# From Stochastic Games to Robust Action Reinforcement Learning [1]

TT Nguyen

A2I2@Deakin

July 19, 2019

[1] Mostly based on[Maitra and Parthasarathy, 1970, Tessler et al., 2019]; see Reference for complete list of related works.

# Contents

1. Motivation

2. Background
   - Markov Decision Process (MDP)
   - Stochastic games (two-player zero-sum games)

3. From Stochastic Game to Robust Action RL [Tessler et al., 2019]
   - Probabilistic Action Robust MDP
   - Noisy Action Robust MDP
   - Heuristics to scale Action Robust RL: Actor-Critic-Adversary

4. Experiment

5. Discussion

# Motivation

- (general) Reinforcement Learning:
  - Models sequential decision making under uncertainty (with vast applications in practice)
  - Learn to maximize expected reward via interactions with an environment.
- But what if the environment dynamic changes over time?
  - e.g., autonomous vehicles: some environment variables such as vehicle mass, tire pressure and road conditions might vary over time.
- A robust algorithm should
  take into account this **perturbation** during optimization process.
- How to make a RL algorithm generalize under small perturbation? Hint: incorporating zero-sum game into decision making:
  - Environment dynamic change as an adversary.
  - The goal is to perform well even under the most adversarial scenario.

1. Motivation
2. Background
   - Markov Decision Process (MDP)
   - Stochastic games (two-player zero-sum games)
3. From Stochastic Game to Robust Action RL [Tessler et al., 2019]
   - Probabilistic Action Robust MDP
   - Noisy Action Robust MDP
   - Heuristics to scale Action Robust RL: Actor-Critic-Adversary
4. Experiment
5. Discussion

1. Motivation

2. Background
   - Markov Decision Process (MDP)
   - Stochastic games (two-player zero-sum games)

3. From Stochastic Game to Robust Action RL [Tessler et al., 2019]
   - Probabilistic Action Robust MDP
   - Noisy Action Robust MDP
   - Heuristics to scale Action Robust RL: Actor-Critic-Adversary

4. Experiment

5. Discussion

# MDP: A bit of history

- **MDP**: First introduced by [Bellman, 1957], solved with linear programming.

Figure: Richard E. Bellman 1920-1984 [image src: Wikipedia]

- [Howard, 1960] introduced **Policy Iteration** to solve MDPs, but its worst-case analysis remained unsolved for approx. 25 years!
- Most recent bound [Hansen et al., 2013] on Howard's Policy Iteration: $O(\frac{m}{1-\gamma} \log \frac{n}{1-\gamma})$ where $m$ is the number of states, and $n$ is number of actions.

# MDP: notation

- Characterized by the 5-tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$
  - $\mathcal{S}$: State space.
  - $\mathcal{A}$: Action space (compact metric space).
  - $P(s'|s, a)$: Transition kernel (weakly continuous in $a$).
  - $R(s, a)$: Reward function (continuous in $a$).
  - $\gamma \in (0, 1)$: Discounted factor.
- A stationary policy $\pi : \mathcal{S} \to \mathcal{A}$.
- Value function of a policy $\pi$:

$$v^\pi(s) = \mathbb{E}^\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s \right], \forall s \in \mathcal{S} \quad (1)$$

- Goal:

$$\pi^*(s) \in \arg\max_{\pi \in \mathcal{P}(\Pi)} v^\pi(s), \forall s \in \mathcal{S} \quad (2)$$

# MDP: Fundamental Result

- Bellman operator:

$$T^{\pi} : \mathbb{R}^{|\mathcal{S}|} \rightarrow \mathbb{R}^{|\mathcal{S}|},\ \ T^{\pi} v = r^{\pi} + \gamma P^{\pi} v$$

where $P^{\pi}_{i,j} = P(s_{t+1} = i | s_t = j, a_t = \pi(s_t)), r^{\pi}(s) = r(s, \pi(s))$.

> **Theorem**: There exists an optimal policy which is stationary and deterministic, i.e., $\pi^* \in \Pi$.

