Action Robust Reinforcement Learning and Applications in Continuous Control

Chen Tessler [*1] **Yonathan Efroni** [*1] **Shie Mannor** [1]

Abstract

A policy is said to be robust if it maximizes the reward while considering a bad, or even adversarial, model. In this work we formalize two new criteria of robustness to action uncertainty. Specifically, we consider two scenarios in which the agent attempts to perform an action $\mathbf{a}$, and (i) with probability α, an alternative adversarial action $\bar{\mathbf{a}}$ is taken, or (ii) an adversary adds a perturbation to the selected action in the case of continuous action space. We show that our criteria are related to common forms of uncertainty in robotics domains, such as the occurrence of abrupt forces, and suggest algorithms in the tabular case. Building on the suggested algorithms, we generalize our approach to deep reinforcement learning (DRL) and provide extensive experiments in the various MuJoCo domains. Our experiments show that not only does our approach produce robust policies, but it also improves the performance in the absence of perturbations. This generalization indicates that action-robustness can be thought of as implicit regularization in RL problems.

1. Introduction

Recent advances in Reinforcement Learning (RL) have demonstrated its potential in real-world deployment. However, since in RL it is normally assumed that the train and test domains are identical, it is not clear how a learned policy would generalize under small perturbations. For example, consider the task of robotic manipulation in which the task is to navigate towards a goal. As the policy is trained on a specific parameter set (mass, friction, etc...), it is not clear what would happen when these parameters change, e.g., if the robot is slightly lighter/heavier.

[*]Equal contribution [1]Department of Electrical Engineering, Technion Institute of Technology, Haifa, Israel. Correspondence to: Chen Tessler <chen.tessler@campus.technion.ac.il>, Yonathan Efroni <jonathan.e@campus.technion.ac.il>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

The advantage of robust policies is highlighted when considering imperfect models, a common scenario in real world tasks such as autonomous vehicles. Even if the model is trained in the real world, certain variables such as traction, tire pressure, humidity, vehicle mass and road conditions may vary over time. These changes affect the dynamics of our model, a property which should be considered during the optimization process. Robust MDPs (Nilim & El Ghaoui, 2005; Iyengar, 2005; Wiesemann et al., 2013) tackle this issue by solving a max-min optimization problem over a set of possible model parameters, an uncertainty set, e.g., the range of values which the vehicle's mass may take - the goal is thus to maximize the reward, with respect to (w.r.t.) the worst possible outcome.

Previously, Robust MDPs have been analyzed extensively in the theoretical community, in the tabular case (Nilim & El Ghaoui, 2005; Iyengar, 2005; Xu & Mannor, 2007; Mannor et al., 2012; Wiesemann et al., 2013) and under linear function approximation (Tamar et al., 2013). However, as these works analyze uncertainty in the transition probabilities: (i) it is not clear how to obtain these uncertainty sets, and (ii) it is not clear if and how these approaches may be extended to non-linear function approximation schemes, e.g., neural networks. Recently, this problem has been tackled, empirically, by the Deep RL community (Pinto et al., 2017; Peng et al., 2018). While these approaches seem to work well in practice, they require access and control of a simulator and are not backed by theoretical guarantees - a well known problem in adversarial training (Barnett, 2018).

Our approach tackles these problems by introducing a natural way to define robustness - robustness w.r.t. action perturbations - a scenario in which the agent attempts to perform an action and due to disturbances, such as noise or model uncertainty, acts differently than expected. In this work, we consider two distinct robustness criteria: given an action provided by the policy (i) the *Probabilistic Action Robust MDP (PR-MDP, Section 3)* criterion considers the case in which, with probability α, a different possibly adversarial action is taken; and (ii) the *Noisy Action Robust MDP (NR-MDP, Section 4)* criterion, in which a perturbation is added to the action itself. These two criteria are strongly correlated to real world uncertainty; the former correlates to abrupt interruptions such as a sudden push and the latter correlates to a constant interrupting force. For instance, if the

robot is heavier, this may be seen as an adversary applying force in the opposite direction (Başar & Bernhard, 2008).

In Section 6, we extend our approach to Deep RL, perform extensive evaluation across several MuJoCo (Todorov et al., 2012) environments and show the ability of our approach to produce robust policies. We empirically analyze the differences between the PR-MDP and NR-MDP approaches, and demonstrate their ability to produce robust policies under abrupt perturbations and mass uncertainty. Surprisingly, we observe that even in the absence of perturbations, solving for the action robust criteria results in improved performance[1].

2. Preliminaries

2.1. Markov Decision Process

We consider the framework of infinite-horizon discounted Markov Decision Process (MDP) with continuous action space. An MDP is defined as the 5-tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ (Puterman, 1994), where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a compact and convex action metric space. We assume $P \equiv P(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$ is a transition kernel and is weakly continuous in $\mathbf{a}$, $R \equiv r(\mathbf{s}, \mathbf{a})$ is a reward function continuous in a, and $\gamma \in (0, 1)$. Let $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ be a stationary policy, where $\mathcal{P}(\mathcal{A})$ is the set of probability measures on the Borel sets of $\mathcal{A}$. We denote Π as the set of stationary deterministic policies on $\mathcal{A}$, i.e., if $\pi \in \Pi$ then $\pi : \mathcal{S} \to \mathcal{A}$, and $\mathcal{P}(\Pi)$ as the set of stationary stochastic policies. Let $v^\pi \in \mathbb{R}^{|\mathcal{S}|}$ be the value of a policy π, defined in state $\mathbf{s}$ as $v^\pi(\mathbf{s}) \equiv \mathbb{E}^\pi[\sum_{t=0}^\infty \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0 = \mathbf{s}]$, where $\mathbf{a}_t \sim \pi(\mathbf{s}_t)$ is a random-variable, $\mathbb{E}^\pi$ denotes expectation w.r.t. the distribution induced by π and conditioned on the event $\{\mathbf{s}_0 = \mathbf{s}\}$.

