Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/158/2025.TJ23484

Optimization of Automatic Driving Safety Strategy Based on Reinforcement Learning Algorithm of PPO

Guanxu Bai

School of Advantage Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
Guanxu.Bai22@Student.xjtlu.edu.cn

Abstract: With the development of the automobile industry and artificial intelligence, autonomous driving is an important research topic and the future development trend, However, there are still defects in the decision-making ability of autonomous driving in the informed environment and the safe driving ability in complex environments. In order to solve this problem, based on the Proximal Policy Optimization (PPO) strategy of reinforcement learning, this study proposes two novel algorithms: Soft-constrained PPO and Hard-constrained PPO to optimize the policy of safe reinforcement learning. Soft constraints mean that by introducing new assessment criteria, the reward function is modified. The hard constraint is to force the unsafe training to stop by setting the maximum risk control threshold. After giving the algorithm, a comparative experiment is carried out, and the three models are trained in the same environment (highway-V0). It is found that the new proposed algorithm not only improves the performance, but also effectively controls the unsafe behaviors in the autonomous driving environment, such as lane deviation and collision.

Keywords: Reinforcement Learning, Automatic Driving, Proximal Policy Optimization

1. Introduction

In recent years, with the rapid improvement of the automotive industry and artificial intelligence, automatic driving algorithms based on reinforcement learning have been widely reached and used. For example, Automatic Braking Systems (ABS), Lane-Keeping Systems (LKS), and Adaptive Cruise Control (ACC) are already used in driver assistance systems [1]. There are four important modules in autonomous driving, which are perception, planning, decision making and control [2]. The logic of the autonomous driving algorithm is that the vehicle analyzes the scene and environment through the information obtained by external sensors, such as GPS positioning and radar, and controls the specific modules of the car, such as the braking system, accelerator, transmission, etc., to realize the driving behavior of the car, including forward, reverse, lane change, overtaking, parking, etc [3][4]. The safety of autonomous driving is an important evaluation condition to consider the performance of algorithms, because it is directly related to the personal safety of drivers and passengers and their degree of trust in the autonomous driving system.

Most of the existing autonomous driving algorithms use reinforcement learning methods for training. Reinforcement learning is an important branch of machine learning, and it is very close to human thinking [5]. In general, reinforcement learning is the process of using an agent, letting it interact with the environment, adjusting the next action based on feedback, and finally getting the optimal solution. Reinforcement learning has led to remarkable achievements in a variety of domains,

DOI: 10.54254/2755-2721/158/2025.TJ23484

including economics [6], computer games [7], board games [8], linguistics [9] and robotics [10][11]. In reinforcement learning, in order to improve the safety performance of autonomous driving, the Safe reinforcement learning (safe RL) algorithm is defined to ensure that the agent can operate within a safe range [12].

Today's autonomous driving algorithms still have flaws. Firstly, most of the experimental data is trained from static data sets, which affects the ability to adapt to dynamic environments and continuous learning to some extent [13]. However, in the complex dynamic environment in reality, the existing autonomous driving algorithms are not enough to completely replace the role of the driver [14]. The decision difficulty in the simulation environment is very different from that in the high-speed driving state.

In this research paper, in order to improve the safety performance in autonomous driving, this study designed two safety reinforcement learning algorithms based on the Proximal Policy Optimization (PPO) algorithm, namely soft constraint PPO and hard constraint PPO, and designed a control experiment to test the performance of the algorithms.

The structure of this paper is organized as follows: after giving the introduction, the second paragraph expounds on the basic theory of reinforcement learning, including the Markov decision process and Markov Decision Processes (MDP) based PPO algorithm logic. In the third paragraph, the methodology of the experiment was given, including the algorithm logic of soft constraint PPO and hard constraint PPO. Then, the next paragraph gives the experimental design, experimental conclusions and data analysis, and finally gives the conclusion.