# MDP: Policy Iteration [Howard, 1960]

**Policy iteration (using iterative policy evaluation)**

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
   $\quad \Delta \leftarrow 0$
   $\quad$ For each $s \in \mathcal{S}$:
   $\quad\quad v \leftarrow V(s)$
   $\quad\quad V(s) \leftarrow \sum_{s',r} p(s', r \,|\, s, \pi(s))\big[r + \gamma V(s')\big]$
   $\quad\quad \Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   *policy-stable* $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   $\quad$ *old-action* $\leftarrow \pi(s)$
   $\quad \pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \,|\, s, a)\big[r + \gamma V(s')\big]$
   $\quad$ If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

Figure: Figure credit: [Sutton and Barto, 1998]

1. Motivation
2. Background
   - Markov Decision Process (MDP)
   - Stochastic games (two-player zero-sum games)
3. From Stochastic Game to Robust Action RL [Tessler et al., 2019]
   - Probabilistic Action Robust MDP
   - Noisy Action Robust MDP
   - Heuristics to scale Action Robust RL: Actor-Critic-Adversary
4. Experiment
5. Discussion

# Stochastic game: A bit of history

- First introduced by [Shapley, 1953] (a bit earlier than MDP)

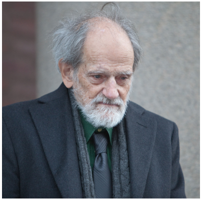


Figure: Lloyd Shapley 1923-2016, Nobel Prize, "the greatest game theorist of all time." [image src: Wikipedia]

- [Maitra and Parthasarathy, 1970] extended Sharley's games to infinite action space and state space.
- [S.S. Rao, 1973] introduced Policy Iteration for stochastic games.
- [Hansen et al., 2013] gave a convergence bound of PI for stochastic games: $O(\frac{m}{1-\gamma}\log\frac{n}{1-\gamma})$.

# Stochastic game

- Widely used to model long-term sequential decision making in stochastic and adversarial environments.
- Two players:
  - Player 1: plays $a \in \mathcal{A}$ according to policy $\pi$.
  - Player 2: plays $\bar{a} \in \bar{\mathcal{A}}$ according to policy $\bar{\pi}$
  - Reward function: $r(s, a, \bar{a})$.
  - Player 1 attempts to <u>maximize</u> the total expected discounted reward.
  - Player 2 attempts to <u>minimize</u> the total expected discounted reward.
- Transition kernel $P(s'|s, a, \bar{a})$ depend on both players.
- Value function (total expected gain):

$$v^{\pi,\bar{\pi}}(s) = \mathbb{E}^{\pi,\bar{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, \bar{a}_t)|s_0 = s\right], \forall s \in \mathcal{S} \tag{3}$$

# Stochastic games (con't)

- An optimal policy $\pi^*$ for player 1:

$$v^{\pi^*,\bar{\pi}'}(s) \geq \inf_{\bar{\pi}} \sup_{\pi} v^{\pi,\bar{\pi}}(s), \forall s, \bar{\pi}'. \tag{4}$$

- An optimal policy $\bar{\pi}^*$ for player 2:

$$v^{\pi',\bar{\pi}^*}(s) \leq \sup_{\pi} \inf_{\bar{\pi}} v^{\pi,\bar{\pi}}(s), \forall s, \pi'. \tag{5}$$

$$\texttt{Dualty\_gap}(s) = \inf_{\bar{\pi}} \sup_{\pi} v^{\pi,\bar{\pi}}(s) - \sup_{\pi} \inf_{\bar{\pi}} v^{\pi,\bar{\pi}}(s) \geq 0 \tag{6}$$

# Stochastic games (con't)

Theorem ([Maitra and Parthasarathy, 1970]): If

- $\mathcal{S}, \mathcal{A}$ and $\bar{\mathcal{A}}$ are compact metric spaces.
- $P(s'|s, a, \bar{a})$ weakly continuous on $\mathcal{S} \times \mathcal{A} \times \bar{\mathcal{A}}$.
- $r(s, a, \bar{a})$ is bounded (?) and continuous on $\mathcal{S} \times \mathcal{A} \times \bar{\mathcal{A}}$.

