

---

# Non-Rectangular Robust MDPs with Normed Uncertainty Sets

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Robust policy evaluation for non-rectangular uncertainty set is generally NP-hard,  
2 even in approximation. Consequently, existing approaches suffer from either  
3 exponential iteration complexity or significant accuracy gaps. Interestingly, we  
4 identify a powerful class of  $L_p$ -bounded uncertainty sets that avoid these complexity  
5 barriers due to their structural simplicity. We further show that this class can be  
6 decomposed into infinitely many sa-rectangular  $L_p$ -bounded sets and leverage  
7 its structural properties to derive a novel dual formulation for  $L_p$  robust Markov  
8 Decision Processes (MDPs). This formulation provides key insights into the  
9 adversary's strategy and enables the development of an efficient robust policy  
10 evaluation algorithm for these  $L_p$  normed non-rectangular robust MDPs.

## 11 1 Introduction

12 Robust Markov Decision Processes (MDPs) effectively handle uncertainties in environmental pa-  
13 rameters, making them indispensable for high-stakes domains such as robotics, finance, healthcare,  
14 and autonomous driving, where failures can have catastrophic consequences [23, 12, 29, 24, 15].  
15 They also outperform standard MDPs in terms of generalization, ensuring robust performance across  
16 diverse scenarios [36, 37, 25]. Consequently, extensive research has been conducted on robust  
17 MDPs [22, 10, 35, 16, 3, 24, 15, 13, 33, 34, 32, 7, 19, 17, 38, 30, 1, 14], primarily focusing on  
18 rectangular uncertainty sets that leverage the contractive robust Bellman operator. However, practical  
19 robust MDPs often feature non-rectangular uncertainty sets, where rectangular relaxations can result  
20 in overly suboptimal solutions [20, 8, 35]. Intuitively, non-rectangular uncertainty set could be  
21 thought of as an  $n$ -dimensional sphere of unit radius, and its rectangular relaxation is the smallest  
22  $n$ -dimensional cube encapsulating the sphere. The ratio between the sphere and the encapsulating  
23 cube is exponential in the dimension ( $O(2^{-n})$ ) [27]). This suggests that the rectangular relaxation  
24 of the non-rectangular uncertainty set, contains many additional environments. Moreover, most of  
25 the additional environments would lie near the corners representing big differences from the center  
26 in many coordinates – scenarios unlikely to occur in the real world, as aptly captured by the paper  
27 titled "*Lightening doesn't strike twice, robust MDPs* [21]". These improbable, highly perturbed  
28 environments can lead to a significant gap between the solutions of non-rectangular robust MDPs and  
29 their rectangular relaxations.

30 While non-rectangular robust MDPs capture much better interdependencies across the states, they  
31 lack the existence of contractive robust Bellman operators, which makes the problem very difficult  
32 to solve with standard dynamic programming techniques [8]. This makes non-rectangular robust  
33 MDPs a crucial yet challenging area of study, with only a limited body of work existing on the topic  
34 [35, 20, 8].

35 The key challenge for non-rectangular robust MDPs is robust policy evaluation . That is, given  
36 oracle access to the robust gradient (robust policy evaluation ), the robust policy gradient method

Table 1: Related Work on Robust Policy Evaluation for Non-Rectangular Uncertainty Sets.

Method	Uncertainty Set	Iteration Complexity	Accuracy	Irreducibility Assumption 1 of [20]	NP-Harness Result of [35]
[8]	<b>Reward</b> $L_p$ Normed	$O(\log \epsilon^{-1})$	$\epsilon$	No	No
Algorithm 3.1 of [20]	General Kernel Set	$O(2^q \log \epsilon^{-1})$	$\epsilon$	No	Yes
Algorithm 3.2 of [20]	General Kernel Set	$O(\epsilon^{-2})$	$\delta_d(2\epsilon + \delta_{\mathcal{P}})$	Yes	Yes
<b>Ours</b>	Kernel $L_p$ Normed	$\log(\epsilon^{-1})$	$\epsilon$	No	<b>No</b>

The constants  $q, \delta_d, \delta_{\mathcal{P}}$  can be as large as  $O(S^2 A)$ ,  $O(2^S)$  and  $O(S\sqrt{A})$  respectively.

can efficiently achieve an  $\epsilon$ -close globally optimal policy with an iteration complexity of  $O(\epsilon^{-4})$  [32]. However, the robust policy evaluation for general convex non-rectangular uncertainty sets is strongly NP-hard, even for approximations [35]. Despite this hardness, [20] proposed two methods for non-rectangular robust policy evaluation for general convex uncertainty sets: One with exponential iteration complexity in the state-action space, and another that approximates the solution but with tolerances so large that the results are meaningless in the worst case (see Table 1). These pioneering approaches remain computationally prohibitive or overly imprecise, consistent with the NP-hardness result.

Interestingly, the NP-hardness result in [35] applies specifically to kernel uncertainty sets with certain polyhedral structures (see Appendix for details). For  $L_p$ -bounded uncertainty sets, [8] showed that robust policy evaluation can be done efficiently, though this is limited to reward uncertainty, a much simpler case compared to kernel uncertainties. This raises a critical open question: **Are there useful classes of kernel uncertainty sets that avoid this NP-Hardness barrier?**

We identify a specific class of non-rectangular uncertainty sets, bounded by an  $L_p$ -ball around a nominal kernel, and demonstrate that it effectively circumvents the NP-hardness result of [35]. Moreover, we show that robust policy evaluation for this non-rectangular  $L_p$ -bounded uncertainty set is equivalent to robust policy evaluation over an infinite collection of sa-rectangular  $L_p$ -bounded uncertainty sets. While robust policy evaluation for each sa-rectangular set is computationally tractable [19, 17], managing this infinite collection poses significant challenges. To overcome this, we leverage the property that the worst kernel for each sa-rectangular uncertainty set is a rank-one perturbation of the nominal kernel [17]. This insight enables us to express the robust policy evaluation problem (or robust return) in a novel dual form, providing a clearer understanding of the adversary’s behavior. Furthermore, this dual formulation facilitates the development of a binary search method for robust policy evaluation, achieving an iteration complexity of  $O(\log \epsilon^{-1})$  for approximating the robust return up to  $\epsilon$  tolerance.

In summary, robust MDPs are critical for handling uncertainties in high-stakes domains, yet existing methods are largely confined to rectangular uncertainty sets, limiting real-world applicability. Non-rectangular uncertainty sets, though more realistic, often face NP-hard challenges in robust policy evaluation. This work identifies a promising class of non-rectangular  $L_p$ -bounded kernel uncertainty sets, demonstrating that they circumvent existing NP-hardness results and enable efficient robust policy evaluation. By connecting robust evaluation for these sets to an infinite collection of sa-rectangular uncertainty sets and leveraging their structure, we propose a computationally efficient binary search method with logarithmic iteration complexity. This approach not only advances the understanding of non-rectangular robust MDPs but also opens the door to future investigation into broader classes of non-rectangular uncertainty sets in robust MDPs.

## 72 2 Preliminary

73 A robust Markov Decision Process (RMDP) can be described as a tuple  $(\mathcal{S}, \mathcal{A}, \gamma, \mu, R, \mathcal{U})$ , where  
 74  $\mathcal{S}$  is the state space,  $\mathcal{A}$  is the action space,  $\mu$  is an initial distribution over states  $\mathcal{S}$ ,  $\gamma$  is a discount  
 75 factor in  $[0, 1)$ ,  $R$  is a reward function mapping  $\mathcal{S} \times \mathcal{A}$  to  $\mathbb{R}$ , and  $\mathcal{U}$  set of transition kernel  $P$  that  
 76 maps  $\mathcal{S} \times \mathcal{A}$  to  $\Delta_{\mathcal{S}}$  [15, 24]. A policy  $\pi : \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$  is a decision rule that maps state space to  
 77 a probability distribution over action space. Let  $\Pi = (\Delta_{\mathcal{A}})^{\mathcal{S}}$  denote set of all possible policies.  
 78 Further,  $\pi(a|s)$ ,  $P(s'|s, a)$  denotes the probability of taking action  $a$  in state  $s$  by policy  $\pi$ , and the  
 79 probability of transition to state  $s'$  from state  $s$  under action  $a$  respectively. In addition, we denote  
 80  $P^{\pi}(s'|s) = \sum_a \pi(a|s)P(s'|s, a)$  and  $R^{\pi}(s) = \sum_a \pi(a|s)R(s, a)$  as short-hands. The return of a  
 81 policy  $\pi$ , is defined as  $J_P^{\pi} = \langle \mu, v_P^{\pi} \rangle = \langle R^{\pi}, d_P^{\pi} \rangle$  where  $v_P^{\pi} := D^{\pi} R^{\pi}$  is value function,  $d_P^{\pi} = \mu^{\top} D^{\pi}$   
 82 is occupancy measure and  $D^{\pi} = (I - \gamma P^{\pi})^{-1}$  is occupancy matrix [26]. As a shorthand, we denote  
 83 the state-action occupancy measure as  $d_P^{\pi}(s, a) = d_P^{\pi}(s)\pi(a|s)$  and the usage shall be clear from the  
 84 context. For an uncertainty set  $\mathcal{U}$ , the robust return  $J_{\mathcal{U}}^{\pi}$  for a policy  $\pi$ , and the optimal robust return  
 85  $J_{\mathcal{U}}^*$ , are defined as:

$$J_{\mathcal{U}}^{\pi} = \min_{P \in \mathcal{U}} J_P^{\pi}, \quad \text{and} \quad J_{\mathcal{U}}^* = \max_{\pi} J_{\mathcal{U}}^{\pi},$$

86 respectively. The objective is to determine an optimal robust policy  $\pi_{\mathcal{U}}^*$  that achieves the optimal  
 87 robust performance  $J_{\mathcal{U}}^*$ . Unfortunately, even robust policy evaluation (i.e., finding the worst-case  
 88 transition kernel  $P_{\mathcal{U}}^{\pi} \in \arg \min_{P \in \mathcal{U}} J_P^{\pi}$ ) is strongly NP-hard for general (non-rectangular) convex  
 89 uncertainty sets [35]. This makes solving non-rectangular robust MDPs a highly challenging problem.

90 To make the problem tractable, a common approach is to use s-rectangular uncertainty sets,  $\mathcal{U}^s =$   
 91  $\times_{s \in \mathcal{S}} \mathcal{P}_s$ , where the uncertainty is modeled independently across states [35]. These sets decompose  
 92 state-wise, capturing correlated uncertainties within each state while ignoring inter-dependencies  
 93 across states. A further simplification is the sa-rectangular uncertainty set,  $\mathcal{U}^{sa}$ , where uncertainties  
 94 are assumed to be independent across both states and actions. Formally,  $\mathcal{U}^{sa} = \times_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_{s,a}$ ,  
 95 where  $\mathcal{P}_{s,a}$  are independent component sets for each state-action pair [15, 24, 33, 34].

96 A  $L_p$ -bounded uncertainty sets,  $\mathcal{U}_p^{sa}$  and  $\mathcal{U}_p^s$ , which are centered around a nominal transition kernel  $\hat{P}$   
 97 are defined as  $\mathcal{U}_p^{sa} = \{P \mid \sum_{s'} P_{sa}(s') = 1, \|P_{sa} - \hat{P}_{sa}\|_p \leq \beta_{sa}\}$ , and  $\mathcal{U}_p^s = \{P \mid \sum_{s'} P_{sa}(s') =$   
 98  $1, \|P_s - \hat{P}_s\|_p \leq \beta_s\}$ , where radius vector  $\beta$  is assumed small enough to ensure all kernels within the  
 99 uncertainty sets are valid [13, 7, 19, 17]. Interestingly, for  $L_p$  bounded uncertainty set, adversarial  
 100 (worst) kernels is a rank one perturbation of the nominal kernel that is used later in the paper [17].

101 **Dual Formulation.** The primal formulation of an MDP is defined as:

$$\max_{v \in \mathcal{V}} \langle \mu, v \rangle, \quad \text{with its dual:} \quad \max_{d \in \mathcal{D}} \langle d, R \rangle,$$

102 where  $\mathcal{V} = \{v \mid v = R^{\pi} + \gamma P^{\pi} v, \pi \in \Pi\}$  represents the set of value functions. The dual formulation  
 103 relies on the state-action occupancy measure  $d$ , where  $d \in \mathcal{D} \subset \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$  satisfies the non-negativity  
 104 constraint ( $d \succeq 0$ ) and the flow conservation constraint:  $\sum_a d(s, a) - \gamma \sum_{s', a'} d(s', a') P(s \mid$   
 105  $s', a') = \mu(s), \quad \forall s \in \mathcal{S}$ . The feasible set  $\mathcal{D}$  forms a convex polytope [2], whereas the set of value  
 106 functions,  $\mathcal{V}$ , is a polytope that is generally non-convex [6]. This dual formulation offers several  
 107 advantages, including efficient handling of constraints and the ability to solve the problem using  
 108 linear programming techniques.

109 For robust MDPs, the geometry of robust value functions is significantly more intricate compared to  
 110 standard MDPs [31]. While the dual formulation for standard MDPs is well-established, this work is  
 111 the first to derive a dual formulation for this specific class of non-rectangular robust MDPs, providing  
 112 critical insights and laying the foundation for the development of robust policy evaluation methods.

## 113 3 Method

114 In this section, we introduce  $L_p$ -bounded non-rectangular uncertainty set, and demonstrate that  
 115 its rectangular relaxation may yield highly sub-optimal solutions. Then, we establish that this  
 116 uncertainty set avoids the NP-Hardness results established in [35]. Subsequently, we show that the  
 117 robust evaluation for this uncertainty set is equivalent to robust evaluation over an infinite collection  
 118 of sa-rectangular robust MDPs. This equivalence leads to a novel dual formulation and, ultimately,

a binary search method for robust policy evaluation. We begin with defining non-rectangular  $L_p$ -bounded uncertainty set as:

$$\mathcal{U}_p = \left\{ P \mid \|P - \hat{P}\|_p \leq \beta, \sum_{s'} P(s' \mid s, a) = 1 \right\},$$

where  $\hat{P}$  is the nominal kernel,  $\beta$  is uncertainty radius, and  $\|P - \hat{P}\|_p = (\sum_{s,a,s'} (P(s' \mid s, a) - \hat{P}(s' \mid s, a))^p)^{\frac{1}{p}}$ . The simplex constraint ensures that the transition kernel  $P$  satisfies the unity-sum-rows property, as discussed in [19]. Following previous works [7, 19, 17], we assume the radius  $\beta$  is sufficiently small to ensure all the kernels within the uncertainty sets are valid transition kernels. As discussed in [19], this assumption can be lifted by imposing boundary constraints ( $0 \leq P(s \mid s, a) \leq 1$ ) at the expense of additional complexity and without yielding significant additional insights. Throughout the paper, we use  $d^\pi, v^\pi, J^\pi, D^\pi$  as shorthand for  $d_{\hat{P}}^\pi, v_{\hat{P}}^\pi, J_{\hat{P}}^\pi$ , and  $D_{\hat{P}}^\pi$ , respectively, w.r.t. nominal kernel  $\hat{P}$ .

**Why non-rectangular RMDPs.** Note that the non-rectangular uncertainty sets allow noise in one state to be coupled with noise in other states. Before delving into solving them, we first discuss their importance. Why are uncertainty sets modeled with non-rectangular sets  $\mathcal{U}_p$  (e.g.,  $L_2$ -balls) better than rectangular ones?