The goal is to find a policy π^*, yielding the optimal value v^*, i.e., for all $\mathbf{s} \in \mathcal{S}$, $\pi^*(\mathbf{s}) \in \arg\max_{\pi' \in \mathcal{P}(\Pi)} \mathbb{E}^{\pi'}[\sum_{t=0}^\infty \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \mid \mathbf{s}_0 = \mathbf{s}]$, and the optimal value is $v^*(\mathbf{s}) = v^{\pi^*}(\mathbf{s})$. It is known, and quite surprising, that there always exists an optimal policy which is stationary and deterministic, meaning $\pi^* \in \Pi$, e.g., (Puterman, 1994)[Theorem 6.2.10].

We note that in all following results we assume continuity of the dynamics and reward in actions. For the exact definitions see Appendix A.1 , Assumption 1.

2.2. Zero-Sum Games

As opposed to the standard MDP framework, in a two player zero-sum game, the reward function and transition kernels are functions of both players $\mathbf{a} \in \mathcal{A}$ and $\bar{\mathbf{a}} \in \bar{\mathcal{A}}$, where $\mathcal{A}, \bar{\mathcal{A}}$ are compact sets. Assuming the policy of player 1 is π and $\bar{\pi}$ of player 2, the value of the game is defined $\forall s \in \mathcal{S}$, $v^{\pi,\bar{\pi}}(\mathbf{s}) \equiv \mathbb{E}^{\pi,\bar{\pi}}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t, \bar{a}_t) \mid s_0 = s]$. Maitra & Parthasarathy (1970) generalized result of Shapley (1953) and established that, under proper conditions, the zero sum game has value for any $s \in \mathcal{S}$, i.e.,

$$v^*(\mathbf{s}) = \max_{\pi \in \mathcal{P}(\Pi)} \min_{\bar{\pi} \in \Pi} \mathbb{E}^{\pi,\bar{\pi}}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t, \bar{a}_t) \mid s_0 = s],$$
$$= \min_{\bar{\pi} \in \mathcal{P}(\Pi)} \max_{\pi \in \Pi} \mathbb{E}^{\pi,\bar{\pi}}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t, \bar{a}_t) \mid s_0 = s].$$

Note that, in the general case, the optimal maximizing policy is selected from the set of stochastic policies. Policies which attain this value, π^* and $\bar{\pi}^*$ for the maximizer and minimizer players, respectively, are said to be in Nash-Equilibrium. In such a scenario, neither player may improve it's outcome further, e.g., $\forall \pi, \bar{\pi} \in \mathcal{P}(\Pi)$, $v^{\pi,\bar{\pi}^*} \leq v^* \leq v^{\pi^*,\bar{\pi}}$.

3. Probabilistic Action Robust MDP

In this section we introduce the Probabilistic Action Robust MDP (PR-MDP), which can be viewed as a zero-sum game between an agent and an adversary. We refer to the optimal policy of the max-agent in PR-MDP as the optimal probabilistic robust policy. Furthermore, we establish that the game has a well defined value and analyze some properties of this criterion. Lastly, we formulate Policy Iteration (PI) schemes that solve the PR-MDP, and show that they inherit properties corresponding to single agent PI schemes.

Definition 1. *Let $\alpha \in [0, 1]$. A Probabilistic Action Robust MDP is defined by the 5-tuple of an MDP (see Section 2.1). Let $\pi, \bar{\pi}$ be policies of an agent an adversary. We define their probabilistic joint policy $\pi^{\text{mix}}_{P,\alpha}(\pi, \bar{\pi})$ as $\forall s \in \mathcal{S}$, $\pi^{\text{mix}}_{P,\alpha}(\mathbf{a} \mid \mathbf{s}) \equiv (1-\alpha)\pi(\mathbf{a} \mid \mathbf{s}) + \alpha\bar{\pi}(\mathbf{a} \mid \mathbf{s})$.*

Let π be an agent policy. As opposed to standard MDPs, the value of the policy is defined by $v^\pi_{P,\alpha} = \min_{\bar{\pi} \in \Pi} \mathbb{E}^{\pi^{\text{mix}}_{P,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$, where $\mathbf{a}_t \sim \pi^{\text{mix}}_{P,\alpha}(\pi(\mathbf{s}_t), \bar{\pi}(\mathbf{s}_t))$. The optimal probabilistic robust policy is the optimal policy of the PR-MDP

$$\pi^*_{P,\alpha} \in \arg\max_{\pi \in \mathcal{P}(\Pi)} \min_{\bar{\pi} \in \Pi} \mathbb{E}^{\pi^{\text{mix}}_{P,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]. \quad (1)$$

The optimal probabilistic robust value is $v^*_{P,\alpha} = v^{\pi^*_{P,\alpha}}_{P,\alpha}$.

Simply put, an optimal probabilistic robust policy is optimal w.r.t. a scenario in which, with probability α, an adversary takes control and performs the worst possible action. This approach formalizes a possible inability to control the system and to perform the wanted actions.

In-order to obtain the optimal probabilistic robust policy, one needs to solve the zero-sum game as defined in (1) (see Appendix B.1 for a formal mapping). It is well known (Straffin, 1993) that any zero-sum game has a well defined value on the set of stochastic policies, but not always on the set of deterministic policies. Interestingly, and similarly to regular MDPs, the optimal policy of the PR-MDP is a

[1] Our code can be found in the following repository: https://github.com/tesslerc/ActionRobustRL

deterministic one as the following proposition asserts (see proof in Appendix B.2).

Proposition 1. *For PR-MDP, there exists an optimal policy which is stationary and deterministic, and strong duality holds in* Π*,*

$$v^*_{P,\alpha} = \max_{\pi\in\Pi}\min_{\bar{\pi}\in\Pi} \mathbb{E}^{\pi^{\text{mix}}_{P,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t,\mathbf{a}_t)]$$
$$= \min_{\bar{\pi}\in\Pi}\max_{\pi\in\Pi} \mathbb{E}^{\pi^{\text{mix}}_{P,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t,\mathbf{a}_t)].$$

3.1. Probabilistic Action Robust and Robust MDPs

Although the approach of PR-MDP might seem orthogonal to the that of Robust MDPs, the former is a specific case of the latter. By using the PR-MDP criterion, a class of models is implicitly defined, and the probabilistic robust policy is optimal w.r.t. the worst possible model in this class.