Then Nash-equilibrium exists for the stochastic game:

$$v^*(s) = \max_{\pi} \min_{\bar{\pi}} v^{\pi,\bar{\pi}}(s) = \min_{\bar{\pi}} \max_{\pi} v^{\pi,\bar{\pi}}(s) \tag{7}$$

# Policy iteration to solve two-player zero-sum game [Hansen et al., 2013]

Alternate between two stages until convergence condition:

- Step 1: Given a fixed adversary policy $\bar{\pi}_k$, calculate the optimal counter policy (similar to solving for single-agent optimal policy)

$$\pi_k \in \arg\max_{\pi \in \Pi} v^{\pi,\bar{\pi}_k} \tag{8}$$

- Step 2: Performe 1-step minimax policy iteration

$$\bar{\pi}_{k+1} \in \arg\min_{\bar{\pi} \in \mathcal{P}(\Pi)} \max_{\pi \in \Pi} T^{\pi,\bar{\pi}} v^{\pi_k,\bar{\pi}_k} \tag{9}$$

Convergence guarantee:

$$\|v_k - v^*\|_\infty \leq \gamma \|v_{k-1} - v^*\|_\infty \tag{10}$$

# Main contribution of [Tessler et al., 2019]

- Specify specific instances of stochastic games for designing robustness and generalization in RL.
- Incorporate Policy Iteration for solving minimax problem.
- Experimental prototype for testing robustness (i.e., model uncertainty in this case)
- Connection to distributional robustness MDP.

# Probabilistic Action Robust MDP (PR-MDP)

- **Category**: A special instance of zero-sum game.
- **Scenario**: Stochastic perturbation in the policy space, i.e., With a probability $\alpha$, an adversary takes control and perform the most adversarial action:

$$\boxed{\pi^{\texttt{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)(a|s) = (1-\alpha)\pi(a|s) + \alpha\bar{\pi}(a|s)}$$

- **Implication**: Someone suddenly pushes the robot

# Probabilistic Action Robust MDP (PR-MDP) (con't)

- PR-MDP is a special instance of stochastic games, thus exists a Nash equilibrium.
- Due to the special structure of PR-MDP problem, the optimal policies are deterministic:

$$v^*_{P,\alpha} = \max_{\pi \in \Pi} \min_{\bar{\pi} \in \Pi} v^{\pi^{\mathtt{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)} = \min_{\bar{\pi} \in \Pi} \max_{\pi \in \Pi} v^{\pi^{\mathtt{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)} \tag{11}$$

# Policy iteration for PR-MDP

- Due to the special structure of PR-MDP, PI for PR-MDP is easier than the general two-player zero-sum game:
  - Update for adversary policy does not involve minimax but still converges to the optimal policy

**Algorithm 1** Probabilistic Robust PI

**Initialize:** $\alpha, \bar{\pi}_0, k = 0$
**while** not changing **do**
  $\pi_k \in \arg\max_{\pi'} v^{\pi^{\text{mix}}_{P,\alpha}(\pi', \bar{\pi}_k)}$
  $\bar{\pi}_{k+1} \in \arg\min_{\bar{\pi}} r^{\bar{\pi}} + \gamma P^{\bar{\pi}} v^{\pi^{\text{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)}$
  $k \leftarrow k + 1$
**end while**
**Return** $\pi_{k-1}$

$$\bar{\pi}_{k+1}(s) = \arg\min_{a \in \mathcal{A}} \left( r(s, a) + \gamma \sum_{s'} p(s'|s, a) v^{\pi^{\text{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)}(s') \right) \tag{12}$$