2. Related work

2.1. Markov decision processes

In reinforcement learning, Markov decision processes are used with high frequency [15]. A five-tuple (S, A, P, R, γ) can be used to describe a Markov decision process, respectively: S is a finite set representing the possible states in all environments. A is a finite set of actions representing all possible actions of the agent. $P(s' \mid s, a)$ is a state transition probability function that illustrates the likelihood of moving from state s to state s' after taking an action a. R(s, a) stands for reward function, providing the immediate feedback received after performing an action in the given stage. $R \in [0,1)$ is the discount factor, which determines the persent value of future rewards.

The decision process of MDP is dynamic, in the initial state, the agent chooses an action in A to execute, after execution, it obtains state S1 according to the state transition equation, and then continues to execute the next action, and repeats [16]. The long-term objective function is defined as follows.

$$J(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(S_t, a_t)] \quad (1)$$

Markov decision processes (MDP) provide the theoretical foundation for mathematical formalization and provide the modeling structure for many policy optimization algorithms, such as Q-Learning and PPO.

2.2. Proximal policy optimization

The PPO algorithm belongs to one of the policy gradient methods, and its powerful function has been widely recognized and used as one of the benchmark algorithms in reinforcement learning [17]. Compared with the traditional policy gradient algorithm, PPO has an efficient and stable update mechanism, which can avoid the trap of large-scale and unstable policy update.

DOI: 10.54254/2755-2721/158/2025.TJ23484

The core idea of PPO is to restrict the policy update within a "proximal" region, avoiding drastic changes that could lead to performance collapse. Rather than using complex trust region constraints, PPO introduces a clipped surrogate objective function:

$$L^{clip}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\overline{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\overline{A}_t)] \quad (2)$$

Here, the $r_t(\theta) = \frac{\pi_{\theta(a_t|s_t)}}{\pi_{\theta_{old}(a_t|s_t)}}$ is the probability radio between the new policies and the old, and is a function of an estimator of the advantage, reflecting the relative quality of an action at a given stage. The chip function is used to prevent the ratio from deviating too much from 1, aiming to ensure a conservative and stable policy.

PPO is often used in conjunction with Generalized Advantage Estimation (GAE), which provides a trade-off between bias and variance in advantage computation:

$$\overline{A}_t = \sum_{l=0}^{T-t}(\gamma\lambda)^l \delta_{t+l}, where\ \delta_t = r_t + \gamma V(S_{t+1}) - V(S_t) \quad (3)$$

3. Methology

Considering that in real-world high-risk task environments, such as autonomous driving, intelligent navigation, etc., optimizing the original reward function is not enough to ensure sufficient safety of the policy. In the case of high task complexity or insufficient exploration period, the agent may obtain high rewards by finding shortcuts, resulting in serious security risks.

In order to solve this problem, this paper designed two different types of security constraint mechanisms based on the PPO framework, namely soft constraint PPO policy and hard constraint PPO policy.

3.1. Soft-Constrained PPO

For Soft-Constrained PPO policy, by modifying the form of the reward function, the unsafe behavior (e.g., collision, lane departure) is added to the total reward in the form of a penalty term. To be specific, Define the environment raw reward as R_base, The number of collisions as N_col, The number of deviations as N_dev, The penalty weights are set to be, respectively are λ_c and λ_d, and the fusion coefficient $\alpha \in [0, 1)$, so that the fixed reward function is shown as:

$$R\_total = (1-\alpha) \times R\_base - \alpha \times (\lambda\_c \times N\_col + \lambda\_d \times N\_dev) \quad (4)$$

Here, α controls the weight between the task goal and the safety penalty. A small value of α encourages reward for the task, while a large value of α favors safe behavior. The reward function will be dynamically calculated at each time step as the basic reward signal in the optimization of the PPO policy.

