2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10160698

Conflict-constrained Multi-agent Reinforcement Learning Method for Parking Trajectory Planning

Siyuan Chen, Meiling Wang, Yi Yang, Wenjie Song*

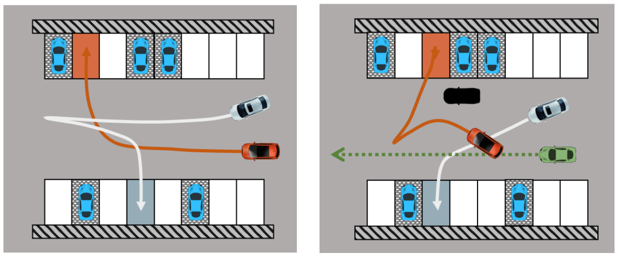
Abstract— **Automated Valet Parking (AVP) has been extensively researched as an important application of autonomous driving. Considering the high dynamics and density of real parking lots, a system that considers multiple vehicles simultaneously is more robust and efficient than a single vehicle setting as in most studies. In this paper, we propose a distributed Multi-agent Reinforcement Learning(MARL) method for coordinating multiple vehicles in the framework of an AVP system. This method utilizes traditional trajectory planning to accelerate the learning process and introduces collision conflict constraints for policy optimization to mitigate the path conflict problem. In contrast to other centralized multi-agent path finding methods, the proposed approach is scalable, distributed, and adapts to dynamic stochastic scenarios. We train the models in random scenarios and validate in several artificially designed complex parking scenarios where vehicles are always disturbed by dynamic and static obstacles. Experimental results show that our approach mitigates path conflicts and excels in terms of success rate and efficiency.**

I. INTRODUCTION

Automated Valet Parking(AVP) is one of the most advanced technologies for improving parking efficiency and security [1]. It is recognized by the industry as the first L4 autonomous driving scenario to be implemented. Compared with road driving, autonomous parking requires a higher degree of precision to park in tight spaces with static and dynamic obstacles.

So far, the research on AVP trajectory planning mainly focus on the sampling or optimization approaches based on known maps to obtain a continuous trajectory [2] [3]. However, the computational complexity of traditional planning method depends on the environmental settings, and the performance varies considerably in different environments.

Learning-based approaches are gradually showing strong advantages in the field of autonomous driving. Among them, deep reinforcement learning(DRL) has shown promising performance on various planning and decision making tasks including robotics [4] and autonomous driving [5]. Compared with traditional rule-based trajectory planning methods [6], DRL have the potential for better scalability and generalization in complex scenarios due to the high representational capability of neural networks [7] [8] [9].



(a) (b)

Fig. 1: The illustration of multi-vehicle parking scenario. (a) Multi-vehicle parking random scenario. (b) Complex scenes with static(black) and dynamic(green) obstacle.

The method for single vehicle planning would somehow consider the prediction of other vehicles. However, it is difficult to predict in such irregular and high-dynamic parking scenarios. A more appropriate way is to consider it as a multi-agent problem. Then comes naturally the multi-agent path finding(MAPF) problem. As shown in Fig.1, the challenge for multi-vehicles parking planning is the conflict among them. Conflict-based search(CBS) is such a method to overcome the conflict by searching in two layers, checking for conflicts and optimising paths respectively [10] [11]. But it is a centralised planning approach where the trajectories of multiple vehicles are replanned in a uniform manner, such that the computation increases with the number of vehicles. It also relies on vehicle-to-X(V2X) technology and requires higher investment in infrastructure. To deal with the conflicts between multiple vehicles, MARL is rapidly being applied from robotics to autonomous driving as a scalable framework for dealing with interaction problems [12]. By sharing policy or perception information, MARL can solve cooperative or adversarial problems.

In this paper, we adapt MARL to multi-vehicle autonomous parking problem. Compared with single-vehicle AVP, ensuring the trajectory safety of multiple vehicles is more challenging. At the same time, the efficiency and safety of multi-vehicle parking planning should be balanced, because conservative policy may lead to unnecessary avoidance. In addition, the MARL training framework poses scalability problems for multi-vehicle parking systems. Thus, we focus on trajectory planning for multi-vehicle parking problem and proposes a safe and efficient trajectory planning method based on MARL.

