Proceedings of Machine Learning Research vol 278:1–15, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Urat: Universal regularized adversarial training in robust reinforcement learning

Jingtang Chen 2021205608@QQ.COM
Fuzhou University, Fuzhou, China

Haoxiang Chen HMYHC7@NOTTINGHAM.EDU.CN
University of Nottingham, Nottingham, England

Zilin Niu* CARL_NZL@163.COM
The High School Affiliated to Renmin University of China, Beijing, China

Yi Zhu YISTEVEN.ZHU@MAIL.UTORONTO.CA
University of Toronto, Toronto, Canada

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

With the increasing maturity of reinforcement learning (RL)technology, its application areas have been widely expanded to several cutting-edge scientific fields, such as artificial intelligence, robotics, intelligent manufacturing, self-driving cars, and cognitive computing. However, the complexity and uncertainty of the real world pose serious challenges to the stability of RL models. For example, in the field of autonomous driving, unpredictable road conditions and variable weather conditions can adversely affect the decision-making process of intelligent driving algorithms, leading them to make irrational decisions. To address this problem, this study proposes a training method called Universal Regularized Adversarial Training in Robust Reinforcement Learning (Urat), which aims to enhance the robustness of the robustness of DRL strategies against potential adversarial attacks. In this study, we introduce a powerful attacker for targeted adversarial training of DRL intelligence. In addition, we innovatively incorporate a robust strategy regularizer into the algorithm to facilitate the learning of strategies by intelligences that can effectively defend against various attacks. The methods in this study have been tested adversarially in several OpenAI Gym environments, including HalfCheetah-v4, Swimmer-v4, and Arcbot-vl.The test results show that the Urat training method can effectively improve the robustness of DRL strategies and achieve robust performance in complex and uncertain environments. This research result not only provides a new perspective in the field of reinforcement learning but also provides theoretical support and technical guarantee for intelligent decision-making in practical application scenarios such as autonomous driving.

Keywords: Universal Regularized Adversarial Training (Urat), Robust Reinforcement Learning, Adversarial Attacks, Policy Regularization, Deep Deterministic Policy Gradient (DDPG), Double Deep Q-Network (DDQN)

1. Introduction

Reinforcement Learning(RL), a key paradigm in artificial intelligence, has revolutionized several industries by learning optimal decision-making strategies through interaction with the environment Reinforcement learning plays an important role in the development of self-driving cars, involving trajectory optimization, motion planning, dynamic path planning, controller optimization, and scene-based highway learning strategies. For example, AWS DeepRacer uses reinforcement learning models to control throttle and direction, while USV(Unmanned Surface Vehicle) utilizes deep

© 2025 J. Chen, H. Chen, Z. Niu & Y. Zhu.

reinforcement learning algorithms to handle lane follow-up tasks (Woo et al., 2019; Garza-Coello et al., 2023). AlphaGo Zero is a famous application of reinforcement leaning in games, as it was able to learn Go from scratch and self-learn through play (Shin et al., 2021). Reinforcement learning is also used in financial trading to determine the best time to hold, buy, or sell a stock. For instance, IBM has adopted a reinforcement learning platform that adapts a reward function based on the profit and loss of a financial trade (Ansari et al., 2022). However, in the real world, there are many uncertainties that pose challenges to the application of reinforcement learning models. In the case of autonomous driving, for example, which relies on a variety of sensors for environmental sensing, the sensors may be affected by bad weather or lighting conditions, and unexpected road conditions such as traffic accidents may occur during the driving process (Vargas et al., 2021). This research aims to address this challenge and pave the way for more resilient and reliable RL-based solutions. Specifically, we focus on the robust stability of the model, which emphasizes the ability of the intelligences to consistently deliver stable and positive results even in the presence of adversarial noise in the environment that may lead to undesirable outcomes. The successful integration of deep reinforcement learning (DRL) into robotics,autonomous systems, and control applications is highly dependent on the development of robust and stable strategies for the continuous state action RL domain.

To ensure the pursuit of our objectives within the framework of adversarial training, we determine the "maximum" attacks that are dedicated to deep reinforcement learning (DRL) agents. In the course of the training process, our attacks are used to interfere with an agent's decision-making mechanism, thus making it more robust. Our work is inspired by a method which uses a shadow mode adversary learning approach to obtain an adversary policy that performs suboptimally (Pattanaik et al., 2017). Nevertheless, such self-defined strong adversarial attacks often drive training patterns to randomness, and the learning procedure itself might struggle to maintain good evaluation when extermal attacks arrive at a later point. Consequently, the trained agent can hardly generalize its outcomes and adaptively improve its strategies under strong adversarial attacks. This reaffirms the need to regulate the agent's choice of action by keeping it restricted to a specific range for the desired good performance. We introduce a regularizer, which is L2 norm-based regularization to quantify the distribution of decisions in the policy space with respect to a linearly changing weighting of adversarial noise (Zhang et al., 2020). The ultimate purpose is for the agent to survive and evolve to be able to generate more complex, robust, and well-adjusted strategies.

Specifically, the primary contributions of this paper are twofold. First, we have integrated a custom-designed regularizer into the dominant RL algorithms to help the RL agent adapt the adversial attack we promoted, marking a substantial advancement in the stability of robust reinforcement learning. Second, the selection of DDPG and DDQN ensures the versatility of our algorithm, accommodating both continuous and discrete environments.

