FullCycle: Full Stage Adversarial Attack For Reinforcement Learning Robustness Evaluation

Zhenshu Ma, Xuan Cai, Changhang Tian, Yuqi Fan, Kemou Jiang, Gangfu Liu, Xuesong Bai*, Aoyong Li, Yilong Ren, Haiyang Yu
State Key Laboratory of Intelligent Transportation Systems, Beihang University

Abstract

Recent advances in deep reinforcement learning (DRL) have demonstrated significant potential in applications such as autonomous driving and embodied intelligence. However, these large-scale, multi-parametric DRL models remain vulnerable to adversarial examples, while their prolonged training durations incur substantial temporal and economic costs.Current methods primarily focus on adversarial attacks during isolated training phases, whereas practical implementations may face interference across all training stages. To address this gap, we propose FullCycle, a full stage adversarial attack method that systematically assesses DRL robustness by injecting perturbations throughout the complete training pipeline. Experimental results reveal that introducing FullCycle adversarially perturbs algorithm convergence speed and agent performance to varying degrees. This work establishes a novel paradigm for robustness evaluation in reinforcement learning systems.Codes are open: https://github.com/C-137-Mzs/fullcycle.

1. Introduction

In recent years, the integration of deep reinforcement learning with large language models has demonstrated significant potential in various fields such as autonomous driving, embodied intelligence, image-text generation, and biological structure prediction.

However, large-scale, multi-parameter deep reinforcement learning models are often susceptible to adversarial examples. As shown in Fig.1, these adversarial instances originate from multiple sources: reward settings, state acquisition, action selection, training duration, interactive environments, and the architecture of reinforcement learning algorithms themselves. Taking state acquisition as an example, in tasks that integrate computer vision—such as obstacle avoidance by autonomous vehicles or object grasping by robotic arms—the reinforcement learning algorithm cannot acquire fully accurate state observations within the interactive environment. This uncertainty arises from sensor errors or dataset biases. Additionally, attacks targeting the environment can also affect model performance; for instance, manipulating environmental dynamic parameters (e.g., friction coefficients, gravity parameters) in a dynamics environment can lead to policy poisoning, significantly degrading the model's performance in continuous control tasks[2]. In other words, most reinforcement learning agents trained in simulated environments struggle to maintain stable performance, often resulting in models that appear to perform well during training but suffer severe degradation or even fail to function effectively when applied in real-world scenarios, leading to time and economic losses. Therefore, conducting robustness evaluations on existing foundational reinforcement learning models is essential.

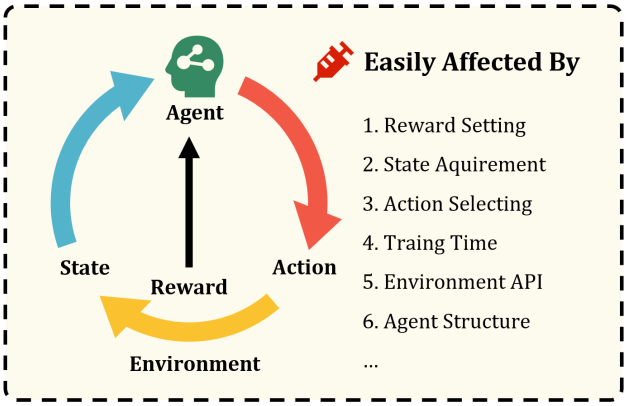


Figure 1. Reinforcement Learning Algorithms could be affected by many factors

A considerable body of work focuses on introducing disturbances to the agent's state observations or applying adversarial perturbations at a single stage[6, 10, 13, 14]. However, in reality, adversaries may attack multiple stages of reinforcement learning algorithms, making single-stage adversarial attacks insufficient for comprehensively assessing the robustness of these algorithms. Therefore, we propose

FullCycle, a full-stage adversarial attack method aimed at evaluating the robustness of reinforcement learning algorithms through perturbing state observation acquisition, capping reward accumulation, and restricting action selection. We conducted perturbation experiments on three reinforcement learning algorithms within the Highway simulation environment and found that perturbations at different stages have high-confidence attack effects on fine-tuned models. This discovery aids in identifying signs of malicious injection into models during the early stages of large-scale training, thereby minimizing time and economic losses.