In Figure 1, we illustrate this by capturing the uncertainty set using non-rectangular  $\mathcal{U}_2$  (circle/sphere) balls and rectangular (square/cube) balls. The blue dots represent possible environments, with the origin being the nominal environment. Points farther away from the origin indicate larger perturbations. Specifically, points near the corners of the square/cube represent environments with large perturbations in nearly all dimensions or coordinates simultaneously. The likelihood of such simultaneous perturbations is very low, and this issue becomes even more pronounced in higher dimensions. This phenomenon is well discussed in the paper *Lightning Doesn't Strike Twice: Coupled RMDPs*[21].

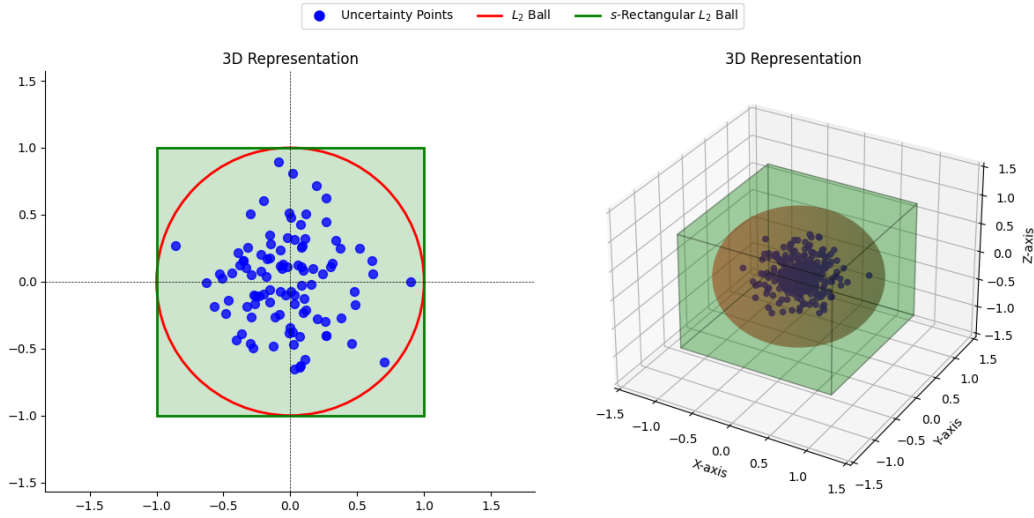


Figure 1: Modeling Uncertainty with Non-Rectangular and Rectangular  $L_2$ -Balls.

Moreover, as shown in the result below, most of the volume of a high-dimensional cube lies near its corners outside the embedded sphere. This implies that rectangular robust MDPs are overly conservative, as their uncertainty sets focus on environments near the corners—corresponding to highly unlikely extreme perturbations.

**Proposition 3.1.** Let  $\mathcal{U}_2^{sa}$  and  $\mathcal{U}_2^s$  denote the smallest  $sa$ -rectangular and  $s$ -rectangular sets, respectively, that contain  $\mathcal{U}_2$ . Then:

$$\frac{\text{vol}(\mathcal{U}_2)}{\text{vol}(\mathcal{U}_2^{sa})} = O(c_{sa}^{-SA}), \quad \text{and} \quad \frac{\text{vol}(\mathcal{U}_2)}{\text{vol}(\mathcal{U}_2^s)} = O(c_s^{-S}),$$

where  $\text{vol}(X)$  denotes the volume of the set  $X$ ,  $S = |\mathcal{S}|$ ,  $A = |A|$  and  $c_s, c_{sa} > 1$  are constants.

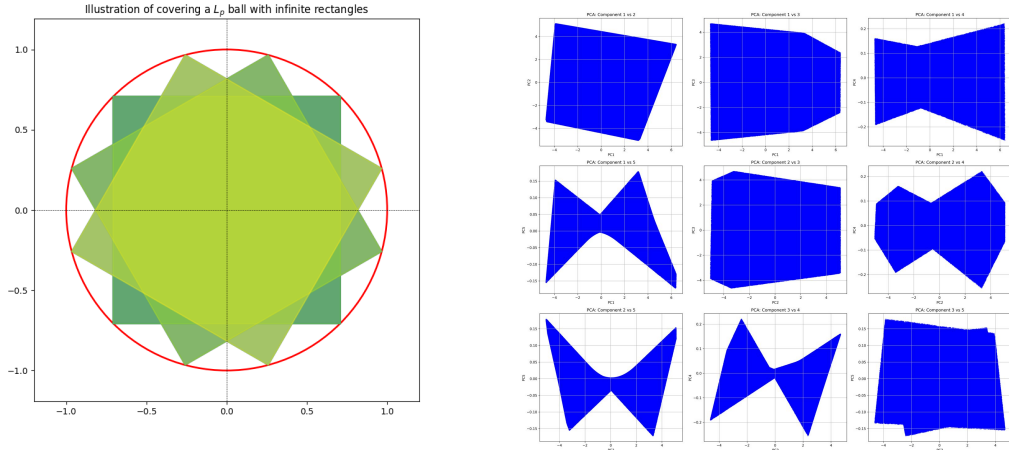
148 The result follows from comparing the  $n$ -dimensional sphere's volume  $c_n r^n$  ( $c_n \rightarrow 0$ ) [27], to the  
 149 enclosing cube's volume  $2^n r^n$  (side  $2r$ ), resulting in a ratio of  $O(2^n)$ .

150 In summary, real-world uncertainty sets are often non-rectangular and highly coupled. Their rectan-  
 151 gular relaxations (the smallest rectangular uncertainty sets encapsulating the original non-rectangular  
 152 sets) introduce exponentially more additional environments, many of which correspond to highly  
 153 perturbed kernels that are improbable in practice. As a result, relaxed rectangular robust MDPs can  
 154 produce overly conservative and suboptimal solutions compared to their non-rectangular counterparts.

155 **Complexity.** While non-rectangular robust MDPs better capture real-world uncertainty sets, robust  
 156 policy evaluation (even approximation) has been proven NP-hard for general uncertainty sets defined  
 157 as intersections of finite hyperplanes [35]. Specifically, [35] reduces an Integer Program (IP) with  
 158  $m$  constraints to robust MDPs where the uncertainty set consists of intersections of  $m$  half-spaces  
 159 ( $m$ -linear constraints). This polyhedral structure is fundamental to the hardness proof, consequently,  
 160 it does not extend to our uncertainty sets  $\mathcal{U}_p$  for  $p > 1$ . For the case of  $\mathcal{U}_1$ , the IP reduction does  
 161 apply, but since  $\mathcal{U}_1$  is defined by a single global constraint ( $\|P - \hat{P}_1\|_1 \leq \beta$ ), this implies that the  
 162 corresponding IP has only one simple constraint which is efficiently solvable. A detailed discussion  
 163 can be found in Appendix B.2.

164 Intuitively, the NP-Hardness result primarily applies to polyhedral uncertainty sets with numerous  
 165 vertices. This leaves room for the possibility that many uncertainty sets defined by a small number  
 166 of global constraints, such as norms or distances, might fall outside the scope of this hardness and  
 167 could potentially be tractable. However, we leave this intriguing question for future exploration: is  
 168 NP-Hardness merely the tip of the iceberg?

169 **Divide and Conquer.** The above discussion highlights the potential tractability of  $L_p$ -robust MDPs,  
 170 prompting us to address the challenge of solving them. A key insight is that the non-rectangular  
 171 uncertainty set  $\mathcal{U}_p$  can be expressed as a union of sa-rectangular sets  $\mathcal{U}_p^{sa}(b)$  with varying radius  
 172 vectors  $b$ , as formalized in the result below. Each sa-rectangular set can be efficiently solved  
 173 individually, paving the way for a more manageable approach to the overall problem.



(a) Illustration of Proposition 3.2: N-dimensional sphere can be written as infinite union of n-dimensional inscribing cubes.

(b) Projections of set  $\mathcal{D}$  along principal components, for  $S = 3$ ,  $A = 2$  with 10 millions samples, strongly suggesting non-convexity.

Figure 2:

174 **Proposition 3.2.** [Decomposition] The non-rectangular uncertainty set  $\mathcal{U}_p$  can be expressed as:

$$\mathcal{U}_p = \bigcup_{b \in \mathcal{B}} \mathcal{U}_p^{sa}(b),$$

175 where  $\mathcal{B} = \{b \in \mathbb{R}_+^{S \times A} \mid \|b\|_p \leq \beta\}$ , and  $\mathcal{U}_p^{sa}(b) = \{P \mid \|P(\cdot \mid s, a) - \hat{P}(\cdot \mid s, a)\|_p \leq$   
 176  $b(s, a), \forall (s, a)\}$ , is sa-rectangular uncertainty set with radius vector  $b$ .

The proof of the above result intuitively generalizes the idea that a circle (or  $n$ -dimensional sphere) can be covered by an inscribed square (or  $n$ -dimensional rectangles) touching its boundaries and a continuum of its rotated versions, as shown in Figure 2(a). This offers a significant simplification to the problem at hand, as it implies that non-rectangular policy evaluation (difficult) can be decomposed into sa-rectangular uncertainty sets (easier) as:

$$J_{\mathcal{U}_p}^\pi = \min_{b \in \mathcal{B}} \min_{P \in \mathcal{U}_p^{\text{sa}}(b)} J_P^\pi. \quad (1)$$

In essence, we have simplified a complex problem into an infinite number of more manageable ones. However, the task remains incomplete. Although a closed-form expression exists for  $J_{\mathcal{U}_p^{\text{sa}}}^\pi = J^\pi - \sum_{s,a} d^\pi(s,a) b_{sa} \sigma_q(v_{\mathcal{U}_p^{\text{sa}}(b)}^\pi)$ , where  $q$  is the Hölder conjugate of  $p$  (i.e.,  $\frac{1}{p} + \frac{1}{q} = 1$ ) and  $\sigma_p$  is the generalized standard deviation (GSTD) defined as  $\sigma_p(v) = \min_{\omega \in \mathbb{R}} \|v - \omega \mathbf{1}\|_p$  [19], this approach is still computationally impractical. The core challenge lies in solving  $\max_{b \in \mathcal{B}} \sum_{s,a} d^\pi(s,a) b_{sa} \sigma_q(v_{\mathcal{U}_p^{\text{sa}}(b)}^\pi)$ , which remains a formidable task. To circumvent this, we leverage the dual formalism, which is elaborated in the next section.

### 3.1 Dual Formulation of Robust MDPs

Here, we present a dual formulation for robust MDPs specifically for  $L_p$ -bounded uncertainty sets. While this formulation is inherently more intricate than the classical dual formulation for standard MDPs [26], it forms the foundation for all subsequent results in this work.

Now, leveraging results from [17], we know that the worst-case kernel for sa-rectangular uncertainty sets,  $P_{\mathcal{U}_p^{\text{sa}}(b)}^\pi = \hat{P} - bk^\top$ , can be expressed as a rank-one perturbation of the nominal kernel, where  $k \in \mathcal{K} := \{k \mid \|k\|_p \leq 1, \mathbf{1}^\top k = 0\}$ . Consequently, the adversary can restrict their focus to rank-one perturbations, enabling us to reformulate the robust return as:

$$J_{\mathcal{U}_p}^\pi = \min_{b \in \mathcal{B}} \min_{k \in \mathcal{K}} J_{\hat{P} - bk^\top}^\pi = \min_{b \in \mathcal{B}} \min_{k \in \mathcal{K}} \mu^\top D_{\hat{P} - bk^\top}^\pi R^\pi,$$

where the last equality stems from  $J_P^\pi = \mu^\top D_P^\pi R^\pi$ . Further, leveraging Lemma 4.4 from [17] or directly applying the Sherman–Morrison formula [4] (see Proposition D.1), the robust return can be expressed as:

$$J_{\mathcal{U}_p}^\pi = \min_{b \in \mathcal{B}, k \in \mathcal{K}} \left[ \mu^\top D^\pi R^\pi - \gamma \mu^\top D^\pi b^\pi \frac{k^\top D^\pi R^\pi}{1 + \gamma k^\top D^\pi b^\pi} \right],$$

where  $b_s^\pi := \sum_a \pi(a|s) b_{sa}$ . The following result introduces a more concise and interpretable form of this robust return expression.

**Lemma 3.3.** [Penalized Robust Return] *The robust return can be expressed as:*

$$J_{\mathcal{U}_p}^\pi = J^\pi - \gamma \max_{b \in \mathcal{B}, k \in \mathcal{K}} \frac{\langle k, v_R^\pi \rangle \langle d^\pi, b^\pi \rangle}{1 + \gamma \langle k, v_b^\pi \rangle},$$

where  $v_b^\pi = D^\pi b^\pi$  represents the value function with uncertainty radius  $b$  as the reward vector.

For the first time, the above result expresses the robust return in terms of the nominal return  $J^\pi$  and a penalty term involving only nominal values ( $d^\pi$ ,  $v_R^\pi = v^\pi$ , and  $v_b^\pi$ ). Notably, the denominator term  $1 + \gamma \langle k, v_b^\pi \rangle$  is strictly positive (see appendix for details). In the subsequent subsections, we delve deeper into evaluating this penalty term and analyzing the nature of the optimal  $(k, b)$  for a given policy  $\pi$ , revealing the adversary. Finally, by maximizing the robust return  $J_{\mathcal{U}_p}^\pi$  over policies, we get a dual formulation, as stated below.

**Theorem 3.4** (Dual Formulation). *The optimal robust return is the solution to*

$$J_{\mathcal{U}_p}^* = \max_{D \in \mathcal{D}} \min_{k \in \mathcal{K}, b \in \mathcal{B}} \left[ \mu^T D R - \gamma \mu^T D b \frac{k^T D R}{1 + \gamma k^T D b} \right]$$

where  $\mathcal{D} = \{ D^\pi H^\pi \mid \pi \in \Pi \}$ ,  $D^\pi = (I - \gamma \hat{P}^\pi)^{-1}$  and  $H^\pi : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \rightarrow \mathbb{R}^{\mathcal{S}}$  is a policy averaging linear operator defined as  $H^\pi R := R^\pi$ .

The dual formulations for the sa-rectangular and s-rectangular cases differ notably in their definitions of  $\mathcal{B}$ . In the sa-rectangular case,  $\mathcal{B} = \{\beta\}$ , whereas in the s-rectangular case,  $\mathcal{B} = \{b \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \mid$

215  $\|b_s\|_p \leq \beta_s\}$ . These distinctions are elaborated in the appendix. The result above frames the dual  
 216 of robust MDPs as a min-max problem, offering valuable and insightful perspectives. However, as  
 217 Figure 2(b) suggests (with further details in the appendix), the set  $\mathcal{D}$  may be non-convex, which  
 218 complicates the problem. Despite this, we believe that the dual formulation holds potential for future  
 219 work, providing deeper insights and enabling the development of improved algorithms. In this work,  
 220 we keep our focus on the robust policy evaluation while policy improvement is addressed via existing  
 221 robust policy gradient method with proven guarantees [32], discussed further in Appendix C.

### 222 3.2 Robust Policy Evaluation

223 Now, we directly attempt to evaluate the penalty term in Lemma 3.3 which leads to a binary search-  
 224 based robust policy evaluation algorithm. The key idea is to identify a bisection function:

$$F(\lambda) = \max_{b \in \mathcal{B}} \|E_\lambda^\pi b\|_q,$$

225 where  $E_\lambda^\pi := \gamma \left( I - \frac{\mathbf{1}\mathbf{1}^\top}{S} \right) [ D^\pi R^\pi \mu^\top D^\pi - \lambda D^\pi ] H^\pi$ . Note that  $E_\lambda^\pi$  is constructed using quantities  
 226 that are computationally straightforward, and  $H^\pi : \mathbb{R}^{\mathcal{S} \times \mathcal{A}} \rightarrow \mathbb{R}^{\mathcal{S}}$  represents the policy-averaging  
 227 linear operator, defined by  $(H^\pi R)(s) := \sum_a \pi(a|s) R(s, a)$ .