This algorithm does not need to modify the training environment, only by introducing a new penalty structure at the reward function calculation, it can realize the constraints on safe behavior. The algorithm is expressed as follows:

Algorithm 1: Soft-Constrained PPO

Input: Environment env, initial policy πθ, value function Vϕ,
balance factor α ∈ [0,1], collision penalty λc, deviation penalty λd,
total training iterations N

Output: Trained policy πθ

DOI: 10.54254/2755-2721/158/2025.TJ23484

```
Initialize πθ and Vϕ
for iteration = 1 to N do
    Collect trajectories { (st, at, rt, st+1) } using policy πθ
    for each episode in trajectories do
        Count safety indicators:
            n_col ← number of collisions
            n_dev ← number of deviations
        for each transition (st, at, rt, st+1) do
            r_base ← rt
            safety_penalty ← λc * n_col + λd * n_dev
            r_total ← (1 - α) * r_base - α * safety_penalty
        end for
    end for
    Estimate advantages Ât using GAE and r_total
    Update policy πθ via PPO objective with reward r_total
    Update value function Vϕ using MSE loss
end for
```

3.2. Hard-Constrained PPO

Compared with Soft-PPO, the hard-constrained policy adopts a more rigid control approach. Firstly, a safety threshold is set for the environment. Once the number of collisions or deviations in a certain episode exceeds the safety threshold, the environment will immediately terminate the current round and impose additional negative penalties, thereby explicitly punishing unsafe behaviors. This approach directly sets a hard boundary for the exploration space of the training policy and is suitable for scenarios with strict safety requirements, such as autonomous driving.

Here, the maximum number of allowed collisions is defined as C_max, the maximum allowable number of deviations is defined as D_max, once the number of times in the current episode exceeds the threshold, the code to stop the environment is triggered. This strategy may inhibit part of the behavior of the agent in the early stage, but in the long run, it can greatly reduce the frequency of serious unsafe behavior, which is more practical in high-risk tasks. Here's a pseudo-code demonstration of the algorithm:

Algorithm 2: Hard-Constrained PPO

```
Input: Environment env, initial policy πθ, value function Vϕ,
       collision threshold C_max, deviation threshold D_max,
       penalty constants η_col, η_dev, total training iterations N

Output: Trained policy πθ
Initialize πθ and Vϕ
for iteration = 1 to N do
    Collect trajectories using policy πθ
    for each episode in trajectories do
        Initialize counters: n_col ← 0, n_dev ← 0
        for each timestep t in episode do
            Take action at ~ πθ(st), observe rt, st+1
            if collision occurs then
                n_col ← n_col + 1
            end if
            if deviation occurs then
```

DOI: 10.54254/2755-2721/158/2025.TJ23484

```
            n_dev ← n_dev + 1
        end if

        if n_col ≥ C_max or n_dev ≥ D_max then
            if n_col ≥ C_max then
                rt ← rt − η_col
            end if
            if n_dev ≥ D_max then
                rt ← rt − η_dev
            end if
            Mark episode as terminated
            break
        end if
    end for
end for
Estimate advantages Ât using GAE
Update policy πθ using PPO clipped surrogate objective
Update value function Vϕ using MSE loss with modified returns
end for
```

4. Experiment

4.1. Experimental design

In order to verify the effectiveness of the proposed safety constraint reinforcement learning method, this study established a unified experimental platform in a multi-lane automatic driving simulation environment, and systematically compared three algorithms, namely the standard PPO algorithm, the hard constraint PPO algorithm and the soft constraint PPO algorithm.

The simulation environment used in this experiment is the highway-v0 scenario provided by highweay-env, a highly dynamic simulation platform for intelligent transportation research, which is able to simulate the behavior of multiple vehicles driving coactively on highways. In this experiment, the agent is the master vehicle that is trained, and the rest of the vehicles interfere in the form of basic control mode. This environment has clear sources of risk, such as vehicle speed differences, vehicle action uncertainty, and vehicle density.

The state space employs the "Kinematics" mode inherent in the environment. The observed state at each moment is a two-dimensional tensor of shape (10, 7), respectively indicating the kinematic information between the main vehicle and the nine vehicles ahead, including existence flags, coordinate positions, velocity vectors, and the cosine and sine of the direction angles. For adaptation to neural network modeling, this tensor is flattened into a 70-dimensional vector through a custom feature extractor and then input into the model. The action space is a discrete set, encompassing five fundamental strategic actions: maintaining, accelerating, decelerating, changing lanes to the left, and changing lanes to the right. Such a configuration is concise and lucid, facilitating the comparison of the selection differences of different strategies within the same strategic space.

