Policy Gradient for Rectangular Robust Markov Decision Processes

Anonymous Author(s)
Affiliation
Address
email

Abstract

Policy gradient methods have become a standard for training reinforcement learning agents in a scalable and efficient manner. However, they do not account for transition uncertainty, whereas learning robust policies can be computationally expensive. In this paper, we introduce robust policy gradient (RPG), a policy-based method that efficiently solves rectangular robust Markov decision processes (MDPs). We provide a closed-form expression for the worst occupation measure. Incidentally, we find that the worst kernel is a rank-one perturbation of the nominal. Combining the worst occupation measure with a robust Q-value estimation yields an explicit form of the robust gradient. Our resulting RPG can be estimated from data with the same time complexity as its non-robust equivalent. Hence, it relieves the computational burden of convex optimization problems required for training robust policies by current policy gradient approaches.

1 Introduction

Markov decision processes (MDPs) provide an analytical framework to solve sequential decision-making problems and seek the best performance in a fixed environment. Since the resulting policy can be highly sensitive to parameter values [16], the robust MDP setting alternatively maximizes return under the worst scenario, thus yielding robustness to uncertain environments [18, 10]. In practice, the robust MDP paradigm quantifies the level of uncertainty through a set $\mathcal{U}$ determining the possible range of model perturbations. Then, a policy is said to be robust-optimal if it reaches maximal performance under the most adversarial model within the uncertainty set. Developing efficient solvers for robust MDPs is of great interest, as it can lead to behavior policies with generalization guarantees [31].

If not computationally expensive, robust MDPs can be strongly NP-hard [30]. Thus, to preserve tractability, we commonly assume that $\mathcal{U}$ is convex and s-rectangular, i.e., $\mathcal{U} = \times_{s\in\mathcal{S}}\mathcal{U}_s$ [18, 10, 30]. The latter assumption means that the overall uncertainty should be designed independently for each state. Further simplification may consider (s, a)-rectangular uncertainty sets of the form $\mathcal{U} = \times_{(s,a)\in\mathcal{X}}\mathcal{U}_{(s,a)}$, albeit this naturally leads to more conservative strategies. In any case, planning in robust MDPs can be computationally costly, as it involves successive max-min problems [7, 1, 30]. To address this issue, the works [3, 12] have established an equivalence between robustness and regularization in reinforcement learning (RL) in order to derive efficient robust planning methods for s and (s, a)-rectangular robust MDPs. Indeed, it appears that resorting to proper regularization instead of solving a minimization problem can yield robust behavior without requiring the polynomial time complexity of convex optimization problems [3].

Alternatively to planning, policy gradient algorithms (PG) directly learn an optimal policy by applying gradient steps towards better performance [22]. Due to its scalability, ease of implementation, and adaptability to many different settings such as model-free and continuous state-action spaces [11, 21],

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

PG has become the workhorse of RL. Although regularization techniques such as max-entropy [6] or Tsallis [13] have shown robust behavior without impairing computational cost, they only account for adversarial reward [2, 5, 3]. Differently, robust PG formulations (RPG) formulations aim to address uncertainty to reward *and* transition functions.

Despite their ability to propel robust behavior, RPG methods that target robust optimal policies are still rare in the RL literature. The global convergence of RPG established in [14, 27] further motivates us to come up with a practical method for estimating the gradients. In fact, [14] occult the estimation part, as they assume full access to the policy gradient. Differently, the solution proposed in [27] requires solving convex optimization problems to find the worst model, which represents a time complexity of $O(S^4 A \log \epsilon^{-1})$ for (s, a)-rectangular, or $O(S^4 A^3 \log \epsilon^{-1})$ for s-rectangular uncertainty sets [27, Sec. 4.1]. These worst kernel and reward models are needed to compute RPG using the policy gradient theorem [23]. Other approaches that elicit an expression for RPG rely on a specific type of uncertainty set such as reward uncertainty with known kernel [3], r-contaminated kernel with known reward [29], or (s, a)-rectangular uncertainty [14], whereas we aim to tackle more general robust MDPs.

In this work, we introduce an RPG method for both s and (s, a)-rectangular ball-constrained uncertainty sets, with similar complexity as non-robust PG. Our approach provides a closed-form expression of RPG without relying on an oracle while applying to the most common robust MDPs. To this end, we derive the worst reward and transition functions, thus revealing the adversarial nature of the corresponding uncertainty set. Surprisingly, we also find that the worst kernel is a rank-one perturbation of the nominal kernel. Leveraging this rank-one perturbation enables us to derive a robust occupation measure. We concurrently propose an alternative definition of the robust Q-value together with an efficient way to estimate it. Combining these results enables us to obtain RPG in closed form. Our resulting RPG update requires $O(S^2 A \log \epsilon^{-1})$ computations, thus showing similar time complexity as non-robust PG.

To summarize our contributions: *(i)* We establish the worst reward and transition models in closed-form; *(ii)* We show that the worst-case transition function is a rank-one perturbation of the nominal; *(iii)* We introduce alternative robust Q-values that can be evaluated through efficient Bellman recursion while retrieving the robust value function; *(iv)* We establish an expression of RPG that can be estimated with similar time complexity as non-robust PG. Experiments show that our RPG speeds up state-of-the-art robust PG updates by 2 orders of magnitude.

2 Related work

Although some previous works use gradient methods to learn robust policies, they seek empirical robustness to adversarial behavior rather than robust MDP solutions [19, 26, 4]. In that sense, our study differs from adversarial RL as we explicitly optimize the max-min objective to find a robust optimal policy. Accordingly, the risk-averse approach focuses on the *internal uncertainty* due to the stochasticity of the system, whereas robust RL addresses the *external uncertainty* of the system's dynamics. As a result, common risk-averse objectives can be reformulated as robust problems with specific uncertainty sets [24].

Previous studies that did aim to derive robust policy-based methods are [3, 29, 27]. These are summarized in Table 1, which also displays the complexity of existing approaches. [3] established RPG for s-rectangular reward-robust MDPs, i.e., robust MDPs with uncertain reward but given kernel. Although it applies to general norms, their result does not account for transition perturbation. Differently, in [29], the authors introduced RPG for r-contaminated MDPs, i.e., robust MDPs with uncertainty set $\mathcal{U} := \{R_0\} \times [(1-r)P_0 + r\Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}]$. Although it has similar complexity as non-robust PG, by construction, their setting is limited to (s, a)-rectangularity with known reward and mixed transition. As such, the proof techniques in [29] are tailor-made to the r-contamination framework and do not apply to more general robust MDPs. In fact, we remark that the r-contamination setting is equivalent to the action robustness approach introduced in [26], which emphasizes its limitation to action perturbation. Differently, our RPG holds whenever the worst kernel is a rank-one perturbation of the nominal transition function (see Lemma 4.4).

To address generic robust MDPs, [27] recently introduced RPG for general uncertainty sets. Their gradient update has a complexity of $O(S^6 A^4 \epsilon^{-4})$, which is more expensive than non-robust PG by a factor of $S^4 A^3 \epsilon^{-4}$. They additionally assume access to an oracle gradient of the robust return with respect to the transition model. Avoiding this oracle assumption naturally leads to even higher time complexity. At the same time, the two works [14, 27] guarantee global convergence of projected robust gradient iterates, thus establishing the potential promise of RPG. In fact, equipped with RPG convergence, the remaining challenge in making it practical is to efficiently estimate the gradient. This represents the main focus of our study: We aim to explicit an RPG method that generalizes existing results on specific uncertainty sets [3, 29] while holding for s-rectangular robust MDPs.

Table 1: Time complexity of RPG update according to the type of uncertainty set. For conciseness, the displayed complexity hides logarithmic factors in A and S. Our RPG method has the same complexity as non-robust PG while it generalizes other RPG methods with similar efficiency.

UNCERTAINTY SET $\mathcal{U}$	TIME COMPLEXITY	REFERENCE
$\{R_0\} \times \{P_0\}$	$S^2 A \log \epsilon^{-1}$	[23]
$\{R_0\} \times [(1-r)P_0 + r\Delta_{\mathcal{S}}^{\mathcal{S}\times\mathcal{A}}]$	$S^2 A \log \epsilon^{-1}$	[29]
(s,a)-rectangular ball $\mathcal{U}_p^{\mathsf{sa}}$	$S^2 A \log \epsilon^{-1}$	**This work**
(s,a)-rectangular, convex $\mathcal{U}^{\mathsf{sa}}$	$S^4 A \log \epsilon^{-1}$	Convex optimization
s-rectangular ball $\mathcal{U}_p^{\mathsf{s}}$	$S^2 A \log \epsilon^{-1}$	**This work**
s-rectangular ball $(R_0 + \mathcal{R}_p^{\mathsf{s}}) \times \{P_0\}$	$S^2 A \log \epsilon^{-1}$	[3]
s-rectangular, convex $\mathcal{U}^{\mathsf{s}}$	$S^4 A^3 \log \epsilon^{-1}$	Convex optimization
s-rectangular, convex $\mathcal{U}^{\mathsf{s}}$	$S^6 A^4 \epsilon^{-4}$	[27]
s-rectangular, non-convex $\mathcal{U}^{\mathsf{s}}$	NP-hard	[30]
Non-rectangular, convex $\mathcal{U}$	NP-hard	[30]

3 Preliminaries

Notation: We denote the cardinal of an arbitrary finite set $\mathcal{Z}$ by $|\mathcal{Z}|$. Given two real functions $\mathbf{a}, \mathbf{b} : \mathcal{Z} \to \mathbb{R}$, their inner product is $\langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{Z}} := \sum_{z \in \mathcal{Z}} \mathbf{a}(z)\,\mathbf{b}(z)$, which induces the ℓ_2-norm $\|\mathbf{a}\|_2 := \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle_{\mathcal{Z}}}$. More generally, the ℓ_p-norm of $\mathbf{a}$ is denoted by $\|\mathbf{a}\|_p$ whose conjugate norm is $\|\mathbf{a}\|_q := \max_{\|\mathbf{b}\|_p \leq 1} \langle \mathbf{a}, \mathbf{b} \rangle_{\mathcal{Z}}$ with $q^{-1} = 1 - p^{-1}$. The vector of all zeros (resp. all ones) with appropriate dimensions is denoted by $\mathbf{0}$ (resp. $\mathbf{1}$), and the probability simplex over $\mathcal{Z}$ by $\Delta_{\mathcal{Z}} := \{\mathbf{a} : \mathcal{Z} \to \mathbb{R}_+ \,|\, \langle \mathbf{a}, \mathbf{1} \rangle_{\mathcal{Z}} = 1\}$. Finally, $I_{\mathcal{Z}}$ designates the identity matrix in $\mathbb{R}^{\mathcal{Z} \times \mathcal{Z}}$.

3.1 Markov Decision Processes

A Markov decision process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, \gamma, \mu, P, R)$ such that $\mathcal{S}$ and $\mathcal{A}$ are finite state and action spaces of cardinal S and A respectively, $\gamma \in [0,1)$ is a discount factor and $\mu \in \Delta_{\mathcal{S}}$ the initial state distribution. Denoting $\mathcal{X} := \mathcal{S} \times \mathcal{A}$, the couple (P, R) corresponds to the MDP model with $P : \mathcal{X} \to \Delta_{\mathcal{S}}$ being a transition kernel and $R : \mathcal{X} \to \mathbb{R}$ a reward function. A policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ maps each state to a probability distribution over $\mathcal{A}$, and we denote by Π the set of such functions. For any policy $\pi \in \Pi$, $R^\pi \in \mathbb{R}^{\mathcal{S}}$ is the expected immediate reward defined as $R^\pi(s) := \langle \pi_s, R(s, \cdot) \rangle_{\mathcal{A}}, \quad \forall s \in \mathcal{S}$, where π_s is a shorthand for $\pi(\cdot|s)$. We similarly define the stochastic matrix induced by π as $P^\pi(s'|s) := \langle \pi_s, P(s'|s, \cdot) \rangle_{\mathcal{A}}, \quad \forall s, s' \in \mathcal{S}$, and extend the occupation measure to an arbitrary initial vector $k \in \mathbf{R}^{\mathcal{S}}$ by defining

$$d^\pi_{P,k} := k^\top (I_{\mathcal{S}} - \gamma P^\pi)^{-1}.$$

The performance measure we aim to maximize is the value function $v^\pi_{(P,R)} := (I_{\mathcal{S}} - \gamma P^\pi)^{-1} R^\pi$, or alternatively, the return $\rho^\pi_{(P,R)} := \langle \mu, v^\pi_{(P,R)} \rangle_{\mathcal{S}}$. We denote the optimal value function (resp. optimal return) by $v^*_{(P,R)} = \max_{\pi \in \Pi} v^\pi_{(P,R)}$ (resp. $\rho^*_{(P,R)} = \langle \mu, v^*_{(P,R)} \rangle$). It can be obtained using Bellman operators, which are defined as $T^\pi_{(P,R)} v := R^\pi + \gamma P^\pi v$ and $T^*_{(P,R)} v := \max_{\pi \in \Pi} T^\pi_{(P,R)} v, \quad \forall v \in \mathbb{R}^{\mathcal{S}}$, respectively [20]. For any vector $v \in \mathbb{R}^{\mathcal{S}}$, we associate its Q-function $Q \in \mathbb{R}^{\mathcal{X}}$ such that

$$Q(s,a) = r(s,a) + \gamma \langle P(\cdot|s,a), v \rangle_{\mathcal{S}}, \quad \forall (s,a) \in \mathcal{X}\,.$$

With a slight abuse of notation, we can similarly define a Bellman operator over Q-values as

$$T^\pi_{(P,R)} Q(s,a) := r(s,a) + \gamma \sum_{(s',a') \in \mathcal{X}} P(s'|s,a) \pi_{s'}(a') Q(s',a'), \quad \forall (s,a) \in \mathcal{X}\,.$$

3.2 Robust Markov Decision Processes

In a robust MDP setting, we assume that $(P, r) \in \mathcal{U}$ and aim to maximize return under the worst model from the set. We denote the robust performance of a policy $\pi \in \Pi$ by $\rho^\pi_{\mathcal{U}} := \min_{(P,R)\in\mathcal{U}} \rho^\pi_{(P,R)}$. It is maximal when it reaches $\rho^*_{\mathcal{U}} := \max_{\pi\in\Pi} \rho^\pi_{\mathcal{U}}$ at an optimal robust policy $\pi^*_{\mathcal{U}} \in \arg\max_{\pi\in\Pi} \rho^\pi_{\mathcal{U}}$. When considering the robust value function $v^\pi_{\mathcal{U}} := \min_{(P,R)\in\mathcal{U}} v^\pi_{(P,R)}$, we further need to assume that $\mathcal{U}$ is convex and rectangular so that an optimal robust policy realizing $v^*_{\mathcal{U}} := \max_\pi v^\pi_{\mathcal{U}}$ can be computed in polynomial time [30]. We thus assume $\mathcal{U}$ to be convex and rectangular in the remainder of this work. Specifically, we denote an (s, a)-rectangular uncertainty set by $\mathcal{U}^{\mathsf{sa}} := \times_{(s,a)\in\mathcal{X}}(\mathcal{P}_{(s,a)}, \mathcal{R}_{(s,a)})$. It represents a particular case of s-rectangular uncertainty which we similarly denote by $\mathcal{U}^{\mathsf{s}} := \times_{s\in\mathcal{S}}(\mathcal{P}_s, \mathcal{R}_s)$. In both cases, there exists an optimal robust policy that is stationary, although all optimal ones may be stochastic [30].