# Soft Policy Iteration for PR-MDP

- Use Frank-Wolfe update for adversary policy

**Algorithm 2** Soft Probabilistic Robust PI

**Initialize:** $\alpha, \eta, \bar{\pi}_0, k = 0$
**while** criterion is not satisfied **do**
  $\pi_k \in \arg\max_{\pi'} v^{\pi^{\text{mix}}_{P,\alpha}(\pi', \bar{\pi}_k)}$
  $\bar{\pi} \in \arg\min_{\bar{\pi}'} \left\langle \bar{\pi}', \nabla_{\bar{\pi}} v^{\pi^{\text{mix}}_{P,\alpha}(\pi_k, \bar{\pi})} \big|_{\bar{\pi}=\bar{\pi}_k} \right\rangle$
  $\bar{\pi}_{k+1} = (1-\eta)\bar{\pi}_k + \eta\bar{\pi}$
  $k \leftarrow k+1$
**end while**
**Return** $\pi_{k-1}$

# Soft Policy Iteration for PR-MDP

Turns out the soft policy iteration is equivalent to the policy iteration:

**Proposition 2.** *Let $\pi, \bar{\pi}$ be general policies. Then,*

$$\begin{aligned}&\arg\min_{\bar{\pi}' \in \Pi} r^{\bar{\pi}'} + \gamma P^{\bar{\pi}'} v^{\pi^{\text{mix}}_{P,\alpha}(\pi,\bar{\pi})} \\ &= \arg\min_{\bar{\pi}' \in \Pi} \left\langle \bar{\pi}', \nabla_{\bar{\pi}} v^{\pi^{\text{mix}}_{P,\alpha}(\pi,\tilde{\pi})} \big|_{\tilde{\pi}=\bar{\pi}} \right\rangle.\end{aligned}$$

# Noisy Action Robust MDP (NR-MDP)

- **Scenario**: Stochastic perturbation in the action space, i.e., Each time an agent takes an action, the adversary adds a small perturbation to the action:

$$\pi^{\texttt{mix}}_{N,\alpha}(\pi, \bar{\pi})(a|s) = \mathbb{E}_{b\sim\pi(.|s),\bar{b}\sim\bar{\pi}(.|s)}\left[1_{a=(1-\alpha)b+\alpha\bar{b}}\right] \tag{13}$$

- **Implication**: Accounting for execution error during control, mass uncertainty (the robot is heavier or slighter during test).

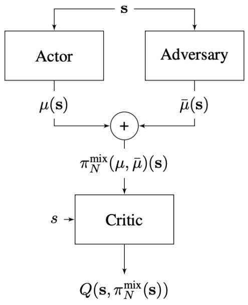
# NR-MDP

Two distince features between Probabilistic action MDP and noisy action MDP:

- Probabilistic action MDP:
  - Optimal counter policy: <u>deterministic</u>
  - 1-step update for adversary: <u>involve solving *min*</u>, i.e., $\arg\min_{\bar{\pi}\in\Pi} T^{\bar{\pi}} v^{\pi^{\mathtt{mix}}_{P,\alpha}(\pi_k,\bar{\pi}_k)}$
- Noisy action MDP:
  - Optimal counter policy: <u>stochastic</u>
  - 1-step update for adversary: <u>involve solving *minimax*</u>, i.e., $\arg\min_{\bar{\pi}\in\mathcal{P}(\Pi)} \max_{\pi\in\Pi} T^{\pi,\bar{\pi}} v^{\pi^{\mathtt{mix}}_{P,\alpha}(\pi_k,\bar{\pi}_k)}$

1. Motivation
2. Background
   - Markov Decision Process (MDP)
   - Stochastic games (two-player zero-sum games)
3. From Stochastic Game to Robust Action RL [Tessler et al., 2019]
   - Probabilistic Action Robust MDP
   - Noisy Action Robust MDP
   - Heuristics to scale Action Robust RL: Actor-Critic-Adversary
4. Experiment
5. Discussion

# Heuristics to scale action robust RL




- To scale beyond tabular case
- Similar to Actor-Critic framework:
    - **Actor**: parameterized policy for Player 1.
    - **Adversary**: parameterized policy for Player 2 (adversarial player)
    - Train the Actor for N gradient steps followed by a single adversarial step.
    - A **Critic** is trained to evaluate the Q-value of the joint policy.