Based on this, PPO-Baseline will use the comparison group as a reference to observe the performance of the algorithm without security constraints. PPO-Soft adjusted the reward function by adding a penalty term, and PPO-HARD forced the termination behavior on unsafe trajectories by setting the maximum number of allowed collisions. Each algorithm adopts a unified network structure and training configuration, and controls variables to ensure fairness. Due to the problem of environment configuration, our strategy has two minor modifications, but they do not affect the impact of the strategy on safe driving. Firstly, the experiment adds a safe driving reward to all strategies to prevent falling into the trap of the optimal strategy. Second, deviation are not considered.

DOI: 10.54254/2755-2721/158/2025.TJ23484

The convergence performance of the policy in reward was recorded at each stage of training, while considering the average reward, the average length, and the final visualization results were generated for analysis.

4.2. Experimental design

Aiming to ensure the stability of the training process and reproducibility of the experimental results, all experiments are conducted with a dixed random seed (SEED = 42). Additionally, a CUDA-compatible GPU is used to accelerate training if available.The core hyperparameters and environment configurations are summarized in Table 1.

Table 1: Experimental setup

Algorithm settings	
Reinforcement learning algorithm	PPO
Total training steps	35000
Discount factor(γ)	0.98
Policy network architecture	Two fully connected layers with 128 and 64 units respectively, using ReLU activation
Learning rate	3e-4
Batch size	256
Rollout steps per update	1024
Number of training epochs per update	20
Environment Configuration(highway-v0)	
Number of lanes	3
Vehicle density	1.2
Episode duration	100 simulation steps
40 simulation steps	Kinematics, including seven features for each vehicle (presence, x, y, vx, vy, cos(heading), sin(heading))
Collision penalty	-3.0
Right lane incentive	+0.2
High-speed driving incentive	+0.8
Lane change penalty	-0.05
Target speed range	25–33 m/s
Off-road termination	Enabled (True)
PPO-Soft	
Collision penalty coefficient	3.0
Lateral deviation penalty coefficient:	1
Reward blending coefficient	$\alpha = 0.3$
Reward adjustment formula	$R = (1 - \alpha)\cdot r - \alpha\cdot(\text{collision\_penalty} + \text{deviation\_penalty})$
PPO-Hard	

DOI: 10.54254/2755-2721/158/2025.TJ23484

Table 1: (continued)

Maximum allowed collisions:	1
Rewards of collisions	-10
Rewards of no unsafy activity	3

4.3. Experimental result

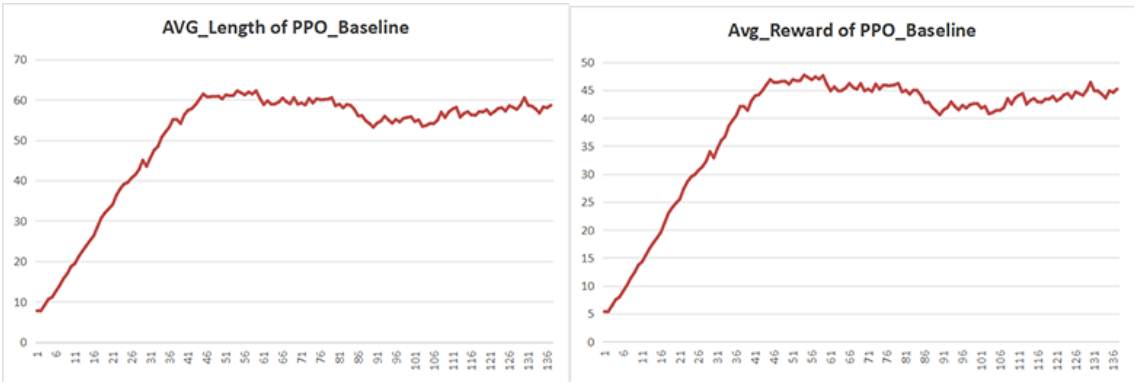

Figure 1: Result of PPO-Baseline (picture credit: original)

