to the mission nature of MVP, the pursuers only obtain a sparse reward after successfully capturing an evader. None of the aforementioned RL-based methods have addressed the sparse reward problem. This issue blurs the direction of the gradient descent of neural networks and seriously affects the strategy optimization. In addition, the lack of dynamic and effective awareness in the above MVP methods limits pursuing efficiency.

Due to powerful capabilities of distribution feature extraction and data generation, generative adversarial networks (GANs) have drawn growing interest in recent years[18] and have been combined with RL to optimize strategies. To address the problem of incomplete observation of traffic information, Wang *et al.* used GANs for traffic data recovery to assist in deep RL (DRL) decision-making[19]. A GAN-assisted human preference-based RL approach[20] was proposed that adopted a GAN to learn human preferences. Li *et al.* designed a conditional deep generative model to predict future trajectory distribution[21]. The adversarial training of GANs was introduced into the policy network and critic network[22] to optimize RL training. Zheng *et al.* developed a reward-reinforced GAN[23] to represent the distribution of the value function. However, mission-critical requirements of MVP pose significant challenges to these methods. The problem of the sparse reward remains unsolved, hindering the RL optimization.

Graph neural networks (GNNs) have an excellent ability to handle unstructured data and are widely applied to modeling multi-agent interactions and feature extraction of traffic information. Liu *et al.* modeled the relationship between agents by a complete graph[24] to indicate the importance of the interaction between two agents. For cooperation among heterogeneous agents, Du *et al.* proposed a heterogeneous graph attention network[25] to model the relationships among these diverse agents. GNNs were employed to model vehicle relationships and extract traffic features to enhance autonomous driving[26,27]. A GNN with spatial-temporal clustering[28] was designed for traffic flow forecasting. However, the single-layer GNN structure in the above methods did not couple the interaction model and traffic information of agents, which affects the RL collaborative game decision-making in complex urban traffic scenes.

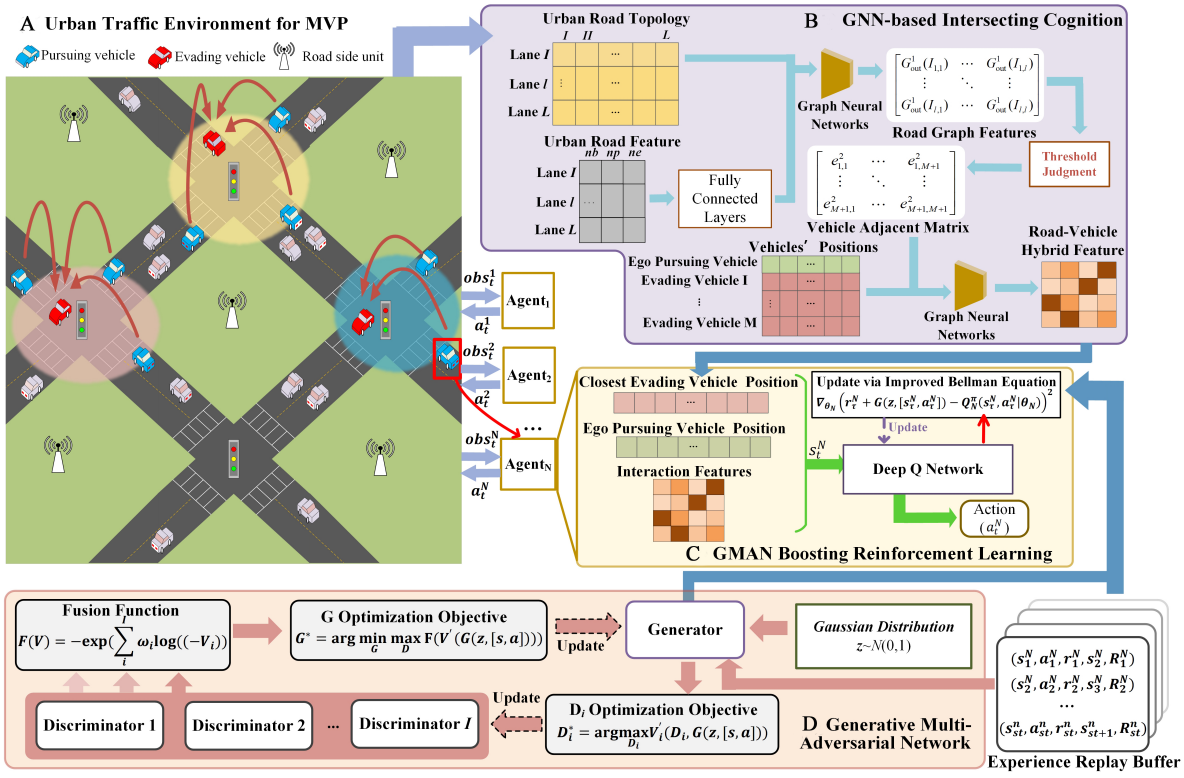

Figure 1. Architecture of DGMARL-MVP. Urban traffic environments for MVP (**A**) provide complex pursuit-evasion scenes and interactive environments for RL. Every pursuing vehicle targets the nearest evading vehicle and launches a collaborative pursuit. GNN-based intersecting cognition (**B**) couples the traffic information and multi-agent interaction features to assist GMAN boosting reinforcement learning (**C**) in decision-making. GMANs (**D**) to guide RL strategy optimization via generating dense rewards, replacing the approximation of Bellman updates. MVP: Multi-vehicle pursuit; GNNs: graph neural networks; GMAN: generative multi-adversarial network.

In summary, as for the existing approaches for MVP, sparse reward and the lack of comprehensive traffic cognition severely limit the collaboratively pursuing efficiency. To address these problems, this paper proposes distributed generative multi-adversarial RL for MVP (DGMARL-MVP) in urban traffic scenes, as shown in Figure 1. Firstly, a generative multi-adversarial network (GMAN) is designed to guide RL strategy optimization via generating dense rewards, replacing the approximation of Bellman updates. The generative multi-adversarial RL can be applied to a wide range of multi-agent systems with sparse rewards to improve task-related performance. Moreover, a proposed GNN-based intersecting cognition promotes deep coupling of traffic information and multi-agent interaction features. The contributions of this paper are summarized as follows.

- This paper proposes DGMARL-MVP in urban traffic scenes. In DGMARL-MVP, a GMAN is designed to improve the Bellman equation by generating the potential dense reward, thereby properly guiding strategy optimization of distributed multi-agent RL (MARL).
- GNN-based intersecting cognition is proposed to promote deep coupling of traffic information and multi-agent interaction features to assist in improving the pursuing efficiency.
- This paper applies DGMARL-MVP to the simulated urban roads with 16 junctions and sets different pursuing difficulty levels with variable numbers of pursuing vehicles and evading vehicles. In the three tested difficulty levels, DGMARL-MVP reduces the pursuit time by 5.47% on average compared to proximal policy optimization (PPO) and improves the pursuing average success rate to 85.67%. Codes are open-sourced at *https://github.com/BUPT-ANTlab/DGMARL-MVP*.

The rest of this paper is organized as follows. Section II describes MVP in an urban traffic scene and models

the MVP problem based on the MDP. Section III presents generative multi-adversarial RL (GMARL) and its training process. Section IV presents distributed GMARL with GNN-based intersecting cognition for MVP. Section V gives the performance of the proposed method. Section VI draws conclusions.