# Action-Robust DDPG algorithm

**Action Robust Reinforcement Learning and Applications in Continuous Control**

**Algorithm 5** Action-Robust DDPG

**Input:** Actor update steps ($N$), uncertainty value $\alpha$ and discount factor $\gamma$
Randomly initialize critic network $Q(\mathbf{s}, \mathbf{a}; \phi)$, actor $f(\mathbf{s}; \theta)$ and adversary $\bar{f}(\mathbf{s}; \bar{\theta})$
Initialize target networks with weights $\phi^-, \theta^-, \bar{\theta}^-$
Initialize replay buffer $R$
**for** episode in $0 \ldots M$ **do**
  Receive initial state $\mathbf{s}_0$
  **for** t in $0 \ldots T$ **do**

Sample an action from the mixed policy

$$\boxed{\text{Sample action } \mathbf{a}_t = \begin{cases} f(\mathbf{s}; \theta_\pi) \text{ w.p. } (1-\alpha) \text{ and } \bar{f}(s; \theta_{\bar{\pi}}) \text{ otherwise} & \text{, PR-MDP} \\ (1-\alpha) f(\mathbf{s}; \theta_\pi) + \alpha \bar{f}(\mathbf{s}; \bar{\theta}_{\bar{\pi}}) & \text{, NR-MDP} \end{cases}}$$

   $\tilde{\mathbf{a}}_t = \mathbf{a}_t$ + exploration noise
   Execute action $\tilde{\mathbf{a}}_t$ and observe reward $r_t$ and new state $s_{t+1}$
   Store transition $(\mathbf{s}_t, \tilde{\mathbf{a}}_t, r_t, \mathbf{s}_{t+1})$ in $R$
   **for** i in $0 \ldots N$ **do**
    Sample batch from replay buffer

Compute the gradient policy of the actor and update it for N steps

$$\boxed{\text{Update actor: } \theta \leftarrow \begin{cases} \nabla_\theta (1-\alpha) Q(\mathbf{s}, f(\mathbf{s}; \theta)) & \text{, PR-MDP} \\ \nabla_\theta Q(\mathbf{s}, (1-\alpha) f(\mathbf{s}; \theta) + \alpha \bar{f}(\mathbf{s}; \bar{\theta})) & \text{, NR-MDP} \end{cases}}$$

Compute the gradient policy of the critic and update it for N steps

$$\boxed{\text{Update critic: } \phi \leftarrow \begin{cases} \nabla_\phi \| r + \gamma [(1-\alpha) Q(\mathbf{s}', f(\mathbf{s}'; \theta^-)) + \alpha Q(\mathbf{s}', f(\mathbf{s}'; \bar{\theta}^-))] \|_2^2 & \text{, PR-MDP} \\ \nabla_\phi \| r + \gamma [Q(\mathbf{s}', (1-\alpha) f(\mathbf{s}'; \theta^-) + \alpha f(\mathbf{s}'; \bar{\theta}^-))] \|_2^2 & \text{, NR-MDP} \end{cases}}$$

   **end for**
   Sample batch from replay buffer

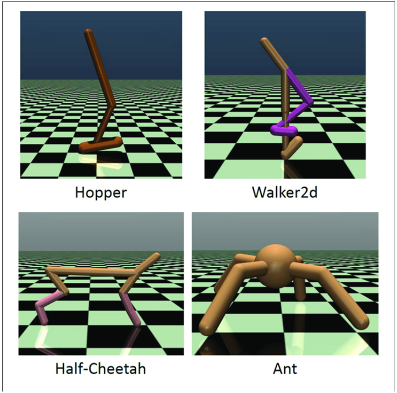
1-step update for adversary

$$\boxed{\text{Update adversary: } \bar{\theta} \leftarrow \begin{cases} \nabla_{\bar{\theta}} \alpha Q(\mathbf{s}, \bar{f}(\mathbf{s}; \bar{\theta})) & \text{, PR-MDP} \\ \nabla_{\bar{\theta}} Q(\mathbf{s}, (1-\alpha) f(\mathbf{s}; \theta) + \alpha \bar{f}(\mathbf{s}; \bar{\theta})) & \text{, NR-MDP} \end{cases}}$$

   Update critic
   Update the target networks:

# Experimental prototype

- Mujoco domain: continuous control problems, e.g., Hopper-v2: Make a two-dimensional one-legged robot hop forward as fast as possible.