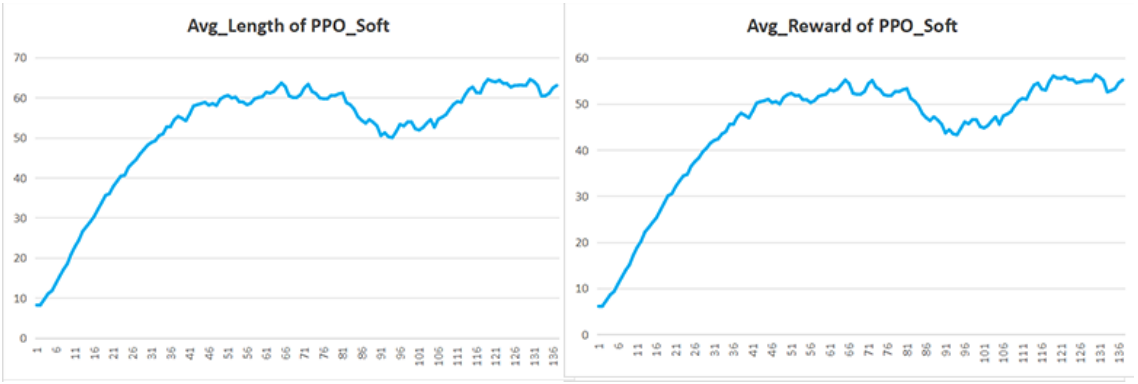

Figure 2: Result of PPO-Soft (picture credit: original)

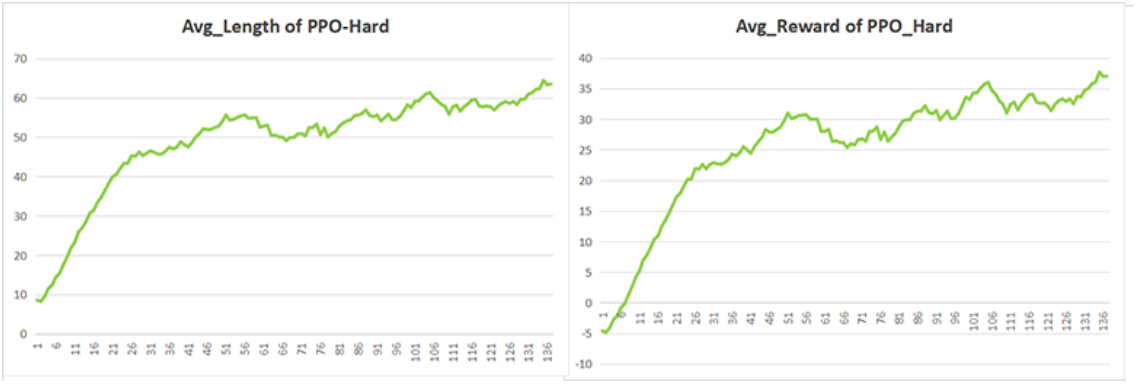

Figure 3: Result of PPO-Hard (picture credit: original)

DOI: 10.54254/2755-2721/158/2025.TJ23484

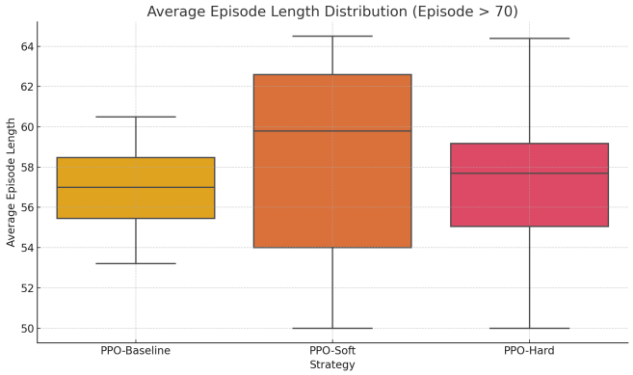

Figure 4: Average episode length distribution (episode>70) (picture credit: original)

In order to comprehensively evaluate the influence of different security strategies on the training performance and security of reinforcement learning, this study conducted simulation experiments on the agreement parameters of the three strategies respectively. Figure 1, Figure 2 and Figure 3 shows the comparison of the trend of average rewards and average length of each training steps(including 256 episode). Figure 4 reflects the average step size comparison of each strategy after the data is close to stability.