Our main contributions are as follows:

1. Proposing, for the first time in the autonomous driving industry, an adversarial attack approach targeting all stages of the training process for foundational reinforcement learning models;

2. Comparing the adversarial attack effects across multiple foundational reinforcement learning algorithms and analyzing the various model metrics potentially influenced by introduced perturbations;

3. Discovering that adversarial attacks at different stages exert high-confidence attack effects on fine-tuned models, enhancing the comprehensiveness of robustness testing for algorithms.

This research underscores the necessity of robustness assessment in reinforcement learning and provides valuable insights for improving the reliability and security of AI systems in practical applications.

2. Related Works

Basic RL Algorithms. DQN is a reinforcement learning algorithm that integrates Q-learning with deep neural networks to address decision-making problems in high-dimensional state spaces. The core idea is to use a neural network to approximate the action value function $Q(s, a)$, which represents the expected return for taking action a in state s.

$$\begin{aligned} &Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \\ &\alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \end{aligned} \tag{1}$$

This update is implemented via a neural network, where the network parameters θ are adjusted based on the loss function:

$$\begin{aligned} L_i(\theta_i) = &\mathbb{E}_{(s,a,r,s') \sim U(D)}[(r + \\ &\gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i))^2] \end{aligned} \tag{2}$$

Here, θ_i^- denotes the parameters of the target network, which are periodically copied from the current network θ_i.

DDQN improves upon DQN by addressing overestimation issues. It separates the selection of the best action from the evaluation of its value to enhance performance.

$$y_i^{DDQN} = r + \gamma Q(s', \arg\max_a Q(s', a; \theta); \theta^-) \tag{3}$$

Here, θ^- represents the parameters of the target network, used to compute the maximum Q-value action for the next state s', but the actual Q-value is evaluated by the current network θ.

DuDQN combines the advantages of Dueling Network Architecture with DDQN to further enhance model performance. Dueling Networks decompose the Q-function into a state value function $V(s)$ and an advantage function $A(s, a)$, allowing better understanding of the importance of different actions in various states.

$$\begin{aligned} &Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + \\ &\left(A(s, a; \theta, \alpha) - \frac{1}{|A|} \sum_{a'} A(s, a'; \theta, \alpha) \right) \end{aligned} \tag{4}$$

Here, $V(s)$ represents the value of state s, $A(s, a)$ indicates the relative advantage of taking action a in state s, and θ, α, β denote the parameters of different parts of the network.

Adversarial Attack for RL Algorithms. Recent advancements in adversarial robustness for reinforcement learning (RL) have addressed critical challenges in safety-critical applications, particularly in the domain of autonomous driving. Wang et al.[11] developed a policy gradient method with global optimality guarantees and complexity analysis, specifically tailored to ensure robust RL under model mismatches, which is crucial for ensuring reliability in unpredictable driving environments. Moving forward, Yang et al.[12] introduced TRACER, a robust variational Bayesian inference approach designed for offline RL that effectively deals with multiple corrupted data types, enhancing the robustness of agents in real-world scenarios where data corruption is common. Miao et al.[9] proposed an iterative scheduled data-switch training framework to improve the noise resistance of neural machine translation models. Mao et al.[8] suggested discrete adversarial training (DAT) to boost visual models' robustness and generalization capabilities by transforming continuous images into symbolic sequences. Li et al.[4] presented ActorRL, a novel distributed reinforcement learning framework aimed at solving multi-agent collaboration problems, significantly increasing system robustness and efficiency. Leo Ardon et al.[1] demonstrated how reinforcement learning can solve NP-hard problems like the capacitated vehicle routing problem, showcasing potential impli-