2. MULTI-VEHICLE PURSUIT IN DYNAMIC URBAN TRAFFIC

This section first introduces the details of the complex urban traffic environment for the MVP problem. Then, the modeling process of the MVP problem is stated as an MDP, and the basic Q-learning algorithm focusing on the update process is introduced.

2.1. Complex urban traffic environment for MVP

This paper focuses on the problem of MVP under the complex urban traffic and constructs a multi-intersection traffic scene. Each road is set to bidirectional two lanes and fixed-phase traffic lights at each intersection. In this scene, there are M pursuing vehicles, N evading vehicles ($M > N$), B background vehicles, and L lanes. Specifically, the background vehicles follow the randomly selected routes, and the evading vehicles randomly select the routes from the preset routes. The pursuing vehicles share the companies and target information via road side units. Each pursuing vehicle integrates multi-source heterogeneous data to make decisions distributedly. If the evading vehicles are not totally captured within st time steps, the MVP task fails. When all evading vehicles are captured or the time steps reach st, the pursuit is *Done*.

Furthermore, the following constraints are set in the MVP environment: (1) All vehicles obey the traffic rules for collision-free driving; (2) 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 consistent; (3) Pursuing vehicles and evading vehicles are randomly initialized at the edge of the traffic map, respectively.

2.2. MDP-based MVP problem formulation

In this paper, the decision-making of each pursuing vehicle only depends on the current state, so the decision process can be modeled as the MDP defined by a tuple $\{S, A, P, R\}$. $s_t \in S$, $a_t \in A$ denote the state and action at time step t. P is the state transition probability from the current state s_t to the next state s_{t+1} by executing the action a_t. $r_t \in R : a_t \times s_t \to \mathbb{R}$ is a real valued reward.

RL provides an excellent solution to MDP games. As an advanced RL algorithm for the problem with discrete action space, Q-learning enables decision-making without exact state transition probability and initial state. For a Q-learning-based agent, the expectation values q of all actions in state s_t are evaluated by $Q^{\pi}(s_t, a)$ function, and the optimal strategy π chooses the action s_t with the greatest expectation to execute. Through the continuous interaction between the agent and the environment, the $Q^{\pi}(s, a)$ function is updated, achieving the purpose of selecting the optimal strategy finally. According to the Bellman equation, the optimal state-action value function can then be derived as

$$Q^{\pi^*}(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim P(.|s_t, a_t)}(r_t + \gamma \max_{a_{t+1}} Q^{\pi}(s_{t+1}, a_{t+1})). \tag{1}$$

And the updating process of Q-learning can be expressed as

$$\begin{cases} Q_{target}(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q^{\pi}(s_{t+1}, a_{t+1}), \\ Q^{\pi}(s_t, a_t) \leftarrow Q^{\pi}(s_t, a_t) + \alpha[\, Q_{target}(s_t, a_t) - Q^{\pi}(s_t, a_t)], \end{cases} \tag{2}$$

where α is the learning rate and γ is the discount factor, indicating the impact of future earnings on the current expectation value. For pursuing vehicle m, the function $Q^{\pi}(s_t, a)$ calculates the expectation values of turning

left, turning right turn and going straight according to the current state s_t to assist the vehicle in selecting the optimal route to pursue the evader.

3. GENERATIVE MULTI-ADVERSARIAL REINFORCEMENT LEARNING

In order to effectively solve the reward sparsity problem of MVP, a GMAN is introduced to improve the Bellman equation by generating suitable potential dense rewards during optimizing RL. Section 3.1 describes the principles and details of how a GMAN generates estimated rewards. Section 3.2 presents GMAN boosting RL and its training process.

3.1. Generative multi-adversarial network for dense reward

As a special game task, the reward of MVP is extremely sparse. Only when the pursuing vehicle captures an evading vehicle can the RL-based agent obtain a reward. The sparse reward blurs the optimization direction of RL, thus seriously hindering the strategy update. In this paper, a conditional generative network G is designed to estimate the potential future rewards and guide the RL training. Therefore, this paper adopts a GMAN to train the generative network G and generate appropriate dense rewards.

Suppose the state of the agent n at time step t is s_t^n, the action is a_t, and the cumulative rewards obtained by n from t until the end of the episode are

$$R_t = \sum_{T=t+1} (\gamma)^{(T-t)} r_T \,. \tag{3}$$

The optimization objective of the generative network is to learn the contribution of the cumulative rewards p_R and make its output $G(z, [s_t, a_t])$ fit R_t, where $z \sim p_z$ is a simple fixed distribution that is easy to draw samples from.

In GMAN, the generating network G performs a max-min game with I discriminators to update parameters. For discriminator D_i, the optimization objective is to distinguish data generated by G from the original data,

$$\arg\max_{D_i} V_i'(D_i, G) = \mathbb{E}_{R\sim p_R}[\log(D_i(R))] + \mathbb{E}_{z\sim p_z}[\log(1 - D_i(G(z, [s, a])))]. \tag{4}$$

In practice, training against a far superior discriminator can impede the learning of the generator. To solve this problem and increase the stability of the generator, a classical Pythagorean mean is chosen as the fusion function F_{soft} to soften $V_i'(D_i, G)$, which is parameterized by λ where $\lambda = 0$ corresponds to the mean and the max is recovered as $\lambda \to \infty$.

$$V = F_{soft}(V') = -\exp(\sum_{i}^{I} \frac{e^{\lambda V_i'}}{\sum_j e^{\lambda V_j'}} \log(-V_i')). \tag{5}$$

Then, the discriminator D_i and generator G are updated by descending their stochastic gradient,

$$\nabla_{\theta_{D_i}} \frac{1}{K} \sum_{k}^{K} [\log(D_i(R)) + \log(1 - D_i(G(z, [s, a])))], \tag{6}$$

$$\nabla_{\theta_G} \frac{1}{K} \sum_{k}^{K} \log(1 - F_{soft}(D_i(G(z, [s, a])))). \tag{7}$$

Therefore, by training with historical experience replay, a GMAN is able to generate potential future rewards $G(z, [s_t^n, a_t^n])$ according to the current state and action of the agent. $G(z, [s_t^n, a_t^n])$ is used to boost the training

process of RL and make its policy forward-looking. Meanwhile, the generated dense reward also effectively promotes RL convergence.

3.2. GMAN boosting reinforcement learning

The sparsity of rewards is a great challenge for the optimization of RL. In MVP, the RL-based agent explores many steps to obtain only one positive or negative reward, which leads to a vague direction of gradient descent for the agent. Therefore, this paper proposes a novel GMAN boosting RL. GMAN boosting RL generates reasonably dense rewards via virtue of the powerful generative power of the generative network. And the generated reward also includes the potential future benefit of the RL decision to improve the learning efficiency and the decision foresight of RL.

