Paper:

# A Decision-Making Method of Multimodal Fusion Based on Deep Reinforcement Learning for Highway Autonomous Driving

**Ping Liu[1], Hongbin Ma*[2], Ying Jin[3], Yazhe Ding[4], Bin Cao[5], Zewen Du[6]**

School of Automation, Beijing Institute of Technology, Beijing, China
mathmhb@bit.edu.cn

**Abstract. This paper focuses on deep reinforcement learning-based decision-making and planning methods for autonomous vehicles to drive on dynamic highways in a high-speed, comfortable, and safe manner. Due to the highly dynamic and complex road environment, the ability to make decisions and plan autonomously as humans has always been a challenging topic for autonomous vehicles. Different from the previous only taking dynamic traffic state as input, a multimodal fusion method is designed in this paper to enrich the input feature information, and the longitudinal acceleration (or deceleration) and lateral rotation angle are used for the output of the decision-making model. The total decision reward for the highway autonomous driving problem is designed into three parts, namely, the non-collision reward, the high-speed driving reward, and the successful-to-destination reward. During the experiment, the training and testing parameters of the decision-making model are set according to the driving parameters of the real highway. The experimental results show that the method accelerates the rapid convergence of the decision-making model by fusing the multi-modal data in the high-speed driving environment, and can make reasonable decisions autonomously in the high-speed environment.**

**Keywords:** Autonomous driving, dynamic environment, decision making, deep reinforcement learning, multimodal fusion

## 1. Introduction

In recent years, autonomous vehicles have received increasing attention for easing traffic jams and improving intercity transportation capacity. As a core component of the autonomous driving field, decision making has received relatively little attention and progress compared to other research (perception and control) [1]. Due to the high dynamic and complex environment characteristics of real expressway scenes, it is generally difficult for autonomous vehicles to plan a feasible route in real time during driving, or even if they make fast decisions, they may not be able to guarantee the efficiency, safety, and comfort in the whole driving process. Therefore, effective decision making becomes one of the most challenging tasks for autonomous driving [2].

Take the one-way four-lane highway scene shown in Fig. 1 as an example, the self-driving vehicle marked in green is the vehicle to be controlled, and the red flag is the destination that the vehicle will reach. In the real highway scene, the vehicle needs to do at least three points: 1) do not collide with other vehicles to ensure the safety of life and property; 2) get to the destination quickly and efficiently to save time and ease traffic jam; 3) comfortable driving to enhance the passenger experience. Considering the actual driving process, the probability of collision will increase when the speed and comfort of the vehicle are improved. Therefore, how to balance the above three situations and determine the optimal driving route is of great research significance for the pursuit of high speed, comfort and safety in the field of autonomous driving in the future.

At present, autonomous vehicle decision and planning research is mainly divided into two categories. One is the traditional planning decision algorithm originated from the field of robotics. It has been widely used in practical life, such as desert vehicles [3], planetary rovers [4] and buses [5]. Traditional graph search algorithms, such as Dijkstra's [6], A* algorithm, Dynamic A Star algorithm (D*) [7], are mainly used to solve the global optimal path planning problem from the start point to the end point on the map. The traditional mainstream technology based on behavior planning is the finite state machine(FSM), which is suitable for simple highway conditions, but cannot be used independently for complex situations. Traditional motion planning-based algorithms, such as artificial potential field(APF) [8] method and sampling planning-based rapidly-exploring random trees (RRT) [9], mainly solve the problem of trajectory generation in a short time in the future. The algorithm mentioned above is completely based on rules, which can ensure safety and reliability within rules. Different from the rule-based planning method, the decision-making method based on AI is more efficient, flexible and intelligent in solving the problem of autonomous driving, especially the deep reinforcement learning, which is based on end-to-end learning of Markov decision process, and successfully combines the high-dimensional perception of deep learning with the decision making ability of reinforcement learning. The dynamic instability of autonomous vehicles constantly shaking during driving is usually caused by the direct use of

