

Dual Formulation for Non-Rectangular L_p Robust Markov Decision Processes

Navdeep Kumar¹, Adarsh Gupta⁴, Maxence Mohamed Elfatihi², Giorgia Ramponi³, Kfir Yehuda Levy¹, Shie Mannor^{1,5}

¹Technion

²École Polytechnique

³University of Zurich

⁴Finsyth AI

⁵NVIDIA Research

Abstract

We study robust Markov decision processes (RMDPs) with non-rectangular uncertainty sets, which capture interdependencies across states unlike traditional rectangular models. While non-rectangular robust policy evaluation is generally NP-hard, even in approximation, we identify a powerful class of L_p -bounded uncertainty sets that avoid these complexity barriers due to their structural simplicity. We further show that this class can be decomposed into infinitely many **sa**-rectangular L_p -bounded sets and leverage its structural properties to derive a novel dual formulation for L_p RMDPs. This formulation provides key insights into the adversary’s strategy and enables the development of the first robust policy evaluation algorithms for non-rectangular RMDPs. Empirical results demonstrate that our approach significantly outperforms brute-force methods, establishing a promising foundation for future investigation into non-rectangular robust MDPs.

1 Introduction

Robust Markov Decision Processes (MDPs) provide a framework for developing solutions that are more resilient to uncertain environmental parameters compared to standard MDPs [1, 2, 3, 4, 5]. This approach is particularly critical in high-stakes domains, such as robotics, finance, healthcare, and autonomous driving, where catastrophic failures can have severe consequences. The study of robust MDPs is further motivated by their potential to offer superior generalization capabilities over non-robust methods [6, 7, 8].

Robust solutions are highly desirable, but obtaining them is a challenging task. In particular, robust policy evaluation has been shown to be strongly NP-hard [9] for general convex uncertainty sets. Consequently, much of the existing work makes rectangularity assumptions, with the most common being s -rectangular uncertainty sets and its special case **sa**-rectangular uncertainty sets [10, 11, 9, 12, 13, 4, 5, 14, 15, 16, 17, 18, 19, 20, 21, 22].

The **s**-rectangularity assumption simplifies the modeling of uncertainty by treating it as independent across states [9]. This assumption is analytically appealing due to the existence of contractive robust Bellman operators, which facilitate computational tractability [9]. However, real world uncertainties are often coupled across the states, and modelling it with **s**-rectangular uncertainty sets, can overly introduce conservatism in solutions (as illustrated in Figure 1 of [23]). In other words, the ratio between the volume of a real world coupled uncertainty set and the smallest **s**-rectangular uncertainty containing it, can be exponential in the state space (as the ratio of volumes of a n-dimension sphere and a n-dimension cube containing it is $O(2^n)$ [24]). In contrast, non-rectangular RMDPs capture much better these interdependencies but lack the existence of any contractive robust Bellman operators, which makes the problem unwieldy [23].

To the best of our knowledge, research on non-rectangular robust MDPs remains limited [23, 17]. While [23] explored robust policy evaluation for non-rectangular uncertainty sets, their work was confined to reward uncertainties. In contrast, this paper addresses kernel uncertainty, which presents a significantly greater level of complexity compared to reward uncertainty. Additionally, [17] demonstrated the convergence of robust policy gradient methods with an iteration complexity of $O(\epsilon^{-4})$ for all types of uncertainty sets, including non-rectangular ones. However, their method depends on oracle access to the robust gradient (worst-case kernel). As a result, robust policy evaluation under non-rectangular kernel uncertainties remains an unresolved challenge.

Further, dual formulation for non-robust MDPs [25] has played a significant role in advancing the field. Unfortunately, no such formulation exists for the robust MDPs.

The key insight of this work is decomposing minimization over a nonrectangular L_p norm-bounded uncertainty sets into minimization over a union of **sa**-rectangular L_p -norm bounded uncertainty sets. For each minimization over rectangular uncertainty set, we have the robust return in close form. Now, we minimize this expression over the set of possible all **sa**-rectangular L_p -norm bounded uncertainty sets that makes up the original nonrectangular set. Using the fact that worst kernel is rank-one perturbation of the nominal kernel in **sa**-rectangular robust MDPs, we derive dual formulation for the L_p robust MDPs. This reveals a very interesting insights over how the adversary chooses the worst kernel. Further, this dual formulation inspires a method for evaluation of robust policy.

Contributions.

- We show that the general NP-hardness result for policy evaluation in non-rectangular RMDPs does not apply to L_p -bounded robust MDPs.
- We derive a novel dual formulation for L_p -RMDPs, providing key insights into the adversary’s strategy and enabling the development of first robust policy evaluation algorithms.
- We experimentally validate our proposed algorithms, demonstrating significant improvements over existing brute-force methods.

This work opens up the avenue for the further investigation of non-rectangular RMDPs otherwise believed too hard.

1.1 Related Work

To the best of our knowledge, there exists no work on non-rectangular robust MDPs with kernel uncertainty. This work is the first to propose an efficient method for robust policy evaluation for a very useful class of uncertainty sets, otherwise thought to be NP-Hard [9].

Rectangular Robust MDPs. In literature, **s**-rectangular uncertainty is a very old assumption [5, 4]. [9] 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 [14, 17]. 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 [18, 15, 19].

There exists many policy gradient based methods for robust MDPs, relying upon contractive robust Bellman operators for the robust policy evaluation [16, 20].

Further, [21, 22] tried to tweak the process, and directly get samples from the adversarial model via pessimistic sampling.

There exist other notions of rectangularity such as k-rectangularity [10] and r-rectangularity [11] which are sparsely studied. However, [26] shows, the 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 [9], in general. For a very specific case, where uncertainty is limited only to reward uncertainty bounded with L_p norm, [23] proposed robust policy evaluation via frequency (occupation measure) regularization, and derived the policy gradient for policy improvement.

Convergence Rate of Robust Policy Gradient . The robust policy gradient method has been shown to converge with iteration complexity of $O(\epsilon^{-4})$ for general robust MDPs [17]. However, it requires oracle access to robust policy evaluation (i.e., the computation of the worst kernel), which can be computationally expensive [17].