To see the equivalence, define the following class of models,

$$\mathcal{P}_\alpha = \{(1-\alpha)P + \alpha P^\pi : \mathcal{P}(\Pi)\}$$
$$\mathcal{R}_\alpha = \{(1-\alpha)r + \alpha r^\pi : \pi \in \mathcal{P}(\Pi)\}.$$

A probabilistic robust policy, which solves (1), is also the solution to the following RMDP (see Appendix B.3),

$$\pi^*_{P,\alpha} \in \arg\max_{\pi'\in\Pi} \min_{P\in\mathcal{P}_\alpha, r\in\mathcal{R}_\alpha} \mathbb{E}^{\pi'}_P[\sum_t \gamma^t r(\mathbf{s}_t,\mathbf{a}_t)],$$

where $\mathbb{E}^\pi_P$ is the expectation of policy π when the dynamics are given by P. This relation explicitly shows that $\pi^*_{P,\alpha}$ is also optimal w.r.t. the worst model in the class $\mathcal{P}_\alpha, \mathcal{R}_\alpha$, which is convex and rectangular uncertainty set (Epstein & Schneider, 2003), and formalizes the fact that PR-MDP is a specific instance of RMDP.

3.2. Policy Iteration Schemes for PR-MDP

In this section, we analyze Policy Iteration (PI) schemes that solve (1). Although a Value-Iteration procedure can be easily derived, we focus on the possible PI schemes. PI schemes are central to the currently used actor-critic approaches in continuous control, which we focus on in our experiments. We present two algorithms, Probabilistic Robust PI (Algorithm 1) and Soft Probabilistic Robust PI (Algorithm 2), and discuss the relation between the two.

The Probabilistic Robust PI (PR-PI, Algorithm 1) is a two player PI scheme adjusted to solving a PR-MDP (e.g., Rao et al. (1973); Hansen et al. (2013)). PR-PI repeats two stages, (i) given a fixed adversary strategy, it calculates the optimal counter strategy, and (ii) it solves the 1-step greedy policy w.r.t. the value of the agent and adversary mixture policy. As suggested in Shani et al. (2018), Section 3.1, stage (i) may be performed by any MDP solver.

The Soft Probabilistic Robust PI (Soft PR-PI, Algorithm 2) is updated using gradient information, unlike the PR-PI. Instead of updating the adversary policy using a 1-step greedy update, the adversary policy is updated using a Frank-Wolfe update (Frank & Wolfe, 1956). The Franke-Wolfe update, similar to the gradient-projection method, finds a policy which is within the set of feasible policies; as, for instance, the gradient may produce policies out of the simplex. It works by finding the valid policy with the highest correlation, i.e., inner product, with the direction of gradient descent and performs a step towards it. As a convex mixture of two policies is a valid policy, the new policy is ensured to be a valid one.

Although the two algorithms might seem disparate, Soft PR-PI merely generalizes the 'hard' updates of PR-PI to 'soft' ones. This statement is formalized in the following proposition, which is a direct consequence of Theorem 1 in Scherrer & Geist (2014), see proof in Appendix B.4.

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

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

Notice that the first single agent, 1-step improvement, has a solution in the set of deterministic policies (since the action space is a compact set and the argument is continuous in the action). Thus, $\bar{\pi}$ in Algorithm 2 is exactly the 1-step greedy policy used in Algorithm 1. This suggests that for $\eta = 1$ Algorithm 2 is completely equivalent to Algorithm 1.

Generally, in two-player PI, the improvement stage amounts to solving a max-min, 1-step, decision problem. In PR-PI it is clearly not the case; in the improvement stage, a single agent, 1-step-greedy policy, is solved. Solving the latter is easier than solving the former, and it is a result of the specific structure of PR-MDP which does not generally hold, as will be demonstrated in Section 4.

The following result shows that in both algorithms the value converges to the unique optimal value of the Nash-Equilibrium (see proof in Appendix B.5).

Theorem 3. *Denote by* $v_k \overset{def}{=} v^{\pi^{\text{mix}}_{P,\alpha}(\pi_k,\bar{\pi}_k)}$*. Then, for any* $\eta \in (0,1]$*, in Algorithm 2,* v_k *contracts toward* $v^*_{P,\alpha}$ *with coefficient* $(1-\eta+\gamma\eta)$*, i.e.,*

$$||v_k - v^*_{P,\alpha}||_\infty \leq (1-\eta+\gamma\eta)||v_{k-1} - v^*_{P,\alpha}||_\infty \ .$$

Due to the equivalence of Algorithms 1 and 2 (when $\eta = 1$), we get as a corollary that PR-PI converges toward the unique Nash-Equilibrium.

Remark 1. *The solution method of the* $\arg\max$ *and* $\arg\min$ *in both Algorithms 1 and 2 can be swapped and the convergence guarantees remain, e.g.,* $\bar{\pi}$ *is the optimal solution to the MDP given* π*, whereas* π *is updated using the 1-step greedy approach w.r.t.* $\bar{\pi}$*.*

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 π_{k-1}

Algorithm 2 Soft Probabilistic Robust PI

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

Remark 2. *Although Soft PR-PI converges slower than the non-soft version, it is reasonable to assume the former will be less sensitive to errors than the latter. Soft PR-PI can be seen as a generalization of Conservative PI (CPI) to solving PR-MDPs. CPI is known to be less sensitive to errors than other PI schemes (Scherrer & Geist, 2014). Nonetheless, the error analysis for Soft PR-PI is substantially different than the one CPI (Kakade & Langford, 2002; Scherrer, 2014). In Soft PR-PI, small changes in the adversarial policy may result in dramatic changes in the agent's policy. Thus, the γ-weighted state occupancy under a measure ν, $d_\nu^{\pi^{\text{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)} = \sum_t \gamma^t \nu P^{\pi^{\text{mix}}_{P,\alpha}(\pi_k, \bar{\pi}_k)}$, may change dramatically between iterations, whereas in CPI the change is smooth. We leave the error analysis for future work.*

4. Noisy Action Robust MDP

In this section we consider an alternative definition for action robustness. Instead of a stochastic perturbation in the policy space, as in Section 3, we consider a perturbation in the action space. To formally study such a perturbation we define the Noisy Action Robust MDP (NR-MDP), which, similarly to the PR-MDP, can be viewed as a zero-sum game (see Appendix C.1 for a formal mapping). We continue by establishing some properties of this MDP while highlighting important differences relative to the approach of PR-MDP.