In GMAN boosting RL, the Bellman equation is modified using the generated reward. The approximation of future reward is replaced by $G(z, [s_t, a_t])$. The equation for target Q after upgrading the Bellman equation is

$$Q_{target}(s_t, a_t) = r_t + G(z, [s_t, a_t]). \tag{8}$$

In this paper, the deep neural network Q is employed to fit the values of actions. Therefore, the loss of the Q network is calculated as

$$loss = \frac{1}{K} \sum_{k} (Q(s_t, a_t) - Q_{target})^2. \tag{9}$$

Distributed on-policy training is adopted in the proposed GMAN Boosting RL. The overall training process is shown in Algorithm 1. For every RL-based agent, the experience collected by the current policy is stored in the replay buffer $\mathcal{G}$. At the start of training, experience $(s_t, a_t, r_t, s_{t+1}, R_t)$ is sampled from $\mathcal{G}$. With the assistance of the generative network, the parameters of RL are updated via Eq. (9). Finally, the discriminators and the generator are trained in turn to help the generator learn the distribution of cumulative rewards under different state-action pairs. Through multiple cycles of training, GMAN boosting RL can have a forward-looking and optimal strategy to handle complex MVP problems with sparse reward.

Algorithm 1: Training Process of GMAN Boosting Reinforcement Learning

Input: RL-based agent Q, generator G, I discriminators, and the experience replay buffer $\mathcal{G}$ collected by Q

```
Sample (s_t, a_t, r_t, s_{t+1}, R_t) from G;
Generate potential dense reward G(z, [s_t, a_t]) by G;
Calculate the loss of Q via Eq. (9) and update Q network;
for i=1:I do
    Update discriminator D_i via Eq. (6);
end
Update generator G via Eq. (7);
Clear Rb;
```

4. DISTRIBUTED GMARL WITH GNN-BASED COGNITION FOR MVP

To enhance comprehensive cognition of complex urban traffic in MVP, a novel double-layer intersecting GNN is proposed to couple the traffic information and multi-agent interaction features. Section 4.1 mainly introduces the details of GNN-based intersecting cognition. DGMARL-MVP is described in Section 4.2. Finally, the decision-making and training flowchart of the proposed DGMARL-MVP is demonstrated in Section 4.3.

4.1. GNN-based intersecting cognition

In this paper, a double-layer intersecting graph network is used with a *road graph* to perceive the traffic condition and a *vehicle graph* to extract efficient information for pursuing vehicles, as shown in Figure 2. And

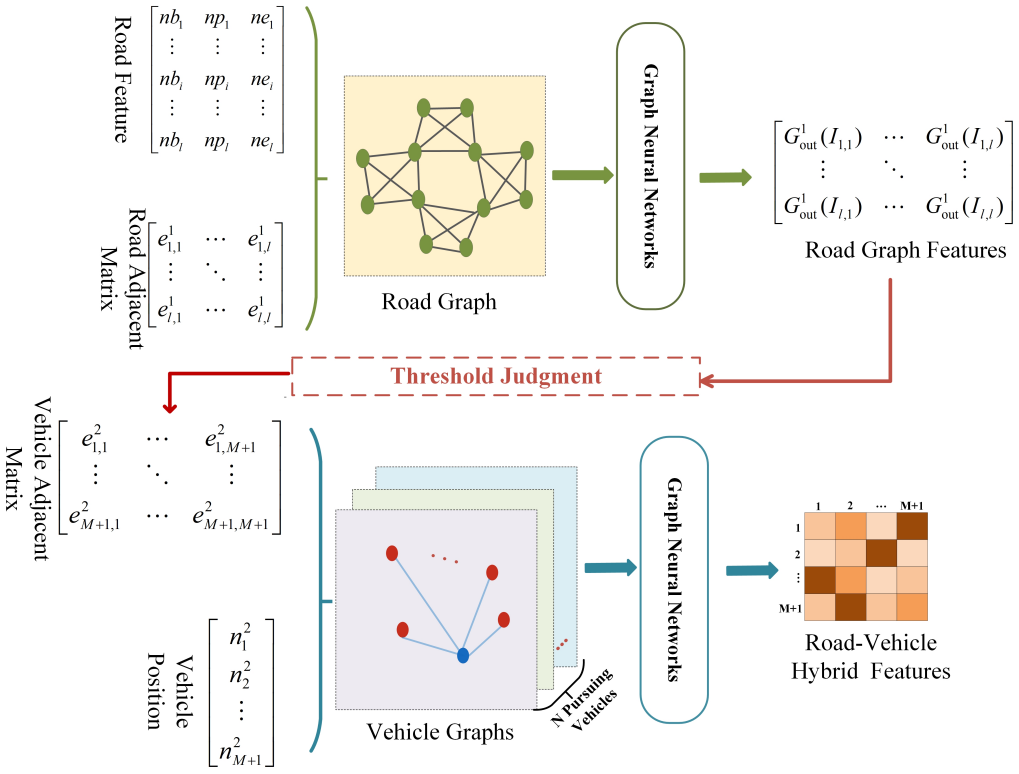

Figure 2. Architecture of GNN-based Intersecting Cognition. GNNs: graph neural networks.

the main idea of intersecting lies in using the perceived traffic information to construct the *vehicle graph*. It enables a deep coupling of road information with vehicle information.

Each lane is modeled as a node on the first road graph, and the topological relationship of the road is regarded as the edge of the graph. More formally, the constructed *road graph* is described as $G^1 = \{N^1, A^1\}$, where $N^1 = \{n_i^1, i \in \{1, 2, \ldots, l\}\}$ is a set of node attributes and $A^1 = \{e_{ij}^1, i, j \in \{1, 2, \ldots, l\}\}$ is a set of edge attributes. Specifically, $n_i^1 = [nb_i, np_i, ne_i]$ represents the node feature consisting of the number of three types of vehicles (including background, pursuing and evading vehicles) in lane i, respectively. l denotes the number of nodes in the constructed graph that is equal to the total number of lanes. e_{ij}^1 denotes the edge value of lane i and lane j. $e_{ij}^1 = 1$when lane i and lane j are connected, while $e_{ij}^1 = 0$ lane i and lane j are not adjacent.

The first *road graph* network consists of fully connected layers (FC) Φ_1^{FC} and graph convolutional network (GCN) Φ_1^G. The node features are firstly input into the FC to assist GCN in understanding their semantic information, represented as $N_{FC}^1 = \Phi_1^{FC}(N^1)$. Then, GCN merges the global traffic node information and produces high-level semantic information. The process can be formulated as follows

$$G\_out_1 = \Phi_1^G \cdot \left(N_{FC}^1,\ A^1 \right) = D_1^{\frac{1}{2}} A^1 D_1^{-\frac{1}{2}} N_{FC}^1 W + b, \tag{10}$$

where D_1 is the degree matrix and $D_{ii}^1 = \sum_j A_{ij}^1$, and W is the trainable weight matrix G_out_1 is set to the shape of $(l \times l)$, representing the relationship between roads based on traffic density.