2 Preliminary

A Markov Decision Process (MDP) can be described as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, P is a transition kernel mapping $\mathcal{S} \times \mathcal{A}$ to $\Delta_{\mathcal{S}}$, R is a reward function mapping $\mathcal{S} \times \mathcal{A}$ to \mathbb{R} , μ is an initial distribution over states in \mathcal{S} , and γ is a discount factor in $[0, 1]$ [25, 27]. A policy $\pi : \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$ is a decision rule that maps state space to a probability distribution over action space. Let $\Pi = (\Delta_{\mathcal{A}})^{\mathcal{S}}$ denote set of all possible policies. Further, $\pi(a|s), P(s'|s, a)$ denotes the probability of taking action a in state s by policy π , and the probability of transition to state s' from state s under action a respectively. In addition, we denote $P^{\pi}(s'|s) = \sum_a \pi(a|s)P(s'|s, a)$ and $R^{\pi}(s) = \sum_a \pi(a|s)R(s, a)$ as short-hands.

The return of a policy π , 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, $d_P^{\pi} = \mu^{\top}D^{\pi}$ is occupation measure and $D^{\pi} = (I - \gamma P^{\pi})^{-1}$ is occupancy matrix [25]. As a shorthand, we denote $d_P^{\pi}(s, a) = d_P^{\pi}(s)\pi(a|s)$ and the usage shall be clear from the context.

A robust Markov Decision Process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, R, \mathcal{U}, \gamma, \mu)$ which generalizes

the standard MDP, by containing a set of environments \mathcal{U} [5, 4]. The reward robust MDPs is well-studied in the precious work of rectangular [18, 19] and non-rectangular [23] uncertainty sets. Hence, in this work , we consider only uncertainty in the kernel which is much more challenging.

For an uncertainty set \mathcal{U} , the robust return $J_{\mathcal{U}}^{\pi}$ for a policy π , and the optimal robust return $J_{\mathcal{U}}^*$, are defined as:

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

respectively. The objective is to determine an optimal robust policy $\pi_{\mathcal{U}}^*$ that achieves the optimal robust performance $J_{\mathcal{U}}^*$. Unfortunately, even robust policy evaluation (i.e., finding the worst-case 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 uncertainty sets [9]. This makes solving non-rectangular robust MDPs a highly challenging problem.

To make the problem tractable, a common approach is to use **s**-rectangular uncertainty sets, $\mathcal{U}^s = \times_{s \in \mathcal{S}} \mathcal{P}_s$, where the uncertainty is modeled independently across states [9]. These sets decompose state-wise, capturing correlated uncertainties within each state while ignoring inter-dependencies across states. This allows the robust value function to be defined as (vector minimum) $v_{\mathcal{U}^s}^{\pi} = \min_{P \in \mathcal{U}^s} v_P^{\pi}$ [9].

A further simplification is the **sa**-rectangular uncertainty set, \mathcal{U}^{sa} , where uncertainties are assumed to be independent across both states and actions. Formally, $\mathcal{U}^{sa} = \times_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_{s,a}$, where $\mathcal{P}_{s,a}$ are independent component sets for each state-action pair [5, 4, 15, 16]. Notably, **sa**-rectangular sets are a special case of **s**-rectangular sets.

Various types of rectangular uncertainty sets have been explored in the literature [15, 28, 29]. In this work, we focus specifically on L_p -bounded uncertainty sets $\mathcal{U}_p^{sa}/\mathcal{U}_p^s$, which are centered around a nominal transition kernel \hat{P} [14, 18, 19, 20], defined as

$$\begin{aligned} \mathcal{U}_p^{sa} &= \{P \mid \sum_{s'} P_{sa}(s') = 1, \|P_{sa} - \hat{P}_{sa}\|_p \leq \beta_{sa}\}, \\ \mathcal{U}_p^s &= \{P \mid \sum_{s'} P_{sa}(s') = 1, \|P_s - \hat{P}_s\|_p \leq \beta_s\}, \end{aligned}$$

with small enough radius vector β is small enough, to ensure all the kernels in the uncertainty sets are valid. Further, symbol q is the Hölder conjugate of p and σ_p is the generalized standard deviation (GSTD)[19] defined as:

$$\frac{1}{p} + \frac{1}{q} = 1, \quad \text{and} \quad \sigma_p(v) := \min_{\omega \in \mathbb{R}} \|v - \omega \mathbf{1}\|_p.$$

One of the most surprising, and useful facts about L_p bounded uncertainty sets, is that the adversarial kernel is a rank one perturbation of the nominal kernel, as stated below.

Proposition 2.1. (*Nature of the Adversary,[20]*) For uncertainty set $\mathcal{U} = \mathcal{U}_p^{sa}/\mathcal{U}_p^s$, the worst kernel is given as

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

where k depends on the robust value function $v_{\mathcal{U}}^{\pi}$ and b is (policy weighted of \mathcal{U}_p^s) radius vector.

This result is insightful however it doesn't characterize the direction of perturbation k in nominal terms.

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:

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

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 [17].

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. Moreover, no prior work has addressed robust gradient evaluation in the context of non-rectangular robust MDPs. This work constitutes the first attempt to compute the robust gradient for such MDPs by leveraging the dual structure of robust MDPs, paving the way for practical robust policy gradient methods.

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

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

where $\mathcal{V} = \{v \mid v = R^\pi + \gamma P^\pi v, \pi \in \Pi\}$ represents the set of value functions. The dual formulation relies on the state-action occupancy measure d , where $d \in \mathcal{D} \subset \mathbb{R}^{|S| \times |A|}$ satisfies the non-negativity constraint ($d \succeq 0$) and the flow conservation constraint: $\sum_a d(s, a) - \gamma \sum_{s', a'} d(s', a') P(s \mid s', a') = \mu(s), \quad \forall s \in \mathcal{S}$. The feasible set \mathcal{D} forms a convex polytope [30], whereas the set of value functions, \mathcal{V} , is a polytope that is generally non-convex [31]. This dual formulation offers several advantages, including efficient handling of constraints and the ability to solve the problem using linear programming techniques.

For robust MDPs, the geometry of robust value functions is significantly more intricate compared to standard MDPs [32]. While the dual formulation for standard MDPs is well-established, this work is the first to derive a dual formulation for robust MDPs. This novel formulation provides critical insights and lays the foundation for the development of robust policy evaluation methods.

3 Method

In this section, we derive the dual formulation for non-rectangular robust Markov decision processes (RMDPs) with uncertainty sets bounded by L_p balls. This dual perspective not only introduces several new research questions but also provides critical insights into the underlying problem. Further, we develop a method to compute the worst kernel, thereby enabling robust policy evaluation.

We begin with defining the non-rectangular L_p -constrained uncertainty set around the nominal kernel \hat{P} as:

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

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} . The simplex constraint ensures that the transition kernel P satisfies the unity-sum-rows property, as discussed in [19]. The kernel radius β is assumed to be small enough to guarantee that all kernels within \mathcal{U}_p are well-defined, consistent with assumptions made in prior works [18, 20, 19].