Similarly to non-robust MDPs, robust MDPs can be solved through Bellman recursion. Indeed, the robust value function $v^\pi_{\mathcal{U}}$ (resp., optimal robust value function $v^*_{\mathcal{U}}$) is known to be the unique fixed point of the γ-contracting robust Bellman operator $T^\pi_{\mathcal{U}} v := \min_{(P,R)\in\mathcal{U}} T^\pi_{(P,R)} v$ (resp., the optimal robust Bellman operator $T^*_{\mathcal{U}} v := \max_{\pi\in\Pi} T^\pi_{\mathcal{U}} v$), both defined for any $v \in \mathbb{R}^{\mathcal{S}}$. Although this ensures linear convergence of robust value iteration, the evaluation of each Bellman operator can still be prohibitive for practical use.

3.2.1 Ball Constrained Uncertainty set

To facilitate the computation of robust Bellman updates, we consider uncertainty sets that are centered around a nominal model (P_0, R_0), i.e., of the form $\mathcal{U} = (P_0, R_0) + (\mathcal{P}, \mathcal{R})$, and constrained according to ℓ_p-norm balls [3, 12, 7, 1]. In the (s, a)-rectangular case, the corresponding uncertainty set is denoted by $\mathcal{U}^{\mathsf{sa}}_p := \mathcal{R}^{\mathsf{sa}}_p \times \mathcal{P}^{\mathsf{sa}}_p = \times_{(s,a)\in\mathcal{X}}(\mathcal{P}_{(s,a)}, \mathcal{R}_{(s,a)})$ where for any $(s, a) \in \mathcal{X}$,

$$\mathcal{R}_{(s,a)} = \{r \in \mathbb{R} \mid \|r\|_p \leq \alpha_{s,a}\}, \quad \text{and} \quad \mathcal{P}_{(s,a)} = \left\{p \in \mathbb{R}^{\mathcal{S}} \mid \langle p, \mathbf{1}\rangle_{\mathcal{S}} = 0, \|p\|_p \leq \beta_{s,a}\right\}.$$

Similarly, an s-rectangular norm-constrained uncertainty is denoted by $\mathcal{U}^{\mathsf{s}}_p := \times_{s\in\mathcal{S}}(\mathcal{P}_s, \mathcal{R}_s)$ where for any $s \in \mathcal{S}$,

$$\mathcal{R}_s = \{r \in \mathbb{R}^{\mathcal{A}} \mid \|r\|_p \leq \alpha_s\}, \quad \text{and} \quad \mathcal{P}_s = \{p \in \mathbb{R}^{\mathcal{X}} \mid \langle p(\cdot, a), \mathbf{1}\rangle_{\mathcal{S}} = 0 \quad \forall a \in \mathcal{A}, \|p\|_p \leq \beta_s\}.$$

In both cases, the noise radius β should be small enough so that transition kernels of the form $P_0 + \mathcal{P}$ are well defined. This normed ball structure on the uncertainty sets enables us to compute robust Bellman updates with similar time complexity as non-robust ones using regularization [3, 12].

First, define the generalized variance function and the mean function as

$$\kappa_q(v) = \min_{w\in\mathbb{R}} \|v - w\mathbf{1}\|_q, \qquad \omega_q(v) \in \arg\min_{w\in\mathbb{R}} \|v - w\mathbf{1}\|_q,$$

respectively, where q is the conjugate value of p (see Tab. 2 for their closed-form expression when $q \in \{1, 2, \infty\}$). Then, we can efficiently evaluate robust value functions by regularizing a standard Bellman operator instead of solving a minimization. We formalize this below.

Proposition 3.1. *([12, Thm. 2-3].) For any policy $\pi \in \Pi$ and any rectangular ℓ_p-ball-constraint uncertainty set, the robust Bellman operator is equivalent to its regularized form:*

$$(T^\pi_{\mathcal{U}} v)(s) = T^\pi_{(P_0,R_0)} v(s) + \Omega_q(\alpha, \beta, v),$$

where $\Omega_q(\alpha, \beta, v) := -\langle \pi_s, \alpha_{s,\cdot} + \gamma\kappa_q(v)\beta^P_{s,\cdot}\rangle_{\mathcal{A}}$ for (s, a)-rectangular uncertainty $\mathcal{U}^{\mathsf{sa}}_p$, and $\Omega_q(\alpha, \beta, v) := -(\alpha_s + \gamma\beta_s\kappa_q(v))\|\pi_s\|_q$ for s-rectangular uncertainty $\mathcal{U}^{\mathsf{s}}_p$.

In the following, we leverage the regularized formulation of robust value functions to explicitly derive RPG for rectangular ℓ_p-ball uncertainty sets.

3.2.2 Robust Gradient Method

Since the robust return can be non-differentiable, we need to follow the projected sub-gradient ascent rule in order to optimize the robust return, namely, update $\pi_{k+1} := \mathbf{proj}_\Pi(\pi_k + \eta\partial_\pi \rho^{\pi_k}_{\mathcal{U}})$ where

$$\partial_\pi \rho^\pi_{\mathcal{U}} := \nabla_\pi \rho^\pi_{(P,R)} \Big|_{(P,R)=(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})}, \tag{1}$$

Table 2: Expressions of the q-mean, the q-variance, and its gradient. We assume that the vector v is sorted, i.e., $v(s_i) \geq v(s_{i+1}), \forall i \in \{1, 2, \cdots, S\}$, and denote $n_l := \lfloor (S+1)/2 \rfloor, n_u := \lceil (S+1)/2 \rceil$.

	$\omega_q(v)$	$\kappa_q(v)$	$\nabla_v \kappa_q(v)$
q	$\arg\min_{w\in\mathbb{R}} \lVert v - w\mathbf{1}\rVert_q$	$\min_{\omega\in\mathbb{R}} \lVert v - \omega\mathbf{1}\rVert_q$	$\frac{\partial \kappa_q(v)}{\partial v(s_i)}$
∞	$\frac{v(s_1)+v(s_S)}{2}$	$\frac{v(s_1)-v(s_S)}{2}$	$\begin{cases} \frac{1}{2} & \text{if } i=1 \\ -\frac{1}{2} & \text{if } i=S \\ 0 & \text{o.w.} \end{cases}$
2	$\frac{\sum_{i=1}^S v(s_i)}{S}$	$\sqrt{\sum_{i=1}^S (v(s_i) - \omega_2(v))^2}$	$\frac{v(s_i)-\omega_2(v)}{\kappa_2(v)}$
1	$\frac{v(s_{n_l})+v(s_{n_u})}{2}$	$\sum_{i=1}^{n_l} (v(s_i) - v(s_{S-i}))$	$\begin{cases} 1 & \text{if } i<n_l \\ -1 & \text{if } i>n_u \\ 0 & \text{o.w.} \end{cases}$

η is the learning rate, $\mathbf{proj}_\Pi$ denotes the orthogonal projection on Π, and $(P^\pi_\mathcal{U}, R^\pi_\mathcal{U})$ is the worst model associated with $\pi \in \Pi$ and $\mathcal{U}$, i.e., $(P^\pi_\mathcal{U}, R^\pi_\mathcal{U}) \in \arg\inf_{(P,R)\in\mathcal{U}} \rho^\pi_{(P,R)}$.

Given oracle access to sub-gradient $\partial\rho^\pi_\mathcal{U}$, projected gradient ascent converges to an ϵ-optimal policy $\pi^*_\mathcal{U}$. Moreover, under similar conditions as in the non-robust setting, projected gradient ascent holds an iteration complexity of $O(S^4A^2\epsilon^{-4})$ [27]. Yet, the sub-gradient in (1) is generally intractable, particularly because general convex uncertainty sets may yield NP-hard complexity. Instead, we propose to focus on ball-constrained uncertainty sets in order to efficiently compute RPG updates.

4 Towards RPG: Expressing the worst quantities

In this section, we provide all the ingredients needed for deriving RPG. Before diving into the gradient expression, we first settle on the general framework of policy gradient. Secondly, in Sec. 4.1, we focus on expressing the worst model according to the nominal explicitly. Surprisingly, we find that the worst transition kernel is a rank-one perturbation of the nominal. This finding enables us to derive the robust occupancy measure, i.e., the occupation measure of the worst kernel in Sec. 4.2. As a last piece, in Sec. 4.3, we propose an alternative definition of robust Q-value and show that it can be estimated from a specific Bellman recursion.

Consider again the projected gradient ascent rule:

$$\pi_{k+1} := \mathbf{proj}_\Pi(\pi_k + \eta\partial_\pi\rho^{\pi_k}_\mathcal{U}).$$

By definition of the sub-gradient in (1) and applying the standard PG theorem [23], it holds that:

$$\partial_\pi\rho^\pi_\mathcal{U} = \sum_{(s,a)\in\mathcal{X}} d^\pi_\mathcal{U}(s)Q^\pi_\mathcal{U}(s,a)\nabla\pi_s(a),$$

where $Q^\pi_\mathcal{U} := Q^\pi_{(P^\pi_\mathcal{U}, R^\pi_\mathcal{U})}$ is the Q-value associated with the worst-case model, and $d^\pi_\mathcal{U} := d^\pi_{P^\pi_\mathcal{U}}$ the occupation measure of the worst transition kernel. In fact, for the uncertainty sets we focus on in this work, the worst Q-value $Q^\pi_\mathcal{U}$ retrieves the common definition of robust Q-value [18, 25] (see the appendix for a detailed discussion). Therefore, for conciseness and with a slight abuse, we shall designate $Q^\pi_\mathcal{U}$ by the robust Q-value, and $d^\pi_\mathcal{U}$ by the robust occupation measure. The remaining question is how to compute these quantities and in particular, can we efficiently find the worst parameters $(P^\pi_\mathcal{U}, R^\pi_\mathcal{U})$? The following part of our study aims to address these questions.

Given an uncertainty set $\mathcal{U}$, let first define the normalized and balanced robust value function as:

$$u^\pi_\mathcal{U}(s) := \frac{\operatorname{sign}(v^\pi_\mathcal{U}(s) - \omega_q(v^\pi_\mathcal{U}))\lVert v^\pi_\mathcal{U}(s) - \omega_q(v^\pi_\mathcal{U})\rVert^{q-1}}{\kappa_q(v^\pi_\mathcal{U})^{q-1}}. \quad (2)$$

By construction, it has zero mean and unit norm, i.e., $\langle u_{\mathcal{U}}^\pi, \mathbf{1}\rangle_{\mathcal{S}} = 0$ and $\|u_{\mathcal{U}}^\pi\|_p = 1$. In fact, as stated in the result below, $u_{\mathcal{U}}^\pi$ is the gradient of the q-variance function, and correlates with the (unnormalized, unbalanced) robust value function according to the same q-variance.

Proposition 4.1. *For any policy $\pi \in \Pi$ and ℓ_p-ball rectangular uncertainty set, the following holds:*

$$u_{\mathcal{U}}^\pi = \nabla_v \kappa_q(v)\Big|_{v=v_{\mathcal{U}}^\pi},$$
$$\langle u_{\mathcal{U}}^\pi, v_{\mathcal{U}}^\pi\rangle = \kappa_q(v_{\mathcal{U}}^\pi).$$

4.1 Worst Kernel and Reward

In the following results, we explicit the relationship between the nominal and the worst-case model for (s,a) and s-rectangular ℓ_p-balls. We will then leverage this relationship to compute the robust Q-values and the robust occupation measure, both necessary for RPG.

Theorem 4.2 ((s,a)-rectangular case)**.** *Given uncertainty set $\mathcal{U} = \mathcal{U}_p^{sa}$ and any policy $\pi \in \Pi$, the worst model is related to the nominal one through:*

$$R_{\mathcal{U}}^\pi(s,a) = R_0(s,a) - \alpha_{s,a} \qquad \textit{and} \qquad P_{\mathcal{U}}^\pi(\cdot|s,a) = P_0(\cdot|s,a) - \beta_{s,a} u_{\mathcal{U}}^\pi.$$

Based on Thm. 4.2, it follows that in the (s,a)-rectangular case, the worst reward function is independent of the employed policy. As we establish in Thm. 4.3 below, this no longer applies under s-rectangularity. In either case, the worst kernel is policy-dependent, discouraging the system to move toward high-rewarding states and directing it to low-rewarding ones instead. Surprisingly, the vector penalty $u_{\mathcal{U}}^\pi \in \mathbb{R}^{\mathcal{S}}$ additionally illustrates that the worst kernel is a rank-one perturbation of the nominal. Indeed, considering the stochastic matrix induced by any policy $\pi \in \Pi$, we have

$$[P_{\mathcal{U}}^\pi - P_0^\pi](s'|s) = -\left(\sum_{a\in\mathcal{A}} \beta_{s,a}\pi_s(a)\right) u_{\mathcal{U}}^\pi(s'), \quad \forall s \in \mathcal{S},$$

so that the perturbation matrix $P_{\mathcal{U}}^\pi - P_0^\pi$ is of rank one. In the sequel, we will leverage this finding to compute the robust occupation measure.