The *vehicle graph* $G^2 = \{N^2, A^2\}$ takes the ego pursuing vehicle and all evading vehicles as nodes $N^2 = \{n_i^2, i \in \{1, 2, \ldots, M+1\}\}$, where M is the number of evading vehicles, n_i^2 is the position embedding of pursuing vehi-

cles or evading vehicles. Obviously, each pursuing vehicle obsesses an independent sub-graph. The adjacency matrix $A^2 = \{e_{ij}^2, i, j \in \{1, 2, \ldots, M+1\}\}$ is deliberately designed and significantly meaningful in this paper. Given that G^1 contains the relationship of each lane based on traffic density in our conception, a threshold ε is set to identify traffic-sensing connections between ego vehicles and each evading vehicle:

$$e_{ij}^2 = \begin{cases} 1 & G\_out_1\left(I_{i,j}\right) > \varepsilon, \\ 0 & G\_out_1\left(I_{i,j}\right) < \varepsilon, \end{cases} \tag{11}$$

where I_1 represents the number of the located lane of the ego pursuing vehicle, and I_j represents that of the evading vehicle j. The calculating process of the *vehicle graph* is the same as the *traffic graph* network. The output of vehicle graph G^2 is $G\_out_2 = \Phi_2^G\left(N_{FC}^2, A^2\right)$, $N_{FC}^2 = \Phi_2^{FC}(N^2)$ is the output of FC Φ_2^{FC}.

4.2. Distributed GMARL for MVP

In this paper, a deep neural network is adopted to fit $Q^\pi(s_t, a_t)$. For the pursuing vehicle n, the GMARL-based agent is used to plan the optimal pursuit route. The agent contains a neural network with a parameter θ_n, an *online* network $Q_n^\pi(s_t^n, a_t^n|\theta_n)$, to estimate the Q value of the action a_t^n in the current state s_t^n. The architecture of DGMARL-MVP is shown in Figure 3.

Each pursuing vehicle distributedly makes decisions based on its own observations and shared information. For pursuing vehicle n, its state s_t^n consists of three parts, including its own position loc_t^n, the position of the closest evading vehicles loc_t^m, and road-vehicle hybrid features G_out_2. In this paper, the location of vehicle n is denoted by $loc_t^n = \left\{Code_l, pos_t^{n,l}\right\}$. And the length of loc_t^n is denoted by len_{loc}. $Code_l$ denotes the binary code of lane l where vehicle n is located, and $pos_t^{n,l}$ denotes the distance between vehicle n and the starting of lane l at time t.

Due to the constraints of the traffic scene, the action space of the pursuing vehicles contains three elements, i.e., turning left, turning right, and going straight at the next intersection. And the expectation values of turning left $q_t^{n,lef}$, turning right $q_t^{n,rig}$, and going straight $q_t^{n,str}$ are calculated according to the pursuing vehicle n's current state s_t^n. The agent randomly chooses an action with probability ϵ to explore or exploit with probability $1-\epsilon$ by selecting the action with the largest q-value. In particular, if a vehicle encounters a T-junction that only enables going straight and turning right and, unfortunately chooses to turn left, the agent will randomly select the action to execute in the reasonable action space.

To motivate the capture of pursuing vehicles and incentivize efficient training, an elaborately designed reward r_t^n consists of two parts.

1. Only the pursuing vehicle n successfully captures an evading vehicle does it obtain a positive reward R_V.

2. A distance-sensitive reward is set to improve the pursuing efficiency. When a pursuing vehicle reduces the distance from the closest evading vehicle 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

$$r_t^n = \begin{cases} R_V & \text{if successful pursuit,} \\ \sigma\left(d_t^{n,m} - d_{t-1}^{n,m}\right) & \text{else,} \end{cases}, \tag{12}$$

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

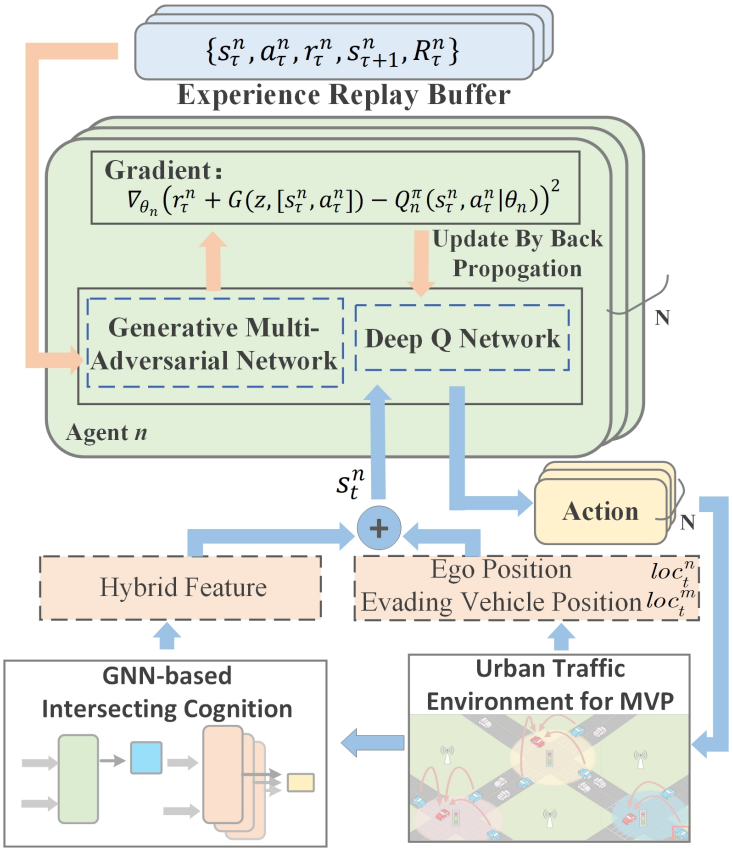

Figure 3. Architecture of Distributed GMARL for MVP. MVP: Multi-vehicle pursuit.

Each agent is updated by gradient descent in distributed training. Specifically, θ_n is iteratively updated through stochastic gradient descent (SGD) using random samples from the experience replay buffer. Suppose the sample data is denoted as $\{s_\tau^n, a_\tau^n, r_\tau^n, s_{\tau+1}^n\}$. The gradient of *online* network $Q_n^\pi(s, a\,|\theta_n)$ is delivered as

$$\nabla_{\theta_n} J = \frac{1}{K}\sum_\tau \nabla_{\theta_n}\left(r_\tau^n + G(z,[s_\tau^n, a_\tau^n]) - Q_n^\pi\left(s_\tau^n, a_\tau^n\,|\theta_n\right)\right)^2 \tag{13}$$

4.3. Decision-making and training process of DGMARL-MVP

This part presents the overall decision-making and online training process of DGMARL-MVP, as shown in Algorithm 2. At the beginning of each episode, the urban pursuit-evasion environment and the local state of all agents are initialized. Then, the road information and the position information of vehicles are fed into the intersecting graph network outputting G_out_2, which extracts integrated features of the traffic situation and agent relationships from multi-source heterogeneous data. Equipped with G_out_2, n agents form their own local state using the other position information, individual position information loc_t^n, and position information of evading vehicles loc_t^m. And each pursuing vehicle n selects an action a_t^n according to the current policy Q_n^π with the local state. The local state will be updated and the reward will be feedback after each agent performs a_t^n. Each transition $(s_t^n, a_t^n, r_t^n, s_{t+1}^n)$ is then stored in the separate replay buffer $\mathcal{G}_n$.The potential future rewards R_t^n are generated by a conditional generative network. R_t^n is appended to the experience replay buffer to estimate and guide the RL training. Then, the training processes of the distributed networks are according to Algorithm 1.