Note that this setting allows noise in one state to be coupled with noise in other states. Before delving into solving it, we first discuss why its important? 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 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*[33].

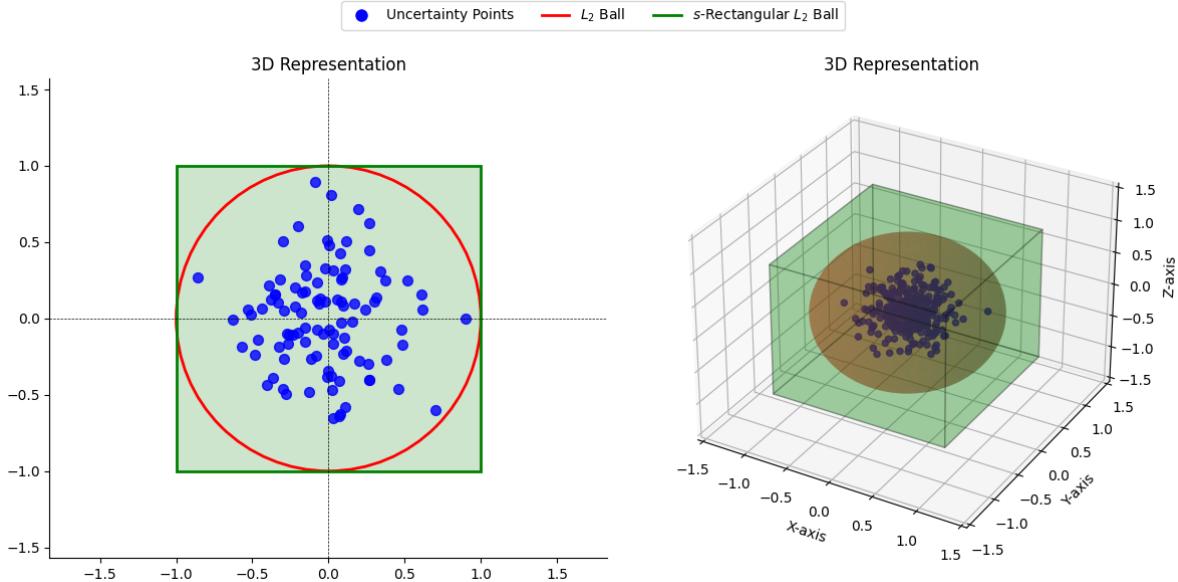


Figure 1: Modeling Uncertainty with Non-Rectangular and Rectangular L_2 -Balls.

To make matters worse, 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,*

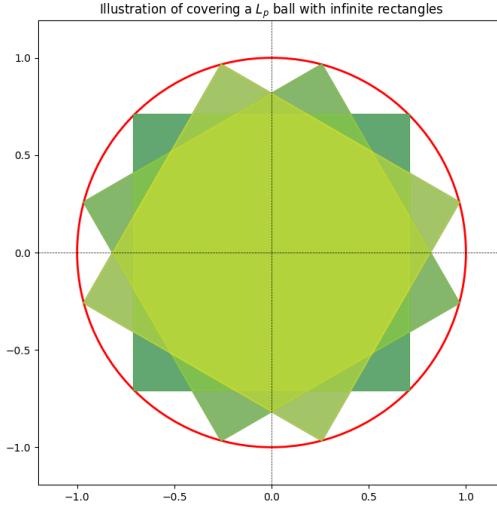


Figure 2: Illustration of Proposition 3.2: N-dimensional sphere can be written as infinite union of n-dimensionssional inscribing cubes.

respectively, that contain \mathcal{U}_2 . Then:

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

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

The result follows from the n -dimensional sphere's volume $c_n r^n$ ($c_n \rightarrow 0$) [24], compared to the enclosing cube's volume $2^n r^n$ (side $2r$), resulting in a ratio of $O(2^n)$.

From the above discussion, we conclude that non-rectangular robust MDPs are less conservative. However, robust policy evaluation (even approximation) has been proven NP-hard for general uncertainty sets defined as intersections of finite hyperplanes [9]. Specifically, [9] reduces an Integer Program (IP) with m constraints to robust MDPs where the uncertainty set consists of intersections of m half-spaces (m -linear constraints). This polyhedral structure is fundamental to the hardness proof, hence, 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 apply, but since \mathcal{U}_1 is defined by a single global constraint ($\|P - \hat{P}_1\|_1 \leq \beta$), it forces the IP to have only one simple constraint which is efficiently solvable. A detailed discussion can be found in Appendix D.2.

We conclude that L_p -robust MDPs are potentially tractable. A key insight is that a non-rectangular uncertainty set \mathcal{U}_p can be expressed as a union of **sa**-rectangular sets $\mathcal{U}_p^{\text{sa}}(b)$ with varying radius vectors b , each of which can be solved more easily on its own.

Proposition 3.2. *Non-rectangular uncertainty \mathcal{U}_p can be written as infinite union of **sa**-rectangular sets $\mathcal{U}_p^{\text{sa}}$, as*

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

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

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. 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:

$$J_{\mathcal{U}_p}^{\pi} = \min_{b \in \mathcal{B}} \min_{P \in \mathcal{U}_p^{\text{sa}}(b)} J_P^{\pi}. \quad (2)$$

In essence, we have reduced a challenging problem into an infinite number of simpler ones. However, our task is not complete yet. While there exists a closed-form expression for $J_{\mathcal{U}_p^{\text{sa}}}^{\pi} = J^{\pi} - \sum_{s,a} d^{\pi}(s,a)b_{sa}\sigma_q(v_{\mathcal{U}_p^{\text{sa}}}^{\pi})$, where $\sigma_q(v_{\mathcal{U}_p^{\text{sa}}}^{\pi})$ represents the q -generalized standard deviation (GSTD) of the robust value function as defined in [19], this formulation is still impractical, as $\max_{b \in \mathcal{B}} \sum_{s,a} d^{\pi}(s,a)b_{sa}\sigma_q(v_{\mathcal{U}_p^{\text{sa}}(b)}^{\pi})$ remains computationally challenging.

To address this issue, we turn to the dual formalism developed in the next section.

3.1 Dual Formulation of Robust MDPs

In this section, we derive, for the first time, a dual formulation for robust MDPs. While it is more complex than the dual formulation for non-robust MDPs and applies specifically to L_p -bounded uncertainty sets, it lays the groundwork for all the subsequent results in the paper.