Definition 2. *Let $\alpha \in [0, 1]$. A Noisy Action Robust MDP is defined by the 5-tuple of an MDP (see Section 2.1). Let $\pi, \bar{\pi}$ be policies of an agent and an adversary. We define their noisy joint policy $\pi^{\text{mix}}_{N,\alpha}(\pi, \bar{\pi})$ as*

$$\forall s \in \mathcal{S}, \mathbf{a} \in \mathcal{A}, \pi^{\text{mix}}_{N,\alpha}(\mathbf{a} \mid \mathbf{s}) \equiv \mathbb{E}_{\substack{\mathbf{b}\sim\pi(\cdot|s) \\ \bar{\mathbf{b}}\sim\bar{\pi}(\cdot|s)}}[\mathbb{1}_{\mathbf{a}=(1-\alpha)\,\mathbf{b}+\alpha\bar{\mathbf{b}}}],$$

1 if action equality holds. 0 otherwise.

the relation is obtained by the fact that $\mathbf{a} \sim \pi, \bar{\mathbf{a}} \sim \bar{\pi}$.

Let π be an agent policy. For NR-MDP, its value is defined by $v^\pi_{N,\alpha} = \min_{\bar{\pi}\in\Pi} \mathbb{E}^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]$, where $\mathbf{a}_t \sim \pi^{\text{mix}}_{N,\alpha}(\pi(\mathbf{s}_t), \bar{\pi}(\mathbf{s}_t))$. The optimal α-noisy robust policy is the optimal policy of the NR-MDP

$$\pi^*_{N,\alpha} \in \arg\max_{\pi\in\mathcal{P}(\Pi)} \min_{\bar{\pi}\in\Pi} \mathbb{E}^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)]. \quad (2)$$

The optimal noisy robust value is $v^*_{N,\alpha} = v^{\pi^*_{N,\alpha}}_{N,\alpha}$.

In simple terms; an optimal noisy robust policy is optimal w.r.t. a scenario, in which an adversary may change the agent's actions by adding bounded perturbations; the action performed on the system is $(1-\alpha)\,\mathbf{a} + \alpha\bar{\mathbf{a}}$, where $\bar{\mathbf{a}}$ is an action drawn from possibly adverserial distribution $\bar{\pi}$. The adversary's ability to add perturbations is controlled through the parameter α. Each value of α defines a new continuous-action NR-MDP, where for $\alpha = 0$ the adversary is unable to affect the system and the decision problem collapses to the standard, non-robust, MDP formulation.

The assumption on the structure of $\mathcal{A}$ is required, in order to ensure that the α-mixture actions are valid actions, an assumption which holds naturally in the domain of continuous control. This approach formalizes a specific meaning for perturbation in the action space.

Although the approach of PR-MDP (Section 3) and NR-MDP are closely related, they are not equivalent and important differences exist between the two. Unlike PR-MDP, for which a *deterministic* stationary optimal policy exists, generally, for NR-MDP it is not the case. The optimal noisy robust policy, in the general case, is a *stochastic policy* (see proof in Appendix C.2).

Proposition 4. *There exists an NR-MDP such that,*

$$\max_{\pi\in\Pi} \min_{\bar{\pi}\in\Pi} \mathbb{E}^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)] < \max_{\pi\in\mathcal{P}(\Pi)} \min_{\bar{\pi}\in\Pi} \mathbb{E}^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})}[\sum_t \gamma^t r(\mathbf{s}_t, \mathbf{a}_t))].$$

Furthermore, strong duality does not necessarily hold on the class of deterministic policies, Π.

The above proposition tells us that while it is often easier to focus on deterministic strategies (policies), when considering the NR-MDP scenario the optimal strategy may be stochastic. A similar notion has been shown to hold in non-cooperative matrix games (Nash, 1951), in which the optimal strategy is stochastic.

4.1. Policy Iteration for NR-MDPs

In section 3.2, we formulated PI schemes to solve PR-MDPs. Unlike two-player zero-sum PI (Rao et al., 1973; Hansen et al., 2013), in PR-PI (Algorithm 1) a *single* agent decision problem is solved, when the adversary policy $\bar{\pi}_{k+1}$ is updated. This structure is indeed unique to the PR-MDP, and does not hold when generalizing two-player zero-sum PI to solve NR-MDP.

Specifically, consider the two-player zero-sum PI that repeats the following two stages:

$$1.\pi_k \in \arg\max_{\pi\in\Pi} v^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi}_k)},$$

$$2.\pi_k \in \arg\min_{\bar{\pi}\in\mathcal{P}(\Pi)} \max_{\pi\in\Pi} r^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})} + P^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})} v^{\pi^{\text{mix}}_{N,\alpha}(\pi_k,\bar{\pi}_k)}.$$

$v^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi}_k)}$ is the value of the joint policy $\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi}_k)$, $r^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})}(s) = \mathbb{E}_{\mathbf{a}\sim\pi,\bar{\mathbf{a}}\sim\bar{\pi}}[r(s,(1-\alpha)\,\mathbf{a}+\alpha\bar{\mathbf{a}})]$, and $P^{\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})}(s,s') = \mathbb{E}_{\mathbf{a}\sim\pi,\bar{\mathbf{a}}\sim\bar{\pi}}[P(s \mid s,(1-\alpha)\,\mathbf{a}+\alpha\bar{\mathbf{a}})]$ are the induced reward and dynamics from by $\pi^{\text{mix}}_{N,\alpha}(\pi,\bar{\pi})$. Following similar lines of proof as in Hansen et al. (2013) or as in Theorem 3, a similar γ-contraction result may be achieved for the NR-MDP, e.g., $||v_k - v^*_{N,\alpha}||_\infty \leq \gamma||v_{k-1} - v^*_{N,\alpha}||_\infty$.