Algorithm 2: DGMARL-MVP Decision-making and Online Training Algorithm

Initialize N DGMARL-based agents with GNN-based cognition, generator G, and discriminators $\{D_i, i \in \{1, 2, \ldots, I\}\}$;
Initialize experience replay buffers $\{\mathcal{G}_n, n \in \{1, 2, \ldots, N\}\}$;
for *e=1:max_epoch* **do**
  Initialize an urban pursuit-evasion environment and obtain initialized state $\{s_t^n\}$;
  for *n=1:N* **do**
    Get G_out_2 by GNN-based intersecting cognition;
    Choose the nearest evading vehicle m as pursuing vehicle n's target;
    $s_t^n \leftarrow [G\_out_2, loc_t^n, loc_t^m]$;
  end
  for *t=1:st* **do**
    for *n=1:N* **do**
      Obtain a_t^n by $Q_n^\pi(s_t^n|\theta_n)$;
    end
    Perform the strategy $\{a_t^n, n \in \{1, 2, \ldots, N\}\}$ and observe $\{s_{t+1}^n, n \in \{1, 2, \ldots, N\}\}$;
    Get G_out_2 by GNN-based intersecting cognition;
    $s_{t+1}^n \leftarrow [G\_out_2, loc_{t+1}^n, loc_{t+1}^{m'}]$;
    Append $(s_t^n, a_t^n, r_t^n, s_{t+1}^n)$ to $\mathcal{G}_n$;
    $\{s_t^n\} \leftarrow \{s_{t+1}^n\}$;
    if *Done* **then**
      break;
    end
  end
  for *n=1:N* **do**
    Calculate cumulative future rewards R_t^n in every time step;
    Append R_t^n to the experience replay buffer;
    Train the agent n via Algorithm 1;
  end
end

In the decision-making process of DGMARL-MVP, collaboration among agents is performed in the information sharing. During the pursuit process, every pursuing vehicle shares its own position and observation information with other pursuing vehicles for collaboration. The shared information is used to develop GNN-based intersecting cognition and gain effective and comprehensive awareness of the agent relationships and traffic situations.

5. EXPERIMENTS AND RESULTS

5.1. Simulator and parameter settings

As a MARL algorithm, DGMARL-MVP collects training data and updates parameters by interacting with the simulated urban traffic environment. This paper constructs a complex urban traffic environment based on SUMO [29] to verify the effect of the DGMARL-MVP. The environment with 3 × 3 grid-pattern urban road structure simulates continuous dynamic random traffic flow. In the simulated closed scene, there are four intersections and eight T-junctions. During the simulation process, the number of background vehicles is fixed, and the background vehicles follow randomly selected routes. Moreover, to evaluate the robustness of DGMARL-MVP, this paper designs three different difficulty levels of MVP tasks with variable numbers of pursuing vehicles N and evading vehicles M, respectively, four pursuing vehicles chasing two evading vehicles (P4-E2), five pursuing vehicles chasing three evading vehicles (P5-E3) and seven pursuing vehicles chasing

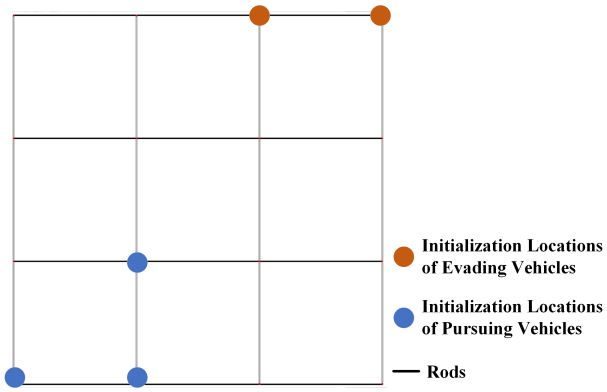

Figure 4. Traffic Simulation Environment for MVP. MVP: Multi-vehicle pursuit.

Table 1. Simulation settings

Parameters	Values
Maximum time steps st	800
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
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	200

Table 2. Parameter settings

Parameters	Values	Parameters	Values
α	10^{-4}	R_V	500
γ	0.9	ε	0
λ	0.5	σ	5
ϵ	0.05	max_epoch	2600

Table 3. Structure of the deep Q network

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

four evading vehicles (P7-E4). All evading and pursuing vehicles randomly select their initialization locations, as shown in Figure 4. The GPU used to train our model is NVIDIA Tesla T4. Notably, all algorithms in our experiments are trained in the same environment and evaluated by averaging various metrics for 100 test epochs, including the average reward, the average time steps, and the success rate. The simulation parameters are shown in Table 1. And the parameter settings of DGMARL-MVP are shown in Table 2. The internal structures of Deep Q Network (DQN) and discriminators are shown in Table 3 and Table 4, respectively.

Table 4. Structure of the discriminator

Layers	Discriminator
Input	(batch size, 1)
Dense Layer 1	(1,128)
Activation Function	$LeakyReLu(negative\_slope = 0.02)$
Dense Layer 2	(128,64)
Activation Function	$LeakyReLu(negative\_slope = 0.02)$
Dense Layer 3	(64,1)
Output	(batch size, 1)

Table 5. Evaluation results

N-M	P4-E2			P5-E3			P7-E4		
Evaluate metrics	Average Reward	Average Time Steps	Success Rate	Average Reward	Average Time Steps	Success Rate	Average Reward	Average Time Steps	Success Rate
DGMARL-MVP	8.688	644.96	0.91	8.827	698.20	0.85	9.213	731.64	0.81
a	7.407	695.33	0.86	8.592	728.53	0.81	8.791	751.32	0.78
b	8.094	684.01	0.87	8.692	714.36	0.82	8.959	739.55	0.80
DQN	6.953	736.09	0.81	7.195	763.31	0.72	8.122	749.76	0.76
PPO	7.513	717.41	0.86	8.368	731.88	0.80	8.844	745.51	0.78
QMIX	6.339	745.27	0.77	8.645	749.09	0.74	8.725	755.48	0.73

[1] a: DQN equipped with GNN-based intersecting cognition; b: GMARL without GNN-based intersecting cognition.

5.2. Ablation experiments

The ablation experiments are conducted to further demonstrate the effectiveness of the proposed method and examine the impact of the GMAN boosting RL and GNN-based intersecting cognition in the DGMARL-MVP. Specifically, a is the method that DQN is only equipped with GNN-based intersecting cognition, and b is the method that GMAN boosting RL is not equipped with GNN-based intersecting cognition. The results are shown in Table 5.

Compared with a, the average reward of DGMARL-MVP is increased by 17.29%, 2.74%, and 4.80% in P4-E2, P5-E3, and P7-E4 scenes, respectively. DGMARL-MVP exhibits fewer average time steps than a in the same scene. Specifically, compared with a, the average time steps of DGMARL-MVP are reduced by 7.24% in the P4-E2 scene at most, reduced by 2.62% in the P7-E4 scene at least, and 4.68% in all scenes on average. Also, DGMARL-MVP has the highest success rate, which is 5.81%, 4.94%, and 3.85% in P4-E2, P5-E3, and P7-E4 scenes, respectively. These results reveal that the proposed GMAN boosting RL algorithm can effectively alleviate the problem of sparse reward caused by MVP under urban environments and successfully indicate the optimization direction of the RL policy through the suitable potential dense reward generated by the GMAN, thus enhancing the optimality of the agent policy.