The paper is organized as follows. We will provide the introduction and related work in Section 1. The background information has been disclosed in Section 2. The explanation of adversity attacks and their use to improve the stability of robustness is given in Section 3. The results of our research are presented in Section 4. Finally, concluding remarks and future directions have been discussed in Section 5.

2. Related Work

In the early stage of study in the field of Deep Reinforcement Learning (DRL), a study proposes a method that designs gradient information for loss functions to improve intelligence performance in adversarial environments in an attempt to improve their decision-making ability (Pattanaik et al., 2017) . Although robustness is enhanced to some extent, this model tends to have unstable performance in adversarial scenarios where training and assessment are conducted in parallel. Overfit issues often occur when confronted with a complex environment (Ran et al., 2022).

To address this problem, a defensive mechanism based on the Deep Q Network (DQN) is designed to handle the attack on the observation state in the power system (Ran et al., 2022) . Thanks to the introduction of normalized policy, the perturbation is more likely to be observed and resolved properly. While the method demonstrates efficacy in the face of rudimentary attacks, its effectiveness is curtailed when confronted with more sophisticated threats. Currently, DQN has been observed to overestimate the state action values. The unstable training process also leads to limitation of its compatibility, especially restricting its applicability in continuous control tasks (Li, 2023). Likewise, there is a more general regularization method proposed that can be used in a variety of main algorithms, such as Proximal Policy Optimization (PPO), to improve the robustness of the model under white-box attacks , while maintaining good performance in attack-free scenarios at the same time (Zhang et al., 2020; Gu et al., 2021). However, this policy is mainly designed for low-complexity tasks and has limited adaptability under intense attack.

Theoretically, a framework for generalized reinforcement learning is established that gave convergence proofs for robust Q-learning and TDC algorithms (Wang and Zou, 2021). Its estimation of the uncertainty set of the unknown environment through samples maintains the convergence efficiency of the standard algorithm without relying on discount factors. A new perspective for robustness has been inspired but still requires advanced analysis for the actual effect. The ATLA framework shows risilience originating from online training against adversarial agents and adaptive opponents (Zhang et al., 2021) . The core of this method is the simulation of TU games and related alternative training mechanism, but it is highly based on the quality and diversity of the design of the opponents model (Rozemberczki et al., 2022).

This paper investigates the application of non-linear loss functions with gradient information to the enhancement of the robustness of DRL algorithms against the phenomenon of adversarial attacks (Pattanaik et al., 2017). In spite of this, robustness is also an issue that must be tackled by the current approach, which is mainly focused on the decision making process and the model training/evaluation process. The complexity arises from the fact that models are trained with different optimization regimes, which could affect the performance of the model, leading to over fitting.

A strong DQN is particularly concerned with the adversarial conditions of the power system and the prediction of the different current and voltage observations. (Ran et al., 2022)Such research proposes a strong method with a policy regularizer that will strengthen the defense of the implemented strategy. Whether this method is effective for complex, non-simple attacks is debatable; thus, it may not be a viable countermeasure to malicious activities that involve complex attacks. Not only the DQN is subject to overestimating the value of certain actions/states that leads to instabilities in training and generalization, but it is also not suitable for direct inception of these systems into control.

A novel way of defining policy regularization grounded theoretically is proposed and it can be used for various DRL agents, one of which is the PPO (Zhang et al., 2020) . This method is

determined to withstand a succession of strong white-box attacks, some of which are fresh and unexplored, all brought by the authors. The decisive strategy then shows a significantly improved DRL performance regardless of the presence of a competitor. However, this method is mainly focused on moderate attacks, while dealing with tricky and advanced attacks may require more resources.

3. Background

3.1. Deep Q learning (DQN) and Deep Double Q Learning (DDQN)

Deep O learning (DON) developed by (Mnih et al., 2015). demonstrated an outstanding performance with humans falling short in Atari games. Q learning includes a value function-based algorithm, where the value function incorporates the state action value function and the state value function. The Q values belonging to a state-action pair represent the significance of the action that the agent is assessing now, depending on the current observation. This learning agent modifies these Q values with the help of the temporal difference error, which is in order to maximize the long-term return. In addition, the DQN framework is based on the premise that the agent has a deep neural network that approximates the Q function. The model tends to stabilize while training thanks to the experience replay and the target network, DQN algorithm ensures the two. Behind the act of experience replay lays the ability to store the history of sequences of states, actions, rewards, and future states. Such sequences are randomly picked from memory and ideally change the Q network, just as supervised learning does. This kind of method breaks the connections. In the "supervised" learning technique of DQN, to enable the update of Q values for the next states, the target network-another neural network

The target network undergoes an update through a hard transfer of on-line weights after a predetermined number of iterations. The two networks constitute a substantial part of the theory's stability. For this "supervised" learning-style update, DQN incorporates another neural network, termed the target network, responsible for providing Q values for subsequent states. The target network undergoes an update through a hard transfer of online weights after a predetermined number of iterations. This dual-network architecture contributes significantly to the stability of the training process. On the other hand, DQN algorithm often can results in an overestimation of Q-values (Van Hasselt et al., 2016). However, to remedy that, it was proposed to use Double Deep Q-Learning (DDQN). Hence, the decision of the online network is to take a particular action;however, the value update of that action is taken from the target network. This adjustment, thus, effectively compensates for the artificial inflation of the value function.