In such an algorithm, stage (1) is performed by solving an MDP, as in PR-PI. However, stage (2) requires solving a 1-step min-max problem. For general reward and transition probabilities it cannot be solved by solving a single-agent decision problem, as in the second stage of PR-PI (Algorithm 1). Furthermore, the solution of stage (2) cannot be achieved by a single-call to a gradient oracle as in Proposition 2 (we elaborate the discussion in Appendix C.3).

Regardless of these differences, in Section 6, we will use the approach of Soft PR-PI and offer DRL algorithms to solve both the PR and NR MDPs. While the approach we consider in Section 6 should be understood as a heuristic for solving NR-MDP, it is based on Algorithm 2, which guarantees convergence for PR-MDP in the error-free case.

5. Related Work

Robust RL: Traditional works in RL, such as Nilim & El Ghaoui (2005) and Iyengar (2005) have provided efficient algorithms for solving Robust MDPs, with uncertainty in the transition probabilities. Mannor et al. (2012) extended their approach to non-rectangular uncertainty sets, e.g., coupled uncertainty sets. However, these approaches are limited to solutions in the tabular case. Additionally, a connection between robustness and generalization has been suggested (Xu et al., 2009; Xu & Mannor, 2012), while it is not clear how this holds in RL, we believe that there lies a similar yet complex connection between the two concepts.

Control: Obtaining robust policies in continuous control problems has been extensively investigated in the past. Most closely related to our work, are max-min Robust Control approaches (e.g., Bemporad et al. (2003); Kerrigan & Maciejowski (2004); de la Pena et al. (2006). In this line of work, a control policy which is robust w.r.t. deterministic perturbations is calculated. There, the max-min problem is solved via Linear program, Quadratic program or by an explicit tree-search. Here, we focus on PI, and gradient based, schemes to solve a more specific problem; action robust policies. Furthermore, and to the best of our knowledge, in this line of works, discussion on the existence of strong-duality does not exists (i.e., as Proposition 1 and 4 assert for PR- and NR-MDPs).

Robust Supervised Learning: Similar to the Robust MDPs framework, robustness to adversarial examples/attacks (Szegedy et al., 2013) is a measure of robustness in supervised learning. A method of learning robust classifiers is through Generative Adversarial Networks (Goodfellow et al., 2014). Similar to our approach, when using GANs for robustness, an adversary learns to create small perturbations in the input data in an attempt to cause a misclassification (Xiao et al., 2018; Samangouei et al., 2018; Kurakin et al., 2018). While these methods work well in practice, they generally lack convergence proofs and should thus be treated as heuristics.

6. Experiments

6.1. Method

Our approach adapts the Soft PR-PI algorithm to the high dimensional scenario. While in the tabular case we may use an MDP solver, which produces the optimal policy; when considering parametrized policies, e.g., neural networks, a dual-gradient approach is taken. In this approach, both the Actor and the Adversary are trained using gradient descent; as it is hard to measure convergence - we train the actor for N gradient steps followed by a single adversary step.

We focus on a robust variant of DDPG which we call Action-Robust DDPG (AR-DDPG, see Appendix D, Algorithm 5). DDPG (Lillicrap et al., 2015) trains an actor to predict an action for each state $\mu_\theta : \mathcal{S} \to \mathcal{A}$ (i.e., a deterministic policy). In AR-DDPG we train two networks, deterministic policies, the actor and adversary, denoted by μ_θ and $\bar{\mu}_{\bar{\theta}}$. Similarly to DDPG, a critic is trained to estimate the q-function of the joint-policy. For PR-MDP (Definition 1), the joint policy is

$$\pi^{\text{mix}}_{P,\alpha}(u \,|\, s;\theta,\bar{\theta}) = (1-\alpha)\delta(u-\mu_\theta(s)) + \alpha\delta(u-\bar{\mu}_{\bar{\theta}}(s)), \tag{3}$$

whereas for NR-MDP (Definition 2), the joint policy is,

$$\pi^{\text{mix}}_{N,\alpha}(u \,|\, s;\theta,\bar{\theta}) = \delta(u-((1-\alpha)\mu_\theta(s) + \alpha\bar{\mu}_{\bar{\theta}}(s))), \tag{4}$$

where $\delta(\cdot)$ is the Dirac delta function.

The following result generalizes DPG (Silver et al., 2014)

for both PR and NR-MDPs. i.e., it establishes how to update θ and $\bar{\theta}$ using a deterministic gradient based method.

Proposition 5. *Let $\mu_\theta, \bar{\mu}_{\bar{\theta}}$ be the agent's and adversary's deterministic policies, respectively. Let $\pi(\mu_\theta, \bar{\mu}_{\bar{\theta}})$ be the joint policy given the agent and adversary policies. i.e., for PR-MDP $\pi = \pi^{\text{mix}}_{P,\alpha}$ (3), and for NR-MDP $\pi = \pi^{\text{mix}}_{N,\alpha}$ (4).*