228 **Lemma 3.5** (Robust Policy Evaluation). *Let  $\lambda^*$  be a fixed point of the function  $F(\lambda)$ , then the robust*  
 229 *return can be expressed as:*

$$J_{\mathcal{U}_p}^\pi = J^\pi - \lambda^*.$$

230 *And  $\lambda^*$  can be efficiently computed using binary search Algorithm 1 as  $F(\lambda) > \lambda \iff \lambda > \lambda^*$ .*

231 *Proof.* The proof can be found in Appendix (see Lemma F.2).  $\square$

232 The result enables a direct computation of the robust return by iteratively refining  $\lambda$  until convergence,  
 233 leveraging the monotonicity properties of  $F(\lambda)$ . Further, the bisection property of  $F$  established in  
 234 the result, directly implies the linear convergence rate of Algorithm 1, as stated result below.

---

#### Algorithm 1 Binary Search for Robust Policy Evaluation

---

**Initialize:** Upper limit  $\lambda_u = \frac{1}{1-\gamma}$ , lower limit  $\lambda_l = 0$

- 1: **while** not converged:  $n = n + 1$  **do**
  - 2:   **Bisection value:**  $\lambda_n = (\lambda_l + \lambda_u)/2$
  - 3:   **Bisection:**  $\lambda_l = \lambda_n$  if  $F(\lambda_n) > \lambda_n$ , else  $\lambda_u = \lambda_n$ .
  - 4:   **Update robust return:**  $J_n = J^\pi - \lambda_n$ .
  - 5: **end while**
- 

235 **Theorem 3.6.** *Algorithm 1 converges linearly, i.e.,*

$$J_n - J_{\mathcal{U}_p}^\pi \leq O(2^{-n}).$$

236 We conclude that robust evaluation can be performed efficiently with linear iteration complexity. How-  
 237 ever, each iteration involves solving the subproblem  $\max_{x \in \mathcal{B}} \|Ax\|_q$ , as part of Algorithm 1. For sim-  
 238 plicity, we focus on the specific case where  $p = 2$ , resulting in the problem:  $\max_{\|x\|_2 \leq 1, x \succeq 0} \|Ax\|_2$ .  
 239 To address this, we propose a modified eigenvalue-based algorithm (Algorithm 2). This method has  
 240 a time complexity of  $O(S^3 A^3)$  and demonstrates excellent practical performance. Specifically, to  
 241 achieve comparable results to those obtained using the numerical solver ‘scipy.minimize’ it takes  
 242 significantly less time, by an order of magnitude. Further details on this method, including theoretical  
 243 insights and empirical evaluations, are provided in Appendix G. Additionally, the performance of  
 244 robust policy evaluation Algorithm 1, is further validated experimentally in Section 5.

---

#### Algorithm 2 Spectral method for computing $\max_{x \in \mathcal{B}} \|Ax\|_2$

---

- 1: Compute eigenvector  $v_i$  and eigenvalues  $\lambda_i$  of  $A^\top A$
  - 2: WLOG let  $\|v_i^+\|_2 \geq \|v_i^-\|_2$  where  $v_i^+ = \max(v_i, 0)$ ,  $v_i^- = -\min(v_i, 0)$
  - 3: Compute best score :  $j = \arg \max_i \lambda_i \langle v_i, \frac{v_i^+}{\|v_i^+\|_2} \rangle$ .
  - 4: **Output:** Approximate maximum value  $\beta \|A \frac{v_j^+}{\|v_j^+\|_2}\|_2$ .
-

## 4 Revealing the Adversary

We provide the first insights into the structure of the worst-case kernel in non-rectangular robust MDPs, addressing an unexplored area in the literature. The following result reveals that, similar to rectangular uncertainty sets [17], the worst-case transition kernel is a rank-one perturbation of the nominal kernel, but with a more complex structure.

**Theorem 4.1** (Worst-Case Kernel). *For a policy  $\pi$  and uncertainty set  $\mathcal{U}_p$ , the worst-case transition kernel is:*

$$P_{\mathcal{U}_p}^\pi = \hat{P} - bk^\top,$$

where  $(k, b)$  solves:

$$\max_{k \in \mathcal{K}, b \in \mathcal{B}} \frac{J_b^\pi \langle k, v_R^\pi \rangle}{1 + \gamma \langle k, v_b^\pi \rangle}.$$

The above result follows directly from Lemma 3.3. It highlights the adversary’s strategic use of  $k$ ,  $b$ , and their interaction with the value functions  $v_R^\pi$  and  $v_b^\pi$ , revealing a more nuanced structure compared to the rectangular case. The adversary’s objectives in selecting the worst-case kernel are twofold:

- **Maximizing Trajectory Uncertainty ( $J_b^\pi$ ):** The adversary seeks to increase the agent’s visits to high-uncertainty states, enhancing its ability to steer the agent toward disadvantageous outcomes.
- **Optimizing the Perturbation Direction ( $k$ ):** The adversary selects  $k$  to maximize  $k^\top v_R^\pi$ , thereby pushing the agent into low-reward trajectories, while simultaneously minimizing  $k^\top v_b^\pi$  to ensure the agent remains exposed to high-uncertainty states.

These insights provide a deeper understanding of the adversary’s behavior and offer practical guidance for designing more resilient robust algorithms to counteract such strategies effectively.

### Message to Practitioners

The adversary focuses solely on rank-one perturbations of the nominal kernel, iteratively boosting its influence by pushing the agent into high-uncertainty states, then leveraging that influence to steer the agent toward low-reward trajectories, ultimately driving the agent to the lowest possible return.

## 5 Experiments: Robust Policy Evaluation

We conduct a numerical comparison of our Algorithm 1 and CPI (Algorithm 3.2 from [20], reproduced as Algorithm 3 in the appendix) for robust policy evaluation. The experiments are performed using a randomly generated nominal kernel  $\hat{P}$ , reward function  $R$ , and policy  $\pi$ . An uncertainty set  $\mathcal{U}_2$  is constructed using the nominal kernel with a fixed uncertainty radius  $\beta$ .

Figure 3 demonstrates the convergence behavior of both methods, presenting results based on the number of iterations (left panel) and computation time (right panel). The left panel shows the robust return achieved per iteration, while the right panel depicts the robust return as a function of wall-clock time. Note that the x-axes of the figure have a logarithmic scale in order to clearly capture the slow convergence of the CPI method.

- **Our Algorithm 1.** We apply our Binary Search Algorithm 1 to perform robust policy evaluation with the given nominal kernel  $\hat{P}$  and uncertainty radius  $\beta$ . Each iteration of the algorithm involves computing  $F(\lambda)$ , for which our Spectral Algorithm 2 is employed. Our algorithm converges very quickly requiring only a few iterations.
- **Algorithm 3.2 of [20].** We run Algorithm 3 with precomputed values of  $d^\pi$  and  $A^\pi$ . The step sizes are chosen to be either a small constant or dynamically adjusted, as described in the algorithm. Note that Line 3 of the algorithm involves solving  $\arg \min_{P \in \mathcal{U}_2} \langle x, P \rangle$ . This constrained optimization is solved using a numerical method (`scipy.minimize`). This gradient based method improves very slowly and converges very far from the true robust return as the uncertainty set  $\mathcal{U}_2$  is very non-rectangular.



286 • **Brute Force Benchmark.** To approximate the true robust return, we generate a large  
 287 number of random samples  $\{P_i \mid i \leq n\}$  from  $\mathcal{U}_2$  and estimate the empirical minimum,  
 288  $\min_i J_{P_i}^\pi$ , as a proxy for the robust return. Note this method requires exponential number of  
 289 samples to reasonably cover the entire uncertainty set. Hence the values obtained in Figure  
 290 3, are an approximate upper bound on the true robust return.

291 The results in Figure 3 reflect a general trend observed across a wide range of experiments conducted  
 292 with state space sizes ranging from  $S = 5$  to  $S = 190$  and uncertainty radius  $\beta \in \{0.005, 0.01, 0.05\}$ .  
 293 Our proposed algorithm consistently demonstrates superior performance, converging in significantly  
 294 fewer iterations and less computation time while the computational demands of the CPI algorithm  
 295 grow substantially with larger state spaces. Hence, our method exhibits more favorable scaling  
 296 properties, making it practical for high-dimensional problems.

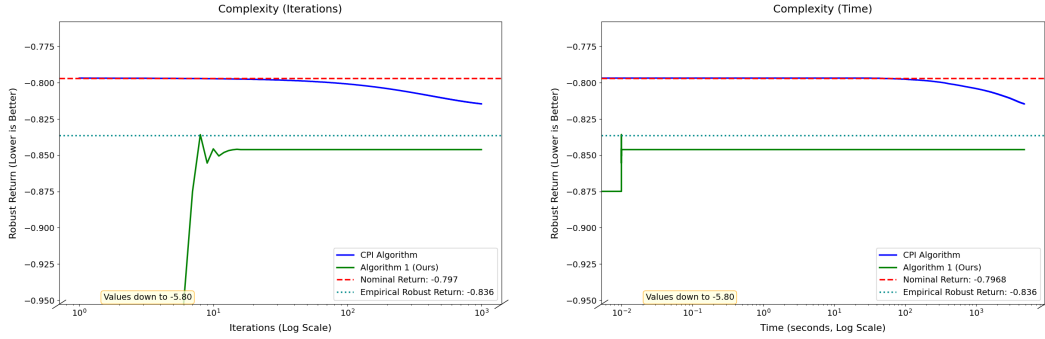


Figure 3: Comparison of Algorithm 1 (Ours) and the CPI Algorithm for  $\beta$  (Uncertainty Radius)  $= 0.05$ ,  $S = 10$ ,  $A = 8$ ,  $\gamma = 0.9$ , and a convergence tolerance of  $10^{-4}$ .

297 The codes, detailed explanations, and additional experiments are available at <https://anonymous.4open.science/r/non-rectangular-rmdp-77B8>. System details for the experiments are as follows: **Operating System:** macOS Sequoia (Version 15.4.1), **Chip:** Apple M2, **Cores:** 8 (4 performance and 4 efficiency), **Memory:** 16 GB (LPDDR5).

## 301 6 Discussion

302 We studied robust Markov decision processes (RMDPs) with non-rectangular  $L_p$ -bounded uncertainty sets, balancing expressiveness and tractability. We showed that these uncertainty sets can be  
 303 decomposed into infinitely many *sa*-rectangular sets, reducing robust policy evaluation to a min-max  
 304 fractional optimization problem (dual form). This novel dual formulation provides key insights into  
 305 the adversary and leads to the development of an efficient robust policy evaluation algorithm. Theory  
 306 and experiments demonstrate the effectiveness of our approach, significantly outperforming the  
 307 existing methods. These findings further pave the way for scalable and efficient robust reinforcement  
 308 learning algorithms.

310 **Limitations.** Similar to [7, 19, 17], we have considered small enough uncertainty radius to ensure  
 311 positivity of the kernel. As discussed in [19], imposing this additional positivity constraints (or  
 312 dealing with nominal kernel with zero transition probability to some states ) would significantly  
 313 complicate the analysis without yielding significant additional insights. However, we leave a thorough  
 314 investigation of this topic for future work.

315 **Future Work.** Our results naturally extend to uncertainty sets that can be expressed as a finite union  
 316 of  $L_p$  balls. Furthermore, any uncertainty set can be approximated using a finite number of  $L_p$  balls,  
 317 with smaller balls providing a better approximation. However, the number of balls required for an  
 318 accurate approximation may grow prohibitively large. While this work is limited to  $L_p$  norms, it may  
 319 be possible to generalize our approach to other types of uncertainty sets. A key challenge in such an  
 320 extension would be identifying the structure of the worst-case kernel and developing corresponding  
 321 matrix inversion techniques.

## References

- [1] M. A. Abdullah, H. Ren, H. B. Ammar, V. Milenkovic, R. Luo, M. Zhang, and J. Wang. Wasserstein robust reinforcement learning, 2019.
- [2] E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.
- [3] J. A. Bagnell, A. Y. Ng, and J. G. Schneider. Solving uncertain markov decision processes. Technical report, Carnegie Mellon University, 2001.
- [4] M. S. Bartlett. An Inverse Matrix Adjustment Arising in Discriminant Analysis. *The Annals of Mathematical Statistics*, 22(1):107 – 111, 1951.
- [5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.
- [6] R. Dadashi, A. A. Taïga, N. L. Roux, D. Schuurmans, and M. G. Bellemare. The value function polytope in reinforcement learning, 2019.
- [7] E. Derman, M. Geist, and S. Mannor. Twice regularized mdps and the equivalence between robustness and regularization, 2021.
- [8] U. Gadot, E. Derman, N. Kumar, M. M. Elfatih, K. Levy, and S. Mannor. Solving non-rectangular reward-robust mdps via frequency regularization, 2023.
- [9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, first edition edition, 1979.
- [10] V. Goyal and J. Grand-Clément. Robust markov decision process: Beyond rectangularity, 2018.
- [11] J. Grand-Clément, N. Si, and S. Wang. Tractable robust markov decision processes, 2024.
- [12] G. A. Hanasusanto and D. Kuhn. Robust data-driven dynamic programming. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [13] C. P. Ho, M. Petrik, and W. Wiesemann. Partial policy iteration for l1-robust markov decision processes, 2020.
- [14] C. P. Ho, M. Petrik, and W. Wiesemann. Robust  $\phi$ -divergence MDPs. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [15] G. N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, May 2005.
- [16] D. L. Kaufman and A. J. Schaefer. Robust modified policy iteration. *INFORMS J. Comput.*, 25:396–410, 2013.
- [17] N. Kumar, E. Derman, M. Geist, K. Y. Levy, and S. Mannor. Policy gradient for rectangular robust markov decision processes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [18] N. Kumar, K. Wang, K. Levy, and S. Mannor. Policy gradient for reinforcement learning with general utilities, 2023.
- [19] N. Kumar, K. Wang, K. Y. Levy, and S. Mannor. Efficient value iteration for s-rectangular robust markov decision processes. In *Forty-first International Conference on Machine Learning*, 2024.
- [20] M. Li, D. Kuhn, and T. Sutter. Policy gradient algorithms for robust mdps with non-rectangular uncertainty sets, 2024.
- [21] S. Mannor, O. Mebel, and H. Xu. Lightning does not strike twice: Robust mdps with coupled uncertainty. *CoRR*, abs/1206.4643, 2012.