3.2. Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) incorporates both an actor and a critic in its learning process (Lillicrap, 2015) . The critic serves to evaluate the policy formulated by the actor. The weights of both the critic and actor networks are iteratively refined using gradient descent optimization.

In updating the critic network, DDPG draws upon the established concepts of experience replay and target networks, similar to those employed in Deep Q-learning. However, a notable distinction lies in the mechanism for updating the target network. Unlike Deep Q-learning, which relies on a "hard" transfer of weights to the target network after a fixed number of iterations, DDPG adopts a

"soft" transfer approach. In this method, the weights of the target network are gradually adjusted by incrementing them by a small fraction towards the weights of the online network. This incremental approach contributes to smoother and more stable learning dynamics within the reinforcement learning framework.

3.3. Potential conditions that need stable robustness

- Sudden Situations in Autonomous Driving:While driving autonomous vehicles, reinforcement learning algorithms are most likely to be deployed in decision making, which is path planning and avoiding obstacles, for example. However, unexpected sensors could provide similar adversarial perturbations or external environment changes, resulting in the vehicle making wrong judgements about obstacles or road lines, which also represent a dangerous situation. An example reflecting this is the sudden emergence of obstacles(like a bunch of small animals, fallen branches, or debris) or the switch of road markings by a human attacker, in the road or lane change detection systems, respectively, causing crushes or delays in stopping that could also lead to accidents. All of these represent a vital part of the RL models that need to be developed for very complicated and fluid environments.

- Sudden Situations in Smart Factory Automation:In the production lines commonly seen in smart factories, machine operations and resource optimization are artificially pronounced by means of RL. With sensor data on the production line that were manipulated by attackers, the system may make erroneous decisions that can cause a visualization of material locations and material quantity and also cause the whole production process to stop and a decrease in product qualities. An insignificant change in sensor data could result in an inappropriate item pick-up by default, which, in turn, leads either to the shutdown of a production line or overproduction of defective products due to this single mistake. That accelerated nature of such turnarounds stresses that resilience is of utmost importance for the business processes run by the RL model in the critical industrial systems

- Sudden Situations in Drone Delivery:It is quite common that reinforcement-leaning models are used in such situations when the drones are delivering, and it is the models that are responsible for path planning and obstacles avoidance. Whether drones come into contact with hackers during their journey who apply silent signals or natural sounds to distort the drone's position, this could make them divert to avoid obstacles, fail to detect land space, and take incorrect leading to damages on crucial goods. For example, what if a drone approaches a high-rise building and makes a misjudgment due to glass reflections from the building, leading to a collision?An advanced RL model should be robust and by far able to learn from the sensor data to guarantee success. The illustrations create most coveted characteristics for robust RL models demands from the implementations of model in intricate environments.

Therefore, during the training process, we choose to generate adversarial trajectories for the agent and employ existing DRL algorithms in order to obtain a robust strategy. However, our study shows that in most scenarios, simple adversarial training methods (e.g. adding adversarial states to the replay buffer)lead to unstable training, which reduces the performance of the agent or fails to effectively improve the robustness against strong attacks, and thus we obtain more robust strategies by regularizing the loss function to penalize the difference in the decision making of the intelligences in the perturbation process during the training process of the intelligences.

4. Methodology

In this section, we provide a comprehensive exposition of the methodology adopted for the adversarial training process, focusing specifically on the strategies for adversarial attacks and the formulation of a robust and stable policy. The significance of adversarial training in robust optimization stems from its ability to equip the agent with the capability to make optimal decisions in accordance with a rational policy, even within an environment characterized by state uncertainties. Our approach integrates a gradient-based adversarial training technique into two preeminent algorithms: Deep Deterministic Policy Gradient (DDPG)and Double Deep Q Network (DDQN). This integration guarantees that our algorithm remains versatile, including the ability to work in both discrete and continuous action spaces. Since both DDPG and DDQN work with the experience replay buffer, which has been used to stabilize the training process and break the correlation of observations by minimizing the effect of similar observations, our model demonstrates a higher probability of convergence in more situations, even with a minimum training time.

4.1. Adversarial Attack

Moreover, we incorporate a regularization method that targets training the model to guarantee its stability and robustness. This chapter of the paper focuses on a method of generating adversary tasks for RL agents and identifying a task structure. The attacks influence the current entry state, which can lead to the coordination of the situation and the application of a sub-optimal policy. In this section, adversarial training will be defined as a method based on the value function approach in the field of reinforcement learning.

Definition 1 (Adversarial Attack) *An adversarial attack is defined as any perturbation that elevates the likelihood of an agent selecting the least favorable action in a given state. In the realm of trained reinforcement learning(RL)agents, the least favorable action refers to the one yielding the minimum Q-value.*

Based on Definition 1, we have introduced an adversarial training approach. This method is exclusively compatible with algorithmic frameworks that utilize a value network to evaluate actions. Specifically, the Double Deep Q-Network(DDQN)employs a target network for Q-value estimation, while the Deep Deterministic Policy Gradient (DDPG)uses a critic network for the same purpose.

We have adopted a gradient-based generative adversarial attack method, which has shown superior effectiveness compared to the traditional cost function employed in the Fast Gradient Sign Method (FSGM) to pinpoint the least optimal action (Huang et al., 2017).