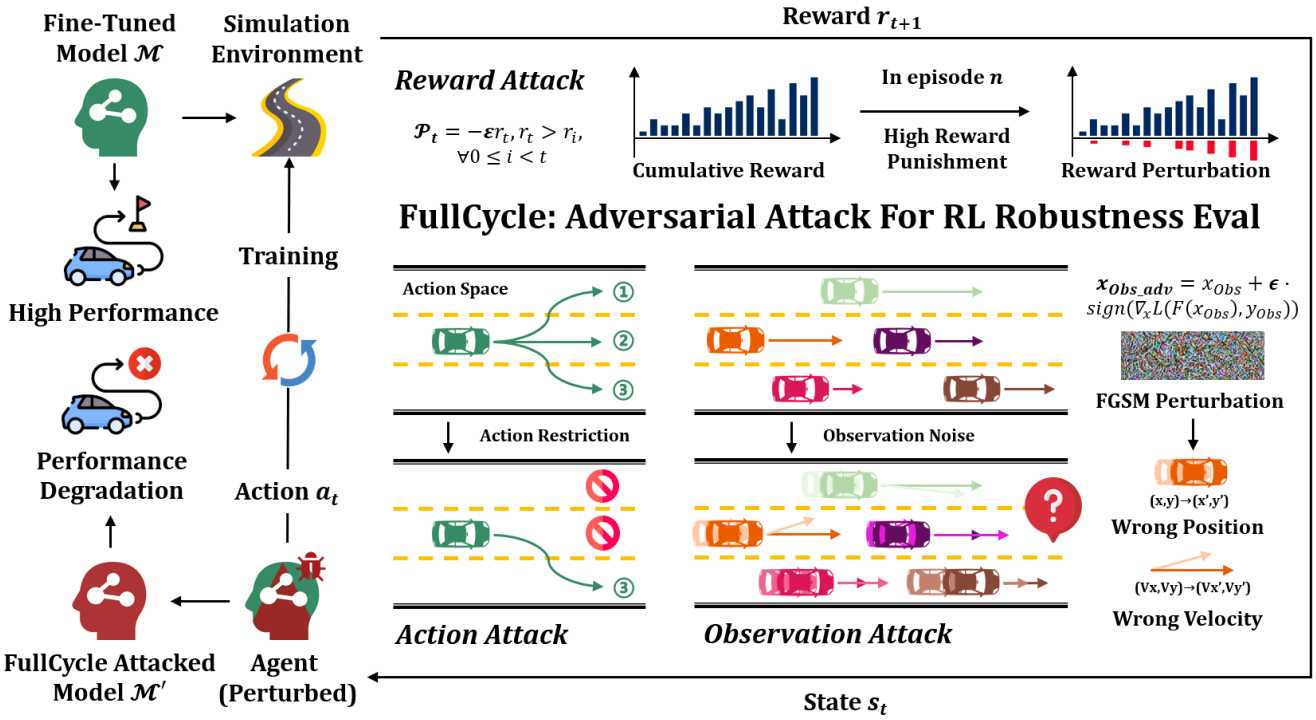


Figure 2. We proposed FullCycle, a full stage adversarial attack method to evaluate the robustness of reinforcement learning algorithms through perturbing state observation acquisition, capping reward accumulation and restricting action selection.

cations for logistics within transportation sectors. Additionally, improvements in sample-efficient deep reinforcement learning through uncertainty estimation by Vincent Mai et al.[7], and the development of AdaPool, a diurnal-adaptive fleet management framework using change point detection by Marina et al.[3], indirectly contribute to the field by addressing aspects relevant to fleet management and operation efficiency.

3. Method

Fig.2 illustrates the framework and key ideas of our evaluation. FullCycle mainly consists of observation attack, action attack and reward attack.

3.1. Observation Attack By State Perturbation

In reinforcement learning, the agent relies on observation data provided by the environment to make decisions. However, these observations may be subject to adversarial perturbations, which can mislead the agent into making suboptimal decisions[5]. To evaluate and enhance the robustness of the agent against such perturbations, we employ an observation attack method based on the Fast Gradient Sign Method (FGSM).