- [22] S. Mannor, O. Mebel, and H. Xu. Robust mdps with k-rectangular uncertainty. *Math. Oper. Res.*, 41(4):1484–1509, nov 2016.
- [23] S. Mannor, D. Simester, P. Sun, and J. N. Tsitsiklis. Bias and variance in value function estimation. In *Proceedings of the Twenty-First International Conference on Machine Learning, ICML ’04*, page 72, New York, NY, USA, 2004. Association for Computing Machinery.
- [24] A. Nilim and L. E. Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.*, 53:780–798, 2005.
- [25] C. Packer, K. Gao, J. Kos, P. Krähenbühl, V. Koltun, and D. Song. Assessing generalization in deep reinforcement learning, 2018.
- [26] M. L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.
- [27] D. J. Smith and M. K. Vamanamurthy. How small is a unit ball? *Mathematics Magazine*, 62(2):101–107, 1989.
- [28] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.
- [29] A. Tamar, S. Mannor, and H. Xu. Scaling up robust mdps using function approximation. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 181–189. JMLR.org, 2014.
- [30] K. Wang, U. Gadot, N. Kumar, K. Levy, and S. Mannor. Robust reinforcement learning via adversarial kernel approximation, 2023.
- [31] K. Wang, N. Kumar, K. Zhou, B. Hooi, J. Feng, and S. Mannor. The geometry of robust value functions. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22727–22751. PMLR, 17–23 Jul 2022.
- [32] Q. Wang, C. P. Ho, and M. Petrik. Policy gradient in robust mdps with global convergence guarantee, 2023.
- [33] Y. Wang and S. Zou. Online robust reinforcement learning with model uncertainty, 2021.
- [34] Y. Wang and S. Zou. Policy gradient method for robust reinforcement learning, 2022.
- [35] W. Wiesemann, D. Kuhn, and B. Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.
- [36] H. Xu and S. Mannor. Robustness and generalization, 2010.
- [37] C. Zhao, O. Sigaud, F. Stulp, and T. M. Hospedales. Investigating generalisation in continuous deep reinforcement learning, 2019.
- [38] R. Zhou, T. Liu, M. Cheng, D. Kalathil, P. Kumar, and C. Tian. Natural actor-critic for robust reinforcement learning with function approximation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [NA].
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

## 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The paper is theoretical in nature, the claims are reflected in abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

## 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Discussed in the last section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proof in appendix and assumption in the main text.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Paper is theoretical in nature, toyish experiments are just for the sake of completeness.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often

one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Codes are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [NA]

Justification: No training , no tests sets.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: Not applicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Provided in the main text.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer:

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets



Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

## 16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The paper is theoretical in nature, hence not relevant. LLM is used for only writing and editing only english text.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## Notations and Definitions

For a set  $\mathcal{S}$ ,  $|\mathcal{S}|$  denotes its cardinality.  $\langle u, v \rangle := \sum_{s \in \mathcal{S}} u(s)v(s)$  denotes the dot product between functions  $u, v : \mathcal{S} \rightarrow \mathbb{R}$ .  $\|v\|_p^q := (\sum_s |v(s)|^p)^{\frac{q}{p}}$  denotes the  $q$ -th power of  $L_p$  norm of function  $v$ , and we use  $\|v\|_p := \|v\|_p^1$  and  $\|v\| := \|v\|_2$  as shorthand. For a set  $\mathcal{C}$ ,  $\Delta_{\mathcal{C}} := \{a : \mathcal{C} \rightarrow \mathbb{R} | a_c \geq 0, \forall c, \sum_{c \in \mathcal{C}} a_c = 1\}$  is the probability simplex over  $\mathcal{C}$ .  $\text{var}(\cdot)$  is variance function, defined as  $\text{var}(v) = \sqrt{\sum_{s \in \mathcal{S}} (v(s) - \bar{v})^2}$  where  $\bar{v} = \frac{\sum_{s \in \mathcal{S}} v(s)}{|\mathcal{S}|}$  is the mean of function  $v : \mathcal{S} \rightarrow \mathbb{R}^d$ .  $\mathbf{0}, \mathbf{1}$  denotes all zero vector and all ones vector/function respectively of appropriate dimension/domain.  $\mathbf{1}(a = b) := 1$  if  $a = b$ , 0 otherwise, is the indicator function. For vectors  $u, v$ ,  $\mathbf{1}(u \geq v)$  is component wise indicator vector, i.e.  $\mathbf{1}(u \geq v)(x) = \mathbf{1}(u(x) \geq v(x))$ .  $A \times B = \{(a, b) | a \in A, b \in B\}$  is the Cartesian product between set  $A$  and  $B$ .

## A Related Work

**Rectangular Robust MDPs.** In the literature, the sa-rectangular uncertainty is a very old assumption [15, 24]. [35] introduced s-rectangular uncertainty sets and proved its tractability, in addition to the intractability of the general non-rectangular uncertainty sets. The most advantageous aspect of the s-rectangularity, is the existence of contractive robust Bellman operators. This gave rise to many robust value based methods [13, 32]. Further, for many specific uncertainty sets, robust Bellman operators are equivalent to regularized non-robust operators, making the robust value iteration as efficient as non-robust MDPs [7, 33, 19]. There exists many policy gradient based methods for robust

Table 2: Useful Notations

Notation	Definition	Remark
$p, q$	$\frac{1}{p} + \frac{1}{q} = 1$	Holder’s conjugates
$\sigma_p$		Standard deviation w.r.t. $L_p$ norm
$v^\pi, v_{P,R}^\pi$	$(I - \gamma P^\pi)^{-1} R^\pi$	Value function
$D^\pi, D_{P,R}^\pi$	$(I - \gamma P^\pi)^{-1}$	Occupancy matrix
$d^\pi, d_{P,\mu}^\pi$	$\mu^T (I - \gamma P^\pi)^{-1}$	Occupancy measure
$\mathcal{U}, \mathcal{U}_p^{\text{sa}}, \mathcal{U}_p^{\text{s}}, \mathcal{U}_p$		Uncertainty sets

MDPs, relying upon contractive robust Bellman operators for the robust policy evaluation [34, 17]. Further, [38, 30] try to refine the process, and directly get samples from the adversarial model via pessimistic sampling. There exist other notions of rectangularity such as k-rectangularity [22] and r-rectangularity [10] which are sparsely studied. However, [11] shows, theses uncertainty sets are either equivalent to s-rectangularity or non-tractable.

**Non-Rectangular Reward Robust MDPs.** Policy evaluation for robust MDPs with non-rectangular uncertainty set is proven to be a Strongly-NP-Hard problem [35], in general. For a very specific case, where uncertainty is limited only to reward uncertainty bounded with  $L_p$  norm, [8] proposed robust policy evaluation via frequency (occupation measure) regularization, and derived the policy gradient for policy improvement.

**Approximate Policy Evaluation for Non-Rectangular Kernel RMDPs.** [20] provides the following two policy evaluation methods for robust MDPs for general uncertainty sets.

- Langevin dynamics based Algorithm 3.1 of [20]: This Langevin dynamics based Markov Chain Monte Carlo method solves the robust policy evaluation problem to global optimality with arbitrary small accuracy  $\epsilon$ . The iteration complexity of the algorithm is  $O(2^q \log \frac{1}{\epsilon})$  which is exponential in the dimension of the uncertainty set  $q$ . The algorithm is well suited only for small dimensional uncertainty. For a general case the dimension  $q = S^2 A$  can be very large, this makes the algorithm very computationally inefficient as expected from the hardness result from [2].
- CPI sytle Algorithm 3.2 of [20] (presented in Algorithm 3): This CPI based algorithm computes the robust policy with iteration complexity of  $O(\frac{1}{\epsilon^2})$  with an accuracy of  $\delta_d(2\epsilon + \delta_P)$ , where  $\delta_d$  is mismatch-coefficient and  $\delta_P$  is measure of non-rectangularity of the uncertainty set. However, the mismatch coefficient may not exist without an irreducibility assumption (Assumption 1 in [20]), moreover even under Assumption 1, the constant  $\delta_d = O(2^S)$  can be exponentially large for ladder MDPs which have large diameter (more details provided below). In addition, the non-rectangularity constant  $\delta_P$  can be as large as  $O(\sqrt{S})$ . Hence, a large  $\delta_d \delta_P > \frac{2}{1-\gamma}$  makes the bound meaningless, as the sub-optimality is upper bounded by  $\frac{2}{1-\gamma}$ . To summarize, this approach is efficient only for small diameter MDPs and almost rectangular uncertainty sets.
- Our Method: We provide a robust policy evaluation method for  $L_2$ -robust MDPs with an iteration complexity of  $O(\log \frac{1}{\epsilon})$  and with an accuracy of  $\epsilon$ . This is possible as we showed that the NP-hardness result of [2] doesn’t apply to this case.  
We don’t require the irreducibility Assumption 1 of [20] which can be very limiting. Further, the  $L_p$  robust MDPs may have very large tolerance  $\delta_d \delta_P$  hence the Algorithm 3.2 from [20] is not applicable.

**Difficult MDPs for Algorithm 3.2 of [20] :**

- **MDP with high mismatch coefficients** : Consider an MDP with only one action and state-space  $\{s_i | 1 \leq i \leq S\}$ . Let  $s_1$  be the starting state. Let the kernel be defined as

$$P_x(s_{\max\{i+1, S\}} | s_i) = x, \quad P_x(s_i | s_i) = 1 - x.$$

Now let the uncertainty set be  $\mathcal{P} = \{P_x | x \in [0.4, 0.6]\}$ . Note that for this case,  $\log(\delta_d) \geq \log(\frac{d^{P_{0.6}}(s_S | s_1)}{d^{P_{0.4}}(s_S | s_1)}) = O(S)$ .

- **High non-rectangularity coefficient** : This is inspired from the fact that

$$\delta = \max_{||a|| \leq 1} \left[ \max_{b \in \mathcal{B}_f} \langle a, b \rangle - \max_{b \in \mathcal{B}} \langle a, b \rangle \right],$$

where  $\mathcal{B} = B(0, 1)$  is a unit ball around origin, and  $\mathcal{B}_f = [-1, 1]^n$  is the smallest rectangular cube containing  $\mathcal{B}$ . Then choosing  $a = \{\frac{1}{\sqrt{n}}\}^n$ , we have  $\max_{b \in \mathcal{B}_f} \langle a, b \rangle = \sqrt{n}$  and  $\max_{b \in \mathcal{B}} \langle a, b \rangle = 1$ . This implies  $\delta \geq \sqrt{n} - 1$ .

From definition in page 11 of [20], we have

$$\delta_{\mathcal{P}} = \max_{P \in \mathcal{P}_f} \langle \nabla V, P \rangle - \max_{P \in \mathcal{P}} \langle \nabla V, P \rangle$$

where  $\mathcal{P}_f$  is the smallest s-rectangular uncertainty containing  $\mathcal{P}$ . Here,  $P \in \mathbb{R}^{SA \times S}$ , this suggests  $\delta_{\mathcal{P}}$  can be of the order of  $O(S\sqrt{A})$ .

The discussion is summarize in the Table 1.

---

**Algorithm 3** CPI Algorithm 3.2 of [20] for Robust Policy Evaluation

---

**Input:** Nominal kernel  $\hat{P}$ , policy  $\pi$ , Uncertainty set  $\mathcal{U}$ .

- 1: **while** not converged:  $n = n + 1$  **do**
- 2: Define :  $f(P) := \frac{1}{1-\gamma} \sum_{s,a,s'} d_{\hat{P}}^{\pi}(s) \pi(a|s) A_{\hat{P}}^{\pi}(s, a, s') P(s'|s, a)$ ,  
where  $A_{\hat{P}}^{\pi}(s, a, s') := \gamma \left[ P(s'|s, a) v_{\hat{P}}^{\pi}(s') - \sum_{s''} P(s''|s, a) v_{\hat{P}}^{\pi}(s'') \right]$ .
- 3: Compute  $P^* \in \arg \min_{P \in \mathcal{U}} f(P)$ .
- 4: Update the estimated worst kernel:  $P_{n+1} = (1 - \alpha_n) P_n + \alpha_n P^*$ ,  
where  $\alpha_n = -\frac{(1-\gamma)^3}{4\gamma^2} f(P^*)$
- 5: **end while**

**Return:** Robust return  $J_{P_{\infty}}^{\pi}$ .

---

**Robust Policy Gradient Methods.** The absence of contractive robust Bellman operators renders the development of value-based methods for robust MDPs particularly challenging. Consequently, policy gradient methods naturally emerge as a viable alternative. The update rule is given by:

$$\pi_{k+1} = \text{Proj}_{\pi \in \Pi} \left[ \pi_k - \eta_k \nabla_{\pi} J_{P_k}^{\pi_k} \right], \quad (2)$$

where  $J_{P_k}^{\pi_k} - J_{\mathcal{U}}^{\pi_k} \leq \epsilon \gamma^k$  and learning rate  $\eta_k = O(\frac{1}{\sqrt{k}})$ . This approach guarantees convergence to a global solution within  $O(\epsilon^{-4})$  iterations [32].

However, this update rule depends on oracle access to the robust gradient, which is highly challenging to obtain because robust policy evaluation is an NP-hard problem.

## B On the Non-Rectangular Uncertainty Sets

### B.1 Why non-rectangular RMPDs

**Proposition B.1.** Let  $\mathcal{U}_2^{sa}, \mathcal{U}_2^s$  be the smallest sa-rectangular set and s-rectangular set containing  $\mathcal{U}_2$  then

$$\frac{\text{vol}(\mathcal{U}_2)}{\text{vol}(\mathcal{U}_2^{sa})} = O(c_{sa}^{-SA}), \quad \text{and} \quad \frac{\text{vol}(\mathcal{U}_2)}{\text{vol}(\mathcal{U}_2^s)} = O(c_s^{-S}),$$

where  $\text{vol}(X)$  is volume of the set  $X$  and  $c_s, c_{sa} > 1$  are some constants.

826 *Proof.* Volume of  $n$ -dimension sphere of radius  $r$  is  $c_n r^n$  where  $c_n \leq \frac{8\pi^2}{15}$  [27]. And to cover an  
 827  $n$ -dimension sphere of radius  $r$ , we need a cube of radius  $2r$  whose volume is  $(2r)^n$ . Hence the first  
 828 result  $\frac{\text{vol}(\mathcal{U}_2)}{\text{vol}(\mathcal{U}_2^{ss})} = O(2^{-SA})$  immediately follows.

829 Now, the volume of the set of  $X = \times_{s \in \mathcal{S}} X_s$  where  $X_s$  is an  $A$ -dimension sphere of radius  $r$ ,  
 830 then the volume of  $X$  is  $(c_A r)^S$ . And the volume of an  $SA$  dimensional sphere is  $c_{SA} r^{SA}$ , where  
 831  $\lim_{n \rightarrow \infty} c_n \rightarrow 0$  [27]. Hence the ratio of their volume is  $O((c_A)^S)$ , implying the other result.  $\square$

## 832 B.2 Complexity

### Reduction of Integer Program to Robust MDP

**0/1 Integer Program (IP):** For  $g, c \in \mathcal{Z}^n, \zeta \in \mathcal{Z}, F \in \mathcal{Z}^{m \times n}$ ,

$$\exists x \in \{0, 1\}^n \quad \text{s.t.} \quad Fx \leq g \quad \text{and} \quad c^\top x \leq \zeta?$$

is a NP-Hard problem [9], [35] which reduces into the following robust MDP.

**Robust MDP:**

1. State Space  $\mathcal{S} = \{b_j, b_j^0, b_j^1 \mid j = 1, \dots, n\} \cup \{c_0, \tau\}$ , where  $\tau$  is a terminal state.
2. Singleton Action Space:  $\mathcal{A} = \{a\}$ .
3. Uncertainty set:  $\mathcal{U} = \{P_\xi \mid \xi \in [0, 1]^n, F\xi \leq g\}$
4. Discount factor  $\gamma \in [0, 1)$ ; Uniform initial state distribution  $\mu$ .
5. Big reward  $M \geq \frac{\gamma A n \sum_i c_i}{2\epsilon^2}$  where  $\epsilon \ll 1$  helps in rounding.
6. Transitions and rewards are illustrated in Figure 4

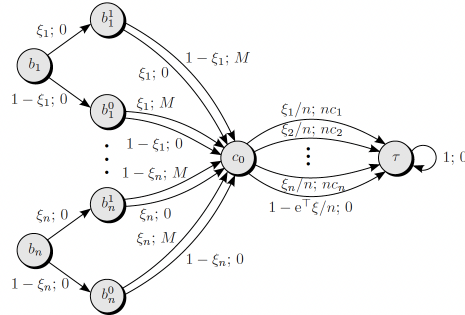


Figure 4: MDP  $P_\xi$ , and  $R$ (Figure 5 of [35]).

