Efficient Adversarial Training without Attacking: Worst-Case-Aware Robust Reinforcement Learning

DEPARTMENT OF COMPUTER SCIENCE

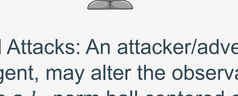
Yongyuan Liang*, Yanchao Sun*, Ruijie Zheng, Furong Huang

UNIVERSITY OF MARYLAND

Motivation and Background

- ❖ A well-trained DRL policy can be particularly vulnerable to bounded perturbations on input observations.
- ❖ There is a crucial need to improve the robustness of RL policies against adversarial attacks, especially the worst-case attacks.
- ➢ Besides promoting the robustness of DNN approximators, it is also important to learn a policy with stronger intrinsic robustness.

Both the green policy and the red policy arrive home without rock collision, when there is no attack.



The green policy is more robust to adversarial attacks since it stays away from the 💣

- ❑ Bounded Adversarial Attacks: An attacker/adversary, during the deployment of the agent, may alter the observation s_t to $\widetilde{s_t} \in \mathcal{B}_\epsilon(s_t)$,
- ❑ where $\widetilde{s_t} \in \mathcal{B}_\epsilon(s_t)$ is a l_p norm ball centered at s_t with radius ϵ.

Challenges

➢ How to correctly characterize the long-term vulnerability?

❖ Existing Regularization-based robust methods

- Regularizes the policy network (improve DNN robustness) to output similar actions under state perturbations
- **Neglect the intrinsic vulnerability** under the environment dynamics, and thus may still fail under strong attacks.

➢ How to efficiently train a robust agent without requiring much more effort than vanilla training?

❖ **SOTA** Alternating Training with Learned Adversaries (ATLA)

- Alternately trains an RL agent and an RL attacker
- Requires extra samples from the environment, and the attacker's RL problem may even be more difficult and sample expensive to solve.

‼ Double the computational burden and sample complexity

➢ Our GOAL: efficiently improves the long-term robustness of RL

Proposed Method

❖ Mechanism 1: Worst-attack Value Estimation

💡 ***Worst-attack Bellman Operator***:

$$(\underline{\mathcal{T}}^\pi Q)(s,a) \coloneqq \mathbb{E}_{S' \backsim P(s,a)}[R(s,a) + \gamma min_{a' \epsilon \mathcal{A}_{adv}(s',\pi)} Q(s',a')$$

💡 ***Estimating worst-attack value by minimizing the estimation loss***:

$$\mathcal{L}_{est}\left(\underline{Q}_\phi^\pi\right) \coloneqq \frac{1}{N}\sum_{t=1}^{N}(\underline{y}_t - \underline{Q}_\phi^\pi(s_t,a_t))^2,$$
$$where\ \underline{y}_t = r_t + \gamma min_{a' \epsilon \mathcal{A}_{adv}(s_{t+1},a')} \underline{Q}_\phi^\pi(s_{t+1},a')$$

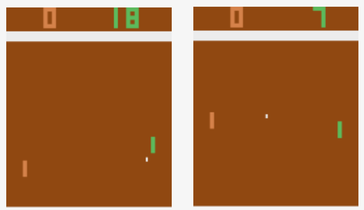
❖ Mechanism 2: Worst-case-aware Policy Optimization

💡 ***Minimizing the worst-attack policy loss below***:

$$\mathcal{L}_{wst}\left(\pi_\theta; \underline{Q}_\phi^\pi\right) \coloneqq -\frac{1}{N}\sum_{t=1}^{N}\sum_{a \epsilon \mathcal{A}} \pi_\theta(a|s_t)\, \underline{Q}_\phi^\pi(s_t,a),$$
$$where\ \underline{Q}_\phi^\pi\ is\ the\ worst\ attack\ critic\ learn\ via\ \mathcal{L}_{est}$$

💡 ***We illustrate how to implement $\mathcal{L}_{wst}$ for PPO and DQN***

❖ Mechanism 3: Value-enhanced State Regularization



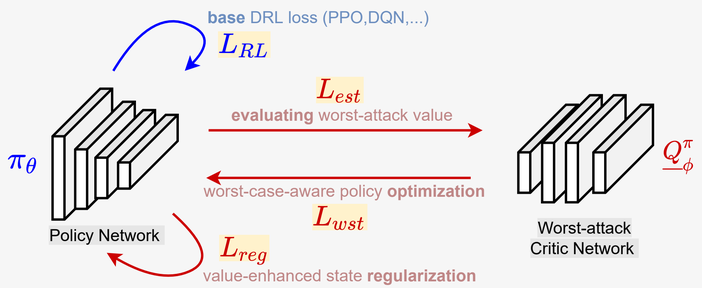
✅ Characterize state importance of $s \in \mathcal{S}$:
$w(s) =$
$max_{a_1 \in \mathcal{A}} Q^\pi(s,a_1)$
$- min_{a_2 \in \mathcal{A}} Q^\pi(s,a_2)$

✅ States in Pong with (left) high weight $w(s)$ and (right) low weight $w(s)$

💡 ***By incorporating the state importance weight w(s), we regularize the policy network loss:***

$$\mathcal{L}_{reg}(\pi_\theta) \coloneqq \frac{1}{N}\sum_{t=1}^{N} w(s_t) max_{\widetilde{s_t} \in \mathcal{B}_\epsilon(s_t)} Dist(\pi_\theta(s_t), \pi_\theta(\widetilde{s_t})),$$

➢ WocaR-RL: A Generic Robust Training Framework

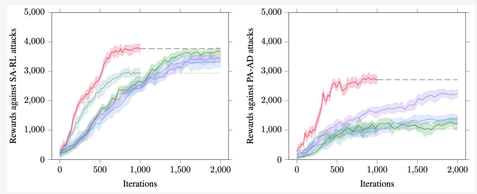


Experiment Results

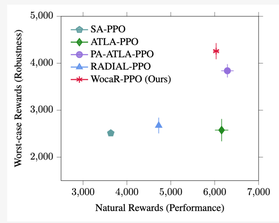
**See our paper for full experiment results on 4 MuJoCo environments and 4 Atari games*

❖ Can WocaR-RL learn policies with better robustness?

State-of-the-art robustness under existing strong attacks (on Walker2d)



❖ Can WocaR-RL maintain natural performance?



Average episode natural rewards v.s. Average worst rewards (on Halfcheetah) WocaR-PPO gains more robustness without losing too much natural performance

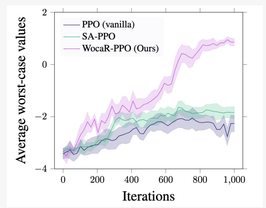
❖ Can WocaR-RL learn more efficiently during training?

Model	Hopper		Ant	
	Time (h)	Steps(m)	Time (h)	Steps (m)
SA-PPO	3.0	2.0	8.9	10.0
ATLA-PPO	5.6	5.0	12.8	10.0
PA-ATLA-PPO	5.2	5.0	12.3	10.0
RADIAL-PPO	3.2	4.0	10.2	10.0
WocaR-PPO (Ours)	**2.3**	**2.0**	**8.7**	**7.5**

❖ Verifying Algorithm Effectiveness

**Detailed ablation studies for each part of our algorithm are included.*

Worst-attack value estimation matches the trend of actual worst-case reward.



Our agent learns more interpretable "robust behaviors": lower down its body during walking 😈

Easter egg 🐣🌟