# Experimental prototype (con't)

- Prototype for **model uncertainty**:
  Change the robot mass parameter during evaluation (note that when robot mass changes, the physic laws result in a different environment dynamic)
    - First, train the algorithm on 5 different seeds.
    - Second, evaluate the final policy on 100 episodes, without adversarial disturbance on different robot mass.

# Experiment 1: Hyperparameter Ablation in a single domain

- Evaluate in Hopper-v2
- In PR-MDP: small $\alpha \rightarrow$ more optimistic; large $\alpha \rightarrow$ more conservative
- In NR-MDP, robust behaviour is not stable because no clear correlation btw $\alpha$ and robust performance.
- Adversary induces enough noise for exploration.

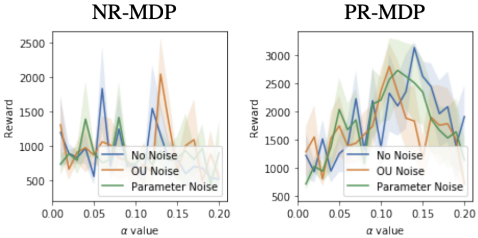



*Figure 1.* Hopper-v2: Performance of both the NR and PR-MDP criteria as a function of the uncertainty $\alpha$.

# Experiment 2: Test robust performance across unseen domains

- Use the best hyperparamter from experiment 1 to test on other domains
- Both NR-MDP and PR-MDP outperform baseline.
- Hyperparameters transfer quite well across domains.

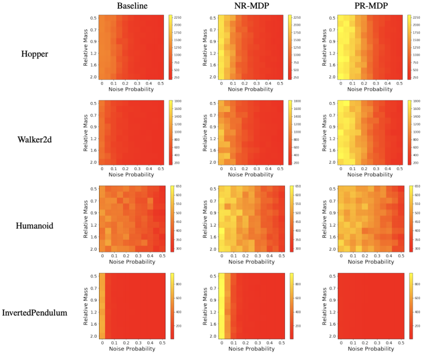



*Figure 2.* Robustness to model uncertainty. Noise probability denotes the probability of a randomly sampled noise being played instead of

# Experiment 3: Test off-policy action robustness

- The goal is to confirm if the improved robust performance comes from the added perturbance or from solving the minimax.
- Prototype: Instead of sampling action from the mixed policy, sample from the Actor's policy (and the rest are the same):

**Algorithm 5** Action-Robust DDPG

**Input:** Actor update steps ($N$), uncertainty value $\alpha$ and discount factor $\gamma$
Randomly initialize critic network $Q(\mathbf{s}, \mathbf{a}; \phi)$, actor $f(\mathbf{s}; \theta)$ and adversary $\bar{f}(\mathbf{s}; \bar{\theta})$
Initialize target networks with weights $\phi^-, \theta^-, \bar{\theta}^-$
Initialize replay buffer $R$
**for** episode in $0 \ldots M$ **do**
  Receive initial state $\mathbf{s}_0$
  **for** t in $0 \ldots T$ **do**

Instead of sample from the mixed policy, sample from the Actor's policy

$$\boxed{\text{Sample action } \mathbf{a}_t = \begin{cases} f(\mathbf{s}; \theta_\pi) \text{ w.p. } (1-\alpha) \text{ and } f(s; \theta_{\bar{\pi}}) \text{ otherwise} & \text{, PR-MDP} \\ (1-\alpha) f(\mathbf{s}; \theta_\pi) + \alpha \bar{f}(\mathbf{s}; \bar{\theta}_{\bar{\pi}}) & \text{, NR-MDP} \end{cases}}$$