In addition, it can be obtained that the proposed DGMARL-MVP shows a higher average reward than b from Table 5, exactly 7.38%, 1.55%, and 2.84% higher than that of b in P4-E2, P5-E3, and P7-E4 scenes, respectively. Also, the average timesteps of DGMARL-MVP are separately 5.71%, 2.26%, and 1.07% less than that of B in P4-E2, P5-E3, and P7-E4 scenes, respectively. As for the success rate, DGMARL-MVP also shows a delightful superiority over b. Concretely, compared with b, the success rate of DGMARL-MVP increases by 4.60% in the P4-E2 scene at most, 1.25% in the P5-E3 scene at least, and 3.17% in all scenes on average. This phenomenon demonstrates that the proposed GNN-based intersecting cognition can effectively assist pursuing vehicles dealing with multi-source heterogeneous data from complex dynamic environments and enable agents to adaptively extract the situation information of other vehicles and the interaction features among agents so as to improve the pursuing efficiency.

Furthermore, Figure 5 depicts the bar chart comparison of the three metrics, average reward, time steps, and success rate, for DGMARL-MVP, a, and b in P4-E2, P5-E3, and P7-E4 scenes, offering a more intuitive illustra-

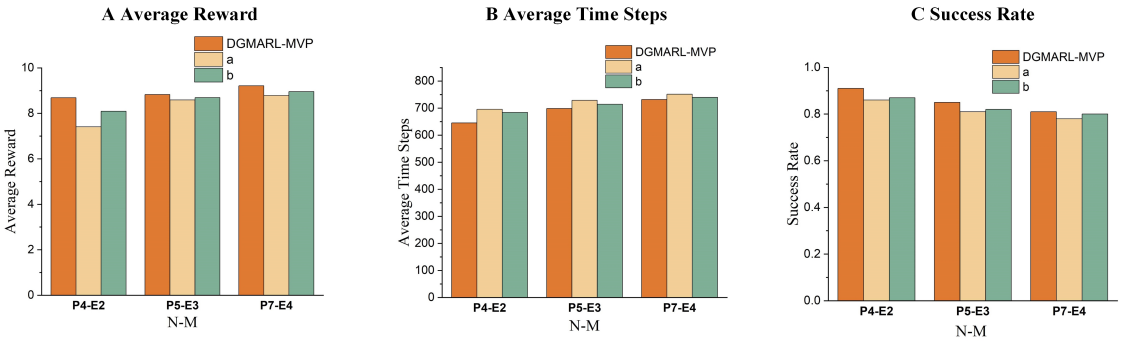

Figure 5. Results of Ablation Experiments. (A): Average Reward of Ablation Experiments. (B): Average Time Steps of Ablation Experiments. (C): Success Rate of Ablation Experiments.

tion of the effectiveness of the proposed modules. It is evident that DGMARL-MVP has the best performance for all metrics in any scene. The ablation experiments confirm that the proposed GMAN boosting RL algorithm can generate appropriate potential dense rewards, which makes RL more forward-looking in policy updating and correctly guides the optimization direction of RL policy, thereby improving the stability of distributed multi-agent system and enhancing the optimality of agent decision-making. Meanwhile, the proposed GNN-based intersecting cognition can adequately couple the interaction features of agents with traffic information and enhance their ability to handle multi-source heterogeneous data so as to promote the adaptability of the agents to the dynamic environment and improve the pursuing efficiency.

5.3. Comparison with other methods

This part demonstrates the performance of applying DGMARL-MVP and other algorithms to three scenes of MVP problems. This paper uses DQN, QMIX[30], and PPO for comparison. The details are shown in Table 5.

In the MVP problem of P4-E2, three metrics show consistency in performance evaluation. It is clear that DGMARL-MVP is noticeably the strongest performer on all of the metrics, which indicates the superiority of our proposed DGMARL-MVP. The success rate is an appreciable 91%, which is 12.35% higher than DQN, 5.81% higher than QMIX, and 5.81% higher than PPO. In the evaluation of the three metrics, the proposed DGMARL-MVP shows the most significant advantage in the average reward metric, with 15.64% higher than the sub-optimal algorithm PPO. This indicates that the proposed DGMARL-MVP can provide better guidance for agents in the pursuing process, in other words, make better local decisions and also lead to better final results. For other comparison algorithms, QMIX performs the worst of all the algorithms on all metrics in this scene, and PPO shows a sub-optimal performance.

Upgrading the difficulty to P5-E3, the proposed DGMARL-MVP algorithm still shows superior performance among other comparison algorithms. This superiority is specifically manifested in that our algorithm is 2.1%, 4.60%, and 6.25% better than the sub-optimal algorithm on average reward, average time steps, and success rate, respectively. The largest performance gap can be seen on the average reward metric compared with DQN, which is an exceedance of 22.68%. From this data, DGMARL-MVP performs better on the metric of the success rate than the other two metrics, which indicates that our algorithm DGMARL-MVP has high stability in a relatively difficult scene, resulting in an improvement in success rate. In this scene, other comparison methods have shown instability on three metrics to some extent. DQN performs worst on all metrics. QMIX is inferior to PPO in terms of success rate by 2.35%, although it is suboptimal in terms of rewards.

The difficulty setting of P7-E4 is approximately the same as that of P5-E3, but the increase of vehicles in both the pursuing team and the evading team increases the difficulty of global scheduling. However, from Table 5,

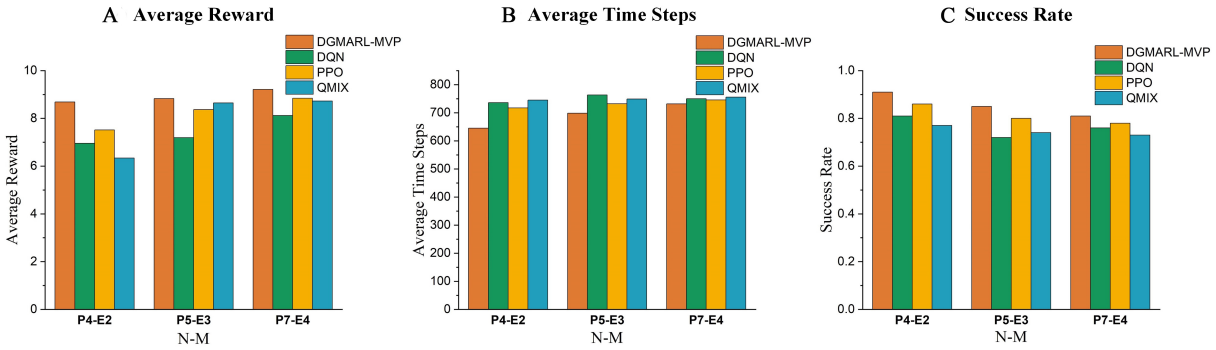

Figure 6. Results of Comparison Experiments. (A): Average Reward of Comparison Experiments. (B): Average Time Steps of Comparison Experiments. (C): Success Rate of Comparison Experiments.

the proposed DGMARL-MVP still shows satisfactory optimal performance among three other algorithms. In terms of the average reward metric, DGMARL-MVP is 4.17% better than the sub-optimal algorithm PPO and 13.43% better than the worst algorithm DQN. In terms of the average time steps, the difference between DGMARL-MVP and other algorithms is not that huge, but compared with the sub-optimal algorithm, there is also a 1.86% improvement with 14.12 time steps, which illustrates that DGMARL-MVP can steadily take its decision-making advantages in more difficult global scheduling scenes. In addition, DGMARL-MVP performs better by 7.05% on the success rate than the other algorithms on average. PPO shows the suboptimal performance on all three metrics.