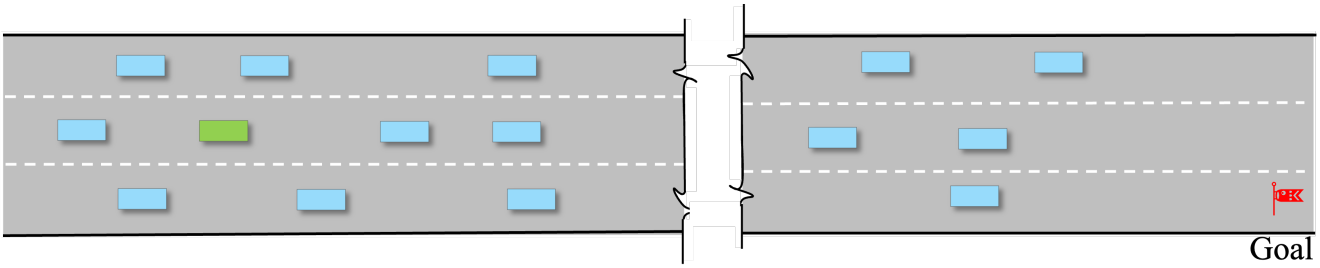



**Fig. 1.** A typical scenario of decision making in the complex highway environment.

DRL methods.

The purpose of this study is to develop a DRL-based integrated decision-making and trajectory planning framework for highway autonomous driving. This paper takes the highway scene shown in Fig. 1 as training and test cases. Our contributions are as follows:

(1) We designed the end-to-end input representation part of the vehicle, and built a multi-modal fusion network to enrich the input features;

(2) The DDPG decision network which can output continuous strategy is selected as the backbone network considering the connectivity of autonomous driving decision.

(3) The reward function is designed with the goal of high speed, safety and comfort, so as to simulate the basic situation of human driving in real scenes.

Our experimental simulations are based on the parameters of real highway scenarios, and the results show that our proposed architecture provides significant quantitative improvements.

The rest of the study is organized as follows: the "Related work" section introduces previous work on this topic. The DDPG background, our new approach to problem formulation, and details of the DDPG model are described in the "Approach" section. Simulation experiments and results are described in the "Simulation results and analysis" section. Finally, conclusions and future work are presented in the section "Conclusions and future work".

## 2. Related work

In the application scenario of autonomous driving, decision-making is also called planning, which can be divided into three types according to the task level: route planning, behavior planning and motion planning. In this section, the traditional rule-based decision planning method is first introduced, and a more flexible and intelligent learning-based method is introduced by analyzing and summarizing the limitations of its application in autonomous driving scenarios, and then the research in the field of autonomous driving is deeply analyzed.

Generally speaking, autonomous vehicle path planning algorithms fall into three categories. The first is distance navigation, also known as global planning, which plans a driving path from the starting point to the end point for the map according to the driving task of the autonomous vehicles. The second is behavior planning, which mainly solves the problem of what behavior the autonomous vehicle should do in the future (a few seconds or ten seconds). The behavior includes: turning left, turning right, changing lanes, overtaking, following and parking, etc. And all actions here are not responsible for giving details of action execution. At present, the FSM is a mainstream behavior planning method [10]. The state machine defines all possible behaviors of autonomous vehicles (overtaking, lane changing, etc.), each state is a decision on the action of the vehicle (ie, an action), and there will be jumps between states under certain conditions. The proposed behavior tree is aimed at the description of complex behavior [11]. The third is motion planning, that is, local planning, which is to plan the actual driving trajectory of the vehicle in a short period of time in the future through the real-time operating state of the autonomous vehicle and the state of the environment. In the local planning process, not only environmental risks, but also the dynamics and driving states of the vehicle must be considered. At present, there are four common methods for motion planning of autonomous vehicles: 1) combinatorial methods. This method solves global planning problems by dividing the entire road spaces into smaller spaces, planning in small space, and then connecting them. 2) potential field methods. In the autonomous driving scenario, it is assumed that there is an anti-gravity field inside each obstacle on the road. However, this method cannot find the optimal solution in some cases. 3) optimal control. This method can complete motion planning and control tasks simultaneously in the process of autonomous driving. 4) sampling-based planning methods, such as PRM [12], RRT, FMT, polynomial trajectory generation algorithm, Hybrid A*, etc. The optimal trajectory is obtained by generating multiple executable paths that satisfy the physical characteristics for the autonomous vehicle, and then the optimization algorithm performs optimization according to the cost function. The above three traditional planning algorithms have their own advantages, but they cannot flexibly cope with the complex and highly dynamic environment.