Our adversarial attack framework consists of several steps. Initially, random noise n_i is introduced iteratively to the present state s_i;, aiming to determine the optimal noise direction via gradient analysis $grad\_dir$. Upon identification, the specific noise that minimizes the value function estimation is selected as the adversarial noise. This noise is then superimposed onto the currently observed state. The perturbed state s_{adv} subsequently misdirects the agent toward selecting a suboptimal action. It is worth noting that during experimentation, noise is drawn from a beta distribution, with noise generation repeated n times. The beta distribution parameter is configured as (1,1), yielding an average value of 0. For a complete outline of the adversarial attack tailored for DDQN, please consult Algorithm 1.

In this algorithm, the representation of the Q network serves as a tool to approximate the expected cumulative reward for executing a specific action in each state. Meanwhile, a delayed copy

Algorithm 1: Gradient Based Attack based on DDQN

Input: Q network (Q), Target Q network (Q^{target}), Number of times to sample noise (n), Current state (s), Parameters of beta distribution (α, β), Adversarial attack magnitude constraint (ϵ)

Output: Adversarial state s_{adv}

Function `GradientBasedAttack` (Q, Q^{target}, n, s, α, β, ϵ) **:**

- $a^* = \arg\max_a Q(s,a)$, $Q^* = \max_a Q^{target}(s,a)$;
- $\pi^{target} = softmax(Q^{target})$;
- $grad = \nabla_s J(s, \pi^{target})$;
- $grad\_dir = \frac{\nabla_s J(s,\pi^{target})}{||\nabla_s J(s,\pi^{target})||}$;
- **for** $i = 1 : n$ **do**
  - $n_i \sim \beta(\alpha, \beta)$;
  - $s_i = s - n_i \times grad\_dir$;
  - $a_{adv} = \arg\max_a Q(s_i, a)$;
  - $Q^{target}_{adv} = Q^{target}(s_i, a_{adv})$;
  - **if** $Q^{target}_{adv} < Q^{target}(s, a_{adv})$ **then**
    - $Q^{target}_{adv} = Q^{target}(s, a_{adv})$;
    - $s_{adv} = s_i$;
  - **else**
    - do nothing;
  - **end**
- **end**

return s_{adv}

of the Q network, denoted Q^{target} acts as a stabilizing factor during training by maintaining a consistent network structure. Regarding noise generation, the beta distribution parameters, represented by α, β, play a key role in shaping and controlling the beta distribution while ϵ functions as a constraint on the magnitude of the adversarial attack. Furthermore, π^{target} is a probability distribution over actions derived from the target Q network utilizing the softmax function. Particularly, to find the direction of the steepest descent gradient $grad$, Function $J(s, \pi)$is defined as:

$$J(s,\pi) = -\sum_{i=1}^{n} p_i log \pi_i \tag{1}$$

whose minimization represents optimal adversarial attack on RL agent. Where $\pi_i = \pi(a_i|s), p_i = P(a_i)$ the adversarial probability distribution P is given by

$$P(a_i) = \begin{cases} 1, & \text{if } a_{\text{worst}} = 1 \\ 0, & \text{otherwise} \end{cases}$$

Regarding the gradient-based attack for DDPG, we utilize the critic network for defining the value function while actions are taken based on the actor network (Silver et al., 2014). Consequently, the target function for the attack is the target Q network(critic), representing a value function determined by a trained commentator network. Algorithm 2 outlines adversarial attacks designed for DDPG, mirroring the structure of Algorithm 1.

Algorithm 2: Gradient Based Attack based on DDPG

Input: Target Q network (critic) (Q^{target}), Actor network (U), Number of times to sample noise (n), Current state (s), Parameters of beta distribution (α, β), Adversarial attack magnitude constraint (ϵ)

Output: Adversarial state s_{adv}

Function `GradientBasedAttack` (Q^{target}, U, n, s, α, β, ϵ) **:**

$a^* = U(s)$, $Q^* = Q^{target}(s, a^*)$;

$grad = \nabla_s Q^{target}(s, a)$;

$grad\_dir = \frac{\nabla_s Q^{target}(s,a)}{||\nabla_s Q^{target}(s,a)||}$;

for $i = 1 : n$ **do**

$n_i \sim beta(\alpha, \beta)$;

$s_i = s - n_i \times grad\_dir$;

$a_{adv} = U(s_i)$;

$Q^{target}_{adv} = Q^{target}(s, a_{adv})$;

if $Q^{target}_{adv} < Q^*$ **then**

$Q^* = Q^{target}_{adv}$;

$s_{adv} = s_i$;

else

do nothing;

end

end

return s_{adv}

4.2. Universal regularized adversarial training

This section discusses our choice of using a robust policy regularizer for DDPG and a hinge-like robust policy regularizer for DDQN.

Our regularizer is proposed to be applied to the DDPG (Deep Deterministic Policy Gradient) since that method is suitable for such policy-based training systems. We find the implementation of the Double Deep Q-Network (DDQN) technique, which is a value-based approach, appropriate as well, and we incorporate our regularizer into the loss function. This regularizer penalizes the total distance between the distributions of smoothed strategies or the predictions of the value function. In doing so, we created robustness in the selection of actions, particularly in the case of factors that can be considered disturbances in the environment. We refer to this methodology as URAT (Universal Regularized Adversarial Training).