The main contributions of this work are threefold:

1. A MARL-based approach to parking planning is pro-

This work was partly supported by National Natural Science Foundation of China (Grant No. 61903034, 61973034 and 62173042), Program for Changjiang Scholars and Innovative Research Team in University (T2014224), Beijing Institute of Technology Research Fund Program for Young Scholars. (Corresponding author: Wenjie Song (email: songwj@bit.edu.cn))

The authors are with School of Automation, Beijing Institute of Technology, Beijing 100081, P.R.China.

posed. By encoding heterogeneous traffic information (i.e. target points and vehicle states), the method implicitly simulate inter-vehicle interactions in a parking scene and can cope with static and dynamic obstacles during planning. The model framework is scalable and can cope with different numbers of vehicles.

2. To avoid track conflicts, the conflict constraint between multiple vehicles is added in the reinforcement learning strategy optimization process, which ensures the safety and the efficiency of multi-vehicle planning.

3 The performance of the proposed approach is demonstrated via experimental evaluation on different kinds of parking lots (parallel and vertical) and various conflict tasks. Experimental results show that our approach can effectively deal with various tricky parking problems and with the capability of versatility.

II. RELATED WORK

In this section, we review existing work on autonomous parking systems, reinforcement learning-based planning methods, and policy optimization techniques under constraints.

A. Autonomous Parking Trajectory Planning

Traditional planning methods based on sampling and search, such as A* [13], dynamic window approach [14], are widely used in parking trajectory planning. However, its space and time complexity are greatly affected by the environment. To achieve fast and safe trajectory planning, Li [3] describes the problem as an optimal control problem. The coarse trajectory guiding a homotopic route is used to replace the difficult collision avoidance constraint with the within-corridor constraints. Aiming at the scenario of multi-vehicle parking, [15] proposed a control method for coordinating multiple vehicles in the framework of AVP system. The system relies on infrastructure servers and vehicle-to-infrastructure (V2I) communication interfaces, which is difficult to implement single-vehicle deployment.

B. Reinforcement Learning-based Planning

DRL based trajectory planning approaches can take a large number of training experiences into account and are advantageous in tackling complex scenarios with high efficiency and robustness. Deep neural networks are usually used to approximate agents policy and value function. Some people propose to learn the navigation policy in a completely end-to-end fashion, which directly maps raw sensor data to the agents action [16].

For multi-agent path planning, PRIMAL [17] uses image-based representation and target goal as input sources. However, non-cooperative dynamic obstacles and temporal information are not considered in their work. Besides, the centralized training approach takes a long time even with the help of imitation learning. Liu. etc [18] proposes a decentralized partially observable multi-agent path planning with evolutionary reinforcement learning (MAPPER) method to learn an effective local planning policy in mixed dynamic environments, which model dynamic obstacles behavior with an image-based representation and decompose the long-range navigation task into many easier waypoint-conditioned subtasks. Han. etc [19] developed a novel reward function based on the reciprocal velocity obstacle(RVO) or velocity obstacle(VO) areas and expected collision time, which encourages robots to learn reciprocal local collision avoidance behaviors under diverse situations.

C. Constrained Reinforcement Learning

Constrained Reinforcement Learning (CRL) is an active research area in safe RL aiming to learn effective and safe. CRL problems is usually formulated as constrained Markov decision process (CMDP) [20]. The constraint optimization method to solve the CRL problem includes the penalty function, Lagrangian methods [21] and Trust region methods [22]. [23] studied the novel reachability constraint in CRL, where the safety value function is constrained, guaranteeing persistent constraint-satisfaction.

III. PRELIMINARIES

A. Markov Decision Process(MDP)

In this paper, we formulate the parking trajectory planning of multiple vehicles as a partially observable Markov decision process(POMDP). It is specified by the tuple $< O, S, A, T, R, \gamma, \rho_0 >$, where O is the observation space; S is the state space; A is the action space; T: $S \times A \times S \rightarrow \mathbb{R}$ is the transition model; R: $S \times A \rightarrow \mathbb{R}$ is the reward model; $\gamma \in [0, 1)$ is the discount factor, ρ_0 is the initial state distribution. At time step t, the agent are in a state s_t, and every agent i takes an action a_t^i according to its policy $\pi_i(a^i|s_t)$ and receive the reward $R_i(s_t^i, a_t^i)$. The environment then transits to a new state $s_{t+1}^i \sim p(|s_t^i, a_t^i)$. The objective in a POMDP is to find a policy π_i that maximizes expected return:

$$\pi_i^* = \arg\max_{\pi_i} \mathbb{E}_{s_0,a_0,o_0,\ldots} \sum_{t=0}^{\infty} \gamma^t R_i\left(s_t^i, a_t^i\right) \tag{1}$$

B. Constrained Policy Optimization

Most RL algorithms allow the agent to freely explore the environment and take any action that increases the reward. However, actions that are highly rewarding can also be highly risky. In autonomous parking scenario, ensuring the safety of the agent is critical. Unlike standard RL, which only maximizes the reward function, the agent must take actions to avoid dangerous situations. In terms of the optimization process, it is solving an optimization problem with constraints.