Highway driving is a complex task, and the neural network in deep reinforcement learning can transform the

entire complex driving process into a black-box model, and realize automatic driving functions through a large amount of data learning and trial-and-error exploration learning. Compared with the rule-based method mentioned above, it has a stronger ability to solve location rules, which makes up for the problem of insufficient expert knowledge coverage. Yu et al. proposed a path planning method based on deep reinforcement learning [13]. The path planning model trained by this method can directly output control action and trajectory sequences according to sensor data and old actions. Du et al. used deep reinforcement learning to carry out trajectory planning for automatic parking of intelligent vehicles, and the parking success rate exceeded 90% [14]. Loiacono et al. proposed an overtaking strategy for autonomous vehicles using Deep Q Network (DQN), which took into account the overtaking decision of vehicles in two situations, namely, a big straight turn or a sharp turn [15]. The results showed that the algorithm could make a reasonable overtaking decision. Sun et al. proposed a deep deterministic policy gradient (DDPG) algorithm for determining vehicle cruise targets on roads with different curvature for heavy commercial vehicles under adaptive cruise conditions [16]. According to the simulation data, Zheng et al. established an automatic driving decision system based on reinforcement learning, modeled the vehicle decision process as markov decision process (MDP), and used reinforcement learning to improve the performance of MDP [17]. The simulation test in high-speed loop environment shows the effectiveness of this method.

Although the automatic driving decision-making system based on deep reinforcement learning has been able to solve some decision-making problems in driving scenarios, these decision-making planning methods are more supplements and improvements to the rule-based decision-making methods. However, the performance of the decision-making model based on deep reinforcement learning is greatly affected by the sample situation and algorithm structure. If the sample quality is low, the decision-making model will have insufficient understanding of the scene and cannot make effective decisions. If the algorithm is too complex, the decision time will be too long and the training will be complicated, and the final decision model cannot meet the limit on the decision time. Especially for complex scenes such as highways, effective samples and low-complexity algorithms are of great significance for high-speed, safe, and comfortable autonomous driving.

## 3. Approach

### 3.1. Background of reinforcement learning

Reinforcement learning, as a kind of machine learning, is used to describe and solve the problem that an agent uses learning strategies to maximize returns or achieve specific goals in the process of interacting with the environment. Reinforcement learning problems are usually modeled as Markov Decision Processes (MDP), in which A tuple $(S,A,P,R,\gamma)$ with measurable states $S$, action space $A$, unknown reward function $R \in \mathbb{R}^{S\times A}$ and unknown dynamics $P \in \mathscr{P}(S)^{S\times A}$ is used to solve MDP. The goals of reinforcement learning is to maximize the future discounted return defined as follows

$$R(s_t,a_t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \quad . . . . . (1)$$

where $r_{t+k}$ is the reward at step $t+k$.