As for the convergence curve, the three strategies all show good convergence, which proves that the three strategies have successfully completed the training. On this basis, there is a significant difference in the performance of the three strategies. Specifically, the reward value of PPO-Baseline starts low and shows a slow upward trend, converging to around 45. This proves that it can still learn basic high-speed driving behavior under the premise of violating safety constraints. However, it has the disadvantages of large fluctuation of reward, and poor stability. For the PPO-Soft strategy, its reward mean value is higher than the Baseline, about 50, while maintaining good convergence. The training curve reaches a stable state in the middle period, showing stronger strategy expression and exploration ability. This shows that under the condition of moderately introducing soft constraints, the agent can improve the overall policy benefit on the basis of ensuring a certain degree of security. Finally, for the PPO-Hard policy, the data showed a trend of rapid growth at the beginning, slow down in the later period, but continued to rise. The rewards value of PPO-Hard increased from negative to positive because we added a very strict safety constraint, which made it faster to avoid unsafe behaviors through a higher penalty mechanism. Due to the extremely high penalty, its reward function data is significantly lower than the other two groups, but the average step size is higher than the other two groups in the later training stage. This shows that the algorithm can first quickly reduce the unsafe behavior through a strong penalty mechanism, and then continue to explore on this basis.

Because of the unique design of the algorithm, REWARDS allows us to see how much each value collids and shifts. For PPO-Baseline, its reward is greater than that of the constrained strategy at the beginning because there is no additional constraint, but in the later stage, even without additional punishment mechanism, its reward function is still lower than Baseline, which proves that there are more unsafe behaviors in PPO-Baseline. The Rewards of Hard-PPO mechanism increase rapidly, which means that the number of unsafe behaviors is decreasing rapidly.

Finally, Figure 4 is the average step size distribution image of the three functions after stabilization. It can be clearly seen that the average step size of PPO-Soft and PPO-Hard strategies is higher than that of PPO-Baseline strategy, and they have higher exploration ability because of larger extreme values. Due to the stronger constraint ability, the data of PPO-Hard strategy is closer, with less fluctuation and more concentrated data. The upper quartile of PPO-Soft is closer to the median and higher, which proves that PPO-Soft has good performance on the basis of secure behavior, although it has a slightly unstable factor, because its lower quartile is the lowest of all the policies.

DOI: 10.54254/2755-2721/158/2025.TJ23484

5. Conclusion

In general, through experiments, this study proves that the two constraint algorithms given: the PPO soft constraint algorithm and the PPO hard constraint algorithm can realize the function of security policy optimization on the basis of improving the performance of the model. Specifically, PPO-Hard has better stability and training feedback, and PPO-Soft reaches the stationary state earlier, and have a strong ability to explore upward, which represents that it has a stronger exploration ability. The introduction of these safety policies significantly improves the agent's exploration ability and strengthens its robustness, two characteristics that are very important for autonomous driving. Still, there are some limitations. First of all, the highway simulation environment configured in this experiment has fixed parameters, such as the number of lanes and vehicle density, in order to control the experimental variables, but it will cause disadvantages that are difficult to reflect the complexity of the comprehensive real traffic situation, such as dynamic traffic flow changes, sensor noise interference, etc. Secondly, the reward function of this experiment is designed manually, focusing on indicators such as speed control, lane keeping and safety behavior, which is the underlying logic of autonomous driving. If it is to be implemented in the real world, a more complex design mechanism is needed. Third, due to the limitation of hardware facilities, the experimental step size is set within the capacity, and more training times can be performed if a better experimental configuration is available. In general, the results of this experiment reach the expected goal, and the safe automatic driving optimization strategy of reinforcement learning based on PPO is successfully designed. Although there are certain limitations, it has made a theoretical basis for subsequent research. Additionally, Based on the logic of soft and hard constraints, researchers can set more safety behavior constraints according to this strategy to improve their reinforcement learning training for autonomous driving in other environments. As the inevitable development trend of the automobile industry in the future, the improvement of safety is crucial if people want to achieve better development. At the same time, as the artificial intelligence learning method closest to human behavioral habits, reinforcement learning also has high development space and potential in the future.

