Caroline Hammond, Aram Moossavi, Wendy Liang, Mark Lovett, and Rose Wenxin Zhao

Project Report
Robust Deep Reinforcement Learning through Mitigating Adversary and Noise in Both States and Human Feedback

COSC 189 Spring 2023
Professor: Yaoqing Yang

DARTMOUTH

2023

Contents

1 Introduction

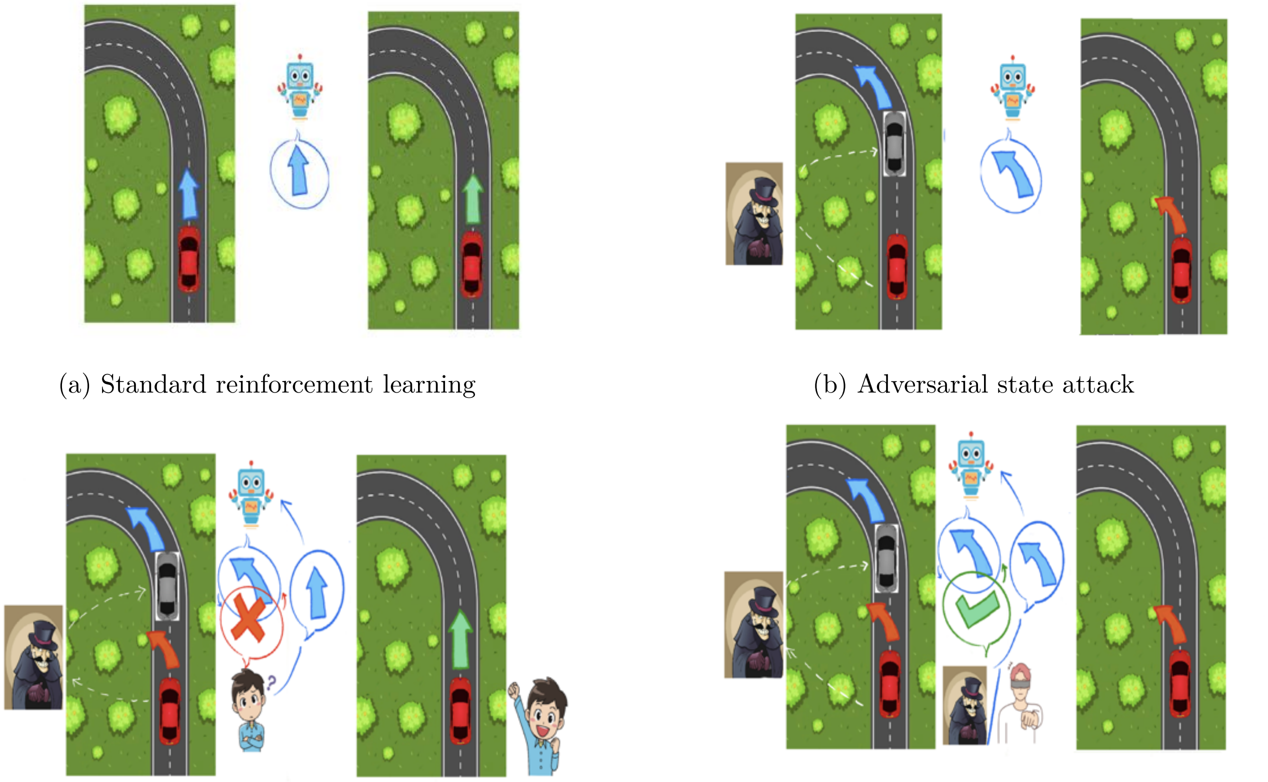
The recent development of deep neural networks has brought a promising outlook to tackling reinforcement learning problems in complex environments. A classical reinforcement problem consists of an agent (i.e. the robot that makes decisions) and an environment (i.e. the context where the agent is in). The agent and environment make interactions as shown in Figure 1a, where the environment displays a state/observation to the agent (e.g. location of the agent), then the agent makes an action, and in turn, receives a reward from the environment. This process can be mathematically modeled as a **Markov Decision Process** (MDP). MDP is a 5-tuple (S, A, P, R, γ) where:

S	set of states	$s \in S$
A	set of actions	$a \in A$
P	state transition function	$p(s'\|s,a) = Pr\{S_{t+1} = s'\|S_t = s, A_t = a\}$
R	reward function	$r(s', s, a) = \mathbb{E}[R_{t+1}\|S_{t+1} = s', S_t = s, A_t = a]$
γ	discount factor for future rewards	$\gamma \in [0, 1]$

The goal of the agent is to devise the best policy π (i.e. strategy of actions) that maximizes the reward quickly. Traditional reinforcement learning achieves this goal by attempting each action and visiting each state repeatedly. This method quickly becomes computationally expensive in an increasingly complex environment with many states. However, such computations can be optimized by deep learning algorithms like proximal policy optimization (PPO), deep deterministic policy gradient (DDPG), and deep Q networks (DQN). Hence, these algorithms led to the emergence of deep reinforcement learning (DRL).

One key challenge of DRL is to devise an effective reward function for each problem. However, Christiano, et al.[1] demonstrated the effectiveness of using (non-expert) human feedback in place of a reward function, which can be very challenging and inaccurate at times, during training Figure 1c. However, for the training to be effective, they rely on the assumption that human inputs are always “helpful” and accurate, which might not be the case in reality.

Another key challenge of DRL is noise and adversarial attacks. To protect against traditional state noises and adversarial attacks to the observations Figure 1b, Zhang, et al.[2] developed a state-adversarial Markov decision process (SA-MDP) and a theoretically principled policy regularization technique, both of which significantly improve the robustness of the agent under strong white-box adversarial attacks. In this paper, we propose a novel DRL approach that applies Zhang's approach against adversaries and noises to both states and human feedback Figure 1d. It aims to reduce the influence of erroneous measurements and adversarial noise, as well as (intentional and unintentional) erroneous human feedback, to achieve improvements in model robustness and training effectiveness.



(a) Standard reinforcement learning

(b) Adversarial state attack

(c) Human feedback from outside perspective

(d) Adversarial human feedback or human feedback from an inside perspective

Figure 1: In (a), the agent is able to observe the real car and direct it to travel straight, which results in a safe ride. In (b), perturbation is added to the state (the real car). Instead of observing the real car, the agent is tricked to observe the adversarial car and direct the real car to turn left, which results in a car crash. In (c), perturbation is added to the state again (the real car), but now a human gets involved. The agent is still tricked by the adversarial car to direct the real car to turn left, but the human teaches that the real car should actually travel straight. In this way, the agent learns and is eventually able to direct the real car to make the correct move (traveling straight), which results in a safe ride. In (d), perturbation is added to both the state (the real car) and the feedback (blind-folded/evil human agent), so the agent is tricked by the adversarial car and also cannot learn from human feedback. In this way, the agent continues to direct the real car to turn left, which results in a car crash. Scenario (d) is what our research aims to deal with.

2 Different Attack Types

In order to determine how well our combined model performs against the SA-MDP and human feedback models individually, we will test our model with many different attack variations. These will include some black-box attacks, such as random attacks in order to form a baseline. However, we will also test our model against many strong white-box attacks. Notably, PPO and DDPG are both very susceptible to Robust Sarsa and Maximal Action Difference attacks. Both of these attacks were shown to be the most damaging attacks in [2] with respect to these architectures. They are independent of the critics, otherwise known as the action-value (Q) or state value functions (V) depending on the framework, that were generated during training. Robust Sarsa uses the fact that the policy is fixed during testing to determine its corresponding critic by using temporal difference updates. Then, the gradient of this critic is used to determine the direction of attack. Maximal Action Difference attacks utilize the idea that, since the best policy against an adversary is the one that minimizes total variation distance (TVD), the best attack would be one that maximizes the TVD for a given policy. More specifically, it maximizes KL Divergence since total variation distance can be bounded by this quantity. Different algorithms are more susceptible to different attacks, so it will be important to test a variety. We suspect that the types of attacks that are effective against PPO and DDPG will also be effective against the A2C deep reinforcement neural network originally used to evaluate the effectiveness of the human feedback mechanism on Atari games. This susceptibility stems from the fact that all three of these networks involve a critic for evaluation. Therefore, even though the Robust Sarsa and Maximal Action Difference attacks have not been directly applied to A2C before, we suspect they will still be useful in our evaluation of an SA-MDP version of A2C.