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=0}^{\infty} R(s_t,a_t)] \quad . . . . . (2)$$

where $s_0 = s$, $a_t \sim \pi(a_t|s_t)$, $s_{t+1} \sim P(s_{t+1}|s_t,a_t)$.

$$Q^{\pi} = R(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim P(s'|s,a)} V^{\pi}(s') \quad . . . . . (3)$$

The optimal action-value function $Q^* = max\, Q^{\pi}(s)$ satisfies the Bellman Optimality Equation:

$$Q^*(s,a) = \mathop{\mathbb{E}}_{s'\sim P(s'|s,a)} max[R(s,a) + \gamma Q^*(s',a')] \quad (4)$$

### 3.2. Problem formulation

**State-representation based on multimodal fusion.** In automatic driving task, state representation is usually used as the input of reinforcement learning algorithm. Usually, a state space $S$, namely the scene where the autonomous driving vehicle is located, is selected as the state representation. There are many ways to describe the state of autonomous vehicles and the surrounding environment (other vehicles), among which the most widely used is to describe the continuous position, direction and speed of nearby vehicles and their own vehicles (autonomous vehicles) in the process of driving on the road. By establishing the attribute list of all vehicles as state representation:

$$S = (s_i)_{i\in[0,N]} \quad . . . . . (5)$$

where $s_i = [x_i\ y_i\ v_i^x\ v_i^y\ \cos\psi_i\ \sin\psi_i]^T$, $i$ is the total number of vehicles in the observation area of the ego-vehicle.

The property known as state feature list, as shown in Fig. 1. It expounds on the driving scenarios needed a minimum quality characteristic information characterization of due to a lack of more abundant information, however, there are two problems with the ultimate autonomous decision-making model first, the default of slow convergence speed in the process of learning strategies, take longer; Second, due to the insufficient learning of the decision model, the comfort level of the safe speed in the driving process of the car cannot be guaranteed.

**Action Space.** The action space applied by deep reinforcement learning in autonomous driving is mainly divided into two categories: discrete action space and continuous action space. Discrete action space is defined as discrete action space on the one hand, which is convenient for fast vehicle control, and on the other hand, it is applicable for the use of discrete deep reinforcement learning algorithms, such as DQN. The control actions defined for

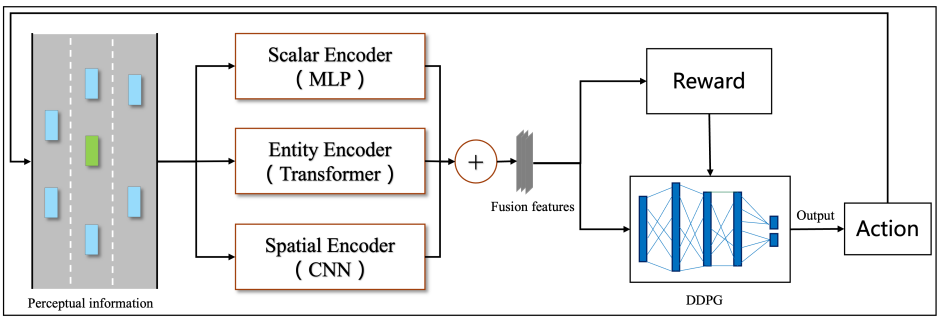



**Fig. 2.** Structure of the decision making approach of this study.

the vehicle at each time step are divided into lateral action space and longitudinal action space. Generally, the definitions are as follows:

$$A_{longi} = \begin{cases} 0, \text{ Slower} \\ 1, \text{ Faster} \\ 2, \text{ Keep} \end{cases} \quad \ldots\ldots\ldots\ldots \quad (6)$$

$$A_{lat} = \begin{cases} 0, \text{ Lane left} \\ 1, \text{ Lane right} \end{cases} \quad \ldots\ldots\ldots\ldots \quad (7)$$

Although discrete action space provides a lot of convenience in autonomous driving decision-making tasks, the actions obtained cannot be applied to real-world driving scenarios. Therefore, in this study, agent is allowed to directly set the low-level controls of the vehicle, namely the throttle $a$ and steering Angle $\delta$.