From [20], we know **sa**-rectangular worst-case kernel $P_{\mathcal{U}_p^{\text{sa}}(b)}^{\pi} = \hat{P} - bk^T$ is a rank-one perturbation of the nominal kernel, where $k \in \mathcal{K} := \{k \mid \|k\|_p \leq 1, \mathbf{1}^T k = 0\}$. Hence, it is enough for the adversary to focus on the rank-one perturbations, allowing us to rewrite (2) as

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

where the last equality comes from $J_{\hat{P}}^{\pi} = \mu^T D_{\hat{P}}^{\pi} R^{\pi}$. Further, as shown in Lemma 4.4 of [20], applying the Sherman–Morrison formula [34] (see Proposition D.1), the robust return can be expressed as:

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

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

Lemma 3.3. *The robust return can be expressed as:*

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

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^{π} and a penalty term involving only nominal values (d^{π} , $v_R^{\pi} = v^{\pi}$, and v_b^{π}). 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 π , revealing the adversary.

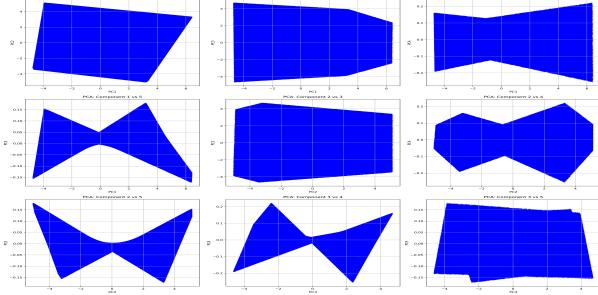


Figure 3: Projections of set \mathcal{D} along principal components, for $S = 3, A = 2$ with 10 millions samples. This figure strongly suggests the non-convexity of the set.

Additionally, by maximizing the robust return $J_{\mathcal{U}_p}^\pi$ over policies, we derive a novel dual formulation, as stated below.

Theorem 3.4. *The optimal robust return is the solution to*

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

where $\mathcal{D} = \{ D^\pi H^\pi \mid \pi \in \Pi \}$ and $H^\pi R := R^\pi, D^\pi = (I - P^\pi)^{-1}$.

Notably, the dual formulations for the **sa**-rectangular and **s**-rectangular cases differ in their definitions of \mathcal{B} : for the **sa**-rectangular case, $\mathcal{B} = \{\beta\}$, whereas for the **s**-rectangular case, $\mathcal{B} = \{b \in \mathbb{R}^{S \times A} \mid \|b_s\|_p \leq \beta_s\}$, as detailed in the appendix.

The result above formulates the dual of robust MDPs as a min-max problem, which is insightful and significant in itself. However, the set \mathcal{D} may be non-convex, as suggested by Figure 3 (details in the appendix), making the problem non-convex. We leave this as an open question for future work:

How can we effectively exploit the above dual form for more insights and better algorithmic design?

In this paper, we first develop a method to approximate the worst-case kernel for robust policy evaluation, as discussed in the next section. We then derive policy gradient methods to facilitate policy improvement.

3.2 Robust Policy Evaluation

In this section, we propose an algorithm for robust policy evaluation and establish its performance guarantees. The following result states that the robust return can be computed using the function:

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

where $E_\lambda^\pi := \gamma \left(I - \frac{\mathbf{1}\mathbf{1}^\top}{S} \right) [D^\pi R^\pi \mu^\top D^\pi - \lambda D^\pi] H^\pi$, and $H^\pi R := R^\pi$ consists of easily computable quantities.

Lemma 3.5. *The robust return can be expressed as:*

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

where the penalty λ^* is a fixed point of $F(\lambda)$. Furthermore, λ^* can be found via binary search since:

$$F(\lambda) > \lambda \quad \text{if and only if} \quad \lambda > \lambda^*.$$

The proof of this result is provided in Appendix (Lemma F.1). Further, the bisection property of F established in the result, directly implies the linear convergence rate of Algorithm 1, as stated below.

Algorithm 1 Binary Search for Robust Policy Evaluation

- 1: **Initialize:** Upper limit $\lambda_u = \frac{1}{1-\gamma}$, lower limit $\lambda_l = 0$
 - 2: **while** not converged: $n = n + 1$ **do**
 - 3: **Bisection value:** $\lambda_n = (\lambda_l + \lambda_u)/2$
 - 4: **Bisection:** $\lambda_l = \lambda_n$ if $F(\lambda_n) > \lambda_n$, else $\lambda_u = \lambda_n$.
 - 5: **Update robust return:** $J_n = J^\pi - \lambda_n$.
 - 6: **end while**
-

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

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

We conclude that robust evaluation can be performed efficiently with linear iteration complexity. However, each iteration requires solving $\max_{x \in \mathcal{B}} \|Ax\|_q$ as a subroutine in Algorithm 1. We focus specifically on the simplified case of $p = 2$, i.e., $\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2$, for which we proposed a modified eigenvalue Algorithm 2. This method has a time complexity of $O(S^3 A^3)$, performs very effectively (i.e., very similar performance to numerical method `scipy.optimize.minimize` and order of magnitude of faster), with more details in Appendix H.

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

- 1: Compute eigenvector v_i and eigenvalues λ_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 $\beta \|A \frac{v_j^+}{\|v_j^+\|_2}\|_2$.
-

Performance of robust policy evaluation Algorithm 1, is further validated experimentally in the later section.

4 Revealing the Adversary

Non-rectangular robust MDPs have been sparsely studied in the literature, and the nature of the adversary remains unexplored. The following result provides the first insights into the adversary's

behavior for non-rectangular uncertainty sets. It establishes that the worst-case transition kernel is a rank-one perturbation of the nominal kernel, similar to the case of rectangular uncertainty sets (see Proposition 2.1). However, its exact structure is significantly more intricate.

Theorem 4.1 (Non-rectangular Worst Kernel). *The worst kernel for the policy π and the uncertainty set \mathcal{U}_p is*

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

where (k, b) is a solution to $\max_{k \in \mathcal{K}, b \in \mathcal{B}} [J_b^\pi \frac{\langle k, v_R^\pi \rangle}{1 + \gamma \langle k, v_b^\pi \rangle}]$.

The result shows that the adversary controls two variables, β and k , with the objective of:

- **Maximizing the average uncertainty in the trajectories** J_β^π , since the more frequently the agent visits high-uncertainty states, the greater the adversary's ability to steer it toward unfavorable states.
- **Choosing the perturbation direction** k to maximize $k^\top v_R^\pi$, forcing the agent into low-reward trajectories, while simultaneously minimizing $k^\top v_b^\pi$, ensuring high exposure to high-uncertainty states.

These insights provide a deeper understanding of the adversary and can aid in designing more resilient robust algorithms.