References

[1] *Naveen, A. M., Ravish, R., & Swamy, S. R. (2022). Distributional reinforcement learning for automated driving vehicle. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) (pp. 1–6).*

[2] *Liu, T., Tian, B., Ai, Y., Chen, L., Liu, F., & Cao, D. (2019). Dynamic states prediction in autonomous vehicles: Comparison of three different methods. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3750–3755).*

[3] *Liu, T., Tian, B., Ai, Y., Chen, L., Liu, F., & Cao, D. (2019). Dynamic states prediction in autonomous vehicles: Comparison of three different methods. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 3750–3755).*

[4] *Liao, J., Liu, T., Tang, X., Mu, X., Huang, B., & Cao, D. (2020). Decision-making strategy on highway for autonomous vehicles using deep reinforcement learning. IEEE Access, 8, 177804–177814.*

[5] *Jia, J., & Wang, W. (2020). Review of reinforcement learning research. In 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 186–191). IEEE.*

[6] *Avramelou, L., Nousi, P., Passalis, N., Doropoulos, S., & Tefas, A. (2023). Cryptosentiment: A dataset and baseline for sentiment-aware deep reinforcement learning for financial trading. In 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) (pp. 1–5).*

[7] *Sure, E. K., & Wang, X. (2022). A deep reinforcement learning agent for general video game AI framework games. In 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 182–186).*

[8] *Li, M., & Huang, W. (2020). Research and implementation of Chinese chess game algorithm based on reinforcement learning. In 2020 5th International Conference on Control, Robotics and Cybernetics (CRC) (pp. 81–86).*

[9] *Yun, L. (2024). Research on dynamic adaptation of English translation based on deep reinforcement learning. In 2024 International Conference on Artificial Intelligence, Deep Learning and Neural Networks (AIDLNN) (pp. 238–241.*

DOI: 10.54254/2755-2721/158/2025.TJ23484

[10] *Singh, A., Chiu, W.-Y., Manoharan, S. H., & Romanov, A. M. (2022). Energy-efficient gait optimization of snake-like modular robots by using multiobjective reinforcement learning and a fuzzy inference system. IEEE Access, 10, 86624–86635.*

[11] *Guan, H. (2020). Analysis on deep reinforcement learning in industrial robotic arm. In 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI) (pp. 426–430).*

[12] *Kulkarni, S., & Patil, D. D. (2025). Reinforcement learning for autonomous systems. In 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL) (pp. 816–820).*

[13] *Zhao, R., Li, Y., Fan, Y., Gao, F., Tsukada, M., & Gao, Z. (2024). A survey on recent advancements in autonomous driving using deep reinforcement learning: Applications, challenges, and solutions. IEEE Transactions on Intelligent Transportation Systems, 25(12), 19365–19398.*

[14] *Naveen, A. M., Ravish, R., & Swamy, S. R. (2022). Distributional reinforcement learning for automated driving vehicle. In 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon) (pp. 1–6).*

[15] *Chen, X. (2019). Research on autonomous data analysis based on deep reinforcement learning. [Master's thesis, University of Electronic Science and Technology of China].*

[16] *Jia, J., & Wang, W. (2020). Review of reinforcement learning research. In 2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC) (pp. 186–191).*

[17] *Sun, Y., Yuan, X., Liu, W., & Sun, C. (2019). Model-based reinforcement learning via proximal policy optimization. In 2019 Chinese Automation Congress (CAC) (pp. 4736–4740).*