Let $J(\pi(\mu_\theta, \bar{\mu}_{\bar{\theta}})) = \mathbb{E}_{\mathbf{s}\sim\rho^\pi}[v^\pi(s)]$ be the performance objective. The gradient of the actor and adversary parameters, for both PR- and NR-MDP is:

$$\nabla_\theta J(\pi(\mu_\theta, \bar{\mu}_{\bar{\theta}})) = (1-\alpha)\mathbb{E}_{\mathbf{s}\sim\rho^\pi}[\nabla_\theta \mu_\theta(\mathbf{s}) \nabla_{\mathbf{a}} Q^\pi(\mathbf{s}, \mathbf{a})] \ ,$$
$$\nabla_{\bar{\theta}} J(\pi(\mu_\theta, \bar{\mu}_{\bar{\theta}})) = \alpha\mathbb{E}_{\mathbf{s}\sim\rho^\pi}[\nabla_{\bar{\theta}} \bar{\mu}_{\bar{\theta}}(\mathbf{s}) \nabla_{\bar{\mathbf{a}}} Q^\pi(\mathbf{s}, \bar{\mathbf{a}})] \ .$$

where for the PR-MDP we have $\mathbf{a} = \mu_\theta(\mathbf{s})$ and $\bar{\mathbf{a}} = \bar{\mu}_{\bar{\theta}}(\mathbf{s})$, and for the NR-MDP $\mathbf{a} = \bar{\mathbf{a}} = (1-\alpha)\mu_\theta(\mathbf{s}) + \alpha\bar{\mu}_{\bar{\theta}}(\mathbf{s})$.

A proof, example algorithm and block diagram are provided in Appendix D.

In order to validate our approach, we consider several MuJoCo domains (Todorov et al., 2012). MuJoCo contains several continuous control problems, such as robotic manipulation, in which we may test the ability of our approach to produce robust policies. Intuitively, our Probabilistic operator is correlative to the occurrence of large *abrupt* forces, e.g., someone suddenly pushes the robot, whereas the Noisy operator is correlative to *mass* uncertainty, e.g., the robot is heavier or lighter.

Our evaluation is split into two parts, we begin by comparing the various hyper-parameters and how they affect the performance of both the NR and PR-MDP approaches. This evaluation is performed extensively on a single domain, the Hopper-v2 task, and the figures are provided in the appendix. We then compare the best performing variants across unseen domains. By doing so we test the transferability of these hyper-parameters across domains.

6.2. Theory versus Practice

Our theoretical approach, Soft PR-PI (Algorithm 2), is proven for the PR-MDP. The algorithm is based on a dynamic programming approach, (i) given a fixed adversary policy, solves the optimal agent's policy, (ii) updates the adversary policy using gradients.

1. While in theory, for the PR criterion, there exists a deterministic optimal policy - this does not necessarily hold for the NR case (Proposition 4). Thus searching over the space of deterministic policies is sub-optimal.
2. Theoretical approaches in general require exact computation, however, in practice, we use function approximation schemes, e.g., deep neural networks. As such, convergence can not be ensured and the approach should be seen as a heuristic.

Regardless of these differences, we based the empirical approach for both PR and NR-MDPs on Algorithm 2.

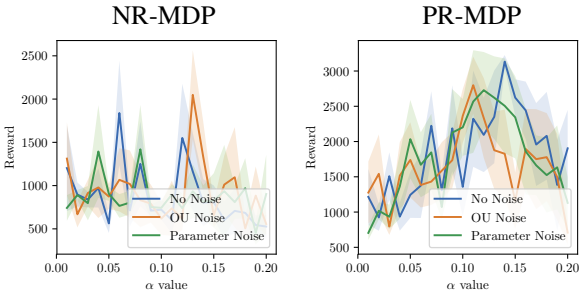

Figure 1. Hopper-v2: Performance of both the NR and PR-MDP criteria as a function of the uncertainty α.

6.3. Hyperparameter Ablation

Table 1. Hyper-parameters considered.

α values	0.01, 0.05, 0.1, 0.15 and 0.2
Actor update steps N	2, 5, 10 and 20

The hyper-parameters we consider are shown in Table 1. In addition, we consider 3 exploration schemes: noiseless (on-policy exploration), Ornstein Ulenbeck (OU, Uhlenbeck & Ornstein (1930)) and Parameter space noise (Plappert et al., 2017). Each configuration, is trained on 5 random seeds and the final policy, once the training is concluded, is evaluated across 100 episodes. The evaluation is performed without adversarial perturbations, on a range of mass values not encountered during training, i.e., we test the ability of the action robust approach to produce policies which are *robust to model uncertainty*. The baseline we compare to, is DDPG with parameter space noise for exploration, which performed best in our experiments.

The extensive comparison is presented in the appendix, however the main conclusion is shown in Figure 1. While there is a clear correlation between the value of α and the performance of the PR-MDP criteria, e.g., an optimal value is attained at $\alpha \in [0.1, 0.15]$ and deviating from this range results in performance deterioration - this is not the case for the NR-MDP. Although the NR-MDP often attains competitive results, it is not clear how the various parameters affect it. We conclude that the for our simple gradient based approach, the PR approach exhibits a more stable behavior than the NR approach.

Specifically, for the PR-MDP we decided to use Parameter space noise with $\alpha = 0.1$ and a ratio of 10:1. Even though there are certain configurations under which the OU noise variant outperformed the Parameter space noise, we decided on the latter as it exhibited higher stability and is thus more likely to transfer easily to new domains. Similarly, a large α provides greater control to the adversary, as such we decided on a more conservative value of 0.1.