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 Policy Improvement

Once a worst kernel for policy is obtained using Algorithm 1, we can compute the policy gradient to update the policy. Alternatively, we can use the policy gradient theorem derived in the result below.

Lemma 5.1. [Policy Gradient] *Given the worst transition kernel $P_{\mathcal{U}_p}^\pi = \hat{P} - bk^\top$, the gradient is given as*

$$\begin{aligned} \nabla_\pi J_{\mathcal{U}_p}^\pi &= d^\pi \circ Q_R^\pi - \gamma \frac{k^\top v_R^\pi}{1 + \gamma k^\top v_b^\pi} d^\pi \circ Q_b^\pi \\ &\quad - \gamma \frac{J_b^\pi(k^\top D^\pi)}{1 + \gamma k^\top v_b^\pi} \circ Q_R^\pi + \gamma^2 \frac{J_b^\pi(k^\top v^\pi)(k^\top D^\pi)}{(1 + \gamma k^\top v_b^\pi)^2} \circ Q_b^\pi, \end{aligned}$$

where $(u \circ v)(s) := u(s)v(s)$.

Note that the above result expresses robust gradient only in nominal terms, displaying the complex interplay of different parameters.

The first term $d^\pi \circ Q_R^\pi$ is a nominal policy gradient, trying to improve the policy's emphasis (weight) for high reward actions. The first part of the second term $\gamma \frac{k^\top v_b^\pi}{1 + \gamma k^\top v_b^\pi} = \sigma_q(v_{\mathcal{U}_p}^\pi)$ is GSTD, hence, always positive, measuring the vulnerability towards the adversary actions. This scales the other term $d^\pi \circ Q_b^\pi$, that is policy gradient w.r.t. the reward being the uncertainty radius. To summarize, the second term discourages the policy's emphasis (weight) in high uncertainty Q-value by the amount proportional to the vulnerability.

The last two terms are much more complex to interpret, showcasing the complexity of robust MDPs .

Theorem 5.2. *Robust policy gradient algorithm 3, convergence to an ϵ -optimal policy in a total of $O(\epsilon^{-8})$ iterations.*

The policy gradient method in [17] requires $O(\epsilon^{-4})$ iterations to converge to the globally optimal robust policy. At the n -th policy gradient step, the guarantee in [17] necessitates an $O(\gamma^{-n})$ -close approximation of the worst-case kernel for the policy π_n , which our Algorithm 1 computes in $O(n)$ iterations. Consequently, the overall iteration complexity to achieve the global optimal robust policy becomes $O\left(\sum_{n=1}^{\epsilon^{-4}} n\right) = O(\epsilon^{-8})$.

Algorithm 3, is a double loop algorithm: That is, the inner loop (Algorithm 1) computes an approximate worst kernel for the fixed policy. On the other hand, the outer loop updates the policy using the gradient obtained using the worst kernel. Alternatively, an actor-critic style algorithm can also be obtained, where the worst kernel and the policy are updated simultaneously. We leave this direction for a future work.

Algorithm 3 Robust Policy Gradient Algorithm

- 1: **while** not converged: $n = n + 1$ **do**
 - 2: Get worst kernel $P = \hat{P} - bk^\top$ for policy π from Algorithm 1 with tolerance $\epsilon = \gamma^n$..
 - 3: Compute gradient G from Lemma 5.1 .
 - 4: Policy update: $\pi \rightarrow Proj \left[\pi + \alpha_n G \right]$.
 - 5: **end while**
-

6 Experiments

In this section, we evaluate the performance of Algorithm 1 for robust policy evaluation, focusing on the case $p = 2$. The algorithm requires computing $F(\lambda)$ at each iteration, which involves solving the constrained matrix norm problem $\max_{x \in \mathcal{B}} \|Ax\|_2$. This can be efficiently handled using our spectral Algorithm 2. Figures 4 and 5 compare four methods for computing the robust return, specifically the penalty term $J^\pi - J_{\mathcal{U}_2}^\pi$:

- **SLSQP from scipy [35] :** This is A semi-brute force approach that uses Lemma 3.3. It computes the penalty term via `scipy.optimize.minimize` to directly optimize over (b, k) . This is equivalent to optimize over only rank-one perturbation of nominal kernel as $a(b, k)$ corresponds to selecting a rank-one perturbation of the nominal kernel $P = \hat{P} - bk^\top$. Note that this method is local, hence can sometime be stuck in very bad local solution.

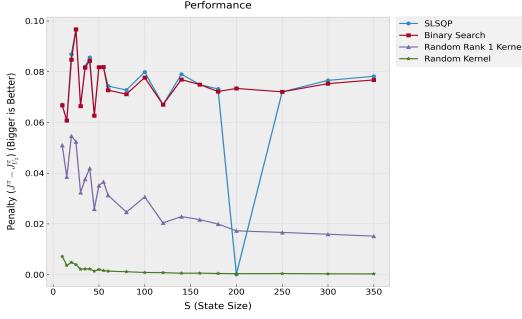


Figure 4: Performance of Robust Policy Evaluation methods with equal amount of time, with fixed action space $A = 8$

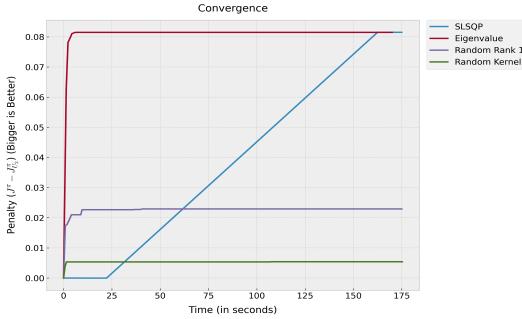


Figure 5: Convergence of Robust Policy Evaluation Methods, with fixed $S = 128, A = 8$.

- **Binary search Algorithm 1 :** Uses the binary search Algorithm 1, and spectral Algorithm 2 for computing $F(\lambda)$ in each iteration.
- **Random Rank-One Kernel Sampling :** A semi-brute force approach that uses Lemma 3.3 to sample random pairs $(b, k) \in \mathcal{B}, \mathcal{K}$, empirically maximizing the penalty term. Since choosing (b, k) corresponds to selecting a rank-one perturbation of the nominal kernel $P = \hat{P} - bk^\top$, the method is named accordingly.
- **Random Kernel Sampling :** A brute-force approach that samples random kernels directly from the uncertainty set \mathcal{U}_2 , computing the empirical minimum as an estimate of the robust return.