834 Robust policy evaluation is proven to be NP-hard for general uncertainty sets defined as intersections  
 835 of finite hyperplanes [35]. Specifically, robust MDPs with uncertainty set  $\mathcal{U}_{hard} := \{P_\xi \mid F\xi \leq g, \xi \in$   
 836  $[0, 1]^n\}$  where  $P_\xi$  is a specially designed kernel with ladder structure with only action (effectively no  
 837 decision) and a terminal state [35].

838 Note that  $F\xi \leq g$  imposes  $m$ -linear constraints on  $\mathcal{U}_{hard}$  while we allow only one global constraint  
 839 on  $\mathcal{U}_p$ . Observe that  $\mathcal{U}_1 = \{P_\xi \mid \mathbf{1}^\top \xi \leq g, \xi \in [0, 1]^n\}$  is the nearest uncertainty to  $\mathcal{U}_{hard}$  as both  
 840 have polyhedral structure. This restricts the class of the IP programmes to have a number of constraints  
 841  $m = 1$  and the row of  $F$  to be all ones. In other words, only IP programmes that can be reduced to  
 842  $\mathcal{U}_1$  are of the following form: For  $, c \in \mathcal{Z}^n, \zeta \in \mathcal{Z}$ ,

$$\exists x \in \{0, 1\}^n \quad \text{s.t.} \quad \mathbf{1}^\top x \leq g, \quad \text{and} \quad c^\top x \leq \zeta?$$

843 **Solution:**

- 844 • Case 1) If  $g < 0$  then **no**.
- 845 • Case 2) If  $g = 0, \zeta \geq 0$  then **yes** and  $g = 0, \zeta < 0$  then **yes**.

846 • If  $g > 0$  then compute the sum of  $g$  smallest coordinates of  $c$ , and this sum is less/equal than  
 847  $\zeta$  then answer is **yes**, otherwise **no**.

848 Further, for IP to be reducible to robust MDPs, the diameter of the uncertainty ( $\max_{P, P' \in \mathcal{U}_{hard}} \|P - P'\|_1 = 2S$ ) has to be large for the practical settings. Loosly speaking, robust MDPs with a  $\mathcal{U}_p$   
 849 uncertainty have one global constraint and a small radius  $\beta$ , which corresponds to a Knapsack  
 850 Problem with a small budget (IP with one constraint and a small  $g$ ) which are much easier to solve  
 851 [5, 9].  
 852

853 We can thus conclude that the hardness result of [35] doesn't apply to our uncertainty case.

### 854 B.3 Decomposition

855 **Proposition B.2.** *Non-rectangular uncertainty  $\mathcal{U}_p$  can be written as an infinite union of  $sa$ -*  
 856 *rectangular sets  $\mathcal{U}_p^{sa}$ , as*

$$\mathcal{U}_p = \bigcup_{b \in \mathcal{B}} \mathcal{U}_p^{sa}(b),$$

857 where  $\mathcal{B} = \{b \in \mathbb{R}_+^{S \times A} \mid \|b\|_p \leq \beta\}$ . Note that all of them share the nominal kernel  $\hat{P}$ .

858 *Proof.* By definition, we have

$$\mathcal{U}_p = \{P \mid \|P - \hat{P}\|_p \leq \beta, \sum_{s'} P(s'|s, a) = 1\} \quad (3)$$

$$= \{P \mid \sum_{s,a} \|P_{sa} - \hat{P}_{sa}\|_p^p \leq \beta^p, \sum_{s'} P(s'|s, a) = 1\} \quad (4)$$

$$= \{P \mid \sum_{s,a} b_{sa}^p \leq \beta^p, \|P_{sa} - \hat{P}_{sa}\|_p^p = b_{sa}^p, \sum_{s'} P(s'|s, a) = 1\} \quad (5)$$

$$= \{P \mid \sum_{s,a} b_{sa}^p \leq \beta^p, \|P_{sa} - \hat{P}_{sa}\|_p^p \leq b_{sa}^p, \sum_{s'} P(s'|s, a) = 1\} \quad (6)$$

$$= \bigcup_{\sum_{s,a} b_{sa}^p \leq \beta^p} \{P \mid \|P_{sa} - \hat{P}_{sa}\|_p^p \leq b_{sa}^p, \sum_{s'} P(s'|s, a) = 1\} \quad (7)$$

$$= \bigcup_{b \in \mathcal{B}} \mathcal{U}_p^{sa}(b). \quad (8)$$

859 □

## 860 C Additional Results: Robust Policy Improvement

861 In the previous section, we identified that the worst-case kernel can be expressed as a rank-one  
 862 perturbation of the nominal kernel. Leveraging this structure, we developed a method to efficiently  
 863 evaluate the robust policy. This method also computes the perturbation ( $\beta k^\top$ ) and, consequently, the  
 864 worst-case kernel.

865 Using the computed worst kernel, we can directly evaluate the gradient with respect to the policy.  
 866 This enables policy improvement through gradient ascent, as detailed in [32]:

$$\pi_{n+1} = \text{proj} \left[ \pi_n + \eta_k \nabla_{\pi} J_{P_n}^{\pi} \Big|_{\pi=\pi_n} \right], \quad (9)$$

867 where  $P_n$  is the worst-case kernel estimate for the policy  $\pi_k$ . This method guarantees global  
 868 convergence with an iteration complexity of  $O(\epsilon^{-4})$  [32].

869 Alternatively, the policy gradient can be derived for the approximate perturbation, as shown in the  
 870 result below.

871 **Policy Gradient Theorem** Once the worst kernel for a policy is computed using Algorithm 1, the  
 872 policy gradient can be used to update the policy. Alternatively, the following policy gradient theorem  
 873 provides a direct way to compute the gradient:

874 **Lemma C.1** (Approximate Policy Gradient Theorem). *Given a transition kernel  $P = \hat{P} - \beta k^\top$ , the*  
 875 *return is expressed as:*

$$J_P^\pi := J_0^\pi - \gamma \frac{J_\beta^\pi \langle k, v_R^\pi \rangle}{1 + \gamma \langle k, v_\beta^\pi \rangle},$$

876 *and the gradient is given by:*

$$\nabla_\pi J_P^\pi = d^\pi \circ Q_R^\pi - \gamma \frac{k^\top v_R^\pi}{1 + \gamma k^\top v_\beta^\pi} d^\pi \circ Q_\beta^\pi - \gamma \frac{J_\beta^\pi (k^\top D^\pi)}{1 + \gamma k^\top v_\beta^\pi} \circ Q_R^\pi + \gamma^2 \frac{J_\beta^\pi (k^\top v^\pi) (k^\top D^\pi)}{(1 + \gamma k^\top v_\beta^\pi)^2} \circ Q_\beta^\pi.$$

877 *Proof.* The expression for the return follows directly from the inverse matrix theorem, as shown in  
 878 [17]. The gradient is then derived using the policy gradient theorem [28] in the format used in [18].

$$\begin{aligned} \nabla_\pi J_P^\pi &= d^\pi \circ Q_R^\pi - \gamma \frac{k^\top D^\pi R^\pi}{1 + \gamma k^\top D^\pi \beta^\pi} d_\mu^\pi \circ Q_\beta^\pi - \gamma \frac{\mu^\top D \beta^\pi}{1 + \gamma k^\top D^\pi \beta^\pi} d_k^\pi \circ Q_R^\pi \\ &\quad + \gamma^2 \frac{\mu^\top D \beta^\pi k^\top D^\pi R^\pi}{(1 + \gamma k^\top D^\pi \beta^\pi)^2} d_k^\pi \circ Q_\beta^\pi, \\ &= d^\pi \circ Q_R^\pi - \gamma \frac{k^\top v_R^\pi}{1 + \gamma k^\top v_\beta^\pi} d^\pi \circ Q_\beta^\pi - \gamma \frac{J_\beta^\pi (k^\top D^\pi)}{1 + \gamma k^\top v_\beta^\pi} \circ Q_R^\pi + \gamma^2 \frac{J_\beta^\pi (k^\top v^\pi) (k^\top D^\pi)}{(1 + \gamma k^\top v_\beta^\pi)^2} \circ Q_\beta^\pi. \end{aligned}$$

879 □

880 The main advantage of this policy gradient formulation is that terms like  $J_\beta^\pi, v_\beta^\pi, Q_\beta^\pi$ , along with the  
 881 nominal terms  $J_R^\pi, v_R^\pi, Q_R^\pi$ , can be efficiently computed using Bellman operators and bootstrapping  
 882 techniques.

883 **Interpretation of Gradient Terms** The approximate policy gradient reveals the interplay of various  
 884 components in robust MDPs:

- 885 • The first term,  $d^\pi \circ Q_R^\pi$ , represents the nominal policy gradient, emphasizing actions with  
 886 high rewards.
- 887 • The second term,  $\gamma \frac{k^\top v_R^\pi}{1 + \gamma k^\top v_\beta^\pi} d^\pi \circ Q_\beta^\pi$ , discourages policies that place significant weight on  
 888 high-uncertainty Q-values, scaled by the vulnerability to adversarial actions.
- 889 • The last two terms, while more complex to interpret, further reflect the intricate dynamics of  
 890 robust MDPs.

891 **Robust Policy Gradient Algorithm** The robust policy gradient algorithm (Algorithm 4) converges  
 892 to an  $\epsilon$ -optimal policy within  $O(\epsilon^{-8})$  iterations.

893 **Theorem C.2.** *The robust policy gradient method from [32] achieves global convergence within*  
 894  *$O(\epsilon^{-4})$  iterations for the policy gradient step. Algorithm 1 computes the worst-case kernel in  $O(n)$*   
 895 *iterations at step  $n$ . The total iteration complexity for global optimality is  $O(\epsilon^{-8})$ .*

896 Algorithm 4 employs a double-loop structure: the inner loop (Algorithm 1) computes the worst-case  
 897 kernel for a fixed policy, while the outer loop updates the policy using the derived gradient. An  
 898 actor-critic style alternative, where the kernel and policy are updated simultaneously, is left for future  
 899 work.

---

#### Algorithm 4 Robust Policy Gradient Algorithm

---

- 1: **while** not converged:  $n = n + 1$  **do**
  - 2:   Compute the worst-case kernel  $P = \hat{P} - \beta k^\top$  for policy  $\pi$  using Algorithm 1 with tolerance  
     $\epsilon = \gamma^n$ .
  - 3:   Compute the policy gradient  $G$  using Lemma C.1.
  - 4:   Update policy:  $\pi \leftarrow \text{proj}[\pi + \alpha_n G]$ .
  - 5: **end while**
-

900 **Extension to KL Entropy Uncertainty Sets.** For the KL uncertainty case, the worst kernel is  
 901 given by  $P_{\mathcal{U}_{KL}^{sa}}^\pi = (I - \gamma \hat{P}^\pi A^\pi)^{-1}$  where  $A^\pi$  is a diagonal matrix [30]. If we can invert this matrix,  
 902 then its possible to build upon it. We leave this for future work.

## 903 D Helper Results

904 **Proposition D.1** (Sherman–Morrison Formula [4]). *If  $A \in \mathbb{R}^{n \times n}$  invertible matrix, and  $u, v \in \mathbb{R}^n$ ,  
 905 then the matrix  $A + uv^T$  is invertible if and only if  $1 + v^T A^{-1} u \neq 0$ :*

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}.$$

**Proposition D.2.**

$$\sigma_q(v) := \min_{w \in \mathbb{R}} \|v - w\mathbf{1}\|_q = \min_{\|k\|_p \leq 1, \mathbf{1}^T k = 0} k^T v$$

906 *Proof.* Follows directly from Lemma J.1 of [19]. □

907 **Proposition D.3.** *For any vector  $\|x\| = 1$ , we have*

$$\max\{\|Proj_{\mathbb{R}_+^n}(x)\|, \|Proj_{\mathbb{R}_+^n}(-x)\|\} \geq \frac{1}{\sqrt{2}},$$

908 *where  $\mathbb{R}_+^n$  is positive quadrant.*

909 *Proof.* For any vector  $\|x\| = 1$ , we have

$$\|x_+\|^2 + \|x_-\|^2 = \|x\|^2 = 1.$$

910 And  $Proj_{\mathbb{R}_+^n}(x) = x_+$  and  $Proj_{\mathbb{R}_+^n}(-x) = x_-$ , the rest follows. □

911 **Proposition D.4.** *For  $\|k\|_p$  and  $k^T \mathbf{1} = 0$ , we have*

$$1 + \gamma k^T (I - \gamma P^\pi)^{-1} b^\pi \geq 0,$$

912 *for all  $\pi$ ,  $\|b\|_p \leq \beta$ ,  $b \succeq 0$ .*

913 *Proof.* This is true from the Sherman–Morrison formula as  $J_{\hat{P}-bk^T}^\pi$  is finite, hence the denominator  
 914 must be strictly greater than zero. □

## 915 E Dual Formulation

916 **Lemma E.1** (Sa-rectangular Duality). *For the sa-rectangular uncertainty set  $\mathcal{U} = \mathcal{U}_p^{sa}(\beta)$  with  
 917 radius vector  $\beta \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ , the robust return can be written as the following optimization problem,*

$$J_{\mathcal{U}}^\pi = J^\pi - \gamma \max_{\|k\|_p = 1, \mathbf{1}^T k = 0} \frac{\mu^T D^\pi \beta^\pi k^T D^\pi R^\pi}{1 + \gamma k^T D^\pi \beta^\pi},$$

918 *where  $\beta_s^\pi = \sum_a \pi(a|s) \beta_{sa}$ .*

919 *Proof.* From [17], we know that the worst kernel  $P_{\mathcal{U}_p^{sa}(\beta)}^\pi$  for the uncertainty set  $\mathcal{U}_p^{sa}(\beta)$  is a rank  
 920 one-perturbation of  $P$ . In other words,

$$P_{\mathcal{U}_p^{sa}(\beta)}^\pi = P + \beta k^T$$



for some  $k \in \mathbb{R}^S$  satisfying  $\|k\|_p = 1$  and  $1^T k = 0$ . This implies that it is enough to look for rank-one perturbations of the nominal kernel  $\hat{P}$  in order to find the robust return. That is,

$$\begin{aligned}
J_{\mathcal{U}_p^{sa}(\beta)}^\pi &= \min_{P \in \mathcal{U}_p^{sa}(\beta)} J_P^\pi \\
&= \min_{P = \hat{P} + \beta k^T, \|k\|_p = 1, 1^T k = 0} J_P^\pi, \quad (\text{looking only at rank one perturbations}) \\
&= \min_{P = \hat{P} + \beta k^T, \|k\|_p = 1, 1^T k = 0} \mu^T D_P^\pi R^\pi \\
&= \min_{P = \hat{P} + \beta k^T, \|k\|_p = 1, 1^T k = 0} \mu^T (I - \gamma P^\pi)^{-1} R^\pi \\
&= \min_{\|k\|_p = 1, 1^T k = 0} \mu^T \left( I - \gamma(P^\pi + \beta^\pi k^T) \right)^{-1} R^\pi \\
&= J^\pi - \gamma \max_{\|k\|_p = 1, 1^T k = 0} \frac{\mu^T D^\pi \beta^\pi k^T D^\pi R^\pi}{1 + \gamma k^T D^\pi \beta^\pi}.
\end{aligned}$$

923

□

**Lemma E.2** (S-rectangular Duality). *For  $\mathcal{U} = \mathcal{U}_p^s$ , the robust return can be written as the following optimization problem,*

$$J_{\mathcal{U}}^\pi = J^\pi - \gamma \min_{\|\beta\|_p \leq \epsilon, \|k\|_p \leq 1, \langle 1, k \rangle = 0} \frac{\langle d^\pi, \beta^\pi \rangle \langle k, v^\pi \rangle}{1 + \gamma k^T D^\pi \beta^\pi},$$

where  $D^\pi = (I - \gamma P^\pi)^{-1}$ ,  $d^\pi = \mu^T D^\pi$  and  $v^\pi = D^\pi R^\pi$ .