Theorem 4.3 (s-rectangular case)**.** *Given uncertainty set $\mathcal{U} = \mathcal{U}_p^{s}$ and any policy $\pi \in \Pi$, the worst model is related to the nominal one through:*

$$R_{\mathcal{U}}^\pi(s,a) = R_0(s,a) - \alpha_s\left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1} \quad \textit{and} \quad P_{\mathcal{U}}^\pi(\cdot|s,a) = P_0(\cdot|s,a) - \beta_s u_{\mathcal{U}}^\pi\left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1}.$$

Similarly to the (s,a)-case, the adversarial kernel is a rank-one perturbation of the nominal. Yet, an extra dependence on the policy through the coefficient $\left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1}$ appears in the s-case, affecting both the worst reward and the worst kernel. Intuitively, it means that the worst model cannot be chosen independently for each action, but must instead depend on the agent's policy. This further explains why optimal policies can all be stochastic in s-rectangular robust MDPs [30].

Thms. 4.2 and 4.3 enable us to derive the worst MDP model in closed form with time complexity $O(S^2 A \log \epsilon^{-1})$, up to logarithmic factors (please see the appendix for a detailed discussion). It thus holds the same complexity as non-robust value iteration, since we additionally need to compute the value function to derive its corresponding regularizer [3, 12]. On the other hand, if we employ convex optimization using value methods instead, obtaining the worst model requires a time complexity of $O(S^4 A \log \epsilon^{-1})$ in the (s,a)-rectangular case, and $O(S^4 A^3 \log \epsilon^{-1})$ in the s-rectangular case [27][Sec. 4.1].

4.2 Robust Occupation Measure

We finally derive the robust occupation measure using nominal values, which will lead to an explicit RPG. Although intractable in general, we show that focusing on ball-constrained uncertainty enables deriving the robust occupation matrix efficiently from the (nominal) occupation measure. We first establish the lemma below, which leverages the fact that the worst transition function is a rank-one perturbation of the nominal and represents our core contribution.

Lemma 4.4. *Let $b, k \in \mathbb{R}^{\mathcal{S}}$ and $P_0, P_1 \in (\Delta_{\mathcal{S}})^{\mathcal{S}}$ two transition matrices. If $P_1 = P_0 - bk^\top$, i.e., P_1 is a rank-one perturbation of P_0, then their occupation matrices $D_i := (I - \gamma P_i)^{-1}, i = 0, 1$ are related through:*

$$D_1 = D_0 - \gamma \frac{D_0 b k^\top D_0}{(1 + \gamma k^\top D_0 b)}.$$

Combining Thms. 4.2 and 4.3 with the above lemma, we obtain the robust occupation in terms of the nominal, as stated in Thm. 4.6 below. Prior to this, we introduce the notion of *expected transition uncertainty* below.

Definition 4.5. *Let $\mathcal{U}$ a rectangular ℓ_p-ball-constrained uncertainty set of transition radius β. For any policy $\pi \in \Pi$, the expected transition uncertainty at any state $s \in \mathcal{S}$ is given by $\beta_s^\pi := \sum_{a \in \mathcal{A}} \pi_s(a) \beta_{s,a}$ if $\mathcal{U} = \mathcal{U}_p^{sa}$, and $\beta_s^\pi := \beta_s \|\pi_s\|_q$ if $\mathcal{U} = \mathcal{U}_p^s$.*

Theorem 4.6. *For any rectangular ℓ_p-ball-constrained uncertainty and $\pi \in \Pi$, it holds that:*

$$d_{\mathcal{U},\mu}^\pi = d_{P_0,\mu}^\pi - \gamma \frac{\langle d_{P_0,\mu}^\pi, \beta^\pi \rangle_{\mathcal{S}}}{1 + \gamma \langle d_{P_0,u_{\mathcal{U}}^\pi}^\pi, \beta^\pi \rangle_{\mathcal{S}}} d_{P_0,u_{\mathcal{U}}^\pi}^\pi. \tag{3}$$

Thm. 4.6 explicitly highlights the relationship between the robust occupation measure and the nominal one. Thus, according to Eq. (3), the standard non-robust occupation measure in the first term needs to be penalized by another one, $d_{P_0,u_{\mathcal{U}}^\pi}^\pi = (u_{\mathcal{U}}^\pi)^\top (I_{\mathcal{S}} - \gamma P_0^\pi)^{-1}$, to obtain the robust occupation measure. Recall that $u_{\mathcal{U}}^\pi$ is the balanced-scaled value function determined by $\pi \in \Pi$ and uncertainty set $\mathcal{U}$. Thus, the penalty term $d_{P_0,u_{\mathcal{U}}^\pi}^\pi$ tends to zero if all coordinates of the robust value function vector converge to the same value.

Nonetheless, our expression (3) does present some challenges. First, the occupation measure appearing in the correction term indicates that instead of taking a fixed initial state distribution, we should start from a *varying* and *signed* measure represented by the balanced value function. Although it suggests putting more weight on worst-performing states, obtaining a non-biased estimator for this occupancy measure remains unclear in model-free learning.

4.3 Robust Q-values

In this section, we focus on the last element needed for RPG and aim to estimate the robust Q-value denoted previously by $Q_{\mathcal{U}}^\pi := Q_{(P_{\mathcal{U}}^\pi, R_{\mathcal{U}}^\pi)}^\pi$. Define its associated value function as $v_{\mathcal{U}}^\pi(s) = \langle \pi_s, Q_{\mathcal{U}}^\pi(s, \cdot) \rangle, \forall s \in \mathcal{S}, \pi \in \Pi$. Based on standard Bellman recursion, it thus holds that:

$$Q_{\mathcal{U}}^\pi(s,a) = R_{\mathcal{U}}^\pi(s,a) + \gamma \langle P_{\mathcal{U}}^\pi(\cdot|s,a), v_{\mathcal{U}}^\pi \rangle_{\mathcal{S}}, \quad \forall (s,a) \in \mathcal{X}, \pi \in \Pi,$$

while $Q_{\mathcal{U}}^\pi$ is the unique fixed point of the γ-contracting operator

$$(\mathcal{L}_{\mathcal{U}}^\pi Q)(s,a) := T_{(P_{\mathcal{U}}^\pi, R_{\mathcal{U}}^\pi)}^\pi Q(s,a), \quad \forall Q \in \mathbb{R}^{\mathcal{X}}. \tag{4}$$

The relations above hold for general uncertainty sets, provided that we have access to the worst model. The s-rectangularity assumption additionally enables us to retrieve the robust value function using the Bellman operator above [30]. Concretely, we have: $v_{\mathcal{U}}^\pi = \min_{(P,R) \in \mathcal{U}} v_{(P,R)}^\pi = v_{(P_{\mathcal{U}}^\pi, R_{\mathcal{U}}^\pi)}^\pi$.

The following result derives a regularized operator equivalent to $\mathcal{L}_{\mathcal{U}}^\pi$, which results in an efficient iteration method to compute the robust Q-value.

Proposition 4.7. *The Bellman operator $\mathcal{L}_{\mathcal{U}}^\pi$ defined in Eq.* (4) *is equivalent to:*

$$(\mathcal{L}_{\mathcal{U}}^\pi Q)(s,a) = T_{(P_0,R_0)}^\pi Q(s,a) + \Omega_q'(\alpha_{s,a}, \beta_{s,a}, v),$$

where $v(s) := \langle \pi_s, Q(s, \cdot) \rangle_{\mathcal{A}}$, $\Omega_q'(\alpha, \beta, v) := -(\alpha_{s,a} + \gamma \beta_{s,a} \kappa_q(v))$ for (s,a)-rectangular uncertainty $\mathcal{U}_p^{sa}$, and $\Omega_q'(\alpha, \beta, v) := -\left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1} (\alpha_s + \gamma \beta_s \kappa_q(v))$ for s-rectangular $\mathcal{U}_p^s$.

5 Robust Policy Gradient

We are now able to derive an RPG by combining our previous results. Notably, unlike previous works that need to sample next-state transitions based on all models from the uncertainty set [19, 15, 4], here, we only need the nominal kernel to get the occupation measures.

Theorem 5.1 (RPG). *For any rectangular ℓ_p-ball-constrained uncertainty, the robust policy gradient is given by:*

$$\partial_\pi \rho^\pi_\mathcal{U} = \sum_{(s,a)\in\mathcal{X}} \left(d^\pi_{P_0,\mu}(s) - c^\pi(s)\right) Q^\pi_\mathcal{U}(s,a)\nabla\pi_s(a), \tag{5}$$

where

$$c^\pi(s) := \frac{\gamma\langle d^\pi_{P_0,\mu}, \beta^\pi\rangle_\mathcal{S}}{1+\gamma\langle d^\pi_{P_0,u^\pi_\mathcal{U}}, \beta^\pi\rangle_\mathcal{S}} d^\pi_{P_0,u^\pi_\mathcal{U}}(s), \quad \forall s\in\mathcal{S}\,.$$

Thm. 5.1 is a direct application of non-robust PG, as its proof simply consists in plugging Eq. (3) into the standard PG expression $\partial_\pi\rho^\pi_\mathcal{U} = \sum_{(s,a)\in\mathcal{X}} d^\pi_{\mathcal{U},\mu}(s)Q^\pi_\mathcal{U}(s,a)\nabla\pi_s(a)$. We obtain a regular PG in the first term, with the robust Q-value instead of the non-robust one, plus a correction term c^π resulting from taking the occupation measure of the worst kernel instead of the nominal. Unlike previous work that uses policy regularization to achieve empirical robustness in PG methods [2, 9], Thm. 5.1 establishes an RPG that accounts for transition uncertainty and targets a robust optimal policy.

5.1 Complexity Analysis

A major concern in solving robust MDPs is time complexity [30]. Similarly, it is of major importance to assess the additional time required for computing an RPG update, compared to its non-robust variant. Although previous work has analyzed the convergence rate of RPG to a global optimum [27], it assumes access to an oracle gradient, thus occulting the computational concerns raised from gradient estimation. In fact, the NP-hardness of non-rectangular and/or non-convex robust MDPs [30] already indicates that their resulting RPG can be intractable.

To compute RPG in Thm. 5.1, we first need to evaluate the robust Q-value. Based on Lemma 4.7 and the Bellman operators introduced there, our evaluation method involves an additional estimation of the variance function κ_p. According to [12], this takes logarithmic time at most, using binary search. As to the compensation term c^π in Eq. (11), it requires computing occupancy measures with respect to two different initial vectors, namely the balanced value function and the initial distribution. Thus, the computational cost for estimating c^π is the same as estimating a non-robust occupancy measure. Tab. 1 summarizes the complexity of different approaches while a detailed discussion can be found in the appendix. We refer to [27][Sec. 4.1] for the complexity of RPG based on convex optimization.

Generalization to arbitrary norms. Until now, we have focused on ℓ_p-norm for concreteness. However, the above results apply to any norm $\|\cdot\|$, at least if the uncertainty set is (s,a)-rectangular, in which case the variance function changes to $\kappa(v) := \min_{\|c\|\leq 1, \mathbf{1}^\top c=0}\langle c, v\rangle$ and the balanced value to $\arg\min_{\|c\|\leq 1, \mathbf{1}^\top c=0}\langle c, v\rangle$. The rank-one perturbation structure of the worst kernel is preserved, so the robust occupation measure can be obtained similarly using Lemma 4.4. The s-rectangular is more involved. We defer its discussion to the appendix and leave its complete derivation for future work.

6 Experiments

In order to test the effectiveness of our RPG update, we evaluate its increased time complexity relative to non-robust PG. In the following experiments, we randomly generate nominal models for arbitrary state-action space sizes. Each experiment was averaged over 100 runs. We refer the reader to the appendix for more details on the radius levels and other implementation choices.

We first focus on ℓ_1-robust MDPs to compare our RPG with a convex optimization approach. Specifically, we consider a robust PG with an optimization solver, which we designate by LP-RPG. Indeed, recall that ℓ_1-ball-constraints induce a linear program (LP) rather than a more general convex optimization problem. Therefore, to compute the robust value function for a given policy, we iteratively evaluate the robust Bellman operator using LP [27, Section 4.1]. Using this approximated value function, we can compute the worst value parameters to apply PG theorem by [23] and deduce an LP-based robust PG update. Differently, our RPG method relies on the regularized formulation of robust value iteration proposed in [3, 12], from which we deduce the normalized-balanced value function as in Eq. (10). We finally apply Thm. 4.6 to compute the robust occupation measure, and Prop. 4.7 to obtain the robust Q-value.

Tab. 3 displays the results obtained for the two alternative methods described above. In all experiments, the standard deviation was typically 2-10% so we omitted it for brevity. As can be seen in Tab. 3, LP-RPG does not scale well compared to RPG, whereas RPG has similar time complexity as PG. Notably, the running time of s-rectangular LP-RPG scales much better with the space size than its (s, a)-rectangular equivalent, which confirms the theoretical complexities from Tab. 1. Yet, since these methods were time-consuming, we repeated these for a few runs only. In fact, LP-RPG is more expensive than RPG by 1-3 orders of magnitude, which illustrates its inefficiency. We emphasize that here, we only focused on ℓ_1-robust MDPs to leverage LP solvers in robust policy evaluation. We expect the computational cost of LP-RPG to scale even more poorly for other ℓ_p-robust MDPs that involve polynomial time-consuming convex programs.