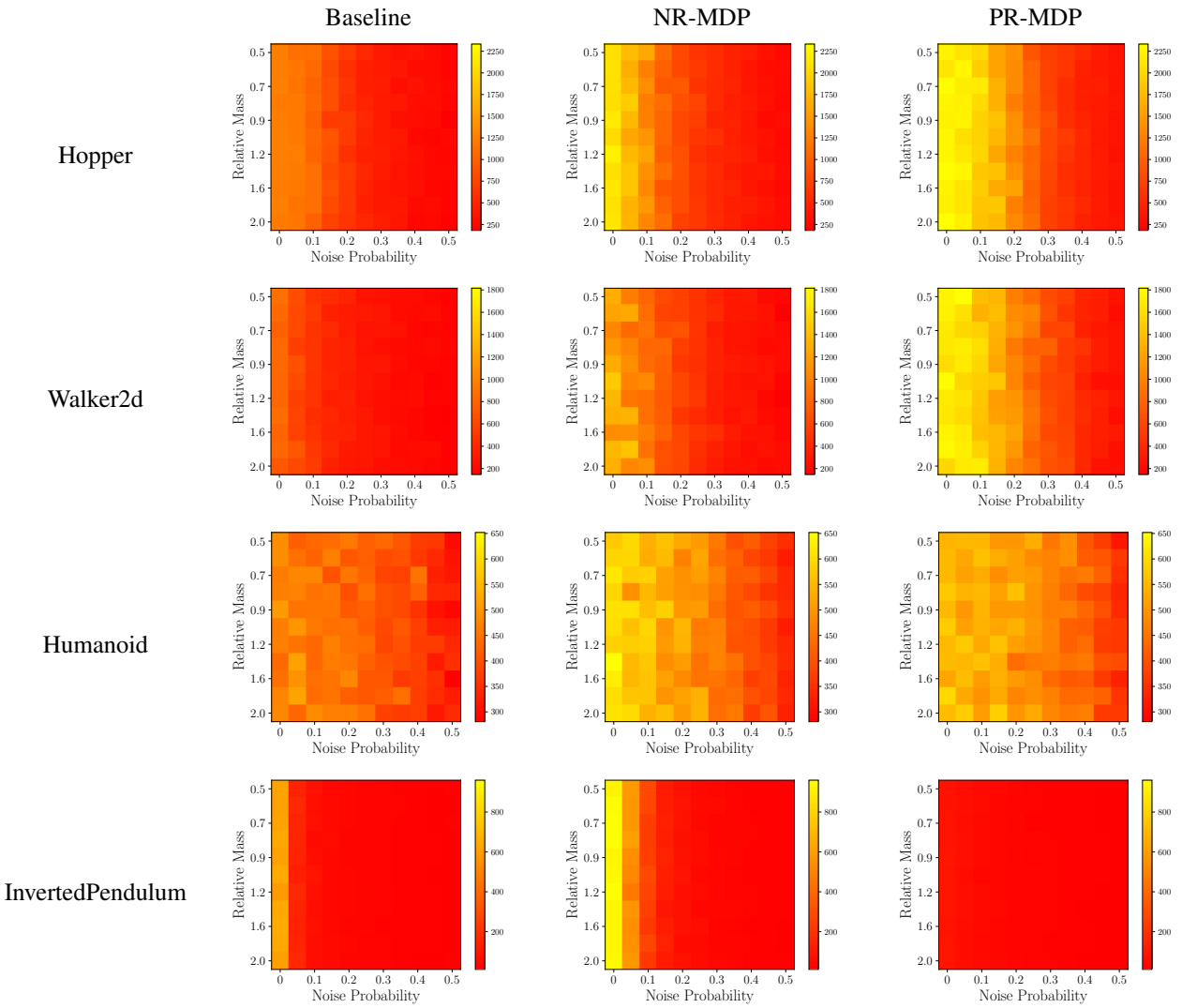

Figure 2. Robustness to model uncertainty. Noise probability denotes the probability of a randomly sampled noise being played instead of the selected action.

For the NR-MDP this selection process is somewhat harder; as slight changes in the hyper-parameters may result in radical changes in the performance. We selected the OU noise combined with $\alpha = 0.1$ and a training ratio of 1:1.

An interesting insight is that in the PR-MDP criteria, the adversary induces enough noise for exploration (Figure 1, PR-MDP - No Noise plot). This can be seen when observing the 'no noise' experiments, which show that the PR-MDP approach outperforms the baseline even without additional exploration noise.

6.4. Testing on various MuJoCo domains

Figure 2 presents our results, on various MuJoCo domains (additional results in Appendix E). It is apparent that while in the Hopper-v2 domain, the PR-MDP outperformed the NR-MDP criterion; this does not hold on all domains. Moreover, in most of the domains, both operators outperform the baseline, both in terms of robustness and in terms of performance in the absence of perturbations. While the optimal parameters may differ across domains; our results show that, in most cases, the parameters transfer across domains and result in improved performance without additional tuning.

Failures: It is also important to acknowledge the scenarios in which our algorithm does not outperform the baseline. Such an example is the InvertedPendulum domain, in which the performance of the PR-MDP was found to be inferior to that of its non-robust counterpart. We find two possible explanations for this phenomenon (i) the parameter tuning is performed on the Hopper domain (as opposed to selecting the optimal hyper-parameters per each domain). As each domain is different, it is plausible that good hyper-parameters

in a certain domain would not be good in all domains. (ii) Specifically in the InvertedPendulum domain, where the task is to balance a pole, an adversary which is too strong (large α value) prevents the agent from successfully solving the task.

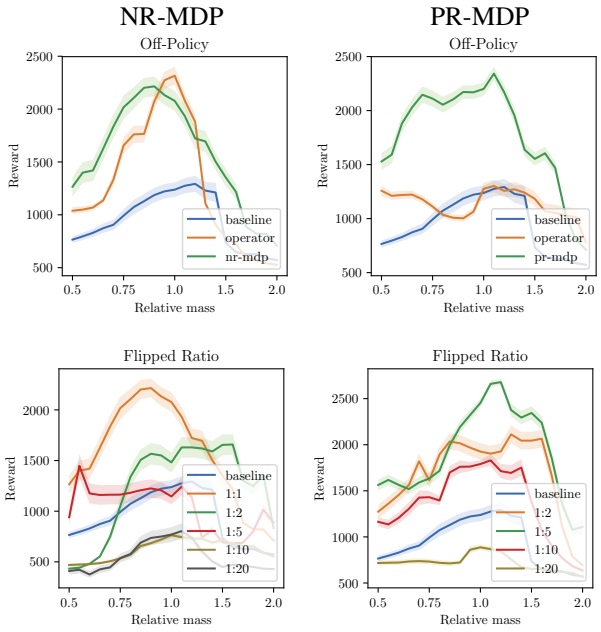

Figure 3. Diving Deeper: (Up) Testing Off-Policy Action-Robustness, and (Down) Solving the MaxMin operator.