*Proof.*

$$\begin{aligned}
J_{\mathcal{U}_p^s(\beta)}^\pi &= \min_{\|P_s - (P)_{sa}\|_p^p = \beta_s^p, 1^T P_{sa} = 1} J_P^\pi \\
&= \min_{\sum_a \beta_{sa}^p \leq \beta_s^p} \min_{\|P_{sa} - (P)_{sa}\|_p = \beta_{sa}, 1^T P_{sa} = 1} J_P^\pi \\
&= \min_{\sum_a \beta_{sa}^p \leq \beta_s^p} J_{\mathcal{U}_p^{sa}(\beta)}^\pi \\
&= \min_{\sum_a \beta_{sa}^p \leq \beta_s^p} \left[ J^\pi - \gamma \max_{\|k\|_p = 1, 1^T k = 0} \frac{\mu^T D^\pi \beta^\pi k^T D^\pi R^\pi}{1 + \gamma k^T D^\pi \beta^\pi} \right] \\
&= J^\pi - \gamma \max_{\sum_a \beta_{sa}^p \leq \beta_s^p, \|k\|_p = 1, 1^T k = 0} \frac{\mu^T D^\pi \beta^\pi k^T D^\pi R^\pi}{1 + \gamma k^T D^\pi \beta^\pi}.
\end{aligned}$$

927

□

The above result formulates the robust return in terms of nominal values only for the first time. This implies the robust objective can be rewritten in the dual form as :

$$J_{\mathcal{U}_p^s}^* = \max_{D \in \mathcal{D}} \min_{k \in \mathcal{K}, b \in \mathcal{B}} \left[ \mu^T D R^\pi - \gamma \mu^T D b^\pi \frac{k^T D R^\pi}{1 + \gamma k^T D b^\pi} \right]$$

where  $\mathcal{D} = \{(I - \gamma P_0^\pi)^{-1} \mid \pi \in \Pi\}$ ,  $\mathcal{K} = \{k \in \mathbb{R}^S \mid \|k\|_p = 1, 1^T k = 0\}$ , and  $\mathcal{B} = \{b \in \mathbb{R}^{S \times \mathcal{A}} \mid \|b_s\|_p \leq \beta_s\}$ .

Comparing the penalty term from the previous results in [19, 17], the dual formulation can be written as

$$J_{\mathcal{U}_p^s}^* = \max_{D \in \mathcal{D}} \min_{k \in \mathcal{K}} \left[ \mu^T D R^\pi - \gamma \mu^T D \beta^\pi \frac{k^T D R^\pi}{1 + \gamma k^T D \beta^\pi} \right]$$

where  $\beta_s^\pi = \|\pi_s\|_q \beta_s$ .

Surprisingly, the optimization here looks as if it is optimized for the same value of  $\beta_s^\pi = \max_{\sum_a \beta_{sa}^p \leq \beta_s^p} \sum_a \pi(a|s) \beta_{sa} = \beta_s \|\pi_s\|_q$  for all values of feasible  $k$ . This suggest that the adversary payoff is maximized by maximizing the expected uncertainty in the trajectories.

938 **Lemma E.3** (Non-rectangular Duality). For  $\mathcal{U} = \mathcal{U}_p$ , the robust return can be written as the following  
 939 optimization problem

$$J_{\mathcal{U}}^{\pi} = J^{\pi} - \gamma \min_{\|\beta\|_p \leq \epsilon, \|k\|_p \leq 1, \langle 1, k \rangle = 0} \frac{\langle d^{\pi}, \beta^{\pi} \rangle \langle k, v_R^{\pi} \rangle}{1 + \gamma \langle k, v_{\beta}^{\pi} \rangle},$$

940 where  $D^{\pi} = (I - \gamma P^{\pi})^{-1}$ ,  $d^{\pi} = \mu^T D^{\pi}$  and  $v^{\pi} = D^{\pi} R^{\pi}$ .

941 *Proof.* Now,

$$\begin{aligned} J_{\mathcal{U}_p(\epsilon)}^{\pi} &= \min_{\|P - P\|_p^p = \epsilon^p, 1^T P_{sa} = 1} J_P^{\pi} \\ &= \min_{\|\beta\|_p^p \leq \epsilon^p} \min_{\|P_{sa} - (P)_{sa}\|_p = \beta_{sa}, 1^T P_{sa} = 1} J_P^{\pi} \\ &= \min_{\|\beta\|_p^p \leq \epsilon^p} J_{\mathcal{U}_p^{sa}(\beta)}^{\pi} \\ &= \min_{\|\beta\|_p \leq \epsilon} \left[ J^{\pi} - \gamma \max_{\|k\|_p = 1, 1^T k = 0} \frac{\mu^T D^{\pi} \beta^{\pi} k^T D^{\pi} R^{\pi}}{1 + \gamma k^T D^{\pi} \beta^{\pi}} \right] \\ &= J^{\pi} - \gamma \max_{\|\beta\|_p \leq \epsilon, \|k\|_p = 1, 1^T k = 0} \frac{\mu^T D^{\pi} \beta^{\pi} k^T D^{\pi} R^{\pi}}{1 + \gamma k^T D^{\pi} \beta^{\pi}}. \end{aligned}$$

942 □

943 The above result formulates the robust return in terms of nominal values only, for the first time.  
 944 Comparing with the existing result, we get a very interesting relation:

$$\sigma_q(v_{\mathcal{U}}^{\pi}) = \max_{\|k\|_p = 1, 1^T k = 0} \frac{k^T v_R^{\pi}}{1 + \gamma k^T v_{\beta}^{\pi}}, \quad (10)$$

945 where  $v_x^{\pi} = (I - \gamma P^{\pi})^{-1} x^{\pi}$ .

946 The LHS is a robust quantity (variance of the robust return) which is express in the terms of purely  
 947 nominal quantities. This is the simplest of all such relations. We believe that the above relation can  
 948 help in theoretical derivations and experiment design but not exactly sure how yet.

## 949 E.1 Intuition on the Adversary

950 **sa-rectangular case.** We know that the  $\sigma(v_{\mathcal{U}}^{\pi})$  represents the penalty for robustness, expressed as:

$$J_{\mathcal{U}}^{\pi} = J^{\pi} - \gamma \langle d^{\pi}, \beta^{\pi} \rangle \sigma_q(v_{\mathcal{U}}^{\pi}).$$

951 Understanding how  $\sigma(v_{\mathcal{U}}^{\pi})$  arises provides insight into the behavior of the adversary as described in  
 952 (10). Furthermore, if  $P = \hat{P} - \beta k^T$ , then:

$$J_P^{\pi} = J^{\pi} - \langle d^{\pi}, \beta^{\pi} \rangle \frac{k^T v_R^{\pi}}{1 + \gamma k^T v_{\beta}^{\pi}}.$$

953 Here,  $k$  represents the direction in which the adversary discourages perturbations in the kernel. The  
 954 optimal direction  $k$  chosen by the adversary maximizes the objective in (10).

955 **s-rectangular uncertainty sets.** Now, we turn our attention to the coupled uncertainty case.

956 **Lemma E.4.** For  $\mathcal{U} = \mathcal{U}_p^s$ , the robust return can be formulated as the following optimization problem:

$$J_{\mathcal{U}}^{\pi} = J^{\pi} - \gamma \min_{\|\beta\|_p \leq \epsilon, \|k\|_p \leq 1, \langle 1, k \rangle = 0} \frac{\langle d^{\pi}, \beta^{\pi} \rangle \langle k, v^{\pi} \rangle}{1 + \gamma k^{\top} D^{\pi} \beta^{\pi}},$$

957 where  $D^{\pi} = (I - \gamma P^{\pi})^{-1}$ ,  $d^{\pi} = \mu^T D^{\pi}$ , and  $v^{\pi} = D^{\pi} R^{\pi}$ .

958 *Proof.* The proof follows similarly to the sa-rectangular case and is detailed in the appendix. The  
 959 key additional step involves decomposing the s-rectangular uncertainty set  $\mathcal{U}_p^s$  into a union of  
 960 sa-rectangular uncertainty sets  $\mathcal{U}_p^{sa}$ .  $\square$

961 By comparing the penalty term from previous results in [19, 17], we obtain:

$$\sum_s d^\pi(s) \|\pi_s\|_q \sigma_q(v_\pi^\pi) = \max_{\sum_a \beta_{sa}^p \leq \beta_s^p, \|k\|_p=1, 1^T k=0} \frac{(d^\pi \beta^\pi)(k^T v^\pi)}{1 + \gamma k^T D^\pi \beta^\pi}.$$

962 This relation is interesting as it connects the robust term on the left-hand side (LHS) with the  
 963 non-robust terms on the right-hand side (RHS).

964 Interestingly, the optimization here suggests that the adversary maximizes the expected uncertainty  
 965 in trajectories, as the same value of  $\beta_s^\pi = \max_{\sum_a \beta_{sa}^p \leq \beta_s^p} \sum_a \pi(a|s) \beta_{sa} = \beta_s \|\pi_s\|_q$  appears for all  
 966 feasible  $k$ .

## 967 F Robust Policy Evaluation

968 **Proposition F.1.** For  $\lambda^* = \max_{x \in C} \frac{g(x)}{h(x)}$ ,  $F(\lambda) := \max_{x \in C} (g(x) - \lambda h(x))$ , we have  
 969  $F(\lambda^*) = 0$  and  $f(\lambda) \geq 0 \iff \lambda^* \geq \lambda$ .

970 *Proof.* • If  $F(\lambda) \geq 0$  then

$$\begin{aligned} & \exists x \text{ s.t. } g(x) - \lambda h(x) \geq 0 \\ \implies & \exists x \text{ s.t. } \frac{g(x)}{h(x)} \geq \lambda, \quad (\text{as } h(x) > 0 \text{ for all } x) \\ \implies & \max_{x \in C} \frac{g(x)}{h(x)} \geq \lambda. \end{aligned}$$

971 • If  $F(\lambda) \leq 0$  then

$$\begin{aligned} & g(x) - \lambda h(x) \leq 0, \quad \forall x \in C \\ \implies & \frac{g(x)}{h(x)} \leq \lambda, \quad \forall x \in C, \quad (\text{as } h(x) > 0) \\ \implies & \max_{x \in C} \frac{g(x)}{h(x)} \leq \lambda \end{aligned}$$

972 • If  $F(\lambda) = 0$  then  $\lambda = \max_{x \in C} \frac{g(x)}{h(x)}$  implied from the above two items.

973  $\square$

974 **Lemma F.2.** The robust return can be expressed as

$$J_{\mathcal{U}_p}^\pi = J^\pi - \lambda^*,$$

975 where the penalty  $\lambda^*$  is a fixed point of  $F(\lambda)$ . Furthermore,  $\lambda^*$  can be found via binary  
 976 search as  $F(\lambda) > 0$  if and only if  $\lambda > \lambda^*$ , where  $F(\lambda) = \max_{b \in \mathcal{B}} \|E^\pi b\|_q$ ,  $E^\pi = \gamma \left( I - \right.$   
 977  $\left. \frac{\mathbf{1}\mathbf{1}^\top}{S} \right) \left[ D^\pi R^\pi \mu^\top D^\pi - \lambda D^\pi \right] H^\pi$ , and  $H^\pi R := R^\pi$ .

978 *Proof.* We want to evaluate the following

$$\lambda^* := \max_{b \in \mathcal{B}, k \in \mathcal{K}} \gamma \frac{k^T D^\pi R^\pi \mu^T D^\pi b^\pi}{1 + \gamma k^T D^\pi b^\pi}.$$

979 This is of the form  $\max_x \frac{f(x)}{g(x)}$ . Then according to Proposition F.1, we have  $f(\lambda^*) = 0$  and  $f(\lambda) > 0$   
 980 if and only if  $\lambda^* > \lambda$ , where

$$\begin{aligned}
f(\lambda) &:= \max_{b \in \mathcal{B}, k \in \mathcal{K}} [\gamma k^T A^\pi b^\pi - \lambda(1 + \gamma k^T D^\pi b^\pi)] \\
&= \max_{b \in \mathcal{B}, k \in \mathcal{K}} k^\top C^\pi b - \lambda, \\
&= \max_{b \in \mathcal{B}, \|k\|_p \leq 1} k^\top \left( I - \frac{\mathbf{1}\mathbf{1}^T}{S} \right) C^\pi b - \lambda, \quad (\text{from Proposition G.2}) \\
&= \max_{b \in \mathcal{B}} \left\| \left( I - \frac{\mathbf{1}\mathbf{1}^T}{S} \right) C^\pi b \right\|_q - \lambda, \quad (\text{Holder's inequality})
\end{aligned}$$

981 where  $A^\pi = D^\pi R^\pi \mu^T D^\pi$ ,  $C^\pi := \gamma \left( A^\pi - \lambda D^\pi \right) H^\pi$ .

982

□

## 983 G Evaluation of $\max_{x,y} xAy$

984 Algorithm 1 requires an oracle access to

$$\max_{\|b\|_p \leq \beta, \|k\|_p \leq 1, \mathbf{1}^T k = 0} k^T Ab,$$

985 where  $k \in \mathbb{R}^S$ ,  $b \in \mathbb{R}^{S \cdot A}$  and  $p \geq 1$ . The above is a bilinear problem, which is NP-Hard, but we have  
986 a very useful structure on domain set ( $L_p$  bounded set).