Table 3: Comparison of the relative running time between RPG and the convex optimization approach (here, LP). Our method is faster than LP-based updates by 1 to 3 orders of magnitude.

		$\{(P_0, R_0)\}$	$\mathcal{U}_1^{\mathsf{sa}}$		$\mathcal{U}_1^{\mathsf{s}}$	
S	A	PG	RPG	LP-RPG	RPG	LP-RPG
10	10	1	1.4	326	1.4	77
30	10	1	1.4	351	1.4	109
50	10	1	1.4	408	1.4	159
100	20	1	1.5	469	1.3	268
500	50	1	1.3	925	1.3	5343

We further compare our RPG to non-robust PG on different ℓ_p-balls. Tab. 4 confirms the comparable time complexity of RPG to non-robust PG, thus demonstrating the effectiveness of our method. We note that for $p \in \{1, 2, \infty\}$, the corresponding regularization quantities can be computed in closed form, whereas they involve a binary search for other values [12]. We thus get a slight running-time increase for $p \in \{5, 10\}$.

Table 4: Relative running time for computing RPG under different types of uncertainty sets.

S	A	$\{(P_0, R_0)\}$	$\mathcal{U}_2^{\mathsf{sa}}$	$\mathcal{U}_2^{\mathsf{s}}$	$\mathcal{U}_5^{\mathsf{sa}}$	$\mathcal{U}_5^{\mathsf{s}}$	$\mathcal{U}_{10}^{\mathsf{sa}}$	$\mathcal{U}_{10}^{\mathsf{s}}$	$\mathcal{U}_\infty^{\mathsf{sa}}$	$\mathcal{U}_\infty^{\mathsf{s}}$
10	10	1	1.5	1.5	4.9	4.7	4.7	4.9	1.5	1.6
30	10	1	1.4	1.5	4.2	4.3	4.2	4.0	1.4	1.4
50	10	1	1.5	1.4	4.5	4.1	4.0	4.0	1.4	1.4
100	20	1	1.4	1.3	2.6	2.5	2.5	2.4	1.3	1.2
500	50	1	1.2	1.2	1.7	1.7	1.7	1.7	1.2	1.3

7 Discussion

This paper introduced an explicit expression of RPG for rectangular robust MDPs. Our approach involved auxiliary results such as deriving the worst model in closed form and showing that it is a rank-one perturbation of the nominal kernel. The resulting RPG extends vanilla PG with additional correction terms that can be derived in closed form as well. Thus, the computational time of RPG is similar to its non-robust variant.

A key assumption that would be interesting to relax is the normed-ball structure of the uncertainty sets considered in this study. Indeed, since the proofs of our technical results rely on norm properties, it is still unclear if and how RPG can generalize to metric-based or f-divergence uncertainty sets. The latter type of uncertainty can be particularly useful for data-driven settings, as the radius can be chosen according to cross-validation or statistical bounds [8]. Another compelling direction would be to explore other variants of RPG using mirror descent or natural policy gradient and examine their compatibility with deep architectures, which would further demonstrate the practical efficiency of our RPG method.

References

[1] Bahram Behzadian, Marek Petrik, and Chin Pang Ho. Fast algorithms for ℓ_∞-constrained s-rectangular robust MDPs. *Advances in Neural Information Processing Systems*, 34:25982–

25992, 2021.

[2] Rob Brekelmans, Tim Genewein, Jordi Grau-Moya, Grégoire Delétang, Markus Kunesch, Shane Legg, and Pedro Ortega. Your policy regularizer is secretly an adversary. *Transactions on Machine Learning Research (TMLR)*, 2022.

[3] Esther Derman, Matthieu Geist, and Shie Mannor. Twice regularized MDPs and the equivalence between robustness and regularization. *Advances in Neural Information Processing Systems*, 34:22274–22287, 2021.

[4] Esther Derman, Daniel J. Mankowitz, Timothy A. Mann, and Shie Mannor. Soft-robust actor-critic policy-gradient. *AUAI press for Association for Uncertainty in Artificial Intelligence*, pages 208–218, 2018.

[5] Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. *International Conference on Learning Representations*, 2022.

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[7] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Partial policy iteration for l_1-robust Markov decision processes. *J. Mach. Learn. Res.*, 22:275–1, 2021.

[8] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Robust ϕ-divergence MDPs. *arXiv preprint arXiv:2205.14202*, 2022.

[9] Hisham Husain, Kamil Ciosek, and Ryota Tomioka. Regularized policies are reward robust. In *International Conference on Artificial Intelligence and Statistics*, pages 64–72. PMLR, 2021.

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

[11] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*. Citeseer, 2002.

[12] Navdeep Kumar, Kfir Levy, Kaixin Wang, and Shie Mannor. Efficient policy iteration for robust markov decision processes via regularization, 2022.

[13] Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Sparse Markov decision processes with causal sparse Tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1466–1473, 2018.

[14] Yan Li, Tuo Zhao, and Guanghui Lan. First-order policy optimization for robust markov decision process. *arXiv preprint arXiv:2209.10579*, 2022.

[15] Daniel Mankowitz, Timothy Mann, Pierre-Luc Bacon, Doina Precup, and Shie Mannor. Learning robust options. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[16] Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

[17] Paul Milgrom and Ilya Segal. Envelope theorems for arbitrary choice sets. *Econometrica*, 70:583–601, 02 2002.

[18] Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[19] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR, 2017.

[20] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[21] John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning, 2017.

[22] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[23] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 99, pages 1057–1063. Citeseer, 1999.

[24] Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. *Advances in neural information processing systems*, 28, 2015.

[25] Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In *International Conference on Machine Learning*, pages 181–189. PMLR, 2014.

[26] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

[27] Qiuhao Wang, Chin Pang Ho, and Marek Petrik. On the convergence of policy gradient in robust mdps. *arXiv preprint arXiv:2212.10439*, 2022.

[28] Yue Wang, Fei Miao, and Shaofeng Zou. Robust constrained reinforcement learning. *arXiv preprint arXiv:2209.06866*, 2022.

[29] Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning. In *International Conference on Machine Learning*, pages 23484–23526. PMLR, 2022.

[30] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

[31] Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86:391–423, 2012.

Contents

A Balanced and Normed Vectors **13**

A.1 Proof of Proposition 4.1 14

B Worst Kernel and Reward **14**

B.1 Proof of Theorem 4.2 14

B.2 Proof of Theorem 4.3 15

C Occupation Matrix **16**

C.1 Proof of Lemma 4.4 16

C.2 Proof of Theorem 4.6 17

D Robust Q-value **17**

D.1 Basic Properties 17

D.2 Evaluation 18

D.3 Convergence 19

E Robust Policy Gradient **20**

F Complexity Analysis **20**

F.1 Helper results for Robust MDPs 21

F.2 Computing the Policy gradient. 22

G Generalization to arbitrary norms **23**

G.1 $\mathtt{sa}$-rectangular robust MDPs. 23

G.2 s-rectangular Case 23

G.3 Generalization to non-norms 24

H Experiments **24**

H.1 RPG by LP 24

A Balanced and Normed Vectors

In this section, we lay down some basic properties of p-normalized-balanced vectors.

First recall the p-variance and the p-mean defined as:

$$\kappa_p(v) = \min_{\omega \in \mathbb{R}} \|v - \omega \mathbf{1}\|_p, \qquad \omega_p(v) = \arg\min_{\omega \in \mathbb{R}} \|v - \omega \mathbf{1}\|_p.$$

Given any $v \in \mathbb{R}^{\mathcal{S}}$, let also the p-balanced-normalized function:

$$u_p(v)(s) := \text{SIGN}(v(s) - \omega_p(v)) \left(\frac{|v(s) - \omega_p(v)|}{\kappa_p(v)} \right)^{p-1}, \quad \forall v \in \mathbb{R}^{\mathcal{S}}, s \in \mathcal{S}.$$

According to [12][Sec. 16.1, Lemma 1], the following holds:

$$\kappa_q(v) = -\frac{1}{\epsilon} \left[\min_{\|c\|_p \le \epsilon, \langle c, \mathbf{1} \rangle = 0} \langle c, v \rangle \right], \tag{6}$$

namely, the p-variance function is the optimal value of a linear optimization under kernel noise constraint. The result below further characterizes the solution to the above problem.

Lemma A.1. *The vector defined as $c^* := -\epsilon u_q(v)$ is an optimal solution to the optimization problem*

$$\min_{\|c\|_p \le \epsilon, \langle c, \mathbf{1} \rangle_{\mathcal{S}} = 0} \langle c, v \rangle.$$

Proof. It suffices to show that c^* satisfies both constraints $\|c^*\|_p \le \epsilon$ and $\langle c^*, \mathbf{1} \rangle_{\mathcal{S}} = 0$, and that it reaches optimal value, i.e., $-\frac{1}{\epsilon}\langle c^*, v \rangle = \kappa_q(v)$. We thus compute:

$$\begin{aligned}
\|c^*\|_p &= \left(\sum_{s \in \mathcal{S}} |c^*(s)|^p \right)^{\frac{1}{p}} \\
&= \left(\sum_{s \in \mathcal{S}} \left| -\epsilon \text{SIGN}(v(s) - \omega_q(v)) \left(\frac{|v(s) - \omega_q(v)|}{\kappa_q(v)} \right)^{q-1} \right|^p \right)^{\frac{1}{p}} \\
&= \left(\left(\frac{\epsilon}{\kappa_q(v)^{q-1}} \right)^p \sum_{s \in \mathcal{S}} \left| \left(\frac{|v(s) - \omega_q(v)|}{\kappa_q(v)} \right)^{q-1} \right|^p \right)^{\frac{1}{p}} \\
&= \frac{\epsilon}{\kappa_q(v)^{q-1}} \left(\sum_{s \in \mathcal{S}} |v(s) - \omega_q(v)|^{(q-1)p} \right)^{\frac{1}{p}} \\
&= \frac{\epsilon}{\kappa_q(v)^{q-1}} \left(\sum_{s \in \mathcal{S}} |v(s) - \omega_q(v)|^{q} \right)^{\frac{1}{p}} && \text{(By assumption, } \frac{p+q}{pq} = 1) \\
&= \frac{\epsilon}{\kappa_q(v)^{q-1}} \kappa_q(v)^{\frac{q}{p}} && \text{(By definition, } \kappa_q(v) = \|v - \omega_q \mathbf{1}\|_q) \\
&= \epsilon, && (\frac{q}{p} - (q-1) = \frac{q - pq + p}{p} = 0)
\end{aligned}$$

so the norm constraint is satisfied. We check the noise constraint by computing:

$$\begin{aligned}
\sum_{s \in \mathcal{S}} c^*(s) &= \sum_{s \in \mathcal{S}} -\epsilon \text{SIGN}(v(s) - \omega_q(v)) \left(\frac{|v(s) - \omega_q(v)|}{\kappa_q(v)} \right)^{q-1} \\
&= \frac{-\epsilon}{\kappa_q(v)^{q-1}} \sum_{s \in \mathcal{S}} \text{SIGN}(v(s) - \omega_q(v)) |v(s) - \omega_q(v)|^{q-1}.
\end{aligned}$$

Now, considering the real function $\varphi : w \to \|v - w\mathbf{1}\|_q$ and taking its derivative, we remark the proportional relation:

$$\sum_{s \in \mathcal{S}} c^*(s) = C \cdot \varphi'(\omega_q(v)),$$

where $C \in \mathbb{R}$ is the proportionality coefficient. By construction, $\omega_q(v)$ is a minimizer of φ, so we must have $\varphi'(\omega_q(v)) = 0$ and c^* satisfies the noise constraint.

We finally show that c^* reaches the optimal value:

$$\begin{aligned}
-\frac{1}{\epsilon}\langle c^*, v\rangle &= -\frac{1}{\epsilon}\langle c^*, v - \omega_q(v)\mathbf{1}\rangle && (\langle c^*, \mathbf{1}\rangle_{\mathcal{S}} = 0) \\
&= \sum_{s\in\mathcal{S}} \frac{|v(s) - \omega_q(v)|^q}{\kappa_q(v)^{q-1}} && \text{(Putting the value of } c^*) \\
&= \frac{\kappa_q(v)^q}{\kappa_q(v)^{q-1}} && (\kappa_q(v) = \|v - \omega_q \mathbf{1}\|_q) \\
&= \kappa_q(v).
\end{aligned}$$

□

A.1 Proof of Proposition 4.1

Proposition. *For any policy $\pi \in \Pi$ and ℓ_p-ball rectangular uncertainty set, the following holds:*

$$u^\pi_{\mathcal{U}} = \nabla_v \kappa_q(v) \Big|_{v = v^\pi_{\mathcal{U}}},$$
$$\langle u^\pi_{\mathcal{U}}, v^\pi_{\mathcal{U}}\rangle = \kappa_q(v^\pi_{\mathcal{U}}).$$

Proof. The second claim directly follows from Lemma A.1 applied to $v := v^\pi_{\mathcal{U}}$, so that by optimality, $\kappa_q(v^\pi_{\mathcal{U}}) = \langle u^\pi_{\mathcal{U}}, v^\pi_{\mathcal{U}}\rangle$. For the first claim, we take the gradient of $\kappa_p(v) := \min_{w\in\mathbb{R}} \|v - w\mathbf{1}\|_p$ w.r.t. v using the envelope theorem [17]. Then, the p-balanced-normalized vector $u_p(v)$ is a sub-gradient of $\kappa_p(v)$, that is,

$$u_p(v) = \nabla \kappa_q(v),$$

which we apply to $v := v^\pi_{\mathcal{U}}$. □

We have the additional properties below:

- The variance function κ_q is translation-invariant in all-ones directions, i.e., for all $\omega \in \mathbb{R}, \kappa_q(v) = \kappa_q(v + \omega\mathbf{1})$. As a result, $\langle \nabla \kappa_q(v), \mathbf{1}\rangle_{\mathcal{S}} = 0$.
- The balanced-normalized vector $u_p(v)$ has unit norm, i.e., $\|u_p(v)\|_p = 1$ by Lemma A.1.