**Reward.** The choice of an appropriate reward function that yields realistic optimal driving behavior is a challenging problem. The setting of the reward function not only needs to consider the needs of real driving scenarios, but also needs to be simple and direct, so that the entire decision-making model can be sufficiently robust by learning enough behaviors. In this study, the reward function is mainly set directly according to human driving needs in the real world. In real highway scenarios, autonomous vehicles must meet safety, efficiency, and comfort characteristics. Therefore, the reward function is defined as:

$$R(s,a) = a\,\frac{\upsilon - \upsilon_{min}}{\upsilon_{max} - \upsilon_{min}} - b\,Collision + c\,Step \quad (8)$$

where $\upsilon, \upsilon_{min}, \upsilon_{max}$ are the current, minimum and maximum speed of the ego-vehicle respectively, $a$, $b$ and $c$ are three coefficients.

### 3.3. Network design and training details

In this study, the network structure is mainly divided into two parts: the policy network and the value network. The network structure is mainly composed of linear layers and activation layers (Relu, Tanh), as shown in Fig. 3. The input of the policy network is multi-modal representation information, and the output is composed of lateral control and longitudinal control. The lateral control is mainly responsible for the heading of the vehicle, and the longitudinal control is mainly responsible for the acceleration control of the vehicle. The value network is used to judge the value of the output action of the policy network.

**Table 1.** Training hyperparameters of the DDPG agent.

| | |
|---|---|
| Discount factor, $\gamma$ | 0.99 |
| Replay memory size, $M_{replay}$ | 8000 |
| Learning rate(PolicyNet), $\eta_1$ | 0.0004 |
| Learning rate(ValueNet), $\eta_2$ | 0.005 |
| Maximum batch size, $M_{max}$ | 128 |
| Target network update frequency, $F_{update}$ | 0.02 |

**Table 2.** Simulation parameters.

| | |
|---|---|
| Simulation road length, $L$ | 1000 |
| Number of lanes, $N$ | 4 |
| Speed limitatione | [25,45] |
| Simulation frequency, $F_s$ | 15 |
| Policy frequency, $F_p$ | 5 |
| Acceleration range, $[-a,+a]$ | [-5, 5] |
| Emergency deceleration, $-a_{max}$ | -5 |
| Other vehicle, $N_{other}$ | 500 |

The hyperparameters of DDPG are shown in Table 1. Among them, the future reward in the reward function is decayed by a discount rate of size $\gamma$. The network trains the model by randomly sampling the batch size from the replay memory not exceeding the maximum batch size, and the network uses the Adam optimizer with a learning rate $\eta$ to update the model parameters. The target network in the DDPG network is also divided into a policy target network and a value target network, and the soft update between the target network and the training network is completed with an update frequency of $N_{step}$.

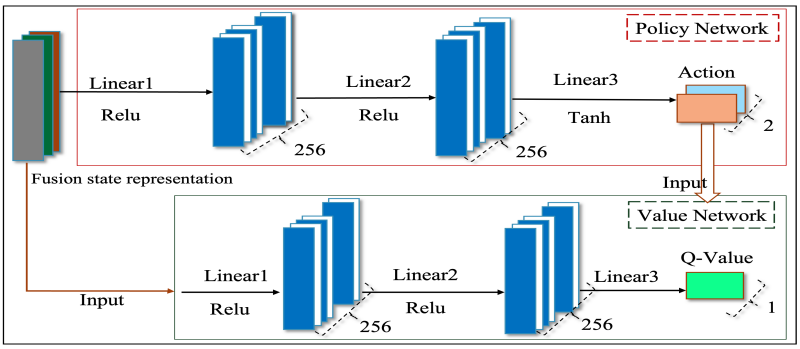



**Fig. 3.** DDPG structure applied in this study.

## 4. Simulation results and analysis

In this section, we first introduce the simulation environment and analyze the characteristics of the simulation environment. Then, the performance of the single-modal feature and the multi-modal feature as the network input to train the DDPG model to reach the destination in terms of success rate and average speed are analyzed.