987 **Proposition G.1.** [Orthogonality Equivalence] Let  $\mathcal{K} = \{k \mid \|k\|_p \leq 1, \mathbf{1}^T k = 0\}$ , and  $\mathcal{W} =$   
988  $\{k^T (I - \frac{\mathbf{1}\mathbf{1}^T}{S}) \mid \|k\|_p \leq 1\}$ . Then we have,

$$\mathcal{K} = \mathcal{W}.$$

989 *Proof.* Now let  $k \in \mathcal{K}$ , then  $k^T (I - \frac{\mathbf{1}\mathbf{1}^T}{S}) = k^\top \in \mathcal{W}$ . Now the other direction, let  $k \in \mathcal{W}$ ,  
990 then  $\langle k^T (I - \frac{\mathbf{1}\mathbf{1}^T}{S}), \mathbf{1} \rangle = 0$  by construction and  $\|k^T (I - \frac{\mathbf{1}\mathbf{1}^T}{S})\|_p \leq \|k\|_p \leq 1$ , this implies  
991  $k^T (I - \frac{\mathbf{1}\mathbf{1}^T}{S}) \in \mathcal{K}$ . □

992 The above result implies that

$$\begin{aligned}
\max_{\|b\|_p \leq \beta, \|k\|_p \leq 1, \mathbf{1}^T k = 0} k^T Ab &= \max_{\|b\|_p \leq \beta, k \in \mathcal{K}} k^T Ab \\
&= \max_{\|b\|_p \leq \beta, k \in \mathcal{W}} k^T Ab, \quad (\text{as } \mathcal{K} = \mathcal{W} \text{ from above Proposition G.1}) \\
&= \max_{\|b\|_p \leq \beta, \|k\|_p = 1} k^\top \left( I - \frac{\mathbf{1}\mathbf{1}^T}{S} \right) Ab, \quad (\text{def. of } \mathcal{W}).
\end{aligned}$$

993 Further, we have equivalence of optimizers

$$\arg \max_{\|k\|_p \leq 1, \mathbf{1}^T k = 0, \|b\|_p \leq \beta} k^T Ab = \left\{ (b^*, (I - \frac{\mathbf{1}\mathbf{1}^T}{S})k^*) \mid (b^*, k^*) \in \arg \max_{\|k\|_p = 1, \|b\|_p \leq \beta} k^\top (I - \frac{\mathbf{1}\mathbf{1}^T}{S}) Ab \right\}.$$

994 **Proposition G.2.** The solving of

$$\max_{\|k\|_p \leq 1, \mathbf{1}^T k = 0, \|b\|_p \leq \beta} k^T Ab, \quad \text{is equivalent to} \quad \max_{\|k\|_p = 1, \|b\|_p \leq \beta} k^\top \left( I - \frac{\mathbf{1}\mathbf{1}^T}{S} \right) Ab.$$

995 *Proof.* Directly follows from the proposition above. □

996 **G.1 Eigenvalue Approach (Spectral Methods)**

997 This section focus on deriving a spectral method for solving the optimization problem:

$$\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2,$$

998 where  $A \in \mathbb{R}^{n \times n}$ . Compute  $A^\top A$ . We perform eigenvalue decomposition of  $A^\top A$ :

$$A^\top A = V\Lambda V^\top,$$

999 where  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  (eigenvalues) and  $V = [v_1, v_2, \dots, v_n]$  (eigenvectors). Further,  
1000 WLOG

$$\lambda_1 \geq \lambda_2 \geq \dots, \quad \text{and} \quad \|v_{+i}\| \geq \|v_{-i}\| \quad \forall i, \quad u_i := \frac{v_i^+}{\|v_i^+\|}$$

1001 where  $v_i^+ = \max(v_i, 0)$ ,  $v_i^- = -\min(v_i, 0)$  denotes positive and negative parts respectively.

1002 • Zero Order Solution:

$$f_0 = \|Au_1\|.$$

1003 • First order solution:

$$f_1 = \max_i \|Au_i\|.$$

1004 • Second order solution:

$$f_2 = \max_{i,j} \max_{t \in [0,1]} \|A \frac{(tv_i + (1-t)v_j)^+}{\|(tv_i + (1-t)v_j)^+\|}\|.$$

1005 • Third order solution:

$$f_3 = \max_{i,j,k} \max_{r,s,t \in [0,1], r+s+t=1} \|A \frac{(rv_i + sv_j + tv_k)^+}{\|(rv_i + sv_j + tv_k)^+\|}\|.$$

1006 Upper bounds on  $\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2$ :

1007 • Zero order upper bound:  $\lambda_1$

1008 • First order upper bound:  $\sqrt{\sum_i \lambda_i c_i}$ , where  $c_i =$   
1009 
$$\begin{cases} \langle v_i, u_i \rangle^2, & \text{if } \sum_{j=1}^i \langle v_j, u_j \rangle^2 \leq 1 \\ 1 - \sum_{j=1}^{i-1} \langle v_j, u_j \rangle^2, & \text{if } \sum_{j=1}^i \langle v_j, u_j \rangle^2 \geq 1, \sum_{j=1}^{i-1} \langle v_j, u_j \rangle^2 \leq 1. \\ 0 & \text{otherwise} \end{cases}$$

1010 **Lemma G.3** (Zero Order Approximation). *The highest projected eigenvector  $u = \frac{v_1^+}{\|v_1^+\|}$  is at least a*  
1011 *half-good solution, i.e.,*

$$\|Au\|_2^2 \geq \frac{\lambda_1}{2} \geq \frac{1}{2} \max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2^2.$$

1012 Further, if  $A$  is rank-one then it is exact, i.e.,

$$\|Au\|_2 = \max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2.$$

1013 *Proof.* We have  $\|v_1^+\| \geq \frac{1}{\sqrt{2}}$  from Proposition D.3. Let  $u = \frac{(v_1)_+}{\|(v_1)_+\|} = \sum_i \sigma_i v_i$ , where  $\sigma_i = \langle u, v_i \rangle$ ,  
 1014 we have

$$\begin{aligned}
 u^T A^T A u &= \left( \sum_i \sigma_i v_i \right) \left( \sum_i \lambda_i v_i v_i^T \right) \left( \sum_i \sigma_i v_i \right) \\
 &= \sum_i \lambda_i \sigma_i^2, \quad (\text{as } v_i \text{ are orthogonal}) \\
 &= \lambda_1 \sigma_1^2 + \sum_{i \neq 1} \lambda_i \sigma_i^2, \\
 &\geq \lambda_1 \sigma_1^2 + \sum_{i \neq 1} \lambda_n \sigma_i^2, \quad (\text{as } \lambda_2 \geq \lambda_3, \dots) \\
 &= \lambda_1 \sigma_1^2 + \lambda_n (1 - \sigma_1^2), \quad (\text{as } \sum_i \sigma_i^2 = 1) \\
 &\geq \frac{1}{2} (\lambda_1 + \lambda_n), \quad (\text{as } \sigma_1 \geq \frac{1}{\sqrt{2}}).
 \end{aligned}$$

1015 Rest follows.

1016

□

**Proposition G.4** (First Order is Better than the First).

$$\|Au_j\|_2^2 \geq \max_i \lambda_i \sigma_i^2 \geq \frac{\lambda_1}{2}$$

1017 where  $j \in \arg \max_i \lambda_i \langle v_i, u_i \rangle$  and  $\sigma_i = \langle v_i, u_i \rangle \geq \frac{1}{\sqrt{2}}$ .

1018 *Proof.* Let  $u_j = \frac{(v_j)_+}{\|(v_j)_+\|} = \sum_i \sigma_i^j v_i$ , where  $\sigma_i^j = \langle u_j, v_i \rangle$ , we have

$$\begin{aligned}
 u_j^T A^T A u_j &= \left( \sum_i \sigma_i^j v_i \right) \left( \sum_i \lambda_i v_i v_i^T \right) \left( \sum_i \sigma_i^j v_i \right) \\
 &= \sum_i \lambda_i (\sigma_i^j)^2, \quad (\text{as } v_i \text{ are orthogonal}), \\
 &\geq \lambda_j (\sigma_j^j)^2, \\
 &= \max_i \lambda_i (\sigma_i^j)^2, \quad (\text{by definition of } j).
 \end{aligned}$$

1019 Rest follows.

1020

□

1021 **Proposition G.5.** Second order solution  $f_2 = \max_{i,j} \max_{t \in [0,1]} \|A \frac{(tv_i + (1-t)v_j)_+}{\|(tv_i + (1-t)v_j)_+\|}\|$  is exactly  
 1022 equal to  $\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2$  when  $A$  is rank two.

1023 This approach is computationally efficient but may not always yield the exact solution, especially  
 1024 when multiple eigenvectors significantly contribute to the optimal  $x$ .

1025 The intuition behind this approach is that the matrix  $A^T A$  can be decomposed into its eigenvalues  
 1026 and eigenvectors, representing the principal directions of the transformation applied by  $A$ . The  
 1027 eigenvector corresponding to the largest eigenvalue provides the direction of maximum scaling for  
 1028  $A$ . However, since the solution is constrained to the nonnegative orthant ( $x \geq 0$ ), we adjust the  
 1029 eigenvectors by only considering their positive parts. The method identifies an approximate solution  
 1030  $u_j$  by selecting and normalizing the positive part of the eigenvector that contributes the most to the  
 1031 objective function.

---

**Algorithm 5** Second Order Spectral Approximation for  $\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2$ 

---

1: Normalize the positive part:

$$u_i = \frac{v_i^+}{\|v_i^+\|_2}.$$

2: Compute scores for all eigenvectors:

$$\text{Score}_i = \lambda_i \langle v_i, u_i \rangle.$$

3: Select  $j = \arg \max_i \text{Score}_i$ .

4: **Output:** Approximate solution  $u_j = v_j^+ / \|v_j^+\|_2$  and approximate maximum value  $\|Au_j\|_2$ .

---

## Notes

- This approach is effective when the largest eigenvalue  $s_1$  dominates the others. It approximates the solution by leveraging the spectral properties of  $A^\top A$ .
- The result might not be exact if multiple eigenvalues contribute significantly, as the approach considers only the contribution of individual eigenvectors.

## G.2 Experimental Verification

This section describes three different methods for solving the optimization problem:

$$\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2,$$

where  $A \in \mathbb{R}^{n \times n}$ . The methods are compared in terms of their computational efficiency and the quality of their solutions.

### G.2.1 Brute Force Random Search

The brute force method randomly samples vectors  $x \in \mathbb{R}^n$  from the nonnegative orthant, normalizes them to satisfy  $\|x\|_2 = 1$ , and evaluates  $\|Ax\|_2$  for each sampled vector. The steps are as follows:

1. Generate  $N$  random vectors  $x_i \geq 0, i = 1, \dots, N$ .
2. Normalize each vector to unit norm:  $x_i \leftarrow x_i / \|x_i\|_2$ .
3. Compute  $\|Ax_i\|_2$  for each vector and select the maximum value.

This method is simple to implement but computationally expensive, as it evaluates  $A$  for a large number of randomly generated vectors. See figure 5

### G.2.2 Numerical Optimization (Scipy Minimize)

This approach uses numerical optimization to directly solve the problem:

$$\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2.$$

The optimization problem is formulated as:

$$\min_x -\|Ax\|_2, \quad \text{subject to } \|x\|_2 \leq 1 \text{ and } x \geq 0.$$

Steps include:

1. Define the objective function as  $-\|Ax\|_2$ .
2. Impose constraints:  $\|x\|_2 \leq 1$  and  $x \geq 0$ .
3. Solve the problem using `scipy.optimize.minimize`, with an initial guess  $x_0$ .

This method provides the exact solution but is computationally more expensive than the spectral method.

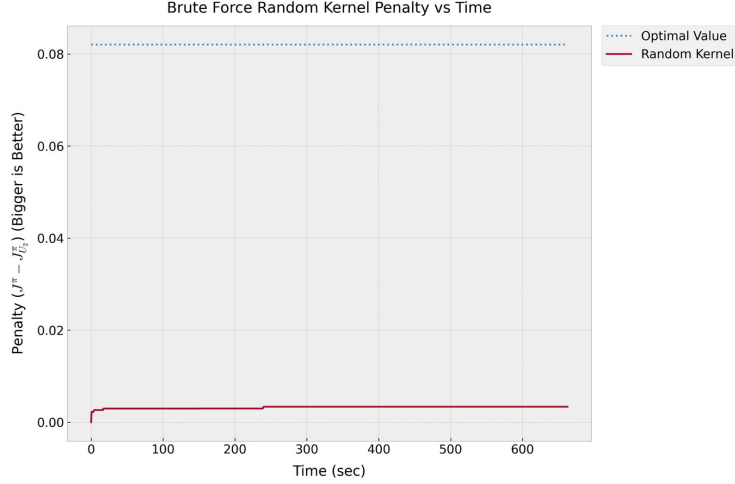


Figure 5: Random Kernel Guess takes exponentially long time to converge. While Algorithm 1 only took 0.14 sec to find the optimal value.

### G.3 Comparison Metrics

The three methods are compared based on:

- **Optimality:** The maximum value  $\|Ax\|_2$  achieved by each method.
- **Time Efficiency:** The computational time required by each method.

### G.4 Results and Observations

The following plots compare the performance of the three methods:

- **Optimality Plot:** Shows that the maximum value obtained with `scipy.minimize` is slightly better than our spectral method, while random search performs poorly.
- **Time Efficiency Plot:** Illustrates that `scipy.minimize` scales much poorly with the dimension, while our spectral method is way faster than both methods.

	Optimal values attained			Time taken		
$n$	Random	Spectral	minimize	Random	Spectral	minimize
10	4.10	4.45	4.46	0.12	0.0007	0.005
20	5.14	6.71	6.82	0.19	0.0003	0.01
50	9.23	11.59	11.93	0.25	0.0007	0.03
100	11.95	16.44	17.19	0.31	0.001	0.28
200	15.74	22.1	23.68	0.44	0.004	2.1
300	19.32	28.58	29.73	0.57	0.012	8.19
500	24.46	36.56	38.47	0.83	0.209	43.49
1000	33.91	51.64	54.25	1.38	0.171	313.6

Table 3: Attained Values and Time Taken.

#### G.4.1 Parameters of Experiments

The experiments were conducted to evaluate the performance of three methods—brute force random search, eigenvalue heuristic, and numerical optimization—on solving the problem:

$$\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2.$$



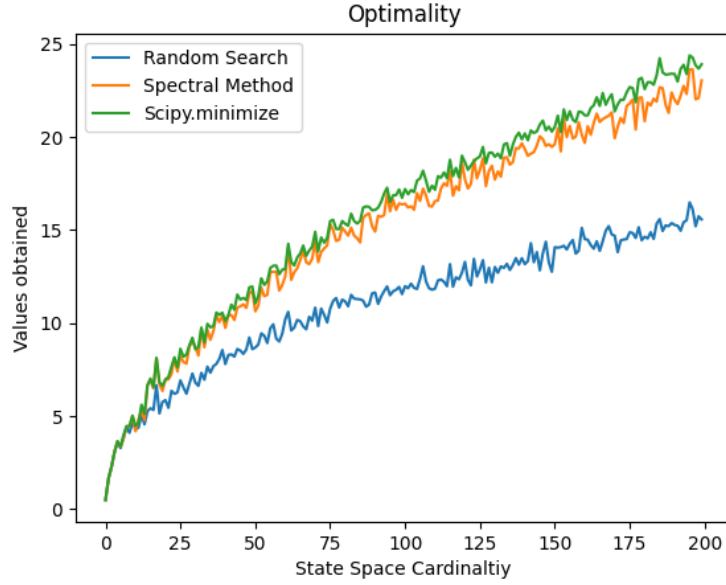


Figure 6: Comparison of optimality across methods.