B Worst Kernel and Reward

Here we present the proofs for the worst/adversarial kernel and reward function characterization.

B.1 Proof of Theorem 4.2

Theorem ((s,a)-rectangular case)**.** *Given uncertainty set $\mathcal{U} = \mathcal{U}^{sa}_p$ and any policy $\pi \in \Pi$, the worst model is related to the nominal one through:*

$$R^\pi_{\mathcal{U}}(s,a) = R_0(s,a) - \alpha_{s,a} \qquad \textit{and} \qquad P^\pi_{\mathcal{U}}(\cdot|s,a) = P_0(\cdot|s,a) - \beta_{s,a} u^\pi_{\mathcal{U}}.$$

Proof. By definition,

$$(P^\pi_{\mathcal{U}^{\mathsf{sa}}_p}, R^\pi_{\mathcal{U}^{\mathsf{sa}}_p}) \in \underset{(P,R)\in\mathcal{U}^{\mathsf{sa}}_p}{\arg\min}\, T^\pi_{(P,R)} v^\pi_{\mathcal{U}^{\mathsf{sa}}_p}.$$

Additionally, since $\mathcal{U}^{\mathsf{sa}}_p = (R_0 + \mathcal{R}) \times (P_0 + \mathcal{P})$, it results that:

$$(R^\pi_{\mathcal{U}^{\mathsf{sa}}_p}, P^\pi_{\mathcal{U}^{\mathsf{sa}}_p}) = (P_0 + P^*, R_0 + R^*)$$

where

$$(P^*, R^*) \in \underset{(P,R)\in\mathcal{P}\times\mathcal{R}}{\arg\min}\, T^\pi_{(P,R)} v^\pi_{\mathcal{U}^{\mathsf{sa}}_p}.$$

By the (s,a)-rectangularity assumption, we get that for all $(s,a) \in \mathcal{X}$,

$$(P^*(\cdot|s,a), R^*(s,a)) \in \underset{(p_{s,a},r_{s,a}) \in \mathcal{P}_{s,a} \times \mathcal{R}_{s,a}}{\arg\min} \left\{ r_{s,a} + \gamma \sum_{s' \in \mathcal{S}} p_{s,a}(s') v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') \right\}$$

It is clear from the above that the worst reward is independent of policy π. Thus, by the ball constraint, it is given by

$$R^*(s,a) = -\alpha_{s,a}, \quad \forall (s,a) \in \mathcal{X}.$$

Differently, the worst kernel depends on the value function which itself depends on the policy. It is given by

$$P^*(\cdot|s,a) = \underset{p_{s,a} \in \mathcal{P}_{sa}}{\arg\min} \left\{ \sum_{s' \in \mathcal{S}} p_{s,a}(s') v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') \right\}, \quad \forall (s,a) \in \mathcal{X}.$$

The optimization is of the form

$$\underset{\|c\|_p \leq \beta, \langle c, \mathbf{1} \rangle = 0}{\arg\min} \langle c, v \rangle,$$

so by Lemma A.1,

$$P^*(s'|s,a) = -\beta_{s,a} \text{SIGN}\left(v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') - \omega_q(v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}) \right) \frac{\left| v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') - \omega_q(v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}) \right|^{q-1}}{\kappa_q(v)^{q-1}}.$$

As a result, we proved that for all $(s,a) \in \mathcal{X}$, $R^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s,a) = R_0(s,a) - \alpha_{s,a}$ and

$$P^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s'|s,a) = P_0(s'|s,a) - \beta_{s,a} \text{SIGN}\left(v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') - \omega_q(v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}) \right) \frac{\left| v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') - \omega_q(v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}) \right|^{q-1}}{\kappa_q(v)^{q-1}}.$$

$\square$

B.2 Proof of Theorem 4.3

Theorem (s-rectangular case)**.** *Given uncertainty set $\mathcal{U} = \mathcal{U}^{\mathsf{s}}_p$ and any policy $\pi \in \Pi$, the worst model is related to the nominal one through:*

$$R^{\pi}_{\mathcal{U}}(s,a) = R_0(s,a) - \alpha_s \left(\frac{\pi_s(a)}{\|\pi_s\|_q} \right)^{q-1} \quad \textit{and} \quad P^{\pi}_{\mathcal{U}}(\cdot|s,a) = P_0(\cdot|s,a) - \beta_s u^{\pi}_{\mathcal{U}} \left(\frac{\pi_s(a)}{\|\pi_s\|_q} \right)^{q-1}.$$

Proof. By definition,

$$(P^{\pi}_{\mathcal{U}^{\mathsf{s}}_p}, R^{\pi}_{\mathcal{U}^{\mathsf{s}}_p}) \in \underset{(P,R) \in \mathcal{U}^{\mathsf{s}}_p}{\arg\min} T^{\pi}_{(P,R)} v^{\pi}_{\mathcal{U}^{\mathsf{s}}_p},$$

and since $\mathcal{U}^{\mathsf{s}}_p = (R_0 + \mathcal{R}) \times (P_0 + \mathcal{P})$, we have

$$(P^{\pi}_{\mathcal{U}^{\mathsf{s}}_p}, R^{\pi}_{\mathcal{U}^{\mathsf{s}}_p}) = (P_0 + P^*, R_0 + R^*)$$

where

$$(P^*, R^*) \in \underset{(P,R) \in \mathcal{P} \times \mathcal{R}}{\arg\min} T^{\pi}_{(P,R)} v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}.$$

By the s-rectangularity assumption, we get that for all $s \in \mathcal{S}$

$$(P^*(\cdot|s,\cdot), R^*(s,\cdot)) = \underset{(p_s,r_s) \in \mathcal{P}_s \times \mathcal{R}_s}{\arg\min} \sum_{a \in \mathcal{A}} \pi_s(a) \left\{ r_{s,a} + \gamma \sum_{s' \in \mathcal{S}} p_{s,a}(s') v^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p}(s') \right\}.$$

Here, the worst reward does depend on policy π and is given by

$$R^*(s,a) = -\alpha_s \frac{\pi_s(a)^{q-1}}{\sum_a \pi_s(a)^{q-1}}, \qquad \forall (s,a) \in \mathcal{X}.$$

As for the worst kernel, it depends both on the value function and the policy. It is given by

$$P^*(\cdot|s,\cdot) = \arg\min_{p_s \in \mathcal{P}_s} \left\{ \sum_{a\in\mathcal{A}} \pi_s(a) \sum_{s'\in\mathcal{S}} p_{s,a}(s') v^{\pi}_{\mathcal{U}^{\mathtt{s}}_p}(s') \right\}.$$

The optimization of interest is of the form

$$\min_{\|c_a\|_p \le \beta_s, \langle c_a, \mathbf{1}\rangle = 0, a\in\mathcal{A}} \left\{ \sum_{a'\in\mathcal{A}} \pi_s(a') \langle c_{a'}, v\rangle \right\},$$

which is equivalent to the following two-fold minimization:

$$\min_{\sum_{a\in\mathcal{A}} (\beta_{s,a})^p \le (\beta_s)^p} \quad \min_{\|c_a\|_p \le \beta_s, \langle c_a, \mathbf{1}\rangle = 0, a\in\mathcal{A}} \left\{ \sum_{a'\in\mathcal{A}} \pi_s(a') \langle c_{a'}, v\rangle \right\}.$$

Thus, rewriting the problem in our context,

$$\begin{aligned}
&\min_{\sum_a (\beta_{s,a})^p \le (\beta_s)^p} \quad \min_{\|p_{sa}\|_p \le \beta_{s,a}, \sum_{s'} p_{sa}(s') = 0} \quad \sum_a \pi_s(a) \langle p_{s,a}, v\rangle \\
=& \min_{\sum_a (\beta_{s,a})^p \le (\beta_s)^p} \sum_a \pi_s(a) \quad \min_{\|p_{sa}\|_p \le \beta_{s,a}, \sum_{s'} p_{sa}(s') = 0} \quad \langle p_{s,a}, v\rangle \\
=& \min_{\sum_a (\beta_{sa})^p \le (\beta_s)^p} \sum_a \pi_s(a)(-\beta_{sa}\kappa_q(v)) && \text{(By Lemma A.1)} \\
=& -\kappa_q(v) \max_{\sum_a (\beta_{sa})^p \le (\beta_s)^p} \sum_a \pi_s(a)\beta_{sa}.
\end{aligned}$$

Computing the optimal β above, the optimization is now the same as in the (s,a)-rectangular case. Hence, we have

$$P^*(s'|s,a) = -\beta_s \frac{\pi_s(a)^{q-1}}{\|\pi_s\|_q^{q-1}} \mathtt{SIGN}(v^{\pi}_{\mathcal{U}^{\mathtt{s}}_p}(s') - \omega_q(v^{\pi}_{\mathcal{U}^{\mathtt{s}}_p})) \frac{\left| v^{\pi}_{\mathcal{U}^{\mathtt{s}}_p}(s') - \omega_q(v^{\pi}_{\mathcal{U}^{\mathtt{s}}_p}) \right|^{q-1}}{\kappa_q(v)^{q-1}},$$

which ends the proof by definition of the balanced value function $u^{\pi}_{\mathcal{U}}$. □

C Occupation Matrix

C.1 Proof of Lemma 4.4

Lemma. *Let $b, k \in \mathbb{R}^{\mathcal{S}}$ and $P_0, P_1 \in (\Delta_{\mathcal{S}})^{\mathcal{S}}$ two transition matrices. If $P_1 = P_0 - bk^\top$, i.e., P_1 is a rank-one perturbation of P_0, then their occupation matrices $D_i := (I - \gamma P_i)^{-1}, i = 0, 1$ are related through:*

$$D_1 = D_0 - \gamma \frac{D_0 b k^\top D_0}{(1 + \gamma k^\top D_0 b)}.$$

Proof. By definition, $D_1 = (I_{\mathcal{S}} - \gamma P_1)^{-1}$ so it follows that:

$$\begin{aligned}
& (I_{\mathcal{S}} - \gamma P_1) D_1 = I_{\mathcal{S}} \\
\iff & I_{\mathcal{S}} + \gamma P_1 D_1 = D_1 \\
\iff & I_{\mathcal{S}} + \gamma (P_0 - bk^\top) D_1 = D_1 && \text{(By assumption, } P_1 = P_0 - bk^\top) \\
\iff & I_{\mathcal{S}} - \gamma bk^\top D_1 = (I_{\mathcal{S}} - \gamma P_0) D_1 \\
\iff & (I_{\mathcal{S}} - \gamma P_0)^{-1} (I_{\mathcal{S}} - \gamma bk^\top D_1) = D_1 && \text{(Multiplying both sides by } (I_{\mathcal{S}} - \gamma P_0)^{-1}) \\
\iff & D_0 (I_{\mathcal{S}} - \gamma bk^\top D_1) = D_1 && \text{(By definition, } D_0 = (I_{\mathcal{S}} - \gamma P_0)^{-1}\ ) \\
\iff & D_0 - \gamma D_0 bk^\top D_1 = D_1. && (7)
\end{aligned}$$

Now, multiplying both sides by k and noticing that $k^\top D_0 b$ is a scalar we get

$$\begin{aligned} & k^\top D_0 - \gamma k^\top D_0 b k^\top D_1 = k^\top D_1 \\ \iff & k^\top D_0 = (1+\gamma k^\top D_0 b) k^\top D_1 \\ \iff & k^\top D_1 = \frac{k^\top D_0}{(1+\gamma k^\top D_0 b)}. \end{aligned} \tag{8}$$

Combining Eqs. (7) and (8) thus yields:

$$D_1 = D_0 - \gamma \frac{D_0 b k^\top D_0}{(1+\gamma k^\top D_0 b)},$$

which concludes the proof. □

C.2 Proof of Theorem 4.6

Theorem. *For any rectangular ℓ_p-ball-constrained uncertainty and $\pi \in \Pi$, it holds that:*

$$d^\pi_{\mathcal{U},\mu} = d^\pi_{P_0,\mu} - \gamma \frac{\langle d^\pi_{P_0,\mu}, \beta^\pi \rangle_{\mathcal{S}}}{1+\gamma \langle d^\pi_{P_0,u^\pi_{\mathcal{U}}}, \beta^\pi \rangle_{\mathcal{S}}} d^\pi_{P_0,u^\pi_{\mathcal{U}}}.$$

Proof. From Thms. 4.2 and 4.3, it holds that:

$$P^\pi_{\mathcal{U}}(s'|s) = P^\pi_0(s'|s) - \beta^\pi_s u^\pi_{\mathcal{U}}(s'), \quad \forall s, s' \in \mathcal{S}.$$

Therefore, setting $P_1 := P^\pi_{\mathcal{U}}$, $P_0 := P^\pi_0$, $b := \beta^\pi$ and $k := u^\pi_{\mathcal{U}}$, we can apply Lemma 4.4 and relate the corresponding occupation matrices. Additionally multiplying both sides of the relation by $\mu^\top \in \mathbb{R}^{1\times\mathcal{S}}$ yields the desired result. □

D Robust Q-value

D.1 Basic Properties

In the literature, robust Q-values are defined in various ways that turn out to be conflicting for s but non-(s,a) rectangular uncertainty sets. In this section, we propose to define the robust Q-value solely based on the worst model. Define the robust Q-value, the robust value function, and the robust occupation respectively as:

$$Q^\pi_{\mathcal{U}} := Q^\pi_{(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})}, \quad d^\pi_{\mathcal{U}} := d^\pi_{(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})}, \quad v^\pi_{\mathcal{U}} := v^\pi_{(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})}.$$

For s-rectangular uncertainty sets (in particular, for (s,a)-rectangular), the above definition of robust value function coincides with the common one, i.e., $v^\pi_{(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})} = \min_{(P,R)\in\mathcal{U}} v^\pi_{(P,R)}$ [30]. If the uncertainty set is additionally (s,a)-rectangular (as in [28] or [3, 12]), the above definition of robust Q-value also coincides with the common one because then,

$$Q^\pi_{\mathcal{U}^{\mathsf{sa}}}(s,a) = \min_{(P,R)\in\mathcal{U}^{\mathsf{sa}}} \left(R(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a) v^\pi_{\mathcal{U}^{\mathsf{sa}}}(s') \right), \quad \forall (s,a) \in \mathcal{X}.$$

Getting back to our own definition, robust Q-value and value functions are related through:

$$\begin{aligned} v^\pi_{\mathcal{U}}(s) &= \langle \pi_s, Q^\pi_{\mathcal{U}}(s,\cdot) \rangle_{\mathcal{A}} \\ Q^\pi_{\mathcal{U}}(s,a) &= R^\pi_{\mathcal{U}}(s,a) + \gamma \sum_{s'\in\mathcal{S}} \pi_s(a) P^\pi_{\mathcal{U}}(s'|s,a) v^\pi_{\mathcal{U}}(s'), \end{aligned}$$

as both quantities are defined based on worst kernel and reward, i.e., $Q^\pi_{\mathcal{U}} := Q^\pi_{(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})}$ and $v^\pi_{\mathcal{U}} := v^\pi_{(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}})}$.