Constrained Policy Optimization (CPO) [22] method introduces Trust Region Policy Optimization [24] (TRPO)'s idea of hard constraint on KL divergence. The objective function of TRPO is:

$$\begin{gathered}\pi_{k+1} = \arg\max_{\pi \in \Pi_\theta} J(\pi) \\ \text{s.t. } D_{KL}\left(\pi, \pi_k\right) \leq \delta,\end{gathered} \tag{2}$$

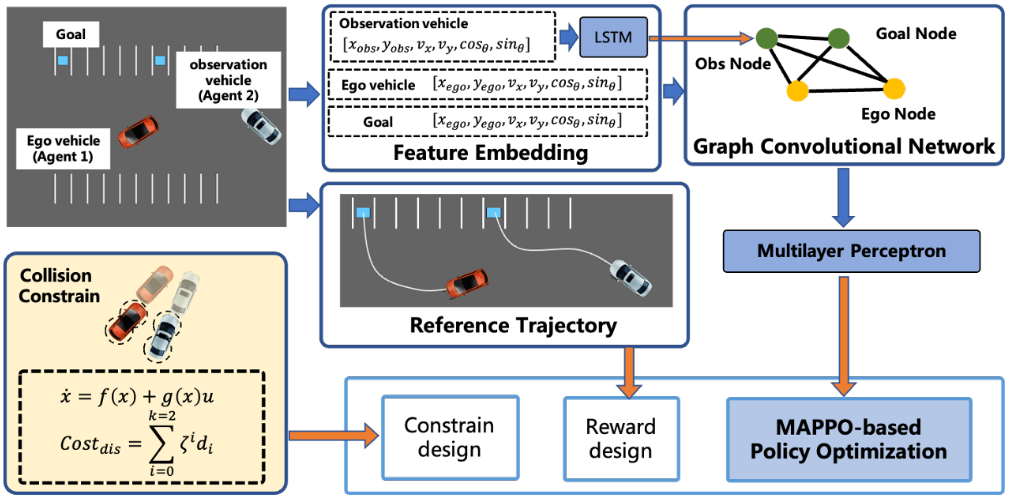

Goal
observation vehicle (Agent 2)
Ego vehicle (Agent 1)
Observation vehicle
$[x_{obs}, y_{obs}, v_x, v_y, cos_\theta, sin_\theta]$
LSTM
Ego vehicle $[x_{ego}, y_{ego}, v_x, v_y, cos_\theta, sin_\theta]$
Goal $[x_{ego}, y_{ego}, v_x, v_y, cos_\theta, sin_\theta]$
Feature Embedding
Goal Node
Obs Node
Ego Node
Graph Convolutional Network
Multilayer Perceptron
Reference Trajectory
Collision Constrain
$\dot{x} = f(x) + g(x)u$
$Cost_{dis} = \sum_{i=0}^{k=2} \zeta^i d_i$
Constrain design
Reward design
MAPPO-based Policy Optimization

Fig. 2: System framework of multiple vehicles parking driving.

by introducing constraints, it becomes:

$$\begin{aligned} \pi_{k+1} = \arg \max_{\pi \in \Pi_\theta} & J(\pi) \\ \text{s.t. } & J_{C_i}(\pi) \le d_i \quad i = 1, \ldots, m \\ & D_{KL}(\pi, \pi_k) \le \delta. \end{aligned} \tag{3}$$

where $J(\boldsymbol{\pi}) \triangleq \mathbb{E}_{s_0 \sim \rho^0, \mathbf{a}_{0:\infty} \sim \boldsymbol{\pi}, s_{1:\infty} \sim p}\left[\sum_{t=0}^{\infty} \gamma^t R\left(\mathrm{s}_t, \mathbf{a}_t\right)\right]$, aiming to max- imise the expected total reward. J_{C_i} is the collision constraint, which is constrained by the cost in Equation.6. $R(s, a)$ is the reward function of the system, which is defined in the reward design in Section IV.