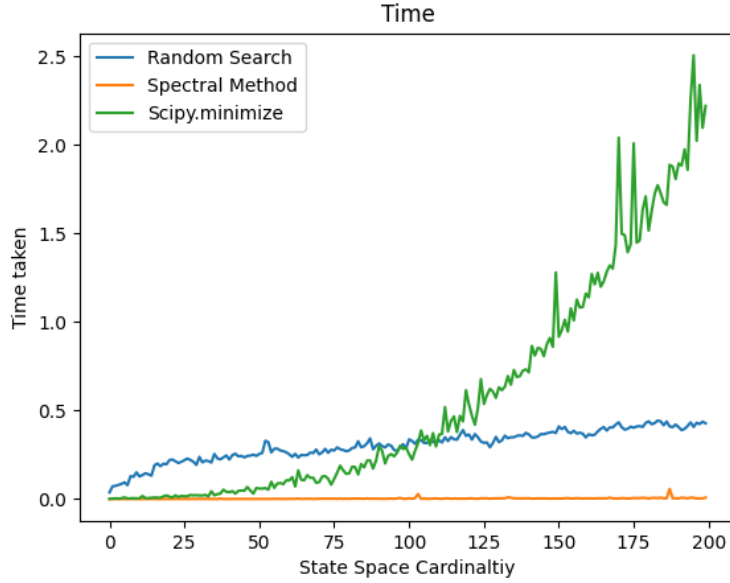


Figure 7: Comparison of computational time across methods

#### 1071 State Space Cardinality and Random matrix Generation

1072 • **State Space Cardinality** ( $n$ ): The dimension of the problem, denoted by  $n$ , represents the  
 1073 state space cardinality. In the experiments,  $n$  varied from 1 to 300 to analyze the scalability  
 1074 of the methods.

1075 • **Matrix Generation:** The matrix  $A \in \mathbb{R}^{n \times n}$  was generated as a random matrix with entries  
 1076 sampled from a standard normal distribution:

$$A_{ij} \sim \mathcal{N}(0, 1), \quad i, j = 1, \dots, n.$$

1077 The same random seed (seed = 42) was used across all runs to ensure reproducibility.

1078 • 10000 random vectors  $x$  were generated for Brute Search Method.

1079 **Process of matrix Evaluation** The goal of the experiments is to maximize  $\|Ax\|_2$  under the  
1080 constraints  $\|x\|_2 \leq 1$  and  $x \geq 0$ . The matrix  $A$  is evaluated by:

- 1081 1. Generating random vectors  $x \in \mathbb{R}^n$  for the brute force method.
- 1082 2. Computing the spectral decomposition of  $A^\top A$  for the eigenvalue heuristic.
- 1083 3. Defining and solving a constrained optimization problem for the numerical optimization  
1084 method.

1085 The results, including the optimal values and computational times, are recorded for each method.

1086 **Evaluation Metrics** The performance of the methods was assessed using the following metrics:

- 1087 • **Optimality:** The maximum value  $\|Ax\|_2$  obtained by each method.
- 1088 • **Computational Efficiency:** The time taken by each method to compute the result.
- 1089 • **Scalability:** The behavior of the methods as  $n$  increases.

1090 This systematic evaluation ensures a fair comparison of the three approaches across varying problem  
1091 sizes.

1092 **Hardware and Software Specifications** The experiments were conducted on the following hard-  
1093 ware and software setup:

- 1094 • **Model Name:** MacBook Pro (2023 model).
- 1095 • **Model Identifier:** Mac14,7.
- 1096 • **Chip:** Apple M2 with 8 cores (4 performance and 4 efficiency cores).
- 1097 • **Memory:** 16 GB Unified Memory.
- 1098 • **Operating System:** macOS Ventura.
- 1099 • **Programming Language:** Python 3.9.
- 1100 • **Libraries Used:**
  - 1101 – numpy for numerical computations.
  - 1102 – scipy for numerical optimization.
  - 1103 – matplotlib for generating plots.
  - 1104 – time for recording computational times.

1105 The experiments were designed to ensure reproducibility by fixing the random seed (`seed = 42`).  
1106 Computational times and results are specific to the above hardware configuration and may vary on  
1107 different systems.

## 1108 H Convexity of $\mathcal{D}$

### 1109 H.1 MDP Configuration

1110 We define an MDP with the following parameters:

- 1111 • **State space size:**  $S = 3$
- 1112 • **Action space size:**  $A = 2$
- 1113 • **Discount factor:**  $\gamma = 0.9$
- 1114 • Random kernel  $P$ , random reward  $R$ , seed 42.
- 1115 • Compute the set  $\mathcal{D} = \{D^\pi H^\pi | \pi\}$  with 10 millions random policies  $\pi$

## 1116 H.2 Dimensionality Reduction via PCA

1117 Given the high-dimensional nature of the  $D^\pi H^\pi$  representations, we apply **Principal Component**  
1118 **Analysis (PCA)** to extract meaningful structure.

- 1119 • We **retain the top 10 components** to capture the dominant variations in the dataset.
- 1120 • The **explained variance ratio** is visualized to assess how much information each component
- 1121 retains.
- 1122 • **2D projections** of the first few principal components are generated for visualization.

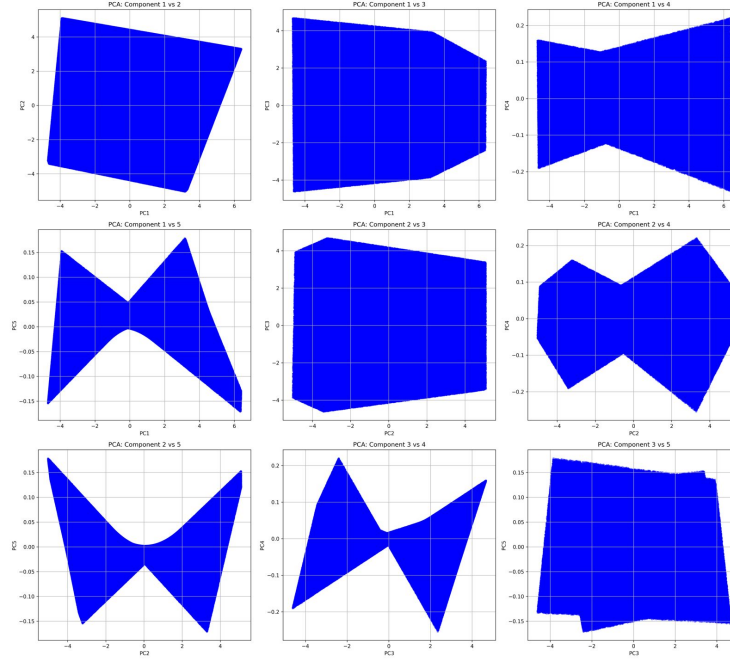


Figure 8: 2D PCA projections of the first 5 components.

## 1123 H.3 Random Linear Projections

1124 To further explore the **geometry of the occupancy measure set**, we apply **random linear projections**  
1125 of the high-dimensional data:

- 1126 • **2D Random Projections:** The data is projected onto **randomly chosen 2D subspaces**.

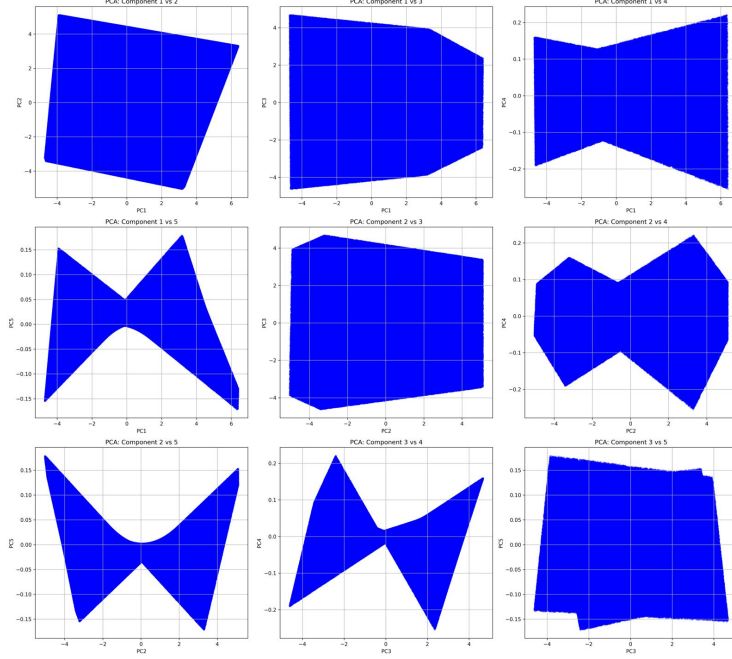


Figure 9: 2D Random Projections of the Data.

## I Experimental Evaluation: Single MDP Comparison

To assess the performance of our proposed binary search algorithm for robust policy evaluation under  $L_2$ -norm bounded uncertainty, we conduct a series of experiments comparing it against existing methods on fixed Markov Decision Process (MDP) instances. The primary objective is to evaluate convergence speed, accuracy relative to an estimated worst-case value, and consistency across different problem configurations. More details of these experiments along with others can be found in the appendix, and codes are available at <https://anonymous.4open.science/r/Kernel-Robust-RL-B742/>

### I.1 Experimental Setup

**Algorithms Compared** We evaluate the following algorithms:

1. **Our Method:** The binary search algorithm presented in this work, which leverages a spectral method for computing the key bisection function  $F(\lambda)$ .
2. **CPI (Frank-Wolfe):** The Conservative Policy Iteration algorithm adapted from [20] for general robust policy evaluation.
3. **SA-Rectangular  $L_2$  VI:** Robust Value Iteration for (s,a)-rectangular  $L_2$  uncertainty, a common baseline representing a structured relaxation.
4. **S-Rectangular  $L_2$  VI:** Robust Value Iteration for (s)-rectangular  $L_2$  uncertainty, another structured relaxation.

**Benchmark Generation** For each MDP instance and policy, we establish an empirical benchmark for the worst-case robust value. This is achieved by sampling 1,000 transition kernels from the  $L_2$  ball of radius  $\beta$  centered at the nominal kernel  $P_{\text{nominal}}$ . Each sampled kernel is projected to ensure it remains a valid stochastic matrix and stays within the  $L_2$  ball. The policy  $\pi$  is evaluated for each sampled kernel, and the minimum value obtained across these samples,  $V_{\text{benchmark}}^{\min}$ , serves as our reference robust value.

**MDP and Policy Configuration** Experiments are conducted on randomly generated MDPs. For each trial, a nominal transition kernel, a reward function, and a uniform initial state distribution  $\mu$  are

generated. A fixed, randomly generated stochastic policy  $\pi$  is then used for robust policy evaluation by all algorithms.

**Experimental Configurations** Two main sets of single MDP comparisons are performed:

1. Varying State Space ( $S$ ):  $S \in \{10, 50, 100, 200\}$ , with actions  $A = 10$  and uncertainty radius  $\beta = 0.01$ .
2. Varying Uncertainty Radius ( $\beta$ ):  $\beta \in \{0.005, 0.01, 0.05, 0.1\}$ , with state space  $S = 100$  and actions  $A = 10$ .

The discount factor is  $\gamma = 0.9$ . Algorithms are run until convergence (tolerance of  $10^{-6}$ ) or a maximum iteration limit (100).

## I.2 Results and Discussion

Figures 10 and 11 present the convergence behavior of the evaluated algorithms on representative MDP instances for the varying state space and varying uncertainty radius configurations, respectively. Each subplot shows the estimated robust value versus algorithm iterations. The horizontal dashed line indicates  $V_{\text{benchmark}}^{\min}$ . An algorithm's final point is marked with a star ( $\star$ ) if its estimated robust value converges to within  $10^{-6}$  of  $V_{\text{benchmark}}^{\min}$ .

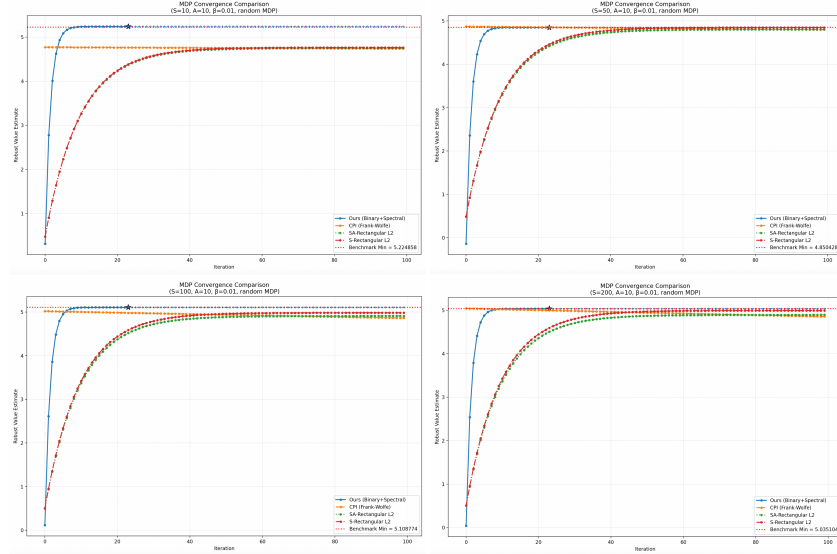


Figure 10: Convergence of robust policy evaluation algorithms for varying state space sizes ( $S$ ). Algorithms whose final value is within  $10^{-6}$  of the benchmark are marked with a star ( $\star$ )

## Observations

- **Convergence Speed and Accuracy of Our Method:** Across all tested configurations, Our Method consistently demonstrates superior performance. It generally converges in fewer iterations and achieves a final robust value remarkably close to  $V_{\text{benchmark}}^{\min}$ , as frequently indicated by the star marker. This suggests efficient and accurate identification of the robust penalty  $\lambda^*$ .
- **CPI Performance:** The CPI algorithm typically converges but often settles at a value slightly higher (less pessimistic) than  $V_{\text{benchmark}}^{\min}$ . While providing a robust estimate, its subproblem, in the version tested, explores extreme points of the set of all stochastic kernels, which may not always precisely align with the worst-case kernel strictly within the  $L_2$  ball.
- **Rectangular Relaxations:** Both sa-rectangular and s-rectangular  $L_2$  VI methods consistently converge to robust values significantly lower than those found by Our Method, CPI,

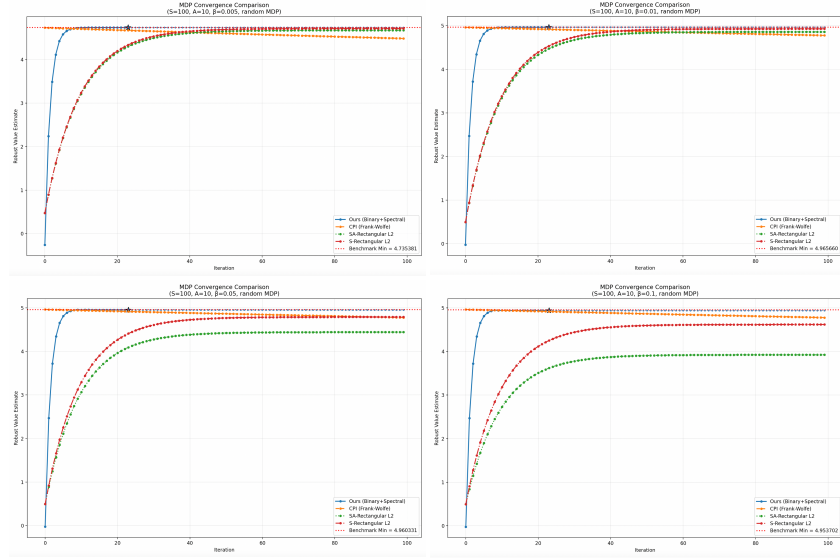


Figure 11: Convergence of robust policy evaluation algorithms for varying uncertainty radius ( $\beta$ ). Algorithms whose final value is within  $10^{-6}$  of the benchmark are marked with a star ( $\star$ )

- 1180 and  $V_{\text{benchmark}}^{\min}$ . This highlights the conservatism inherent in rectangular relaxations when  
 1181 dealing with non-rectangular uncertainty.
- 1182 • **Consistency Across Setups:** The advantages of Our Method in terms of faster and more ac-  
 1183 curate convergence are maintained robustly across different state space sizes and uncertainty  
 1184 radius.