Given an optimal robust policy $\pi^*_{\mathcal{U}}$, we further use $P^*_{\mathcal{U}}, R^*_{\mathcal{U}}, v^*_{\mathcal{U}}, Q^*_{\mathcal{U}}, d^*_{\mathcal{U}}$ as a shorthand for $P^{\pi^*_{\mathcal{U}}}_{\mathcal{U}}, R^{\pi^*_{\mathcal{U}}}_{\mathcal{U}}, v^{\pi^*_{\mathcal{U}}}_{\mathcal{U}}, Q^{\pi^*_{\mathcal{U}}}_{\mathcal{U}}, d^{\pi^*_{\mathcal{U}}}_{\mathcal{U}}$ respectively. For (s,a)-rectangular uncertainty set $\mathcal{U}^{\mathsf{sa}}$, the optimal value function is the best optimal Q-value, that is

$$v^*_{\mathcal{U}^{\mathsf{sa}}}(s) = \max_{a\in\mathcal{A}} Q^*_{\mathcal{U}^{\mathsf{sa}}}(s,a), \quad \forall s \in \mathcal{S}.$$

because an optimal policy deterministically takes the action with the highest Q-value [18, 10]. This does no longer hold for s-rectangular or coupled uncertainty sets, as there, an optimal policy may be stochastic [30]. Still, based on Thms. 4.2 and 4.3, we get the Bellman recursion below.

Proposition D.1. *Let an ℓ_p-ball constrained uncertainty set. Then, for all $(s,a) \in \mathcal{X}$ and $\pi \in \Pi$, the robust Q-value satisfies the following recursion in the (s,a) and s-rectangular case respectively:*

$$\begin{aligned} Q^{\pi}_{\mathcal{U}^{sa}_p}(s,a) =& T^{\pi}_{(P_0,R_0)} Q^{\pi}_{\mathcal{U}^{sa}_p}(s,a) - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v^{\pi}_{\mathcal{U}^{sa}_p}), \\ Q^{\pi}_{\mathcal{U}^{s}_p}(s,a) =& T^{\pi}_{(P_0,R_0)} Q^{\pi}_{\mathcal{U}^{s}_p}(s,a) - \left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1} \left(\alpha_s + \gamma\beta_s\kappa_q(v^{\pi}_{\mathcal{U}^{s}_p})\right). \end{aligned}$$

Proof. We give proof for the (s,a)-rectangular case only. The s-rectangular case follows the exact same lines except that it uses Thm. 4.3 instead of Thm. 4.2. We have:

$$\begin{aligned} Q^{\pi}_{\mathcal{U}}(s,a) &= Q^{\pi}_{(P^{\pi}_{\mathcal{U}},R^{\pi}_{\mathcal{U}})}(s,a) && \text{(By definition)} \\ &= R^{\pi}_{\mathcal{U}}(s,a) + \sum_{s'\in\mathcal{S}} P^{\pi}_{\mathcal{U}}(s'|s,a) v^{\pi}_{\mathcal{U}^{sa}_p}(s') \\ &= R_0(s,a) - \alpha_{sa} + \gamma \sum_{s'\in\mathcal{S}} \left(P_0(s'|s,a) - \beta_{sa} u^{\pi}_{\mathcal{U}^{sa}_p}(s')\right) v^{\pi}_{\mathcal{U}^{sa}_p}(s') && \text{(By Thm. 4.2)} \\ &= R_0(s,a) - \alpha_{sa} + \gamma \sum_{s'\in\mathcal{S}} P_0(s'|s,a) v^{\pi}_{\mathcal{U}^{sa}_p}(s') - \gamma\beta_{sa}\kappa_q(v^{\pi}_{\mathcal{U}^{sa}_p}) && \text{(2d statement of Prop. 4.1)} \\ &= R_0(s,a) + \gamma \sum_{s',a'} P_0(s'|s,a)\pi_{s'}(a') Q^{\pi}_{\mathcal{U}^{sa}_p}(s',a') - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v^{\pi}_{\mathcal{U}^{sa}_p}) \\ &= T^{\pi}_{(P_0,R_0)} Q^{\pi}_{\mathcal{U}^{sa}_p}(s,a) - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v^{\pi}_{\mathcal{U}^{sa}_p}). \end{aligned}$$

□

The above recursion applies the standard Bellman operator on robust Q-values. We can similarly apply it on the robust value function (itself can be computed efficiently based on [3, 12]).

Corollary D.2. *Let an ℓ_p-ball constrained uncertainty set. Then, for all $(s,a) \in \mathcal{X}$ and $\pi \in \Pi$, the robust Q-value satisfies the following recursion in the (s,a) and s-rectangular case respectively:*

$$\begin{aligned} Q^{\pi}_{\mathcal{U}^{sa}_p}(s,a) &= R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v^{\pi}_{\mathcal{U}^{sa}_p}(s') - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v^{\pi}_{\mathcal{U}^{s}_p}), \\ Q^{\pi}_{\mathcal{U}^{s}_p}(s,a) &= R_0(s,a) + \gamma \sum_{s'} P_0(s'|s,a) v^{\pi}_{\mathcal{U}^{sa}_p}(s') - \left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1} \left(\alpha_s + \gamma\beta_s\kappa_q(v^{\pi}_{\mathcal{U}^{s}_p})\right). \end{aligned}$$

D.2 Evaluation

Based on the Bellman recursion above, we now derive robust Q-learning equations to learn a robust Q-value. Precisely, we investigate if the linear operator below is contracting and can be evaluated efficiently:

$$(\mathcal{L}^{\pi}_{\mathcal{U}} Q)(s,a) := R^{\pi}_{\mathcal{U},v}(s,a) + \gamma \sum_{(s',a')\in\mathcal{X}} P^{\pi}_{\mathcal{U},v}(s'|s,a)\pi_{s'}(a') Q(s',a'), \quad \forall Q \in \mathbb{R}^{\mathcal{X}}, \tag{9}$$

where $(P^{\pi}_{\mathcal{U},v}, R^{\pi}_{\mathcal{U},v}) \in \arg\min_{(P,R)\in\mathcal{U}} T^{\pi}_{(P,R)} v$ and $v(s) = \langle \pi_s, Q(s,\cdot)\rangle_{\mathcal{A}}, \quad \forall s \in \mathcal{S}$.

Proposition D.3. *Consider an ℓ_p-ball constrained uncertainty set. Then, for all $Q \in \mathbb{R}^{\mathcal{X}}$ and $\pi \in \Pi$, the operator $\mathcal{L}^{\pi}$ can be evaluated as:*

$$\begin{aligned} (\mathcal{L}^{\pi}_{\mathcal{U}^{sa}_p} Q)(s,a) &= T^{\pi}_{(P_0,R_0)} Q(s,a) - \alpha_{sa} - \gamma\beta_{sa}\kappa_q(v), \\ (\mathcal{L}^{\pi}_{\mathcal{U}^{s}_p} Q)(s,a) &= T^{\pi}_{(P_0,R_0)} Q(s,a) - \left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1} \left(\alpha_s + \gamma\beta_s\kappa_q(v)\right), \end{aligned}$$

where for all $Q \in \mathbb{R}^{\mathcal{X}}$, its corresponding value is $v(s) := \langle \pi_s, Q(s,\cdot)\rangle, \quad \forall s \in \mathcal{S}$.

Proof. We give proof for the (s,a)-rectangular case only. The s-rectangular case follows the exact same lines except that we take the worst model for s-rectangular balls. By definition,

$$
\begin{aligned}
(\mathcal{L}^{\pi}_{\mathcal{U}^{\mathsf{sa}}_p} Q)(s,a) &= \min_{(P,R)\in\mathcal{U}^{\mathsf{sa}}_p} \left\{ R(s,a) + \gamma \sum_{(s',a')\in\mathcal{X}} P(s'|s,a)\pi_{s'}(a')Q(s',a') \right\} \\
&= \min_{R\in\mathcal{R}^{\mathsf{sa}}_p} R(s,a) + \gamma \min_{P\in\mathcal{P}^{\mathsf{sa}}_p} \left\{ \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') \right\} \\
&= R_0(s,a) - \alpha_{s,a} + \gamma \sum_{s'\in\mathcal{S}} P_0(s'|s,a)v(s') - \beta_{s,a}\kappa_q(v) && \text{(By [12])} \\
&= R_0(s,a) - \alpha_{s,a} + \gamma \sum_{(s',a')\in\mathcal{X}} P_0(s'|s,a)\pi_{s'}(a')Q(s',a') - \beta_{s,a}\kappa_q(v) \\
&= T^{\pi}_{(P_0,R_0)} Q(s,a) - \alpha_{s,a} - \beta_{s,a}\kappa_q(v).
\end{aligned}
$$

□

D.3 Convergence

In the rest of this section, we focus on ℓ-ball constrained uncertainty sets of the form $\mathcal{U}^{\mathsf{sa}}_p$ or $\mathcal{U}^{\mathsf{s}}_p$. Let our Q-value iteration $Q_{n+1} := \mathcal{L}^{\pi}_{\mathcal{U}} Q_n$, and denote $v_n(s) = \langle \pi_s, Q_n(s,\cdot)\rangle_{\mathcal{A}}, \forall s \in \mathcal{S}, n \in \mathbb{N}$.

Proposition D.4. *For all $Q \in \mathbb{R}^{\mathcal{X}}$, denote $v(s) := \langle \pi_s, Q(s,\cdot)\rangle_{\mathcal{A}}, \forall s \in \mathcal{S}$. Then, for any policy $\pi \in \Pi$, the Q-value iteration defined according to $Q_{n+1} = \mathcal{L}^{\pi}_{\mathcal{U}} Q_n$ induces*

$$v_{n+1} := \mathcal{T}^{\pi}_{\mathcal{U}} v_n.$$

Proof. By construction, for all $s \in \mathcal{S}$ we have

$$
\begin{aligned}
v_{n+1}(s) &= \langle \pi_s, Q_{n+1}(s,\cdot)\rangle_{\mathcal{A}} \\
&= \langle \pi_s, (\mathcal{L}^{\pi}_{\mathcal{U}} Q_n)(s,\cdot)\rangle_{\mathcal{A}} \\
&= \sum_{a\in\mathcal{A}} \pi_s(a) \left[R^{\pi}_{\mathcal{U},v_n}(s,a) + \gamma \sum_{(s',a')\in\mathcal{X}} P^{\pi}_{\mathcal{U},v_n}(s'|s,a)\pi_{s'}(a')Q_n(s',a') \right] && \text{(By Eq. 9)} \\
&= \sum_{a\in\mathcal{A}} \pi_s(a) \left[R^{\pi}_{\mathcal{U},v_n}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P^{\pi}_{\mathcal{U},v_n}(s'|s,a)v_n(s') \right] && \text{(By definition of } v_n) \\
&= (\mathcal{T}^{\pi}_{\mathcal{U}} v_n)(s),
\end{aligned}
$$

where the last equality holds because $(P^{\pi}_{\mathcal{U},v_n}, R^{\pi}_{\mathcal{U},v_n}) \in \arg\min_{(P,R)\in\mathcal{U}} \mathcal{T}^{\pi}_{(P,R)} v_n$. □

As a result of the above proposition, the value iteration induced by our Q-value iteration rule converges linearly to the robust value function, i.e., $\|v_n - v^{\pi}_{\mathcal{U}}\|_\infty \le \gamma^n \|v_0\|_\infty$. Therefore, Q-value iterates converge to a fixed point. Precisely, $v_n \to_n v^{\pi}_{\mathcal{U}}$ implies that $(P^{\pi}_{\mathcal{U},v_n}, R^{\pi}_{\mathcal{U},v_n}) \to_n (P^{\pi}_{\mathcal{U}}, R^{\pi}_{\mathcal{U}})$, which in turn implies that $Q_n \to_n Q^{\pi}_{\mathcal{U}}$. The result below further characterizes the convergence rate.