6.5. Diving Deeper

We attempt to analyze the behavior of our criteria (Figure 3) by asking two questions: (i) Does the performance increase due to the added perturbations from the adversary, or does the operator itself induce a prior, e.g., regularization, on the policy which leads to improved performance. (ii) How close is the empirical behavior to its theoretical counterpart.

Off-Policy Action Robustness: In previous experiments, during training, the action was drawn from the joint policy of the agent and adversary, where the joint policy is specified in the PR and NR-MDP approaches (see Definition 1,2).

A natural alternative approach is to *act with the actor's policy*, yet, to acquire an action-robust policy in an off-policy fashion. Meaning, use the same algorithms while obtaining the data without the effect of the adversary. A possible advantage of such an approach is minimizing the number of bad actions (since the adversary does not intervene), while still benefiting from the presence of robust learning.

Figure 3 presents the results of this experiment. For the NR-MDP, it seems that the operator itself, i.e., the training is what results in the performance improvement; whereas the adversarial exploration amount to a small increase in stability. Surprisingly, an opposite effect is observed for the PR-MDP. There, the combination of adversarial exploration and the operator are both required in order to attain the performance increase.

Does MaxMin equal MinMax? While so far we trained our agent through N actor updates followed by a single adversary gradient update, this corresponds to the MinMax operator, in theory the opposite should result in an identical performance (Proposition 1) for the PR-MDP approach, and to deteriorate the performance for the NR-MDP approach (Proposition 4).

Experimentally (Figure 3) the results show that as opposed to the theoretical analysis, a 'stronger' adversary does result in performance degradation. This could be due to two possible factors: (i) as we trained for the same number of steps for both scenarios, it means that in this case the actor receives less gradient update steps, and/or (ii) it could be that increasing the convergence of the adversary results in faster convergence to a sub-optimal solution (w.r.t. the actor).

7. Summary

We have presented two new criteria for robustness, the Probabilistic and Noisy action Robust MDP, related each to real world scenarios of uncertainty and discussed the theoretical differences between both approaches. Additionally; we developed the Soft PR-PI (Algorithm 2), a policy iteration scheme for solving PR-MDPs. Building upon the Soft PR-PI algorithm, we presented a deep reinforcement learning approach, which is capable of solving our criteria. We compared both criteria, analyzed how the various hyper-parameters affect the behavior and how the empirical results correlate (and occasionally contradict) with the theoretical approach. Most importantly, we notice that not only does training with our criteria result in robust policies, but our approach improves performance even in the absence of perturbations.

Lastly, for solving an action-robust policy, there is *no need in providing an uncertainty set*. The approach requires only a scalar value, namely α (or possibly a state-dependent $\alpha(s)$), which *implicitly* defines an uncertainty set (see Section 3.1). This is a major advantage compared to standard robust approaches in RL and control, which, to the best of our knowledge, require a distribution over models or perturbations. Of course, this benefit is also a restriction - the Action Robust approach is unable to handle any kind of worst-case perturbations. Yet, due to its simplicity, and its demonstrated performance, it is worthwhile to be considered by an algorithm designer.

8. Acknowledgements

The authors would like to thank Bruno Scherrer, Esther Derman and Nadav Merlis for the fruitful discussions and help during the work on this paper.

References

Barnett, S. A. Convergence problems with generative adversarial networks (gans). *arXiv preprint arXiv:1806.11382*, 2018.

Başar, T. and Bernhard, P. *H-infinity optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media, 2008.

Bemporad, A., Borrelli, F., and Morari, M. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on automatic control*, 48(9):1600–1606, 2003.

de la Pena, D. M., Alamo, T., Bemporad, A., and Camacho, E. F. Feedback min-max model predictive control based on a quadratic cost function. In *American Control Conference, 2006*, pp. 6–pp. IEEE, 2006.

Epstein, L. G. and Schneider, M. Recursive multiple-priors. *Journal of Economic Theory*, 113(1):1–31, 2003.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2): 95–110, 1956.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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

Iyengar, G. N. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, 2005.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pp. 267–274, 2002.

Kerrigan, E. C. and Maciejowski, J. M. Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 14(4):395–413, 2004.

Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., et al. Adversarial attacks and defences competition. *arXiv preprint arXiv:1804.00097*, 2018.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

Mannor, S., Mebel, O., and Xu, H. Lightning does not strike twice: Robust mdps with coupled uncertainty. *arXiv preprint arXiv:1206.4643*, 2012.

Nash, J. Non-cooperative games. *Annals of mathematics*, pp. 286–295, 1951.

Nilim, A. and El Ghaoui, L. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8. IEEE, 2018.

Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pp. 2817–2826, 2017.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

Rao, S., Chandrasekaran, R., and Nair, K. Algorithms for discounted stochastic games. *Journal of Optimization Theory and Applications*, 11(6):627–637, 1973.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

Scherrer, B. Approximate policy iteration schemes: a comparison. In *International Conference on Machine Learning*, pp. 1314–1322, 2014.

Scherrer, B. and Geist, M. Local policy search in a convex space and conservative policy iteration as boosted policy search. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 35–50. Springer, 2014.

Shani, L., Efroni, Y., and Mannor, S. Revisiting exploration-conscious reinforcement learning. *arXiv preprint arXiv:1812.05551*, 2018.

Shapley, L. S. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *ICML*, 2014.

Straffin, P. D. *Game theory and strategy*, volume 36. MAA, 1993.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tamar, A., Xu, H., and Mannor, S. Scaling up robust mdps by reinforcement learning. *arXiv preprint arXiv:1306.6189*, 2013.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 5026–5033. IEEE, 2012.

Uhlenbeck, G. E. and Ornstein, L. S. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

Wiesemann, W., Kuhn, D., and Rustem, B. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018.

Xu, H. and Mannor, S. The robustness-performance tradeoff in markov decision processes. In *Advances in Neural Information Processing Systems*, pp. 1537–1544, 2007.

Xu, H. and Mannor, S. Robustness and generalization. *Machine learning*, 86(3):391–423, 2012.

Xu, H., Caramanis, C., and Mannor, S. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510, 2009.