In addition to CPO, it is possible to solve Equation.3 by using Lagrange multiplier to solve the optimization of linear and quadratic constraints [25]. The Lagrangian method is easy to implement and does not require the calculation of the magnitude of the Hessian H.

$$\min_{\lambda \ge 0} \max_{\theta} L(\lambda, \theta) = \min_{\lambda \ge 0} \max_{\theta} \left[J_R^{\pi_\theta} - \lambda \cdot \left(J_C^{\pi_\theta} - d\right)\right] \tag{4}$$

where L is the Lagrangian and $\lambda \ge 0$ is the Lagrange multiplier.

IV. METHODS

The structure of the proposed multiple vehicles parking trajectory planning system is shown in Fig.2. The whole system is based on MAPPO algorithm [26] as the main framework to optimize the reinforcement learning strategy. Considering the interaction between the main vehicle and surrounding vehicles, LSTM network and graph neural network are introduced to extract features. Moreover, the reference trajectory is introduced to design the reward, and the contradictions and conflicts of multi-vehicle driving are considered to construct the constraint conditions of strategy optimization. In this section, the detailed design of the system is divided into the following three parts.

A. *Problem Statement*

As shown in Fig.1, a target parking space is randomly assigned to each vehicle in the parking lot. The goal of the vehicle is to successfully enter the appropriate parking space while remaining safe to drive. As mentioned before, the tuple of MDP is composed of three elements: state space S, action space A, reward function R.

1) State Space: In the parking trajectory planning task, we need to pay attention to the state of ego vehicle s_{ego}, the target parking space s_{park} and the states of the surrounding vehicles s_{surr}. The state of each one is represented as a six-dimensional vector, i.e. $[x, y, v_x, v_y, cos\_\theta, sin\_\theta]$, where $[x, y]$ represents the coordinates of the vehicle or parking space, $[v_x, v_y]$ is the velocity in the x and y directions and $[cos\_\theta, sin\_\theta]$ represents the cosine and sine of the heading angle. In particular, the velocity $[v_x, v_y]$ of the target parking space s_{park} is set to 0.

2) Action Space: In the parking scenario, we need to control the vehicle in a continuous motion, so the motion space dimension is 2-dimensional, i.e. front steering wheel angle (δ) and acceleration (acc). Action a_i represents the action performed by vehicle i and all the actions at each time step form a joint action $a \in \mathcal{A}$. $\mathcal{A}$ is a set defined as belows:

$$a_i \in \mathcal{A} = \{\delta, acc \mid \delta \in [-1, 1], acc \in [-1, 1]\}$$

where the front wheel steering angle (δ) and acceleration (acc) ranges are mapped to $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$, $\left[-5m/s^2, 5m/s^2\right]$ respectively, based on the actual physical characteristics of the vehicle.

3) Reward Design: The reward function is designed to encourage the agent to learn to park to the corresponding parking space as quickly and accurately as possible without collisions, which is calculated by:

$$R = \omega * \|s_{ego} - s_{park}\| + R_{ref}$$

where ω is $[1.0, 0.3, 0.0, 0.0, 0.2, 0.2]$, representing the weight of the difference between the current and target state of the ego vehicle respectively.

In the unstructured environment, the observation space range of agents is large, and the convergence rate of agents is slow. Inspired by some Multi-Agent Path Finding (MAPF) algorithms [18], they use A* for single-agent path generation and apply an off-route penalty if agents failed to follow the path. Therefore, in order to accelerate the learning of parking trajectory, hybrid A* is used for trajectory pre-planning before the vehicle starts. The nearest point from the vehicle's current position to the preplanned trajectory is calculated, and the distance deviation is calculated as a penalty R_{ref}.