The core idea of this method is to generate adversarial samples that can mislead the agent's decision-making process. Specifically, for a given state x_{obs}, we compute the gradient of the loss function with respect to the input observation and add a small but effective perturbation in the direction of the gradient. The formula is as follows:

$$x_{obs\_adv} = x_{obs} + \epsilon \cdot \text{sign}(\nabla_x L(F(x_{obs}), y_{obs})) \quad (5)$$

In this equation, x_{obs} represents the original observation; ϵ is a parameter controlling the magnitude of the perturbation; $\nabla_x L(F(x_{obs}), y_{obs})$ denotes the gradient of the loss function with respect to the input observation; sign$(\cdot)$ extracts the direction of the gradient rather than its magnitude.

The adversarial sample x_{obs_adv} generated in this way is designed to remain similar to the original observation while maximally disrupting the agent's action selection mechanism.

The specific implementation steps are as follows:

- **State Perception:** First, the agent receives the current state information from the environment.
- **Loss Computation and Gradient Calculation:** Next, the corresponding loss function is computed based on the current policy and objective (e.g., maximizing cumulative rewards). The gradient of this loss with respect to the observation is then calculated.
- **Adversarial Sample Generation:** Using the obtained gradient direction and a predefined perturbation strength ϵ, an adversarial sample is generated.

- **Policy Adjustment:** Finally, the adversarial sample is used in place of the original observation to re-evaluate and select actions.

Through this approach, we can effectively simulate potential adversarial interference scenarios in real-world applications, thereby testing and improving the agent's performance under such challenges. This method not only helps identify vulnerabilities in existing models but also provides important experimental insights for developing more robust learning algorithms.

3.2. Action Attack By Lane Restriction

Traditional action-based attack methods directly modify the agent's actions to force it to perform specific behaviors. However, such approaches are often too "violent" and lack realism. We propose an environment-interference-based action attack method, where obstacles (in the form of vehicles) are randomly placed on the agent's driving path to indirectly influence its decision-making.

The positions of obstacles on each lane are determined by a truncated normal distribution:

$$P_i \sim TN(\mu, \sigma^2, D_{\min}, D_{\max}) \tag{6}$$

The number of obstacles is modeled using a Poisson distribution:

$$N \sim Poisson(\lambda) \tag{7}$$

In these formulas, P_i represents the position of obstacles on the i-th lane, following a truncated normal distribution $TN(\mu, \sigma^2, D_{\min}, D_{\max})$ within the interval $[D_{\min}, D_{\max}]$. Here, μ indicates the mean position (e.g., lane center), and σ^2 controls the spread of obstacle positions. The range $[D_{\min}, D_{\max}]$ limits where obstacles can appear. Compared to a uniform distribution, this better models real-world clustering. Meanwhile, $N \sim Poisson(\lambda)$ defines the total number of obstacles, with λ indicating road congestion or obstacle density.

The specific implementation steps are as follows:

- **Determine the Total Number of Obstacles:** Calculate the total number of obstacles N using the Poisson distribution.
- **Assign Obstacle Positions:** Allocate these obstacles to different lanes and determine their specific positions.
- **Set Obstacle States:** Ensure that the obstacle vehicles remain stationary to simulate roadblocks.
- **Dynamic Updates:** As the agent moves, dynamically adjust the positions and numbers of obstacles to maintain environmental uncertainty.

This approach avoids the "violent" direct manipulation of the agent's actions, while introducing more sophisticated probabilistic models to enhance the realism of the attack.