The aforementioned attacks are specifically engineered to affect the state space, but the introduction of human feedback also incorporates more potential issues. One concept that was mentioned in the original paper surrounding this idea is that human feedback is imperfect. If the humans providing the feedback are not particularly good at the task, their input could hurt more than it could help the performance. Ideally, we would want the feedback to mainly be from experts on the subject or for there to be some metric of weighing the expert feedback higher than the novice feedback. This could help improve the performance of the original model and make our version more robust as well. A way to incorporate this is by utilizing Cohen's Kappa ratio. This statistic is generally used in supervised learning to see the degree to which two classifiers agree in their classifications of similar data. A similar idea could be extended to how much weight human feedback has in the reward function. For instance, if every annotator agreed that one possible move was better than the other, the human feedback would have a large influence on the reward function because there is consensus. While this is not a direct way to weigh expert opinions more than novice ones, it is more likely for a majority rules opinion to favor the correct answer as long as the probability of an average person presenting the correct answer is more than 0.5, which is a reasonable assumption. This value could also incorporate the number of people who gave their input, as a consensus of many people is more statistically significant than a consensus of only a few. To begin, these are the values that we will compare since they can be used to determine two important qualities. The first is whether or not human feedback improves performance against attack. The second is if the introduction of an SA-MDP framework enhances performance of a human feedback reward function even if there is no direct correlation between the reinforcement learning elements that are being effected.

3 Methodology

3.1 A robust model to adversarial changes to the state

In Zhang's work they introduced SA-MDP; SA-MDP is very similar to a traditional Markov decision process with the addition of the component B [2]. B is the set of all state perturbations an adversary could choose from. So rather then observe a state s, an agent would instead observe a perturbed state $\hat{s}$. Naturally this leads to a reduction of an agents performance as it makes convergence of the agent's state-value function $V_\pi(s)$ as well as the agent's policy function $\pi(a,s)$ difficult. An illustration of this concept is given via Figure 1b. The authors then developed the key concept of an optimal adversary. By developing an optimal adversary, the authors could measure the worst case scenario for the RL agent. Then, they could use the worst case scenario as a metric for building a regularizer for the loss function of the neural net contained in the deep RL algorithms. Using their regularizer, they were able to reduce the overall impact of adversarial state changes on the agents overall rewards. From Zhang we have the following bound on how much of an impact the optimal adversary can have on our agents payoffs,

$$\max_{s \in S}\{V_\pi(s) - \tilde{V}_{\pi \circ \nu^*(\pi)(s)}\} \leq \alpha \max_{s \in S} \max_{\hat{s} \in B(s)} (D_{TV}(\pi(\cdot|s), \pi(\cdot|\hat{s})), \tag{1}$$

with α being a constant the upper bound for the impact of the optimal adversary is then given by D_{Tv} or the total variation difference between the agent's policies [2]. Zhang goes further to develop a regularizer for DQN, PPO, and DDPG ($\mathcal{R}_{DQN}, \mathcal{R}_{PPO}, \mathcal{R}_{DDPG}$) which we apply in our model. It is important to note that the regularizers are always independent of the reward function. Thus, perturbations to the reward function are not mitigated by these regularizers; in turn, this also implies that the methods for mitigating adversarial changes to the reward function are ineffective for reducing the loss of performance caused by state perturbations.

3.2 Human feedback

Human preferences or feedback has added another layer of improvement on RL algorithms [1, 3, 4]. This improvement is achieved by fundamentally augmenting the reward function of the agents. To do this a human annotator observes two sequences of state action pairs with length N

$$\sigma_1 = ((s_0, a_0), (s_1, a_1), (s_2, a_2), \dots, (s_N, a_N)),\ \sigma_2 = ((s'_0, a'_0, (s'_1, a'_1), (s'_2, a'_2), \dots, (s'_N, a'_N)).$$

The human then chooses the better of the two based on an implicit metric (i.e. which looks more like X). The human preferences are then stored as a tuple $(\sigma_1, \sigma_2, \mu)$ in a data set D; where μ represents the distribution over option 1 and 2. This process is then implemented by learning a reward function such that

$$\hat{r} = -\min_{\hat{r}} \sum_{(\sigma_1,\sigma_2,\mu)\in D} \mu(1)\log\Big(\hat{P}(\sigma_1 > \sigma_2)\Big) + \mu(2)\log\Big(\hat{P}(\sigma_2 > \sigma_1)\Big). \tag{2}$$

Here, $\hat{P}$ represents the probability a human chooses segment 1 over segment 2 based on perceived awards by the human annotator. According to Christiano, this human feedback produced a more "elegant" result and allowed the agents to learn more complex policies for tasks that were difficult to build an effective reward function for [1].