Algorithm 1: Collision Constrained based MAPPO

```
Initialize the policy actor $\pi_\theta$, critic $V_\psi$, constraint
  value network $Vc_\phi$; for episode = 1, ..., $N_{eps}$ do
    // collect data
    for agent $i = 1, ..., n$ do
        Run policy $\pi_\theta$ for $T$ steps
        Collect data $(o_i, a_i, r_i, cost_{dis-i})$ for $T$ steps,
          which denote the observation, action, reward,
          and collision cost of step i, respectively
        Add data into buffer $buffer_i$
        Compute returns $\hat{R}_t$
        Compute constraint cost returns $\hat{Rc}_t$
    end
    // policy update
    for $k \leftarrow 1$ to $K_{iter}$ do
        Compute advantages estimation $\hat{A}_t$ based on
          the current critic network $V_\psi$
        Compute constraint advantages estimation
          $\hat{Ac}_t$ based on the current constraint value
          network $Vc_\phi$
        Update policy with PPO-Clip objective:
        1. compute actor hybrid advantage with
          constraint: $A^- = \hat{A}_t - \lambda * \hat{Ac}_t$
        2. update actor:
            $r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_k}(a_t \mid s_t)}$
          $L^{CLIP}(\theta) =$
        $\hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)A^-, \operatorname{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)A^-\right)\right]$
        Update critic:
            $L_V = \hat{\mathbb{E}}_t\left[\hat{A}_t - \hat{R}_t\right]^2$
        Update constraint value network:
            $L_{Vc} = \hat{\mathbb{E}}_t\left[\hat{Ac}_t - \hat{Rc}_t\right]^2$
    end
end
```

B. Neural Network Architecture

The network structure used in this paper is shown in Fig.3. We use MAPPO as the core RL algorithm for its stability and computational efficiency. Consider the parking scenario containing the ego vehicle and n surrounding vehicles. The overall structure consists of three parts: observation encoder, graph representation and actor critic network.

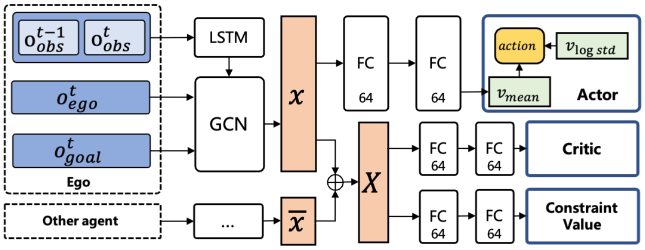


Fig. 3: The neural network architecture of the system.

For parking lots, where there are no lane constraints and vehicle's behaviour is heavily influenced by its relationship with other traffic participants. So the description of vehicle state is particularly important. In this paper, LSTM network is used for feature coding of surrounding vehicles' states. The shared dependencies of different traffic participants can be formalized as a graph, and each car and target parking space can be represented as a node.

The output of the graph neural network is used to calculate the distribution of actions in the actor network π_θ through the fully connected layers. At the same time, global observation is considered for critic network V_ψ and constraint value network Vc_ϕ, and the features of multiple agents are simultaneously used as the input of fully connected layers to obtain the estimation of value.

The training framework proposed in this paper is scalable. In the training, all agents share the same strategy, and the nearest vehicle is used as the observation to adapt to the multi-vehicle environment.

C. Collision Constrain

In the process of multi-agent parking, the distance constraint between the vehicle and the obstacle is beneficial to ensure the safety of the vehicle.

$$\dot{x} = f(x) + g(x)u \tag{5}$$

Based on the vehicle state transition function (Equation.5), we can obtain the state information of the vehicle and make multi-step prediction of the safe distance between ego and surrounding vehicles. The cost of distance is as follows:

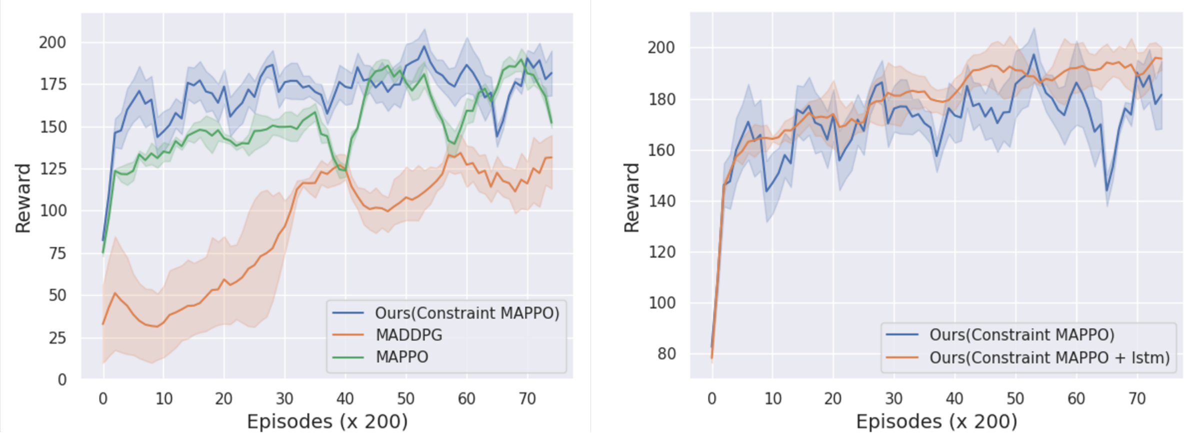
$$Cost_{dis} = \sum_{i=0}^{k_{pre}} \zeta^i \left[1.0 - \operatorname{clip}\left(D_{min}, 0, \mathit{\Gamma}_d\right)/\mathit{\Gamma}_d\right] \tag{6}$$