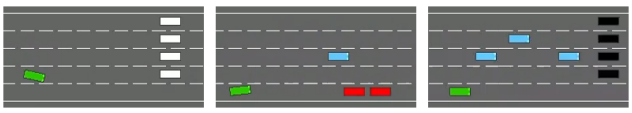


Figure 3. Random obstacles in Highway-env

3.3. Reward Attack By Capping Accumulation

The goal of the reward attack is to limit the agent's ability to obtain high reward accumulation. Specifically, during each episode, we track the maximum reward observed so far r_{max} and modify the current reward r_t if it exceeds or equals this maximum. The process can be expressed as the following system of equations:

$$\begin{cases} r_{\max} \leftarrow \max(r_{\max}, r_t), \\ r_t \leftarrow \begin{cases} (2-\varepsilon)r_{\max} - r_t, & \text{if } r_t \geq r_{\max}, \\ r_t, & \text{otherwise.} \end{cases} \end{cases} \tag{8}$$

By applying this mechanism, the system tracks the maximum reward observed so far within an episode r_{max} and updates it whenever a new reward r_t exceeds this maximum. If r_t is equal to or greater than r_{max}, the reward is modified according to the formula $(2-\varepsilon)r_{\max} - r_t$, where ϵ acts as a penalty factor that controls the strength of the reduction. This ensures high rewards are systematically reduced while maintaining a proportional relationship to r_{max}. Otherwise, the reward remains unchanged. Through this approach, whenever a reward exceeds the tracked maximum, it gets reduced, which directly impacts the total cumulative reward over the entire episode by limiting the agent's ability to accumulate exceptionally high rewards.

4. Experiment

4.1. Environment Preparations

Experiments in this paper were conducted in the Highway-env environment. The specific config and algorithm parameters are shown in Table 1. Discrete-5 means there are 5 discrete actions in action space: *LANE_LEFT, IDLE, LANE_RIGHT, FASTER* and *SLOWER*.

Table 1. Parameters Setting

Parameter	Value	Parameter	Value
lr	0.001	simu_freq	10
γ	0.9	vehicles_count	19
lanes_count	5	batch_size	20
reward_speed	[20, 32]	mem_capacity	200
punish_speed	[16, 22]	train_times	200,000
state_space	Kinematics	action_space	Discrete-5

4.2. Evaluation Indicators

Experiments set five indicators to measure the effectiveness of different attack methods on different RL algorithms.

Average Living Length Per Episode**(ALP)**: This indicator defines the average of the number of actions performed by the intelligences in each episode (excluding the buffer pool), from the beginning to the end (when a collision occurs or when the time cap is reached), which is also the average length of the Markov chain.The ALP reflects the ability of the intelligences to stay on track in the face of an attack. Higher values of ALP indicate that the intelligences are more resistant to interference and are able to maintain a stable selection of actions for a longer period of time.

First Success Episode**(FSE)**: The number of rounds required for the intelligent body to successfully complete the autopilot task for the first time. An intelligent is considered to be successful in a round if it does not have a collision, does not make an irrational action and reaches the upper time limit of the round.FSE is used to assess the learning efficiency and robustness of an intelligent. A lower FSE value indicates that the intelligence is able to quickly adapt to the environment and overcome the effects of an attack to achieve the task goal as early as possible.

Average Speed Per Episode**(ASP)**: The average speed at which the intelligent body travels within each round during the entire training or testing process (excluding the buffer pool).ASP not only reflects the speed control ability of the intelligent body, but also indirectly shows its performance when dealing with attacks. Ideally, the ASP should be as close as possible to the preset reward speed range and remain relatively fast to ensure efficient and safe driving behavior.

Reasonable Rate**(RR)**: The proportion of all actions performed by an intelligent body that are reasonable. By "reasonable" we mean following the rules of the road, staying on the road, and that's about it, RR quantifies the ability of an intelligence to respond appropriately even when under attack. A high RR means that the intelligence can maintain a high level of behavioral normality and security even in the face of challenges.

Success Rate**(SR)**: The percentage of training rounds (excluding the buffer pool) in which the intelligent body did not collide, did not perform an irrational action, and reached the upper time limit of the round.SR is a rather critical indicator, which directly reflects the overall performance of the intelligent body under counter-attack conditions. A high SR indicates that the intelligent body not only works well under normal conditions, but also has sufficient recovery ability and stability when it is subjected to external interference.

4.3. Comparison and Analysis Between Attacks

As shown in Fig 4, three basic algorithms: DQN, DDQN and DuDQN can converge normally after 2000 rounds of episodes without attack, and the loss of each round also shows a stable decreasing trend, and the speed of each round increases steadily to 31km/h or above.