By comparing the performance of all algorithms in these three scenes, the proposed DGMARL-MVP is the most stable algorithm and also performs the best. In spite of this stable performance, there are differences in the performance of DGMARL-MVP in the three scenes. As the difficulty of the scene increases, for example, from P4-E2 upgrading to P5-E3, the success rate of the pursuit decreases by 7.06%, and the average time steps decrease by 8.25%. It illustrates that the negative impact of increasing pursuing difficulty on success rates is indisputable, but the proposed DGMARL-MVP is more stable than other comparison algorithms. It is worth mentioning that DQN, which is the basis of DGMARL-MVP, is the worst performer in all three scenes. From this perspective, it indicates the proposed DGMARL-MVP makes considerable improvements.

In order to show the performance variation and comparison of all algorithms more clearly, a bar chart is used to show the changing trend of the three metrics in three scenes, as shown in Figure 6. Intuitively, as the number of evading vehicles increases, the average reward increases at the same time. In Figure 6A, unexpectedly, the average reward of the proposed DGMARL-MVP presents the highest rewards but a slight increase, and QMIX presents the largest increase. An inference of the reason could be that DGMARL-MVP provides better decisions during the pursuit, resulting in a high reward accumulation. As the difficulty increases in Figure 6B, DGMARL-MVP has a larger increase than other algorithms on the metric of average time steps, which illustrates that DGMARL-MVP presents more advantages in a sample scene than in difficult scenes. In terms of the success rate in Figure 6C, DGMARL-MVP, with the other algorithms, shows a downward trend, although DGMARL-MVP reaches the highest, except DQN, which shows a rise of performance as the scene changes from P5-E3 to P7-E4. The negative impact of increasing pursuing difficulty on success rates is indisputable, but the performance of DGMARL-MVP presents more stable, which shows the generalization of DGMARL-MVP.

5.4. Convergence comparison during training

In order to more convincingly prove the advantages of the proposed method, Figure 7 describes the convergence curves of average reward with training steps for various methods, including DGMARL-MVP, a, PPO, and QMIX, in P4-E3, P5-E3, and P7-E4 scenes. In Figure 7A, in the P4-E2 scene, it can be seen that com-

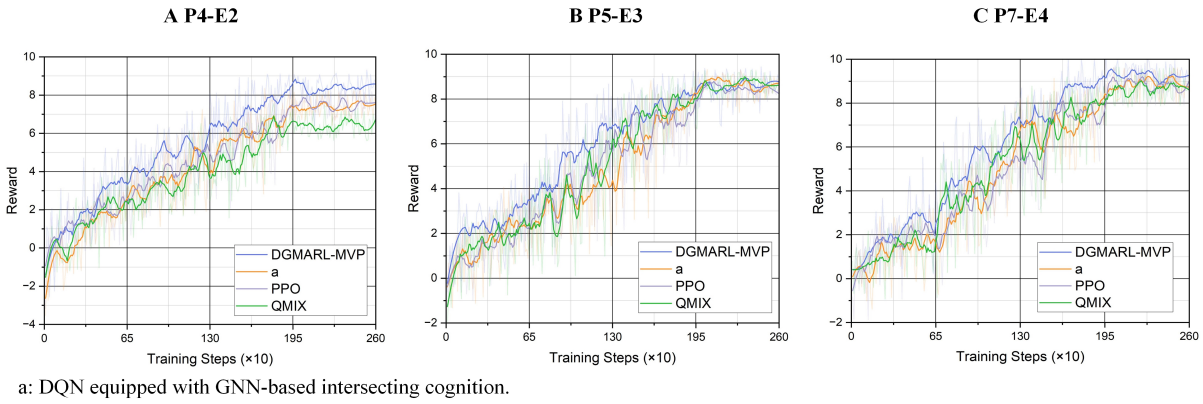

Figure 7. Convergence Process During Training. (A): Convergence Process of Average Reward for Methods in P4-E2. (B): Convergence Process of Average Reward for Methods in P5-E3. (C): Convergence Process of Average Reward for Methods in P7-E4.

pared with a, of which the fluctuation has a slight advantage over other methods in the last stage of training, DGMARL-MVP has better performance in convergence rate and convergence target. For the P5-E3 scene, as shown in Figure 7B, although all the methods showed similar convergence stability at the last stage of training, our method has a better convergence trend, which has a growing trend and a higher convergence target during the training. Figure 7C depicts the convergence curve of average reward with training steps in the P7-E4 scene, presenting that DGMARL-MVP has superior performance over other methods in both convergence rate and convergence trend. In conclusion, Figure 7 illustrates that compared with PPO and QMIX, which are separately the best and the worst of all comparison methods, DGMARL-MVP makes the competitive convergence rate and trend, demonstrating its superiority and effectiveness on MVP under urban environments.

With the horizontal comparison of the convergence of the proposed DGMARL-MVP in three scenes, the convergence performance of DGMARL-MVP in the three scenes is basically the same, and the convergence starts at about 1950 time steps. Compared with the other three algorithms, in all three scenes with different difficulties, the time step of convergence of DGMARL-MVP is basically the same as that of other algorithms. It is worth mentioning that the proposed DGMARL-MVP improved surprisingly quickly at the beginning of the training process, which indicates that our algorithm can better guide the direction of training at the initial stage. For this, it is not surprising that the reward of DGMARL-MVP remains the highest from the beginning to the end of the training in the P4-E2 scene and P7-E4 scene. An exception occurs in the scene of P5-E3; the average reward is reversed by DQN in the later stages of training, but the final result is still the best. In addition, the proposed DGMARL-MVP exhibits much smaller fluctuations in this scene, which shows the stability of the proposed algorithm.

6. CONCLUSIONS

This paper has proposed DGMARL-MVP to address the sparse rewards and insufficient perception of complex traffic situations brought by the MVP under urban traffic environments. In DGMARL-MVP, a GMAN has been designed to generate potential dense rewards and provide proper guidance for distributed RL optimization. Equipped with a GMAN, DGMARL-MVP has effectively solved the problem of optimization direction ambiguity caused by reward sparsity via the enhanced Bellman equation. In addition, this paper has proposed a GNN-based intersecting cognition, where the construction of vehicle graphs encourages a deep coupling between traffic information and multi-agent information. It thoroughly extracted and utilized the multi-source heterogeneous data of urban traffic and the complicated multi-agent interaction features, thus considerably improving pursuing efficiency. Extensive experimental results have demonstrated that the DGMARL-MVP can significantly improve pursuing success rate up to 85.67%. In the future, the impact of additional factors,

such as pedestrians and communication delays, on the design and analysis of MVP methods will be investigated. More real scenes, such as evading vehicles not following traffic rules, will also be considered to design smarter MVP methods.

DECLARATIONS

Acknowledgments

The authors would like to thank the editor-in-chief, the associate editor, and the anonymous reviewers for their valuable comments.