Proposition D.5. *The Q-value iteration $Q_{n+1} := \mathcal{L}^{\pi}_{\mathcal{U}} Q_n$ converges linearly to $Q^{\pi}_{\mathcal{U}}$ for uncertainty set $\mathcal{U} = \mathcal{U}^{sa}_p, \mathcal{U}^{s}_p$ for every policy $\pi \in \Pi$.*

Proof.

$$
\begin{aligned}
\|Q_{n+1} - Q^{\pi}_{\mathcal{U}}\|_\infty =& \|R^{\pi}_{\mathcal{U},v_n} + \gamma P^{\pi}_{\mathcal{U},v_n} v_n - R^{\pi}_{\mathcal{U}} + \gamma P^{\pi}_{\mathcal{U}} v^{\pi}_{\mathcal{U}}\|_\infty, \\
=& \gamma \|P^{\pi}_{\mathcal{U},v_n} v_n - P^{\pi}_{\mathcal{U}} v^{\pi}_{\mathcal{U}}\|_\infty, \qquad (\text{ as } R^{\pi}_{\mathcal{U},v} = R^{\pi}_{\mathcal{U}}, \quad \forall v), \\
=& \gamma \|(P_0 - B^{\pi} u_n)v_n - (P_0 - B^{\pi} u^{\pi}_{\mathcal{U}})v^{\pi}_{\mathcal{U}}\|_\infty, \qquad (\text{ as } R^{\pi}_{\mathcal{U},v} = R^{\pi}_{\mathcal{U}}, \quad \forall v),
\end{aligned}
$$

where $B^{\pi}(s,a) = \beta_{s,a}$ for $\mathcal{U} = \mathcal{U}^{\mathsf{sa}}_p$ and $B^{\pi}(s,a) = \beta_s \left(\frac{\pi_s(a)}{\|\pi_s\|_q}\right)^{q-1}$ for $\mathcal{U} = \mathcal{U}^{\mathsf{sa}}_p$. The equality comes from the worst kernel characterization. This implies

$$
\begin{aligned}
\|Q_{n+1} - Q^\pi_{\mathcal{U}}\|_\infty \leq & \gamma\|P_0(v_n - v^\pi_{\mathcal{U}})\|_\infty + \gamma\|B^\pi(u_n)^\top v_n - B^\pi(u^\pi_{\mathcal{U}})^\top v^\pi_{\mathcal{U}}\|_\infty, \qquad , \\
\leq & \gamma^{n+1}\|v_0 - v^\pi_{\mathcal{U}}\|_\infty + \gamma\|(u_n)^\top v_n - (u^\pi_{\mathcal{U}})^\top v^\pi_{\mathcal{U}}\|, \qquad \text{(as } B^\pi(s,a) \leq 1\text{)}, \\
\leq & \gamma^{n+1}\|v_0 - v^\pi_{\mathcal{U}}\|_\infty + \gamma\|(u_n)^\top (v_n - v^\pi_{\mathcal{U}})\| + \gamma\|(u_n - u^\pi_{\mathcal{U}})^\top v^\pi_{\mathcal{U}}\|, \\
\leq & \gamma^{n+1}\|v_0 - v^\pi_{\mathcal{U}}\|_\infty + \gamma^{n+1}S\|v_0 - v^\pi_{\mathcal{U}}\|_\infty + \gamma\frac{\|\mathcal{R}\|_\infty}{1-\gamma}\|u_n - u^\pi_{\mathcal{U}}\|_\infty.
\end{aligned}
$$

Here, $u_n, u^\pi_{\mathcal{U}}$ is the balanced normalized vector associated with vector v_n and $v^\pi_{\mathcal{U}}$ respectively. Recall, the p-balanced normalized vector u associated with vector v is given by

$$
u(s) := \frac{\text{sign}(v(s) - \omega_q(v)\|v(s) - \omega_q(v)\|^{q-1}}{\kappa_q(v)^{q-1}}, \tag{10}
$$

where $\kappa_p(v) = \min_w \|v - w\mathbf{1}\|_p$ and $\omega_p(v) =_w \|v - w\mathbf{1}\|_p$. It is easy to see that ω_p, κ are Lipschitz function in v. Hence, $\|u_n - u^\pi_{\mathcal{U}}\|_\infty \leq CPol(\|v_n - v^\pi_{\mathcal{U}}\|_\infty)f(\kappa(v^\pi_{\mathcal{U}}), S, A)$, where Pol , f is some polynomial and some function. This implies,

$$
\|Q_{n+1} - Q^\pi_{\mathcal{U}}\|_\infty \leq \gamma^{n+1} f(\kappa(v^\pi_{\mathcal{U}}), \|v_0 - v^\pi_{\mathcal{U}}\|_\infty, S, A).
$$

This concludes our proof. □

E Robust Policy Gradient

Theorem (RPG). *For any rectangular ℓ_p-ball-constrained uncertainty, the robust policy gradient is given by:*

$$
\partial_\pi \rho^\pi_{\mathcal{U}} = \sum_{(s,a)\in\mathcal{X}} \left(d^\pi_{P_0,\mu}(s) - c^\pi(s)\right) Q^\pi_{\mathcal{U}}(s,a)\nabla\pi_s(a), \tag{11}
$$

where

$$
c^\pi(s) := \frac{\gamma\langle d^\pi_{P_0,\mu}, \beta^\pi\rangle_{\mathcal{S}}}{1 + \gamma\langle d^\pi_{P_0,u^\pi_{\mathcal{U}}}, \beta^\pi\rangle_{\mathcal{S}}} d^\pi_{P_0,u^\pi_{\mathcal{U}}}(s), \quad \forall s \in \mathcal{S}\,.
$$

Proof. The proof directly follows from plugging the robust occupation measure of Thm. 4.6 and the robust Q-value into standard policy-gradient theorem [23]. □

F Complexity Analysis

In this section, we study the iteration complexity to compute robust policy gradient different uncertainty set.

Convex non-rectangular Uncertainty set. Robust policy improvement are strongly NP Hard for non-rectangular uncertainty set, even if it is convex [30]. The policy gradient method finds global optimal given oracle access to policy gradient in polynomial time [27]. This implies computation of policy gradient must be of NP Hard.

Non-robust MDPs. For non-robust case, computation of Q-value and occupation is $O(S^2A\log(\epsilon^{-1}))$ each, which are most costly computations. Computing the product of d^π, Q^π and $\nabla\pi$ as in policy gradient is $O(SA)$ operation, which insignificant. Hence, the total cost for computing policy gradient is the same as cost of Q-value. More precisely, lets approximate Q-value with Q and occupation with d, with $\frac{\epsilon}{SA}$ tolerance, that is $\|Q - Q^\pi\|_\infty, \|d - d^\pi\|_\infty \leq \frac{\epsilon}{SA}$. This takes $O(S^2A\log(SA\epsilon^{-1}))$ each. Now then, we have

$$
\sum_{s,a} d'(s)Q'(s,a)\nabla\pi_s(a) = \sum_{s,a}(Q^\pi(s,a) + \epsilon_1(s,a))(d^\pi(s,a) + \epsilon_2(s,a))\nabla\pi_s(a)
$$

where $\epsilon_1(s,a) = Q(s,a) - Q^\pi(s,a)$ and $\epsilon_2(s,a) = d(s,a) - d^\pi(s,a)$. We know, let B be the bound on $\|Q^\pi\|_\infty, \|d^\pi\|_\infty \leq B$. So now we have,

$$\sum_{s,a} d'(s)Q'(s,a)\nabla\pi_s(a) = \sum_{s,a}(Q^\pi(s,a)+\epsilon_1(s,a))(d^\pi(s,a)+\epsilon_2(s,a))\nabla\pi_s(a)$$
$$= \sum_{s,a} Q^\pi(s,a)d^\pi(s,a)\nabla\pi_s(a) + O(\epsilon).$$

So the exact complexity of policy gradient for non-robust MDP is $O(S^2A\log(SA\epsilon^{-1})$.

F.1 Helper results for Robust MDPs

Computing variance and mean functions. Computing $\kappa_p(v)$ and $\omega_p(v)$ to ϵ tolerance requires $O(S\log(S\epsilon^{-1}))$ and $O(S\log(\epsilon^{-1}))$ respectively, via binary search [12].

Computing occupation measure. Let $k \in \mathbb{R}^{\mathcal{S}}$ be any vector. From definition, we have

$$d^\pi_{P,k} = \sum_{n=0}^{\infty}\gamma^n k^\top (P^\pi)^n.$$

We have,

$$\|d^\pi_{P,k} - \sum_{n=0}^{N-1}\gamma^n k^\top (P^\pi)^n\| = \|\sum_{n=N}^{\infty}\gamma^n k^\top (P^\pi)^n\| \leq \|k\|\sum_{n=N}^{\infty}\|\gamma P^\pi\|^n.$$

Since, $\|P^\pi\| \leq 1$ as it is a stochastic matrix, then

$$\|d^\pi_{P,k} - \sum_{n=0}^{N-1} k^\top\gamma^n (P^\pi)^n\| \leq \frac{\|k\|\gamma^N\|P^\pi\|^N}{1-\gamma\|P^\pi\|} \leq \frac{\|k\|\gamma^N}{1-\gamma}.$$

This implies $\sum_{n=0}^{N-1}\gamma^n k^\top (P^\pi)^n$ is $O(\gamma^N)$ approximation of $d^\pi_{P,k}$. Now, take $u_0 = k$ and

$$u_{n+1} := \gamma(u_n)^\top P,$$

then $\sum_{n=0}^{N-1}\gamma^n k^\top (P^\pi)^n = \sum_{n=0}^{N-1} u_n$. And each iteration take $O(S^2)$ iterations, leading to total cost $O(S^2N)$ for N iterations. Computing P^π from P is $O(S^2A)$. We conclude computing the occupation measure has complexity of $O(S^2A + S^2\log(\epsilon^{-1}))$.

Lemma F.1. *We can approximate $d^\pi_{P,k}$ with $\sum_{n=0}^{N-1}\gamma^n (k')^\top (P^\pi)^n$ with complexity $O(S^2A + S^2\log(\epsilon^{-1}))$ with tolerance:*

$$\|d^\pi_{P,k} - \sum_{n=0}^{N-1}\gamma^n (k')^\top (P^\pi)^n\| \leq O(\frac{\|k\|\gamma^N + \|k-k'\|}{1-\gamma})$$

Proof.

$$\begin{aligned}
\|d^\pi_{P,k} - \sum_{n=0}^{N-1}\gamma^n (k')^\top (P^\pi)^n\| &\leq \|d^\pi_{P,k} - \sum_{n=0}^{N-1}\gamma^n (k)^\top (P^\pi)^n\| + \|\sum_{n=0}^{N-1}\gamma^n (k')^\top (P^\pi)^n - \sum_{n=0}^{N-1}\gamma^n (k)^\top (P^\pi)^n\| \\
&\leq O(\frac{\|k\|\gamma^N}{1-\gamma}) + \|\sum_{n=0}^{N-1}\gamma^n (k)^\top (P^\pi)^n - \sum_{n=0}^{N-1}\gamma^n (k')^\top (P^\pi)^n\| \\
&\leq O(\frac{\|k\|\gamma^N}{1-\gamma}) + \|k-k'\|\|\sum_{n=0}^{N-1}\gamma^n (P^\pi)^n - \sum_{n=0}^{N-1}\gamma^n (P^\pi)^n\| \\
&\leq O(\frac{\|k\|\gamma^N}{1-\gamma}) + O(\frac{\|k-k'\|}{1-\gamma}).
\end{aligned}$$

□

Computing Q-value given value function. Let v be ϵ_1 approximation of robust value function $v^\pi_{\mathcal{U}}$, that is

$$\|v - v^\pi_{\mathcal{U}}\|_\infty \leq \epsilon_1.$$

We want to compute Q-value using the relation:

$$\begin{aligned}Q^\pi_{\mathcal{U}}(s,a) &= R^\pi_{\mathcal{U}}(s,a) + \sum_{s,a}\pi_s(a)P^\pi_{\mathcal{U}}(s'|s,a)v^\pi_{\mathcal{U}}(s')\\ &= R_0(s,a) + \gamma\sum_{s'}P_0(s'|s,a)v^\pi_{\mathcal{U}}(s') - \Omega_{\mathcal{U}}(v^\pi_{\mathcal{U}},\pi).\end{aligned}$$

where $\Omega_{\mathcal{U}^{\mathtt{s}}_p}(v^\pi_{\mathcal{U}^{\mathtt{s}}_p},\pi) = \frac{\pi_s(a)^{q-1}}{\|\pi_s\|_q^{q-1}}\,\big(\,\alpha_s + \gamma\beta_s\kappa_q(v^\pi_{\mathcal{U}^{\mathtt{s}}_p})\,\big)$ and $\Omega_{\mathcal{U}^{\mathtt{sa}}_p}(v^\pi_{\mathcal{U}^{\mathtt{sa}}_p},\pi) = \alpha_{sa} + \gamma\beta_{sa}\kappa_q(v^\pi_{\mathcal{U}^{\mathtt{sa}}_p})$. Let Q be approximated from v as

$$Q(s,a) = R_0(s,a) + \gamma\sum_{s'}P_0(s'|s,a)v(s') - \Omega_{\mathcal{U}}(v,\pi).$$

So we have,

$$\begin{aligned}\|Q^\pi_{\mathcal{U}}(s,a) - Q(s,a)\|_\infty &= \gamma\|\sum_{s'}P_0(s'|s,a)(v(s') - v^\pi_{\mathcal{U}})\| + \|\Omega_{\mathcal{U}}(v,\pi) - \Omega_{\mathcal{U}}(v^\pi_{\mathcal{U}},\pi)\|\\ &\leq \gamma\epsilon_1 + \|\Omega_{\mathcal{U}}(v,\pi) - \Omega_{\mathcal{U}}(v^\pi_{\mathcal{U}},\pi)\|\\ &\leq \gamma\epsilon_1 + \|\beta\|_\infty\|\kappa_q(v) - \kappa_q(v^\pi_{\mathcal{U}})\|\\ &\leq \gamma\epsilon_1 + \|\beta\|_\infty S^{\frac{1}{q}}\epsilon_1, \qquad \text{(using lemma F.2)}\\ &= O(S^{\frac{1}{q}}\epsilon_1).\end{aligned}$$

This implies, $\|Q - Q^\pi_{\mathcal{U}}\|_\infty \leq O(S^{\frac{1}{q}}\epsilon_1)$.