3.3 Robust human feedback

As mentioned in the prior section, there are several drawbacks to human feedback. First and foremost, humans are imperfect and make errors. Second, some humans are intentionally malicious. Third, if there is an adversary, there is potential for humans to see tainted results and reinforce the adversary's attacks. These considerations are of particular importance since by this metric in Equation (2), the agent may learn on a tainted reward function. For the last part, we can consider two perspectives: an insider's point of view where the human annotator only has access to the agent's observations, and an outsider's point of view where the annotator has access to either the true states or both the true state and the agent's observations.

3.3.1 Outside perspective

With an outside perspective, we have the potential for unintentional errors or intentional errors; however, human preference is expected to be more robust under an augmented state in this case since the agent will get feedback from the true state rather than the false state. In this case, we have unintentional errors from a human behaving at a lower efficiency than the true reward. We define a new parameter

$$\rho(\sigma_1) = \sum_{(s,a)\in\sigma_1} r_t(s,a)$$

where r_t is a pre-built reward function. Then we define weights

$$k_i = w\left|\frac{|\rho(\sigma_i)| - |\min(\rho(\sigma_1),\rho(\sigma_2))|}{\max|\rho(\sigma_1)|,|\rho(\sigma_2)|}\right|$$

where w is a hyperparameter used as a confidence measure of how accurate our estimated reward function is. We have defined k_i such that the model is punished for choosing against the approximate reward. Thus, if the human annotators are very different from the approximate solution, then the model will have a high loss. We can apply this to a new learned reward function:

$$\hat{r} = -\min_{\hat{r}} \sum_{(\sigma_1,\sigma_2,\mu)\in D} (k_1 + \mu(1)) \log\Big(\hat{P}(\sigma_1 > \sigma_2)\Big) + (k_2 + \mu(2)) \log\Big(\hat{P}(\sigma_2 > \sigma_1)\Big)) \tag{3}$$

. An additional change we could make is using Cohan's Kappa as mentioned in the chapter on attacks. Thus we would record $(\sigma_1, \sigma_2, \mu, \kappa) \in \mathcal{D}$. We then discard the results that have a low Cohan's kappa, and reduce the impact of adversaries in the human annotators. Finally, we can use the learned reward function to add to the traditional reward function

$$r_{true} = \iota r_t + (1 - \iota)\hat{r}$$

where ι is a hyperparameter. Thus, r_{true} is a weighted average of the learned reward function and the estimated reward function. If we expect that humans will out perform the estimated reward function, we would make $\iota \to 0$ and vice versa if we expect that the true reward function is accurate.

3.3.2 Inside perspective

A nice side effect of our new reward function Equation (3) is that it should also be applicable to results where the human annotators are also impacted by the adversarial perturbations on the state space. Moreover, this result is because the difference in the reward function will be significantly large and discourage the model from using the human preferences. In fact, the model would likely put more emphasis on the estimate reward function instead.

4 Discussion

In this project we introduce a newly designed reward function. Our goal is for this reward function to be robust to state changes as well as bad human influencers and annotators. Inspired by Christiano, et al.[1] and Zhang, et al.[2], the aforementioned robustness is achieved by combining the learned reward function from [1] with a regularizer added to the policy function from [2]. Potential downsides to our approach do exist. The introduction of this reward function increases the computational complexity of the model. Also, it adds another hyperparameter that would need tuning. While these drawbacks are somewhat worrisome, they are minimal and as expected. Moreover, the applications that exist for agents with such a reward function are powerful. Take a team of scientists at NASA that are operating a semi-autonomous Mars rover, they may want the rover to take a left due to a certain input from one of its sensors. Unbeknownst to the scientists, that particular sensor is faulty making the desired left turn untenable. Since the rover was trained with this reward function, it is robust to this noise and bad human influence, electing not to perform such a turn and preventing the rover from damage. Not only would such a thing save the time invested by the NASA's scientists on the project but also the American taxpayer's money. A future extension for our work, rather obviously, is to implement and experiment with this reward function. Its wide ranging applications to complex problems such as a Mars rover or driver-less cars make for fascinating follow-up opportunities with high potential impact.

References

[1] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[2] 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.

[3] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[4] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.