Authors' contributions

Made contributions to the research, idea generation, conception, and design of the work and wrote and edited the original draft: Zhang L, Li X, Yang Y, Wang Q

Made contributions to the algorithm design and simulation and developed the majority of the associated code for the simulation environment and the proposed method: Li X, Yuan Z, Yang Y

Participated in part of the experimental data analysis and visualizations and performed data collation and related tasks: Li L, Xu C, Wang Q

Performed critical review and revision and provided administrative, technical, and material support: Zhang L, Li L, Xu C

Availability of data and materials

The codes of this paper are open-sourced and available at *https://github.com/BUPT-ANTlab/DGMARL-MVP*.

Financial support and sponsorship

This work is supported by the National Natural Science Foundation of China (Grants No. 61971096 and No. 62176024), the National Key R&D Program of China (2022ZD01161, 2022YFB2503202), Beijing Municipal Science & Technology Commission (Grant No. Z181100001018035) and Engineering Research Center of Information Networks, Ministry of Education.

Conflicts of interest

All authors declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

REFERENCES

1. Chen C, Zou W, Xiang Z. Event-triggered consensus of multiple uncertain euler–lagrange systems with limited communication range. *IEEE Trans Syst Man Cybern, Syst* 2023;53:5945–54. DOI
2. Boin C, Lei L, Yang SX. AVDDPG – Federated reinforcement learning applied to autonomous platoon control. *Intell Robot* 2022;2:145–67. DOI
3. Zhu Z, Pivaro N, Gupta S, Gupta A, Canova M. Safe model-based off-policy reinforcement learning for eco-Driving in connected and automated hybrid electric vehicles. *IEEE Trans Intell Veh* 2022;7:387–98. DOI
4. Cao Z, Xu S, Jiao X, Peng H, Yang D. Trustworthy safety improvement for autonomous driving using reinforcement learning. *Trans Res Part C-Emer Technol* 2022;138:103656. DOI

5. Patrol guide. section: Tactical operations. procedure no: 221-15; 2016. Available from: https://www1.nyc.gov/assets/ccrb/downloads/pdf/investigations_pdf/pg221-15-vehicle-pursuits.pdf.
6. Qi Q, Zhang X, Guo X. 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. DOI
7. Xu B, Wang Y, Wang Z, Jia H, Lu Z. Hierarchically and cooperatively learning traffic signal control. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35; 2021. pp. 669–77. DOI
8. Li S, Yan Z, Wu C. Learning to delegate for large-scale vehicle routing. *Adv Neural Inf Process Syst* 2021;34. Available from: https://proceedings.neurips.cc/paper_files/paper/2021/file/dc9fa5f217a1e57b8a6adeb065560b38-Paper.pdf.
9. Garcia E, Casbeer DW, Von Moll A, Pachter M. Multiple pursuer multiple evader differential games. *IEEE Trans Automat Contr* 2020;66:2345–50. DOI
10. Xu Y, Yang H, Jiang B, Polycarpou MM. Multiplayer pursuit-evasion differential games with malicious pursuers. *IEEE Trans Automat Contr* 2022;67:4939–46. DOI
11. Lopez VG, Lewis FL, Wan Y, Sanchez EN, Fan L. Solutions for multiagent pursuit-evasion games on communication graphs: finite-time capture and asymptotic behaviors. *IEEE Trans Automat Contr* 2020;65:1911–23. DOI
12. Pan T, Yuan Y. A region-based relay pursuit scheme for a pursuit–evasion game with a single evader and multiple pursuers. *IEEE Trans Syst Man Cybern, Syst* 2023;53:1958–69. DOI
13. Jia S, Wang X, Shen L. A continuous-time markov decision process-based method with application in a pursuit-evasion example. *IEEE Trans Syst Man Cybern, Syst* 2016;46:1215–25. DOI
14. De Souza C, Newbury R, Cosgun A, et al. Decentralized multi-agent pursuit using deep reinforcement learning. *IEEE Robot Autom Lett* 2021;6:4552–59. DOI
15. Zhang R, Zong Q, Zhang X, Dou L, Tian B. Game of drones: multi-uav pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Trans Neural Netw Learn Syst* 2022. DOI
16. Yang Y, Li X, Yuan Z, Wang Q, Xu C, et al. 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. 534–41. DOI
17. Zheng Z, Duan H. UAV maneuver decision-making via deep reinforcement learning for short-range air combat. *Intell Robot* 2023;3:76–94. DOI
18. Durugkar I, Gemp I, Mahadevan S. Generative multi-adversarial networks. In: International Conference on Learning Representations (ICLR); 2017. Available from: https://openreview.net/forum?id=Byk-VI9eg.
19. Wang Z, Zhu H, He M, et al. Gan and multi-agent drl based decentralized traffic light signal control. *IEEE Trans Veh Technol* 2021;71:1333–48. DOI
20. Zhan H, Tao F, Cao Y. Human-guided robot behavior learning: a gan-assisted preference-based reinforcement learning approach. *IEEE Robot Autom Lett* 2021;6:3545–52. DOI
21. Li L, Yao J, Wenliang L, et al. Grin: Generative relation and intention network for multi-agent trajectory prediction. *Adv Neural Inf Process Syst* 2021;34:27107–18. Available from: https://proceedings.neurips.cc/paper_files/paper/2021/file/e3670ce0c315396e4836d7024abcf3dd-Paper.pdf.
22. Xia Y, Zhou J, Shi Z, Lu C, Huang H. Generative adversarial regularized mutual information policy gradient framework for automatic diagnosis. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34; 2020. pp. 1062–69. DOI
23. Zheng C, Yang S, Parra-Ullauri JM, Garcia-Dominguez A, Bencomo N. Reward-reinforced generative adversarial networks for multi-agent systems. *IEEE Trans Emerg Top Comput Intell* 2021;6:479–88. DOI
24. Liu Y, Wang W, Hu Y, et al. Multi-agent game abstraction via graph attention neural network. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. pp. 7211–18. DOI
25. Du W, Ding S, Zhang C, Shi Z. Multiagent Reinforcement Learning With Heterogeneous Graph Attention Network. *IEEE Trans Neural Netw Learn Syst* 2022;PP:1-10.. DOI
26. Liu Q, Li Z, Li X, Wu J, Yuan S. Graph convolution-based deep reinforcement learning for multi-agent decision-making in interactive traffic scenarios. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC). IEEE; 2022. pp. 4074–81. DOI
27. Xiaoqiang M, Fan Y, Xueyuan L, et al. Graph Convolution Reinforcement Learning for Decision-Making in Highway Overtaking Scenario. In: 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA). IEEE; 2022. pp. 417–22. DOI
28. Chen Y, Shu T, Zhou X, et al. Graph attention network with spatial-temporal clustering for traffic flow forecasting in intelligent transportation system. *IEEE Trans Intell Transport Syst* 2022. DOI
29. Lopez PA, Behrisch M, Bieker-Walz L, et al. Microscopic Traffic Simulation using SUMO. In: The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE; 2018. . DOI
30. Rashid T, Samvelyan M, De Witt CS, et al. Monotonic value function factorisation for deep multi-agent reinforcement learning. *J Mach Learn Res* 2020;21:7234–84. DOI