The reward/loss curves of the three algorithms under reward attack are the most similar to those under no-attack state, and the average reward total obtained in a single round is higher, but the average speed stays at a lower level, indicating that the intelligences under reward attack tend to be conservative.

The DQN and DDQN algorithms under the action attack maintain the same convergence rate as in the no-attack state, but DuDQN converges slower under this attack, and all three algorithms have a very unstable loss in this state. Nevertheless, the average speed per round can approach the no-attack state level.

The Observation attack causes the algorithm's training process to exhibit very poor convergence, and its loss curve is also very unstable and the average speed is at a low level.

The FullCycle attack also severely interferes with the convergence of the algorithm, while the average speed of the intelligences in each epoch is steadily at the lowest level of all attacks.

As shown in Table 2, in the absence of all attacks, all models showed good performance on all indicators. This provides a baseline that can be used to compare the impact of different attack methods on the models.

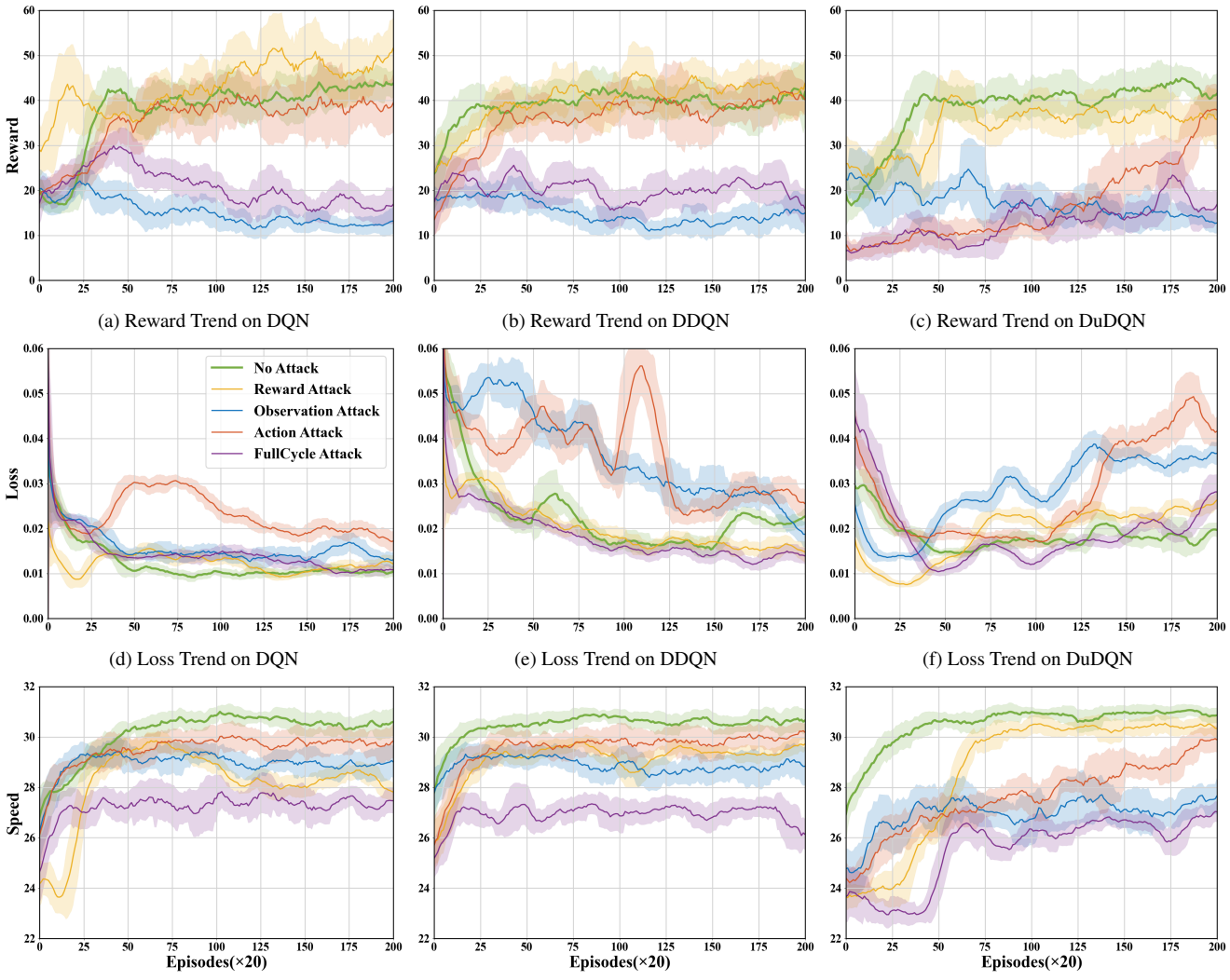
With the reward attack, there is an overall increase in ALP for all three algorithms, and even the RR for the DQN algorithm is higher than the baseline model, and it appears that the reward attack has led to an increase in the performance of the models. In fact, the reward attack causes the intelligences to avoid acquiring high reward values and therefore reduces the tendency of the intelligences to explore, making the intelligences' actions conservative. By limiting the acquisition of high reward values, the success rate (SR) of the intelligent body decreases significantly when it encounters unexplored environments. Thus under reward attacks, even if the intelligent body is able to perform more actions, its ability to complete the task is compromised.