where k_{pre} is the number of frames to predict the conflict constraint, D_{min} is the distance from the nearest obstacle to ego vehicle, and $\mathit{\Gamma}_d$ is the distance threshold to calculate the conflict constraint.

We leverage the Lagrange multiplier method to solve problem, which is a common approach to CRL. As shown in Algorithm.1, we use the neural network with the same structure as critic V_ψ network to fit the value function of conflict constraints Vc_ϕ, and introduce conflict constraints in the process of updating actor network π_θ, as shown in Equation.4.

TABLE I: A performance comparison of different approaches in the random scenarios

Scenario	Success Rate (%)			Total Parking Steps			Average Speed (m/s)		
	(MA)DDPG	(MA)PPO	Ours	(MA)DDPG	(MA)PPO	Ours	(MA)DDPG	(MA)PPO	Ours
One Vehicle	82.5	81.0	**94.2**	122.4	126.3	**117.6**	4.1	4.40	4.26
Multiple Vehicles	60.5	68.2	**82.5**	134.6	119.4	**106.2**	3.8	4.22	4.25



(a) (b)

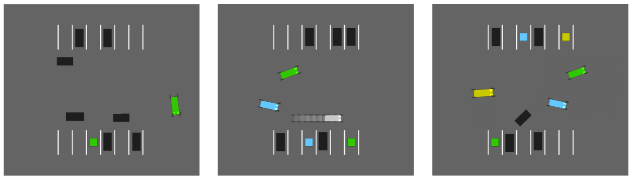
Fig. 4: The learning curve of four methods. The model is tested every 50 epochs to calculate the test episode rewards.

V. Experiments And Results

A. Experiment Setup

In this paper, the OpenAI gym-based parking-env simulator [27] is employed to verify the effectiveness. This paper focuses on multi-vehicle parking trajectory planning and the target parking spaces will be randomly assigned for each agent.

Fig.5 shows different parking scenarios, including static obstacle scenario, dynamic obstacle scenario and multi-vehicle scenario. In the experiment, we tested the proposed method in a variety of environments to verify its adaptability to environmental diversity and the scalability of multi-agent environment. All agents share the same policy and collect the observations for the policy update by the MAPPO algorithm after each epoch. The environment will be reset when there are collisions or over the maximum episode length.



(a) (b) (c)

Fig. 5: Different scenarios in the multi-vehicle parking experiment. The color of the vehicle corresponds to the target parking space. (a) Static obstacle scenario.
(b) Dynamic obstacle(white vehicle) scenario.
(c) Multi-vehicles random scenario.

B. Training Performance

In order to verify the effectiveness of collision constrain based MAPPO algorithm proposed in this paper, we compared it with MADDPG [28] and MAPPO in the scenario shown in Fig.5.

Each trial is trained for 15k epochs with 300 environment steps per epoch. The values of the hyperparameters for the training process are listed in Table.II.

TABLE II: The hyperparameters of the training process

Para.	Val.	Para.	Val.
learning rate	3e-4	optimizer	Adam
max train steps	9e6	gamma	0.99
buffer size	300	K	3
mini batch size	300	ppo epsilon clip	0.2

The training curves are represented by a solid line of the mean value and an error band of the 95% confidence interval. As shown in Fig.4(a), we compare the training effect of the proposed algorithm with MADDPG and MAPPO algorithms and our method shows great advantages in searching for optimal solution. Due to the introduction of the hybrid A* reference trajectory, the reward function is redesigned, and the unreasonable driving actions can be punished more accurately, which can guide the vehicle to quickly learn the stopping trajectory, and the curve convergence speed is faster. At the same time, the new conflict-constrained value function network can effectively evaluate the safety cost between the main vehicle and obstacles. The conflict between the main vehicle and obstacles is constrained, which restricts the choice of unsafe actions of the policy network and improves the safety of driving.