4.2.1. D_{TV} BASED ROBUST POLICY REGULARIZER ON DDPG

DDPG (Deep Deterministic Policy Gradient)learns a deterministic policy π(s). However, this deterministic nature makes the policy susceptible to severe fluctuations in the presence of significant disturbances during the adversarial training process. Such disturbances can potentially destabilize training results. To mitigate this challenge, we propose the introduction of a smooth variant of the policy. By incorporating noise sampled from an independent Gaussian distribution and adding it to the current states, we obtain a noised state. This perturbed state prompts the agent to adapt its policy.

Subsequently, we utilize Equation 3 to calculate the total variation distance D_{TV} between two actions. This approach enhances the robustness of the policy against disturbances, thereby improving the stability of the training process.

$$D_{TV} = \sum_{s \in \Pi_S} \max_{s' \in \Pi_s} \left\| \pi_{\theta_\pi}(s) - \pi_{\theta_\pi}(s') \right\|_2 \tag{2}$$

$$R_{regularizer} = \sqrt{\frac{2}{\pi}} \cdot \left(\frac{1}{\sigma} \right) \cdot \sum_{s \in \Pi_S} \max_{s' \in \Pi_s} \left\| \pi_{\theta_\pi}(s) - \pi_{\theta_\pi}(s') \right\|_2 \tag{3}$$

Where $\pi_{\theta_\pi}(s)$ is the action from the policy network π at state s, and similarly, $\pi_{\theta_\pi}(s')$ is the action from the policy network π at the perturbed state s'. The term $\|\cdot\|_2$ denotes the L2 norm. The symbol Π_S signifies the complete set that encompasses all possible states in the state space, whereas Π_s refers to a set of perturbations specifically designed for a given state s. Furthermore, σ functions as a regularization parameter, controlling the magnitude of the regularization terms.

By introducing a penalty term through Equations (2) and (3), which quantifies and penalizes deviations in policy actions resulting from state perturbations, we can effectively constrain policy variations to a certain degree. This approach serves to strengthen the agent's resilience against significant disturbances, thereby promoting a more stable and robust policy. Our proposed regularizer, which can be seamlessly integrated into the loss function as demonstrated in Equation (4), contributes to the enforcement of smoother policy transitions in the presence of perturbations.

$$\mathcal{L}_{actor} = -\mathbb{E}_{s \sim \mathcal{D}} \left[Q \left(s, \mu \left(s | \theta^\mu \right) \middle| \theta^Q \right) \right] + R_{regularizer} \tag{4}$$

Where $Q\left(s, \mu\left(s|\theta^\mu\right)|\theta^Q\right)$is the estimated value (from the critic network Q) of taking action $\mu\left(s|\theta^\mu\right)$ which suggested by the actor-network μ at state s, under the critic's parameters θ^Q. Drepresents the experience replay buffer, which stores past experiences (state, action, reward, next state).

Apparently, our training object is to obtain a minimized loss function. Meanwhile, by introducing a penalty term based on D_{TV}, the degree of change in policy distribution is also limited to a reasonable extent.

4.2.2. HINGE-LIKE ROBUST POLICY REGULARIZER FOR DQN

In Deep Q-Networks (DQN), actions are typically chosen from a discrete set based on the highest Q-value predictions. Analogously to the regularizer devised for the Deep Deterministic Policy Gradient (DDPG), we have formulated a novel regularizer tailored for Double Deep Q-Networks (DDQN). This regularizer is designed to penalize deviations in Q-network predictions resulting from state perturbations. The regularizer is defined as follows:

$$R_{regularizer} = \sqrt{\frac{2}{\pi}} \cdot \left(\frac{1}{\sigma} \right) \cdot \sum_{s \in \Pi_S} \max_{s' \in \Pi_s} \left\| Q(s, a|\theta) - Q(s', a|\theta) \right\|_2 \tag{5}$$

Where σ is a regularizing parameter to scale the size of regularized terms, Π_S is the set of all possible state in state space S, Π_s represents a set of perturbations for a single state s, $Q(s, a \mid \theta)$, represents the Q-value, which estimates the expected reward if taking action a in state s given the parameters θ of the Q-network. The Q value with perturbed state s' is denoted as $Q(s', a \mid \theta)$.

The loss function for DQN, incorporating our regularizer, is expressed as:

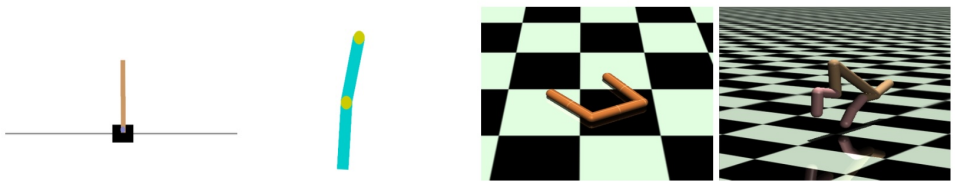
$$\mathcal{L}_{dqn} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q(s,a|\theta) - \left(r + \gamma Q'(s', \arg\max_{a'} Q(s',a'|\theta)|\theta^-)\right)\right)^2\right] + R_{regularizer} \tag{6}$$

In this equation, $Q(s, a \mid \theta)$ represents the estimated Q-value for action a in state s. R is the reward obtained after executing the action a in state s. The discount factor γ balances the significance of rewards over time. $Q'\left(s', \arg\max_{a'} Q(s', a' \mid \theta) \mid \theta^-\right)$ denotes the target Q-value, where the action maximizing the Q-value in the next state s' is chosen based on the current Q-network, but its value is computed using the target network $Q'(\cdot \mid \theta^-)$.