    $\tilde{\mathbf{a}}_t = \mathbf{a}_t$ + exploration noise
    Execute action $\tilde{\mathbf{a}}_t$ and observe reward $r_t$ and new state $s_{t+1}$
    Store transition $(\mathbf{s}_t, \tilde{\mathbf{a}}_t, r_t, \mathbf{s}_{t+1})$ in $R$
    **for** i in $0 \ldots N$ **do**
      Sample batch from replay buffer
      Update actor:

$$\theta \leftarrow \begin{cases} \nabla_\theta (1-\alpha) Q(\mathbf{s}, f(\mathbf{s}; \theta)) & \text{, PR-MDP} \\ \nabla_\theta Q(\mathbf{s}, (1-\alpha) f(\mathbf{s}; \theta) + \alpha \bar{f}(\mathbf{s}; \bar{\theta})) & \text{, NR-MDP} \end{cases}$$

      Update critic:

$$\phi \leftarrow \begin{cases} \nabla_\phi \| r + \gamma [(1-\alpha) Q(\mathbf{s}', f(\mathbf{s}'; \theta^-)) + \alpha Q(\mathbf{s}', f(\mathbf{s}'; \bar{\theta}^-))] \|_2^2 & \text{, PR-MDP} \end{cases}$$

# Experiment 3: Test off-policy action robustness (con't)

- It seems both adversarial exploration and minimax operator are important for improved robustness performance.

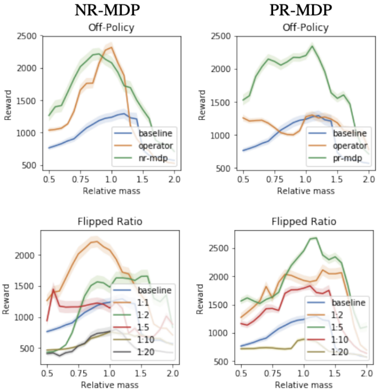



*Figure 3.* Diving Deeper: (Up) Testing Off-Policy Action-Robustness, and (Down) Solving the MaxMin operator.

# Discussion

- Though the paper is a natural approach of game theory for robustness in MDP, this paper is still novel in a sense that: This is the first to incorporate Policy Iteration with Minimax framework and design a good experimental prototype.
- A new perspective: Adversarial disturbance is a "**structural**" noise induced to simulate the worst-case conditions, thus encourage robustness (and even exploration).
- Game theory provides a nice formulation of adversary and convergence conditions, but does not provide a way to compute optimal policy itself.
  - Soften (relax) the conditions usually helps for solving/approximating optimal policies, e.g., work on a nice space which can provide tractable solution
- Related to *distributional robust optimization* and *optimal transport*.

Bellman, R. E. (1957).
Dynamic programming.
*Princeton University Press.*

Hansen, T. D., Miltersen, P. B., and Zwick, U. (2013).
Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor.
*J. ACM*, 60(1):1:1–1:16.

Howard, R. (1960).
Dynamic programming and markov processes.
*MIT Press.*

Maitra, A. and Parthasarathy, T. (1970).
On stochastic games.
*Journal of Optimization Theory and Applications*, 5(4):289–300.

Shapley, L. (1953).
Stochastic games.
*Proc. Nat. Acad. Sci. U.S.A.*, 39:1095–1100.

S.S. Rao, R. Chandrasekaran, K. N. (1973).
Algorithms for discounted games.
*Journal of Optimization Theory and Applications*, pages 627–637.

Sutton, R. S. and Barto, A. G. (1998).
Reinforcement learning: An introduction.
*IEEE Trans. Neural Networks*, 9(5):1054–1054.

Tessler, C., Efroni, Y., and Mannor, S. (2019).
Action robust reinforcement learning and applications in continuous control.
In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6215–6224.