### 4.1. Simulation environments

In this study, we use the four-lane one-way high-speed scene shown in Fig. 1 as the simulation environment, in which the vehicle is required to reach the red mark of the rightmost lane in the figure, and the distance from the start of the vehicle to the target point for $L$. The driving speed of the vehicle in the simulation environment is completely set according to the speed limit in the actual high-speed scene, and obeys ($V_{min}$, $V_{max}$). When the environment is initialized, the vehicle and other vehicles are randomly generated on the map, and the positions of each vehicle do not overlap. During the learning process, the vehicle adopts the algorithm formulated in this paper to drive, while other vehicles adopt the non-intelligent method, that is, keep a constant speed and driving direction during the driving process.

### 4.2. Overall performance of the algorithm

In this study, the purpose of the vehicle is to quickly reach the end point without collision, so the success rate of reaching the end point is defined as the evaluation index of the vehicle's performance. The formula for the specific success rate is as follows:

$$rate = \sum_{n-k}^{n-1} \frac{I_n(togoal == True)}{k} \quad \cdots\cdots \quad (9)$$

where, $n = k+1, k+2, \ldots$, $I_n(togoal == True)$ denotes whether the ego-vehicle exits the off-ramp in the nth episode with 1 for success and 0 for failure. $k = [1, 2, \ldots]$, used to obtain the average value of the last $k$ times the vehicle has reached the end point.

The original curve of the distance traveled by the ego-vehicle during the training process is shown in Fig. 4. By

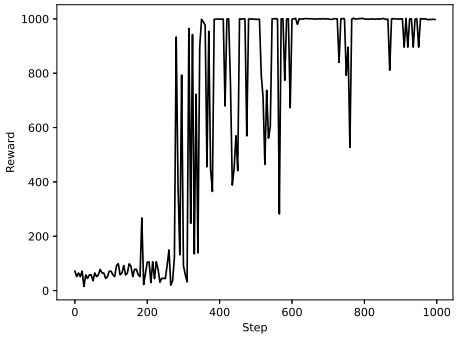



**Fig. 4.** The change curve of the reward during the training period of the improved DDPG network (steptime=5).

observing the curve, it can be found that as the number of iterations increases, the distance traveled by the vehicle is increasing. The distance continues to decrease. When $step = 550$, the ego-vehicle reaches the end point for the first time. When the step is after 1550, the ego-vehicle drives out of the end point stably, which means that the success rate of the car is stable at 100%. During the training process, the change curve of the reward value of the ego-vehicle is shown in Fig. 5. From the figure, we can observe that the change curve of the reward function of the vehicle is proportional to the change curve of the driving distance, which also shows that the reward has an obvious incentive effect for the vehicle to successfully drive out of the destination.

Through ablation experiments, the success rates and average speeds of different models are shown in Table 3. By comparison, it can be found that the average speed of the vehicle after adding the internal reward mechanism is increased by 0.6% compared with the introduction of multi-modal fusion representation, and it is also increased by 13.4% compared with the original DDPG decision model. Therefore, designing an internal reward mechanism for autonomous vehicles has a greater impact on decision-making than the introduction of multimodal fusion representations. Finally, after adding multi-modal features, the success rate of this car is close to 100%. Finally, through

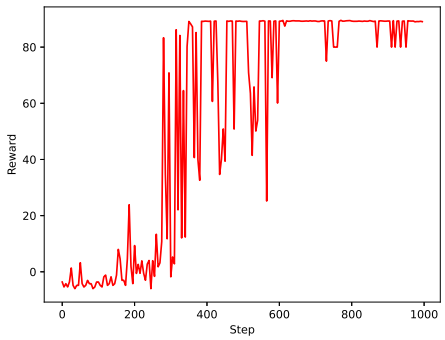



**Fig. 5.** The change curve of the distance between the ego-vehicle and the destination during the training of the improved DDPG network (steptime=5).

**Table 3.** The average success rate($rate_{aver}$) and average speed($v_{aver}$) test results of the ego-vehicle to the end point (steptime=1000).

| model | $rate_{aver}(\%)$ | $v_{aver}(km/h)$ |
|---|---|---|
| DDPG | 86.1 | 25 |
| + internal reward | 99.5 | 35 |
| + multimodal fusion | 98.9 | 32 |
| ours | 100.0 | 36 |

the speed analysis of different models, it is found that after adding internal mechanisms and multi-modal representation states, the vehicle has learned a decision-making method for efficient driving.