By incorporating this regularizer into the DQN loss function, we aim to enhance the robustness of action selection under environmental disturbances, thus improving the agent's performance in dynamic and uncertain scenarios.

5. Results

In this section, we will discuss the results related to adversarial attacks with regularizer and without regularizer. We discovered the improvement in stability robustness over two algorithms(DDQN and DDPG)with regularizer. As figure 1 shows, all experiments have been performed within an OpenAi gym environment as figure with MuJoCo and Classic Control (Brockman, 2016).



(a) Pendulum (b) Arcbot (c) Swimmer (d) Half-Cheetah

Figure 1: OpenAI Gym Environment

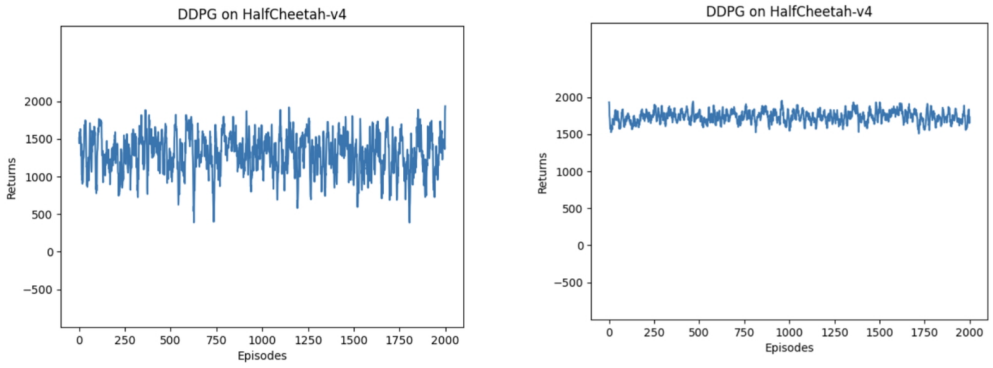
Notably, the choice of the regularization parameter σ significantly impacts experimental outcomes, as it reflects the strength of the imposed penalty and interacts with both the physical characteristics of the environment and the decision-making mechanisms of the learning algorithm. In the majority of continuous control environments, such as *HalfCheetah-v4*, the high-dimensional action and state spaces, along with complex locomotion dynamics, require meticulous tuning. Specifically, σ is set equal to 0.1 in our experiment to impose stronger regularization and effectively suppress the noise from high-dimensional observations while retaining the flexibility needed for coordinated leg movements.

In terms of the *Swimmer-v4* environment, however, characterized by fluid dynamics, the regularization factor is set to be a moderate lever ($\sigma = 0.4$). This setting achieves a balance between maintaining swimming efficiency and reducing torque fluctuations, highlighting the trade-off between regularization strength and hydrodynamic resistance.

For discrete control architectures such as *Acrobot-v1*, a larger value of $\sigma = 1.14$ is used. Due to the simplicity of the discrete action space and the step-like nature of action transitions, a higher tolerance for value variation is appropriate. The increased σ reduces the penalization of Q-value

differences, enabling the agent to maintain stable policy decisions even under noisy state observations.

The line graphs presented depict the State-Adversarial (SA)reward and Universal Regularized Adversarial(Urat)reward of SA-DDPG and Urat DDPG, respectively. Each line graph is derived from the average rewards of at least 15 agents trained using identical parameters. The vertical axis represents the mean reward per training epoch, while the horizontal axis denotes the training epochs. The left graph illustrates the line graph of the SA reward and the right graph displays the line graph of the Urat reward.