In Fig.4(b), we compare the training effect before and after introducing obstacle feature extraction and graph neural network. According to the Reward convergence curve, the feature extraction and graph network designed in this paper can effectively characterize the states of the ego vehicle, obstacles and target parking space, reduce the effective state space, and accelerate the convergence of the training process.

C. Model Evaluation

1) Metrics: We use the following three metrics to measure the driving performance: success rate, total parking steps and average speed. The success rate is a ratio of the successful cases without any collision during the parking period, which describes the policys ability of collision avoidance and trajetory planning. The total parking steps refers to the amount of steps, represented by the iteration step in the simulation when all robots arrive at the goal positions, which reflects the policys efficiency. The average speed of the parking process of the multiple vehicles measures the policys performance on effective velocity selection.

2) Safety and efficiency: We conducted a set of experiments using the scenario examples shown in Fig.5, testing each scenario 100 times at random and counting the three metrics as shown in Table.I. It can be seen that our method performs well on all three metrics. To prevent collisions, the traditional practice of reinforcement learning is to assign a negative reward whenever a collision occurs. Comparing MAPPO with our proposed method in Table.I, the The constraints on vehicle conflicts can effectively avoid collisions and improve the success rate of the system. With the introduction of reference trajectory as a guide and the inclusion of feature extraction and graph structure representation, vehicle driving efficiency is improved.

3) Generalization Evaluation: As shown in Fig.6, we deploy experiments from single-vehicle to multi-vehicle scenarios. The first two figures show the ability of avoiding both static obstacles (black rectangle in Fig.6(a)) and dynamic obstacle (blue rectangles in Fig.6(b)) during parking. In Fig.6(c), Fig.6(d) the two ego vehicles pass through each other safely even the trajectories to their goal are crossed. In particular, the target parking space for the blue vehicle is occupied by the green vehicle in Fig.6(c). Therefore, the blue vehicle waits for the green vehicle to leave before planning to park into the target location. Then comes up with the more complex scenario(Fig.6(e), Fig.6(f)), the proposed method is also able to perform safe trajectory planning. All the experiments demonstrate the scalability of our scheme as well as the conflict avoidance ability. Unlike frameworks, our proposed approach does not rely on Internet of Vehicles (IoV) and allows for distributed deployment.

The generalisability of our approach is not only reflected in the variation of the number of vehicles, but also in the high adaptability to different types of parking spaces. Simulation experiments are carried out in the parking scenarios with parallel parking spaces as shown in Fig.7. The ego vehicle has the ability to handle multiple static obstacles(Fig.7(a)). In the case of multi-vehicle path conflict, it is able to achieve reasonable avoidance. The blue vehicle in Fig.7(b) starts late and actively avoids the green one.

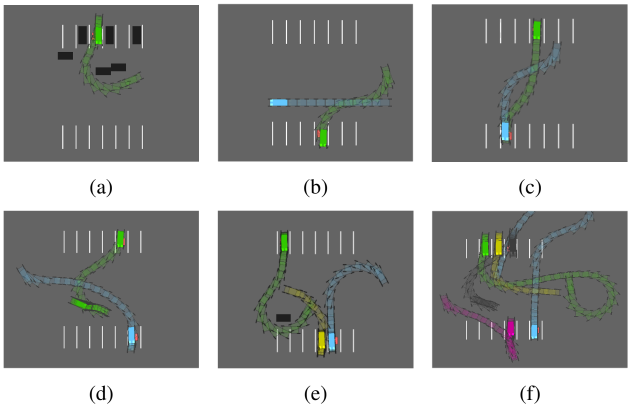


Fig. 6: Motion trajectory diagram under four different scenarios. (a) Single vehicle in the scene with static obstacles. (b) Single vehicle in the scene with dynamic obstacle. (c)(d) Scenarios with conflicting trajectories of two vehicles. (e)(f) Multi-vehicle random scenario.

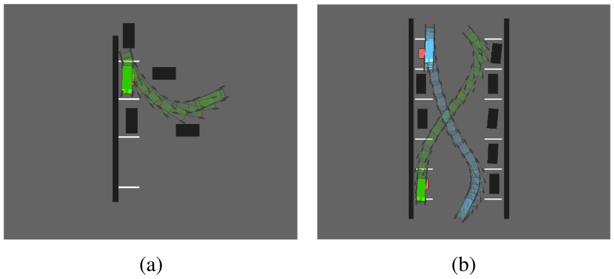


Fig. 7: Experimentation with parallel parking spaces. (a) Static obstacles blocking scenarios. (b) Multi-vehicle path conflict in parallel parking scenario.