When confronted with an observation attack, ALP and SR of all models decreased significantly, indicating that the average survival time and task efficiency of the intelligences were very low, showing the great ability of observation attacks to interfere with the decision-making process of the intelligences. However, the Reasonableness Rate (RR) remained at a high level, indicating that the intelligences were still able to make mostly reasonable decisions, although the overall performance was affected.

The action attack resulted in a significant decrease in ALP and SR and a significant increase in FSE for all models. This means that in order to successfully complete the task, the intelligences need to make more attempts, and

Table 2. Attack performance comparisons Between different RL algorithms

Method	DQN					DDQN					DuDQN				
	ALP	FSE	ASP(km/h)	RR(%)	SR(%)	ALP	FSE	ASP(km/h)	RR(%)	SR(%)	ALP	FSE	ASP(km/h)	RR(%)	SR(%)
No Attack	40.33	125	30.36	93.67	43.92	40.18	100	30.47	100	43.67	41.41	214	30.55	100	49.19
Reward Attack	50.40	146	28.20	100	34.85	46.27	258	29.07	93.49	45.53	41.16	1045	29.33	92.82	41.39
Observation Attack	**15.49**	55	28.90	**90.99**	0.6	**15.67**	127	29.08	89.88	0.64	**15.35**	540	27.34	83.47	0.5
Action Attack	39.33	789	29.57	93.08	13.02	38.62	1443	29.60	**88.76**	5.72	31.25	**5478**	28.83	88.01	2.19
FullCycle Attack	21.94	**1449**	**27.57**	100	**0.07**	21.30	**2388**	**26.94**	100	**0.05**	19.85	2042	**26.67**	**82.4**	**0.05**



(a) Reward Trend on DQN (b) Reward Trend on DDQN (c) Reward Trend on DuDQN

(d) Loss Trend on DQN (e) Loss Trend on DDQN (f) Loss Trend on DuDQN

(g) Speed Trend on DQN (h) Speed Trend on DDQN (i) Speed Trend on DuDQN

Figure 4. Attack performance comparisons Between different attack methods

the average survival time is drastically shortened, showing the effect of the action attack on the stability of the intelligences' actions.

The FullCycle attack, as an adversarial attack method proposed in this paper, shows a unique advantage in verifying the robustness of reinforcement learning algorithms. It not only significantly reduces the ALP and SR of each model, but also substantially increases the FSE, implying that the intelligent body needs more attempts to successfully complete the task. More importantly, this attack resulted in a significant decrease in SR, indicating that the intelligences had difficulty in maintaining normal decision-

making behavior when under attack.The FullCycle attack has significant advantages in evaluating and improving the robustness of reinforcement learning algorithms used in autonomous driving systems.