## 5. Conclusion and future work

In this paper, we develop a multimodal fusion decision architecture based on a deep reinforcement learning approach for complex highway environments. To the best of our knowledge, this is the first DRL-based decision planning architecture with multimodal fusion and an internal reward mechanism. In this work, we propose a DDPG solution. The proposed actor-critic architecture combines value-based and policy-based features, which can solve the continuous action space problem in autonomous driving tasks. The situational awareness of autonomous vehicles is enhanced by the design of fused representations as input to the DDPG network. At the same time, an internal reward mechanism is designed to improve the reward sparsity problem of the vehicle in the learning process. Therefore, the proposal of this architecture solves the problems of insufficient awareness and incentives of the driving environment for autonomous vehicles, and is of great significance to improving road safety and efficiency. There is still some work to be done in the future. For example, the number of parameters of the decision-making model has been increased due to the introduction of the multi-modal fusion network, which reduces the efficiency of the algorithm. In the future, we will consider designing a lighter fusion network. In addition, although more flexible and intelligent than traditional decision-making methods, deep reinforcement learning methods are not interpretable due to the existence of neural networks. A rule-based approach is considered to design interpretable decision-making architectures in the future, and DRL is used as an auxiliary decision-making tool.

**Acknowledgements**

This work was partially funded by the National Key Research and Development Plan of China (No.2018AAA0101000) and the National Natural Science Foundation of China under grant 62076028.

**References:**

[1] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.

[2] Huanjie Wang, Shihua Yuan, Mengyu Guo, Ching-Yao Chan, Xueyuan Li, and Wei Lan. Tactical driving decisions of unmanned ground vehicles in complex highway environments: A deep reinforcement learning approach. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(4):1113–1127, 2021.

[3] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, and David Stavens. *Stanley: The Robot That Won the DARPA Grand Challenge*, pages 1–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[4] Mihail Pivtoraiko and Alonzo Kelly. Fast and feasible deliberative motion planner for dynamic environments. 2009.

[5] C. Fernández, F.J. Espinós, M.C. López, F.J. García-Diego, and C. Cervera. Representation of a mathematical model to predict methane output in dairy goats. *Computers and Electronics in Agriculture*, 91:1–9, 2013.

[6] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[7] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3310–3317 vol.4, 1994.

[8] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505, 1985.

[9] Steven Lavalle and James Kuffner. Rapidly-exploring random trees: Progress and prospects. *Algorithmic and computational robotics: New directions*, 2000.

[10] Paden, Brian, Čáp, Michal, and Yong. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.

[11] Colledanchise, Michele, and Petter Ogren. *Behavior Trees in Robotics and AI: An Introduction*. 2018.

[12] Dolgov, Dmitri, Thrun, and Sebastian. Practical search techniques in path planning for autonomous driving. *AAAI Workshop - Technical Report*, 2008.

[13] Lingli Yu, Xuanya Shao, Yadong Wei, and Kaijun Zhou. Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning. *Sensors*, 18(9), 2018.

[14] Du, Zhuo, Miao, and Qiheng. Trajectory planning for automated parking systems using deep reinforcement learning. *International Journal of Automotive Technology*, 21:881–887, 08 2020.

[15] Loiacono, Daniele, Prete, and Alessandro. Learning to overtake in torcs using simple reinforcement learning. pages 1 – 8, 08 2010.

[16] Sun, Ming, Zhao, and Weiqiang. Ddpg-based decision-making strategy of adaptive cruising for heavy vehicles considering stability. *IEEE Access*, PP:1–1, 03 2020.

[17] Rui Zheng, Chunming Liu, and Qi Guo. A decision-making method for autonomous vehicles based on simulation and reinforcement learning. In *2013 International Conference on Machine Learning and Cybernetics*, volume 01, pages 362–369, 2013.