Figure 4 presents the penalty value ($J^\pi - J_{\mathcal{U}_2}^\pi$) computed by different methods across various state space sizes while keeping the action space ($A = 8$) fixed, with each method given the same computational time. Our binary search Algorithm 1 combined with spectral Algorithm 2 performs significantly better to brute-force (random kernel sampling) and semi-brute (random sampling of rank-one perturbations of the nominal kernel) approaches. Notably, the scipy SLSQP variant performs slightly better on average, but our Algorithm 1 is more reliable. This is expected, as the spectral method used to compute $F(\lambda)$ in Algorithm 1, is global, while scipy SLSQP is a local optimizer and thus more prone to getting stuck in suboptimal solutions (as evident in the figure).

Figure 5 illustrates the convergence of different approaches over time (in seconds) for a fixed state space size ($S = 8$). Algorithm 1 converges rapidly, while the SLSQP variation gradually approaches the same performance. In contrast, the brute-force method shows slow, logarithmic

improvement. This behavior arises because brute-force methods require an exponential number of samples (in the dimensionality of the problem) to adequately explore all directions. In comparison, our binary search Algorithm combined with the spectral Algorithm 2 achieves significantly better efficiency, with a complexity of $O(S^3 A^3 \log \epsilon^{-1})$.

More details of these experiments along with others can be found in the appendix, and codes are available at <https://anonymous.4open.science/r/non-rectangular-rmdp-77B8>.

Our experiments confirm the efficiency of our binary search Algorithm 1 for robust policy evaluation, significantly outperforming brute-force approaches in both accuracy and convergence speed.

7 Discussion

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 decomposed into infinitely many *sa*-rectangular sets, reducing robust policy evaluation to a min-max fractional optimization problem (dual form). This novel dual formulation provides key insights into the adversary and leads to the development of the first robust policy evaluation algorithms. Experiments demonstrate the effectiveness of our approach, significantly outperforming brute-force methods. These findings further pave the way for scalable and efficient robust reinforcement learning algorithms.

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

Another promising direction is to design policy improvement methodologies compatible with deep neural networks.

References

- [1] Shie Mannor, Duncan Simester, Peng Sun, and John 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.
- [2] Grani Adiwena Hanasusanto and Daniel Kuhn. Robust data-driven dynamic programming. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

- [3] Aviv Tamar, Shie Mannor, and Huan 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.
- [4] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Oper. Res.*, 53:780–798, 2005.
- [5] Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2):257–280, May 2005.
- [6] Huan Xu and Shie Mannor. Robustness and generalization, 2010.
- [7] Chenyang Zhao, Olivier Sigaud, Freek Stulp, and Timothy M. Hospedales. Investigating generalisation in continuous deep reinforcement learning, 2019.
- [8] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning, 2018.
- [9] Wolfram Wiesemann, Daniel Kuhn, and Breç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.
- [10] Shie Mannor, Ofir Mebel, and Huan Xu. Robust mdps with k-rectangular uncertainty. *Math. Oper. Res.*, 41(4):1484–1509, nov 2016.
- [11] Vineet Goyal and Julien Grand-Clément. Robust markov decision process: Beyond rectangularity, 2018.
- [12] David L. Kaufman and Andrew J. Schaefer. Robust modified policy iteration. *INFORMS J. Comput.*, 25:396–410, 2013.
- [13] J. Andrew Bagnell, Andrew Y. Ng, and Jeff G. Schneider. Solving uncertain markov decision processes. Technical report, Carnegie Mellon University, 2001.
- [14] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Partial policy iteration for l1-robust markov decision processes, 2020.
- [15] Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty, 2021.
- [16] Yue Wang and Shaofeng Zou. Policy gradient method for robust reinforcement learning, 2022.
- [17] Qiuhan Wang, Chin Pang Ho, and Marek Petrik. Policy gradient in robust mdps with global convergence guarantee, 2023.
- [18] Esther Derman, Matthieu Geist, and Shie Mannor. Twice regularized mdps and the equivalence between robustness and regularization, 2021.

- [19] Navdeep Kumar, Kaixin Wang, Kfir Yehuda Levy, and Shie Mannor. Efficient value iteration for s-rectangular robust markov decision processes. In *Forty-first International Conference on Machine Learning*, 2024.
- [20] Navdeep Kumar, Esther Derman, Matthieu Geist, Kfir Yehuda Levy, and Shie Mannor. Policy gradient for rectangular robust markov decision processes. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [21] Ruida Zhou, Tao Liu, Min Cheng, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Natural actor-critic for robust reinforcement learning with function approximation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [22] Kaixin Wang, Uri Gadot, Navdeep Kumar, Kfir Levy, and Shie Mannor. Robust reinforcement learning via adversarial kernel approximation, 2023.
- [23] Uri Gadot, Esther Derman, Navdeep Kumar, Maxence Mohamed Elfatihi, Kfir Levy, and Shie Mannor. Solving non-rectangular reward-robust mdps via frequency regularization, 2023.
- [24] David J. Smith and Mavina K. Vamanamurthy. How small is a unit ball? *Mathematics Magazine*, 62(2):101–107, 1989.
- [25] Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.
- [26] Julien Grand-Clément, Nian Si, and Shengbo Wang. Tractable robust markov decision processes, 2024.
- [27] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.
- [28] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning, 2019.
- [29] Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Robust ϕ -divergence MDPs. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [30] E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.
- [31] Robert Dadashi, Adrien Ali Taïga, Nicolas Le Roux, Dale Schuurmans, and Marc G. Bellemare. The value function polytope in reinforcement learning, 2019.
- [32] Kaixin Wang, Navdeep Kumar, Kuangqi Zhou, Bryan Hooi, Jiashi Feng, and Shie Mannor. The geometry of robust value functions. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan 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.

- [33] Shie Mannor, Ofir Mebel, and Huan Xu. Lightning does not strike twice: Robust mdps with coupled uncertainty. *CoRR*, abs/1206.4643, 2012.
- [34] M. S. Bartlett. An Inverse Matrix Adjustment Arising in Discriminant Analysis. *The Annals of Mathematical Statistics*, 22(1):107 – 111, 1951.
- [35] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [36] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.
- [37] 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.
- [38] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.
- [39] Navdeep Kumar, Kaixin Wang, Kfir Levy, and Shie Mannor. Policy gradient for reinforcement learning with general utilities, 2023.

A Summary of 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 .