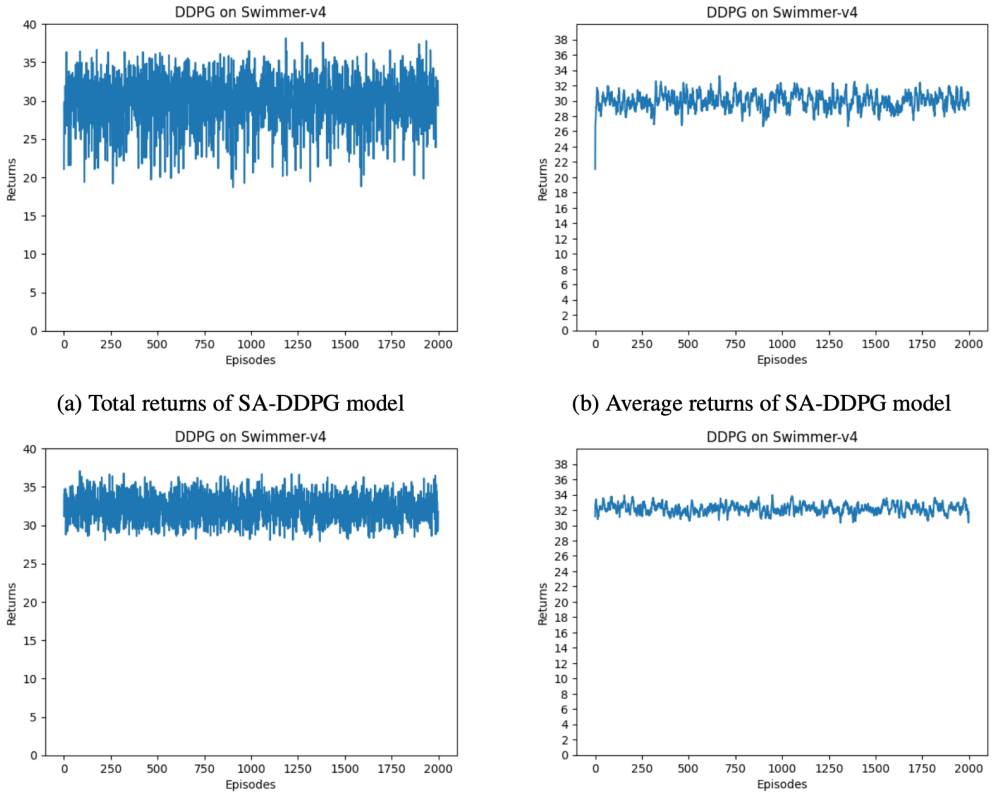
(a) Average returns of SA-DDPG model

(b) Average returns of Urat model

Figure 2: The figure displayed in this figure show the State-Adversarial (SA)rewards alongside the Universal Regularization Adversarial(Urat)rewards of the SA-DDPG model and Urat-DDPG, as replicated in the HalfCheetah-v4 environment. These lines show the average total reward per episode of at least 15 agents whose trained parameters were similar. The vertical line shows the mean of rewards per training episode, while the horizontal indicates the number of training episodes.

In this work, we will be registering Urat-DDPG with a good quality implementation of State-Adversarial (SA)-DDPG being our baseline, although there are similar findings. As figure 2 shows, the regularized SA-DDPG achieves improved performance in the Half-Cheetah-v4 environment. We employ the optimal hyperparameters for SA-DDPG and use the same set of hyperparameters for Urat-DDPG .We run the Half-Cheetah-v4 environment for an extensive number of steps to ensure convergence. And we did the same steps for other environments such as Swimmer-v4 (As figure 3 shows) and Arcbot-v1. The regularizer maintains decision stability by penalizing the differences in actions output by the perturbed policy network. We assess the robustness of the trained agents in adversarial test environments.

Our findings indicate that adversarial training solely through state-adversarial methods can lead to unstable performance, failing to reliably enhance the robustness of the agent in the environment. Our gradient-based attack proves to be highly effective in the environment, yielding significantly lower rewards compared to random attacks. This underscores the importance of evaluating agents with strong attacks. Given the potential for significant performance variations in DDPG training across multiple runs, we demonstrate the consistent robustness of our SA-DDPG by repeatedly training both SA-DDPG and Urat-DDPG in the same environment at least 15 times and attacking all agents obtained. In the figures, we present line graphs that show the best natural attack rewards



(a) Total returns of SA-DDPG model

(b) Average returns of SA-DDPG model

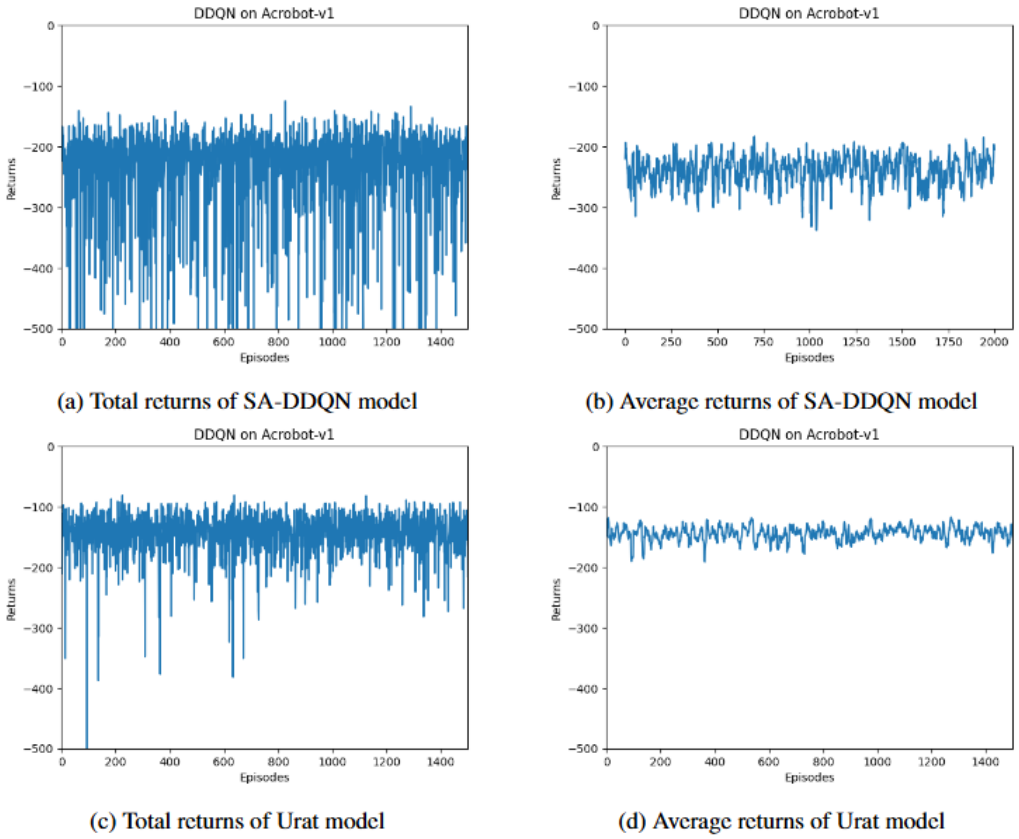
(c) Total returns of Urat model

(d) Average returns of Urat model

Figure 3: The results illustrate the rewards of State-Adversarial(SA)DDPG and the rewards of Universal Regularization Adversarial(Urat)DDPG in the Swimmmer-v4 environment for step-based DDPG. The left part shows the complete reward earned from the experiments with 15 agents using the same parameter values, while the right part shows the average reward per episode for the same amounts of the trained agents. The upper portion of the chart narrates the testing outcomes obtained with the plain SA-DDPG model;conversely, the lower portion of the chart reveals the outcomes for Urat DDPG. The vertical line shows the average reward gained per training episode, and the horizontal one shows the number of training episodes.