5. Conclusion and Limitations

Conclusion. This paper proposes a comprehensive attack framework that exposes the vulnerabilities of deep reinforcement learning systems to multi-stage adversarial perturbations. By injecting disturbances across the training phases (state observation, reward accumulation and action selection), FullCycle significantly degrades algorithm convergence and agent performance, even in fine-tuned models. Our findings highlight the necessity of holistic robustness evaluation for DRL systems and provide actionable insights for securing real-world AI deployments.

Limitations. Our work is limited to the autonomous driving task in Highway-env, and there is a lack of robustness analysis for the tested RL algorithm in other environments and tasks.

References

[1] Leo Ardon. Reinforcement learning to solve np-hard problems: an application to the cvrp, 2022. 2

[2] Jun Guo, Yonghong Chen, Yihang Hao, Zixin Yin, Yin Yu, and Simin Li. Towards comprehensive testing on the robustness of cooperative multi-agent reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 115–122, 2022. 1

[3] Marina Haliem, Vaneet Aggarwal, and Bharat Bhargava. Adapool: An adaptive model-free ride-sharing approach for dispatching using deep reinforcement learning. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, page 304–305, 2020. 3

[4] Guanzhou Li, Jianping Wu, and Yujing He. Harl: A novel hierachical adversary reinforcement learning for automoumous intersection management, 2022. 2

[5] Simin Li, Jun Guo, Jingqiao Xiu, Yuwei Zheng, Pu Feng, Xin Yu, Aishan Liu, Yaodong Yang, Bo An, Wenjun Wu, and Xianglong Liu. Attacking cooperative multi-agent reinforcement learning by adversarial minority influence, 2024. 3

[6] Aishan Liu, Shiyu Tang, Xinyun Chen, Lei Huang, Haotong Qin, Xianglong Liu, and Dacheng Tao. Towards defending multiple p-norm bounded adversarial perturbations via gated batch normalization. *International Journal of Computer Vision*, 132(6):1881–1898, 2024. 1

[7] Vincent Mai, Kaustubh Mani, and Liam Paull. Sample efficient deep reinforcement learning via uncertainty estimation. In *International Conference on Learning Representations*, 2022. 3

[8] Xiaofeng Mao, Yuefeng Chen, Ranjie Duan, Yao Zhu, Gege Qi, Shaokai Ye, Xiaodan Li, Rong Zhang, and Hui Xue. Enhance the visual representation via discrete adversarial training. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2022. 2

[9] Zhongjian Miao, Xiang Li, Liyan Kang, Wen Zhang, Chulun Zhou, Yidong Chen, Bin Wang, Min Zhang, and Jinsong Su. Towards robust neural machine translation with iterative scheduled data-switch training. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5266–5277, Gyeongju, Republic of Korea, 2022. International Committee on Computational Linguistics. 2

[10] Lu Wang, Tianyuan Zhang, Yikai Han, Muyang Fang, Ting Jin, and Jiaqi Kang. Attack end-to-end autonomous driving through module-wise noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 8349–8352, 2024. 1

[11] Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23484–23526. PMLR, 2022. 2

[12] Rui Yang, Jie Wang, Guoping Wu, and Bin Li. Uncertainty-based offline variational bayesian reinforcement learning for robustness under diverse data corruptions. In *Advances in Neural Information Processing Systems*, pages 39748–39783. Curran Associates, Inc., 2024. 2

[13] Tianyuan Zhang, Lu Wang, Jiaqi Kang, Xinwei Zhang, Siyuan Liang, Yuwei Chen, Aishan Liu, and Xianglong Liu. Module-wise adaptive adversarial training for end-to-end autonomous driving, 2024. 1

[14] Xinwei Zhang, Tianyuan Zhang, Yitong Zhang, and Shuangcheng Liu. Enhancing the transferability of adversarial attacks with stealth preservation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2915–2925, 2024. 1