Table 1: Useful Notations

Notation	Definition	Remark
p, q	$\frac{1}{p} + \frac{1}{q} = 1$	Holder's conjugates
σ_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

B More on Setting, Assumptions, Results, Discussion

Extension to KL Entropy Uncertainty Sets. For the KL uncertainty case, the worst kernel is given by $P_{\mathcal{U}_{KL}^{\text{sa}}}^\pi = (I - \gamma \hat{P}^\pi A^\pi)^{-1}$ where A^π is a diagonal matrix [22]. If we can invert this matrix, then its possible to build upon it. We leave this for future work.

B.1 Revealing the Adversary

The worst kernel for the policy π and the uncertainty set $\mathcal{U}_p^{\text{sa}}/\mathcal{U}_p^{\text{s}}$, is given as

$$P_{\mathcal{U}_p^{\text{sa}}}^\pi(\cdot|s, a) = \hat{P}(\cdot|s, a) - \beta_{sa} k, \quad \text{and} \quad (3)$$

$$P_{\mathcal{U}_p^{\text{s}}}^\pi(\cdot|s, a) = \hat{P}(\cdot|s, a) - \beta_s \left(\frac{\pi(a|s)}{\|\pi_s\|_q} \right)^{q-1} k, \quad (4)$$

where k is a solution of $\max_{k \in \mathcal{K}} \frac{k^T v_R^\pi}{1 + \gamma k^T v_\beta^\pi}$. The relation sheds the light on the working of the adversary in (13). Basically, k is the direction in which the adversary discourages the perturbation of the kernel. And the optimal direction k that the adversary chooses is the one

that maximizes the potential gain in the reward ($\langle k, v_R^\pi \rangle$) and minimizes the average uncertainty on the trajectories ($\langle k, v_\beta^\pi \rangle$).

Compared to the existing characterization of the adversary in [20], see Proposition C.2 , where the direction $k = u_{\mathcal{U}}^\pi$ is in robust terms, this is the first time, we have a complete overview of the working of the adversary in nominal terms.

Theorem B.1 (Non-rectangular Worst Kernel). *The worst kernel for the policy π and the uncertainty set \mathcal{U}_p is*

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

where (k, b) is a solution to $\max_{k \in \mathcal{K}, b \in \mathcal{B}} [J_b^\pi \frac{\langle k, v_R^\pi \rangle}{1 + \gamma \langle k, v_b^\pi \rangle}]$.

Moreover, we can see that the adversary wants to optimize three things: A) J_β^π , $k^\top v_R^\pi$ and $k^\top v_\beta^\pi$ with two variables β and k .

1. Maximizing the average uncertainty in the trajectories J_β^π . This makes sense, as the more the agent visits states with high uncertainty, the higher is the ability of the adversary to undermine it.
2. Choosing the perturbation direction k that discourages the agent to transition into good value states (w.r.t. nominal value function v_R^π). This observation was also seen in a previous study [20], for sa and s rectangular uncertainty sets but it was w.r.t. robust value function $v_{\mathcal{U}}^\pi$. Whereas, the result has the exact finding except it is w.r.t. nominal value function, which is known contrast to the robust value function.
3. Choosing the uncertainty radius vector β and the perturbation direction k , such that the $k^\top v_\beta^\pi$ is minimized. This can be done, by putting negative entries of k at maximal entries of v_β^π , and positive entries of k^\top at states which has minimum uncertainty value function. In other words, we want to perturb the kernel such that visitation of high uncertainty states in the long term is maximized.

C Background

C.1 Robust MDPs

A robust Markov Decision Process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, R, \mathcal{U}, \gamma, \mu)$ which generalizes the standard MDP, by containing a set of environments \mathcal{U} [5, 4]. The reward robust MDPs is well-studied in the precious work of rectangular [18, 19] and non-rectangular [23] uncertainty sets. Hence, in this work , we consider only uncertainty in kernel which is much more challenging.

The robust return of a policy π , is its performance over the uncertainty set \mathcal{U} , defined as $J_{\mathcal{U}}^\pi = \min_{P \in \mathcal{U}} J_P^\pi$. The objective is to find an optimal robust policy $\pi_{\mathcal{U}}^*$ that achieves the optimal robust performance $J_{\mathcal{U}}^*$, defined as

$$J_{\mathcal{U}}^* = \max_{\pi} J_{\mathcal{U}}^\pi. \tag{5}$$

Unfortunately, a general solution to the robust objective (5), is proven to be strongly NP-hard for both non-convex sets and convex ones [9]. The robust value function $v_{\mathcal{U}}^{\pi}$ and optimal robust value function $v_{\mathcal{U}}^*$ [5, 4], for any uncertainty set \mathcal{U} can be defined state wise, for all π , as

$$v_{\mathcal{U}}^{\pi}(s) = \min_{P \in \mathcal{U}} v_P^{\pi}(s), \quad v_{\mathcal{U}}^*(s) = \max_{\pi \in \diamond} v_P^{\pi}(s).$$

C.2 Rectangular Robust MDPs

Unfortunately, a general solution to the robust objective (5), is proven to be strongly NP-hard for both non-convex sets and convex ones [9]. Hence, it is common practice to take the **sa**-rectangular uncertainty sets \mathcal{U}^{sa} , where ambiguity in each state and action are independent [5, 4, 15, 16]. Formally defined as $\mathcal{U}^{\text{sa}} = \times_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_{s,a}$ is decomposed over state-action-wise where $\mathcal{P}_{s,a}$ are components sets.

However, many classes of uncertainty sets arise in practice, where ambiguities in a given state are correlated. This type of uncertainty sets are captured by **s**-rectangular uncertainty sets $\mathcal{U}^{\text{s}} = \mathcal{P} = \times_{s \in \mathcal{S}} \mathcal{P}_s$, which can be decomposed state-wise as [9]. Note that **sa**-rectangular uncertainty sets are a special case of it.

Fortunately, the decoupling structure in **s**-rectangular uncertainty sets allows the existence of a kernel and a reward function that minimizes the value function over the uncertainty set for each state for any given policy. Similarly, it allows the existence of an optimal robust policy that maximizes the robust value in each state [9]. Mathematically, the robust value function is to be rewritten as

$$v_{\mathcal{U}^{\text{s}}}^* = \max_{\pi} v_{\mathcal{U}^{\text{s}}}^{\pi}, \quad v_{\mathcal{U}^{\text{s}}}^{\pi} = \min_{P \in \mathcal{U}^{\text{s}}} v_P^{\pi}.$$

Hence, the robust return can be rewritten as $J_{\mathcal{U}}^{\pi} = \langle \mu, v_{\mathcal{U}}^{\pi} \rangle$, and $J_{\mathcal{U}}^* = \langle \mu, v_{\mathcal{U}}^* \rangle$. Most importantly, this rectangularity implies the existence of contractive robust Bellman operators, which are pivotal same as non-robust MDPs [9]. Specifically, the robust value function $v_{\mathcal{U}}^{\pi}$, and the optimal robust value function $v_{\mathcal{U}}^*$ is the fixed point of the robust Bellman operator $\mathcal{T}_{\mathcal{U}}^{\pi}$ and the optimal robust Bellman operator $\mathcal{T}_{\mathcal{U}}^*$ respectively [9, 5], defined as

$$\mathcal{T}_{\mathcal{U}}^{\pi} v := \min_{P \in \mathcal{U}} T_P^{\pi} v, \quad \text{and} \quad \mathcal{T}_{\mathcal{U}}^* v := \max_{\pi} \mathcal{T}_{\mathcal{U}}^{\pi} v,$$

where $T_P^{\pi} v := R^{\pi} + \gamma P^{\pi} v$ is non-robust Bellman operator [25].