Lemma F.2. *κ_p is Lipschitz function, precisely*

$$\|\kappa_p(v_1) - \kappa_p(v_2)\| \leq S^{\frac{1}{p}}\|v_1 - v_2\|_\infty \leq S^{\frac{1}{p}}\|v_1 - v_2\|_\infty.$$

Proof. Let $w_i \in \arg\min_w\|v_i - w\mathbf{1}\|_p$, then we have

$$\begin{aligned}\|\kappa_p(v_1) - \kappa_p(v_2)\| &= \kappa_p(v_1) - \kappa_p(v_2), \qquad \text{(WLOG, assuming } \kappa_p(v_1) \geq \kappa_p(v_2))\\ &= \min_w\|v_1 - w\mathbf{1}\|_p - \|v_2 - w_2\mathbf{1}\|_p, \qquad \text{(From definition)}\\ &\leq \|v_1 - w_2\mathbf{1}\|_p - \|v_2 - w_2\mathbf{1}\|_p, \qquad \text{(From definition of min operator)}\\ &\leq \|(v_1 - w_2\mathbf{1}) - (v_2 - w_2\mathbf{1})\|_p, \qquad \text{(Reverse triangle inequality)}\\ &= \|v_1 - v_2\|_p\\ &= [\sum_s(v_1(s) - v_2(s))^p]^{\frac{1}{p}} \leq S^{\frac{1}{p}}\|v_1 - v_2\|_\infty.\end{aligned}$$

□

Lemma F.3. *$Q^\pi_{\mathcal{U}^{sa}_p}$ can be approximated to ϵ tolerance with the same complexity as complexity of computing $v^\pi_{\mathcal{U}^{sa}_p}$ to $S^{-\frac{1}{q}}\epsilon$.*

Proof. Compute value function with $S^{-\frac{1}{q}}\epsilon$ tolerance. The rest of operations are insignificant. Rest follows from the above. □

F.2 Computing the Policy gradient.

Let $O^{\mathtt{sa}}_p(\epsilon)$ be the complexity to compute robust value function $v^\pi_{\mathcal{U}^{\mathtt{sa}}_p}$, upto ϵ tolerance, see [12] for details. Calculate Q-value up to ϵ_1 tolerance which requires $O^{\mathtt{sa}}_p(S^{-\frac{1}{q}}\epsilon_1)$ from lemma F.3. Let d_1 and d_2 be ϵ_2 approximation of $d^\pi_{P_0,\mu}$ and $d^\pi_{P_0,k}$ respectively, which is insignificant compared to $O^{\mathtt{sa}}_p(S^{-\frac{1}{q}}\epsilon_2)$. Now let's approximate the gradient with $d_1, d_2, Q, \nabla\pi$ as in Theorem 5.1, which has a complexity of $O(SA)$. Since the uncertainty set $\mathcal{U}$ is compact, all the quantities are bounded. And there are $O(SA)$ operations in the Theorem 5.1, so taking $\epsilon_1, \epsilon_2 = O(\frac{\epsilon}{SA})$, we will get the $O(\epsilon)$ of the gradient. Hence, the total complexity is $O^{\mathtt{sa}}_p(S^{-\frac{1}{q}-1}A^{-1}\epsilon)$ which is $\tilde{O}(S^2A\log(\epsilon^{-1}))$, by hiding log factors, see [12]. A similar analysis follows for the s-rectangular case.

G Generalization to arbitrary norms

Here we focus on the generalization of our result to a general norm from the existing ℓ_p norm. We do it case by case.

G.1 sa-rectangular robust MDPs.

Lets consider sa-rectangular uncertainty set $\mathcal{U} = \mathcal{U}^{\mathtt{sa}}_{\|\cdot\|}$ constrained by $\|\cdot\|$ norm. Precisely, defined as

$$\mathcal{U}^{\mathtt{sa}}_{\|\cdot\|} = (P_0 + \mathcal{P}) \times (R_0 + \mathcal{R}), \qquad \text{where} \qquad (\mathcal{P}, \mathcal{R}) = (\times_{s,a}\mathcal{P}_{sa}, \times_{s,a}\mathcal{P}_{sa}),$$
$$\mathcal{R}_{(s,a)} = \{r \in \mathbb{R} \mid \|r\| \leq \alpha_{s,a}\}, \quad \text{and} \quad \mathcal{P}_{(s,a)} = \left\{p \in \mathbb{R}^{\mathcal{S}} \mid \langle p, \mathbf{1}\rangle_{\mathcal{S}} = 0, \|p\| \leq \beta_{s,a}\right\}.$$

The robust Bellman operator $\mathcal{T}^\pi_{\mathcal{U}}$ can be evaluated as

$$(\mathcal{T}^\pi_{\mathcal{U}} v)(s) = \sum_a \pi_s(a) \Big[R(s,a) - \gamma\beta_{s,a}\kappa_{\|\cdot\|}(v) + \gamma \sum_{s'} P(s'|s,a)v(s') \Big],$$

where variance function is defined as

$$\kappa_{\|\cdot\|}(v) := \min_{\langle u, \mathbf{1}\rangle_{\mathcal{S}}=0, \|u\|\leq 1} \langle u, v^\pi_{\mathcal{U}}\rangle.$$

This can be used to compute the robust value function. Then the worst values can found using robust Bellman operator $\mathcal{T}^\pi_{\mathcal{U}}$ and robust value function $v^\pi_{\mathcal{U}}$ as

$$(P^\pi_{\mathcal{U}}, R^\pi_{\mathcal{U}}) \in \arg\min_{(P,R)\in\mathcal{U}} \mathcal{T}^\pi_{(P,R)} v^\pi_{\mathcal{U}}, \qquad [30].$$

It is easy to see that the worst values are given as

$$R^\pi_{\mathcal{U}}(s,a) = R_0(s,a) - \alpha_{s,a} \quad \text{and} \quad P^\pi_{\mathcal{U}}(\cdot|s,a) = P^\pi_0(\cdot|s,a) - \beta_{s,a}u^\pi_{\mathcal{U}},$$

where normalized-balanced value function $u^\pi_{\mathcal{U}}$ is a solution to

$$\min_{\langle u, \mathbf{1}\rangle_{\mathcal{S}}=0, \|u\|\leq 1} \langle u, v^\pi_{\mathcal{U}}\rangle.$$

Observe that the worst kernel is still a rank-one perturbation of the nominal kernel. Hence, the robust occupation measure can be obtained using the Lemma 4.4 as

$$d^\pi_{\mathcal{U},\mu} = d^\pi_{P_0,\mu} - \gamma \frac{\langle d^\pi_{P_0,\mu}, \beta^\pi\rangle_{\mathcal{S}}}{1 + \gamma\langle d^\pi_{P_0,u^\pi_{\mathcal{U}}}, \beta^\pi\rangle_{\mathcal{S}}} d^\pi_{P_0,u^\pi_{\mathcal{U}}}, \tag{12}$$

where $\beta^\pi(s) = \sum_a \pi_s(a)\beta_{s,a}$. The last ingredient to compute RPG is robust Q-value which can be computed using robust value function and worst values. However, it can be computed directly using the following iterates:

$$\begin{aligned} Q_{n+1}(s) &= \min_{(P,R)\in\mathcal{U}} \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s') \Big] \\ &= R(s,a) - \alpha_{s,a} - \gamma\beta_{s,a}\kappa_{\|\cdot\|}(v) + \gamma \sum_{s'} P(s'|s,a)v(s'), \end{aligned}$$

as Q_n converges to robust Q-value $Q^\pi_{\mathcal{U}}$ linearly.

The proofs of the above claims are easy or similar to the ℓ_p counterparts. Finally, computation of the variance function $\kappa_{\|\cdot\|}$ and normalized-value function $u^\pi_{\mathcal{U}}$ can be done via numerical convex optimization methods for general norms. However for the ℓ_p case, they can be obtained in concrete forms, hence we choose it for the main presentation.

G.2 s-rectangular Case

Generalization of our methods to s-rectangular balls of a general norm is not straightforward and may not be possible for all kinds of norms. The crucial property of ℓ_p norm that we exploited to prove rank-one perturbation is 'decoupling', that is, for $x \in \mathbf{R}^{\mathcal{A}\times\mathcal{S}}$,

$$\|x\|^y_p = \sum_{a\in\mathcal{A}} \|x_a\|^z_w,$$

for some w, y, z. This holds for the ℓ_p norm with $w = y = z = p$. We leave the further analysis of this setting for future work.

G.3 Generalization to non-norms

Further, generalization of our results to distance such as KL, can be tricky. The ability of our methods to compute RPG (particularly robust occupation measure) crucially relies on the rank-one perturbation result, which might not be the case for distance measures such as KL. We leave this analysis for future work.

H Experiments

Parameters. All the nominal transition kernels and reward functions are generated randomly. The number of states and the number of actions are varied. Discount factor $\gamma = 0.9$, reward noise radius $\alpha_{s,a}, \alpha_s = 0.1$, transition noise kernel $\beta_{s,a}, \beta_s = \frac{0.01}{SA}$.

Hardware Experiments are done on the machine with the following configuration: Intel(R) Core(TM) i7-6700 CPU @3.40GHZ, size:3598MHz, capacity 4GHz, width 64 bits, memory size 64 GiB.

Software and codes All the experiments were done in Python using numpy, matplotlib. All codes and results will be made public on GitHub after the publication to preserve anonymity.

Procedure and Results. All the experiments were repeated 100 times, except for Linear Programming (LP) cases as LP methods were very time-consuming. In LP methods, experiments were repeated 5 times except for the case ($S = 500, A = 100$) which was done only once. As this case was prohibitively expensive. Standard deviation in all cases was less than 10%, and typically $1 - 2\%$. This conveniently illustrates the superiority of our methods over LP methods.

Observations

- **Scalability of our methods.** Note that our methods scale very well with large state-action space. It takes a (small) constant times the time required by non-robust MDPs. On the other hand, LP methods explode. Both observations confirm the theoretical time complexity.
- **`sa`-case vs `s` case in LP methods.** We see `s`-case outperforms `sa`-case for small state-action spaces via LP methods. This is opposite to the theoretical time complexity of `s`-case which expensive than `sa`-case. We believe this is due to the internal implementation issues. Note that computing the robust value function is the most expensive step which requires evaluation of the robust Bellman operator. In `sa` case, one evaluation requires solving SA LP programs with S variables each, while for `s`-case, it is S LP programs with SA variables each. To solve LP, scipy.linprog is used, we believe it does some parallelization for large LPs. Hence, we observe less cost for `sa`-case. However, we observe that the cost of `s`-case increases much faster than `s`-case, and eventually under-performing than `sa`-case.

H.1 RPG by LP

We compute RPG using LP in the following steps:

1. **(Robust Value Iteration)** Approximately compute the robust value function $v^\pi_\mathcal{U}$ using the iterates $v_{n+1} := \mathcal{T}^\pi_\mathcal{U} v_n$. Evaluation of robust Bellman operator $\mathcal{T}^\pi_\mathcal{U}$ is done via LPs as described below. This is the most expensive step as it requires evaluating robust Bellman operators $O(\log(\epsilon^{-1}))$ times, and each evaluation requires many LPs.
2. **(Adversarial Values)** Compute the worst values $(P^\pi_\mathcal{U}, R^\pi_\mathcal{U})$ using the robust value function from the following relation:
$$(P^\pi_\mathcal{U}, R^\pi_\mathcal{U}) \in \arg\min_{(P,R)\in\mathcal{U}} \mathcal{T}^\pi_\mathcal{U} v^\pi_\mathcal{U}.$$
This is also solved by LP.
3. **(Policy Gradient Theorem)** We now compute the RPG using policy gradient Theorem [23] w.r.t. the adversarial values computed above, as
$$\partial\rho^\pi_\mathcal{U} = \sum_{s,a} d^\pi_{P^\pi_\mathcal{U}}(s) Q^\pi_{P^\pi_\mathcal{U}, R^\pi_\mathcal{U}}(s,a) \nabla\pi_s(a).$$

Observe that d_P^π can be approximated as $\sum_{n=0}^n (\gamma P^\pi)^n$ for large n enough, and $Q_{P,R}^\pi$ can be approximated by dynamic programming [22]. Notably, this step and the second step are negligible as compared to the first step.

Robust Value Iteration by LP

$\mathtt{sa}$-rectangular robust MDPs. We first consider $\mathtt{sa}$-rectangular L_1 constrained uncertainty set $\mathcal{U}_p^{\mathtt{sa}} = \mathcal{P} \times \mathcal{R}$. Robust Bellman operator is given by

$$(\mathcal{T}_{\mathcal{U}_p^{\mathtt{sa}}}^\pi v)(s) = \max_a \underbrace{\min_{p \in \mathcal{P}_{sa}, r \in \mathcal{R}_{sa}} \left[r + \gamma \sum_{s'} p(s') v(s') \right]}_{\text{LP with } S \text{ variable}}.$$

Note that the above can be solved by A LPs as uncertainty set $\mathcal{U}_p^{\mathtt{sa}} = \mathcal{P} \times \mathcal{R}$ induces linear constraint and the objective is also linear with S variables. Hence, evaluation of $\mathcal{T}_{\mathcal{U}_p^{\mathtt{sa}}}^\pi v$ requires solving SA LPs with S variable each.

s-rectangular robust MDPs. We now consider s-rectangular L_1 constrained uncertainty set $\mathcal{U}_p^{\mathtt{s}} = \mathcal{P} \times \mathcal{R}$. Robust Bellman operator is given by

$$(\mathcal{T}_{\mathcal{U}_p^{\mathtt{s}}}^\pi v)(s) = \underbrace{\min_{p \in \mathcal{P}_s, r \in \mathcal{R}_s} \sum_a \pi_s(a) \left[r(a) + \gamma \sum_{s'} p(s'|a) v(s') \right]}_{\text{LP with } SA \text{ variable}}.$$

Note that the above can be solved by one LP as uncertainty set $\mathcal{U}_p^{\mathtt{s}} = \mathcal{P} \times \mathcal{R}$ induces linear constraint and the objective is also linear with SA variables. Hence, evaluation of $\mathcal{T}_{\mathcal{U}_p^{\mathtt{s}}}^\pi v$ requires solving S LPs with SA variable each.