VI. CONCLUSIONS

In this paper, a novel decentralized multi-agent RL algorithm is proposed to solve the problem of multi-vehicle parking collision in the parking environment, so as to improve the efficiency and safety of autonomous parking. In order to accelerate parking trajectory learning, traditional hybrid A* trajectory planning was introduced, and reward function was designed to accelerate training convergence. Considering the interaction between the ego vehicle and the surrounding dynamic vehicles, an obstacle feature extraction network is developed by encoding the multi-frame obstacle observation information using the timing information, and the features are mapped into the graph structure at a low cost. Comparing MADDPG, MAPPO and our method, the experimental results show that our method can maintain higher success rate and driving efficiency in random transformation scenarios. In the future, we will carry out research on intelligent allocation methods for target parking spaces, where the joint optimisation of space selection and parking trajectories will further improve parking efficiency.

References

[1] H. Banzhaf, D. Nienhüser, S. Knoop, and J. M. Zöllner, "The future of parking: A survey on automated valet parking with an outlook on high density parking," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1827–1834.

[2] W. Liu, Z. Li, L. Li, and F.-Y. Wang, "Parking like a human: A direct trajectory planning solution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3388–3397, 2017.

[3] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong, and X. Peng, "Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[4] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[5] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, "Driving in dense traffic with model-free reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5385–5392.

[6] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[7] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6064–6071.

[8] Z. Du, Q. Miao, and C. Zong, "Trajectory planning for automated parking systems using deep reinforcement learning," *International Journal of Automotive Technology*, vol. 21, no. 4, pp. 881–887, 2020.

[9] L. Junzuo and L. Qiang, "An automatic parking model based on deep reinforcement learning," in *Journal of Physics: Conference Series*, vol. 1883, no. 1. IOP Publishing, 2021, p. 012111.

[10] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[11] J. Li, W. Ruml, and S. Koenig, "Eecbs: A bounded-suboptimal search for multi-agent path finding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, 2021, pp. 12 353–12 362.

[12] S. Bhalla, S. Ganapathi Subramanian, and M. Crowley, "Deep multi agent reinforcement learning for autonomous driving," in *Canadian Conference on Artificial Intelligence*. Springer, 2020, pp. 67–78.

[13] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The international journal of robotics research*, vol. 29, no. 5, pp. 485–501, 2010.

[14] Y. Lei, Y. Wang, S. Wu, X. Gu, and X. Qin, "A fuzzy logic-based adaptive dynamic window approach for path planning of automated driving mining truck," in *2021 IEEE International Conference on Mechatronics (ICM)*. IEEE, 2021, pp. 1–6.

[15] M. Kneissl, A. K. Madhusudhanan, A. Molin, H. Esen, and S. Hirche, "A multi-vehicle control framework with application to automated valet parking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5697–5707, 2020.

[16] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 ieee international conference on robotics and automation (icra)*. IEEE, 2017, pp. 1527–1533.

[17] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. S. Kumar, S. Koenig, and H. Choset, "Primal: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2378–2385, 2019.

[18] Z. Liu, B. Chen, H. Zhou, G. Koushik, M. Hebert, and D. Zhao, "Mapper: Multi-agent path planning with evolutionary reinforcement learning in mixed dynamic environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 11 748–11 754.

[19] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao, and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5896–5903, 2022.

[20] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[21] H. Ma, Y. Guan, S. E. Li, X. Zhang, S. Zheng, and J. Chen, "Feasible actor-critic: Constrained reinforcement learning for ensuring statewise safety," *arXiv preprint arXiv:2105.10682*, 2021.

[22] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*. PMLR, 2017, pp. 22–31.

[23] D. Yu, H. Ma, S. Li, and J. Chen, "Reachability constrained reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 25 636–25 655.

[24] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[25] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," *arXiv preprint arXiv:1805.11074*, 2018.

[26] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative, multi-agent games," *arXiv preprint arXiv:2103.01955*, 2021.

[27] E. Leurent, "An environment for autonomous driving decision-making," https://github.com/eleurent/highway-env, 2018.

[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, 2017.