for these SA-DDPG and Urat-DDPG agents. Our results reveal that the best attack rewards for most Urat-DDPG agents outperform those of SA-DDPG agents in the majority of cases, and the rewards of Urat-DDPG are comparatively more stable.

To ensure the generalizability of our method, as figure 4 shows, we also implemented SA-Double DQN and prioritized experience playback in the Acrobot-vl environment (Pattanaik et al., 2017) . We conducted at least 15 adversarial training sessions for both SA-DDQN and Urat-DDQN and computed the average to ensure the robustness of our experimental results. Using SA-DDQN as the baseline for Urat-DDQN training, we incorporated a regularizer into the loss function to penalize the differences in actions sampled by the Q-network before and after perturbation. We observed that the rewards did not decrease further. In the figures we displayed, it demonstrates that our Urat-DDQN achieves higher and more robust rewards when attacked in most environments, whereas



(a) Total returns of SA-DDQN model

(b) Average returns of SA-DDQN model

(c) Total returns of Urat model

(d) Average returns of Urat model

Figure 4: Line graphs comparing the state-adversarial (SA)rewards and Universal regularized adversarial (Urat) rewards for SA-DDQN and Urat DDQN in Acrobot-vl environment. The upper part represents the total rewards obtained by at least 15 agents trained with the same parameters, while the lower represents the average rewards per episode obtained by the same number of agents. The vertical axis indicates the average rewards per training round, while the horizontal axis represents the training rounds. The left graph displays the line graph for SA rewards, and the right graph shows the other one.

SA-DDQN, which is solely subjected to unrestricted attacks, exhibits highly unstable performance under strong attacks. This is because our method constrains decision making within a controllable range, rather than attacking the agent without limitation during the training process.

6. Conclusion

In this paper, we introduce an adversarial attack to train a reinforcement learning agent and incorporate a regularizer during the adversarial training process, and our experimental results in multiple environments show that adversarial-trained intelligences are prone to making unstable decisions. However, with the addition of the regularizer, the rewards of reinforcement learning in adversarial attacks will stabilize in a relatively high range, implying that our approach achieves robustness and decision quality of the intelligences.

Acknowledgments

Jingtang Chen and Haoxiang Chen are co-first authors of this work, having contributed equally to the research and writing. Zilin Niu and Yi Zhu are co-second authors. Zilin Niu also served as the corresponding author and was responsible for coordinating the collaboration and finalizing the manuscript.

The authors would like to thank Prof.Peter Zhang (Operations Research, Carnegie Mellon University), Dr.Zijian Wang (The University of Hong Kong) and Prof.Mingjian Fu(Fuzhou University) for their valuable guidance and support throughout this research. Their assistance in setting up the research environment, designing the experiments, and addressing critical issues was instrumental to the success of this work.

References

Yasmeen Ansari, Sadaf Yasmin, Sheneela Naz, Hira Zaffar, Zeeshan Ali, Jihoon Moon, and Seungmin Rho. A deep reinforcement learning-based decision support system for automated stock market trading. *IEEE Access*, 10:127469–127501, 2022.

G Brockman. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Lydia A Garza-Coello, Marco Moreno Zubler, Axayacatl Nava Montiel, and Carlos Vazquez-Hurtado. Aws deepracer: A way to understand and apply the reinforcement learning methods. In *2023 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–3. IEEE, 2023.

Yang Gu, Yuhu Cheng, CL Philip Chen, and Xuesong Wang. Proximal policy optimization with policy feedback. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(7):4600–4610, 2021.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

Shengbo Eben Li. Deep reinforcement learning. In *Reinforcement learning for sequential decision and optimal control*, pages 365–402. Springer, 2023.

TP Lillicrap. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

Xiaohong Ran, Wee Peng Tay, and Christopher HT Lee. A robust deep q-network based attack detection approach in power systems. In *2022 4th International Conference on Smart Power & Internet Energy Systems (SPIES)*, pages 995–1000. IEEE, 2022.

Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. In *The 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence*, pages 5572–5579. International Joint Conferences on Artificial Intelligence Organization, 2022.

Minkyu Shin, Jin Kim, and Minkyung Kim. Human learning from artificial intelligence: Evidence from human go players' decisions after alphago. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43, 2021.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Jorge Vargas, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors*, 21(16):5397, 2021.

Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 7193–7206. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/3a4496776767aaa99f9804d0905fe584-Paper.pdf.

Joohyun Woo, Chanwoo Yu, and Nakwan Kim. Deep reinforcement learning-based controller for path following of an unmanned surface vehicle. *Ocean Engineering*, 183:155–166, 2019.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33:21024–21037, 2020.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust reinforcement learning on state observations with learned optimal adversary, 2021. URL https://arxiv.org/abs/2101.08452.