This contractive property of Bellman operators plays a central role in the **s**-rectangular robust MDPs. Unfortunately, these contractive Bellman operators do not exist for non-rectangular uncertainty sets. This makes the non-rectangular robust MDPs very unwieldy.

C.3 Rectangular L_p robust MDPs

Some useful definitions: q is reserved for Holder's conjugate of p that satisfies $\frac{1}{p} + \frac{1}{q} = 1$. The generalized p -variance function σ_p and p -mean function ω_p are defined as

$$\sigma_p(v) = \min_{w \in \mathbb{R}} \|v - w\mathbf{1}\|_p, \quad \omega_p(v) \in \arg \min_{w \in \mathbb{R}} \|v - w\mathbf{1}\|_p.$$

Table 2: p -variance

p	$\sigma_p(v)$	Remark
∞	$\frac{\max_s v(s) - \min_s v(s)}{2}$	Semi-norm
2	$\sqrt{\sum_s (v(s) - \frac{\sum_s v(s)}{S})^2}$	Variance
1	$\frac{\sum_{i=1}^{\lfloor (S+1)/2 \rfloor} v(s_i) - \sum_{i=\lceil (S+1)/2 \rceil}^S v(s_i)}{\sum_{i=1}^S v(s_i)}$	Top half - lower half

where v is sorted, i.e. $v(s_i) \geq v(s_{i+1}) \quad \forall i$.

Their close form for $p = 1, 2, \infty$, is summarized in table 2 . The results below summarizes policy evaluation for L_p -robust MDPs [19].

Let \hat{P} be any nominal transition kernel. In accordance with [19], we define **sa**-rectangular L_p constrained uncertainty set $\mathcal{U}_p^{\text{sa}}$ and **s**-rectangular L_p constrained uncertainty set \mathcal{U}_p^{s} as

$$\begin{aligned}\mathcal{U}_p^{\text{sa}} &= \{P \mid \underbrace{\sum_{s'} P_{sa}(s')}_{\text{simplex condition}} = 1, \|P_{sa} - (\hat{P})_{sa}\|_p \leq \beta_{sa}\} \\ \mathcal{U}_p^{\text{s}} &= \{P \mid \sum_{s'} P_{sa}(s') = 1, \|P_s - (\hat{P})_s\|_p \leq \beta_s\}.\end{aligned}$$

Note that $\mathcal{U}_p^{\text{sa}}$, \mathcal{U}_p^{s} are sets around nominal kernel \hat{P} component wise bounded by radius vectors β . To ensure, all the kernels in \mathcal{U} are valid, we assume the radius vector β is small enough.

Proposition C.1. [19] *The robust return is*

$$\begin{aligned}J_{\mathcal{U}_p^{\text{sa}}}^\pi &= J^\pi - \gamma \sigma_q(v_{\mathcal{U}_p^{\text{sa}}}^\pi) \sum_{s,a} d^\pi(s) \pi(a|s) \beta_{sa}, \\ J_{\mathcal{U}_p^{\text{s}}}^\pi &= J^\pi - \gamma \sigma_q(v_{\mathcal{U}_p^{\text{s}}}^\pi) \sum_s d^\pi(s) \|\pi_s\|_q \beta_s\end{aligned}$$

where $\|\pi_s\|_q$ is q -norm of the vector $\pi(\cdot|s) \in \Delta_{\mathcal{A}}$.

C.4 Nature of Adversary

First, we define the normalized and balanced robust value function for uncertainty set $\mathcal{U} = \mathcal{U}_p^{\text{sa}}, \mathcal{U}_p^{\text{s}}$ as

$$u_{\mathcal{U}}^\pi(s) := \frac{\text{sign}(v_{\mathcal{U}}^\pi(s) - \omega_q(v_{\mathcal{U}}^\pi)) |v_{\mathcal{U}}^\pi(s) - \omega_q(v_{\mathcal{U}}^\pi)|^{q-1}}{\sigma_q(v_{\mathcal{U}}^\pi)^{q-1}}$$

which has zero mean and unit norm, that is,

$$\langle u_{\mathcal{U}}^\pi, \mathbf{1} \rangle = 0, \quad \text{and} \quad \|u_{\mathcal{U}}^\pi\|_p = 1.$$

Further, it can be re-written as a gradient of q -variance and its correlation with robust value function is q -variance.

Property 1. [19] For uncertainty set $\mathcal{U} = \mathcal{U}_p^{\text{sa}}, \mathcal{U}_p^s$, we have

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

Proposition C.2. [20] For any policy π , taking $k = u_{\mathcal{U}}^\pi$, the worst model is related to the nominal one through:

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

where $b(s, a) = \beta_{sa}, \beta_s \left[\frac{\pi(a|s)}{\|\pi_s\|_q} \right]^{q-1}$ for $\mathcal{U}_p^{\text{sa}}$ and \mathcal{U}_p^s respectively.

The above result states that for **sa**-rectangular case, the worst-case reward is independent of the employed policy. However, the worst kernel is policy dependent, and a rank one perturbation of nominal kernel which is a very surprising finding. The adversarial kernel discourages the system to move to high rewarding states, and directs towards low rewarding states.

Compared to the **sa**-rectangular case, in the **s**-rectangular case the worst reward and worst kernel have an extra dependence on the policy term $\frac{\pi(a|s)^{q-1}}{\|\pi_s\|_q^{q-1}}$. This is because the worst values cannot be chosen independently for each action in the **s**-rectangular case, but are instead dependent on the policy. Similarly to the **sa**-case, the adversarial kernel is also a rank-one perturbation of the nominal kernel.

C.5 Robust Policy Gradient

The update rule

$$\pi_{k+1} = \text{Proj}_{\pi \in \Pi} \left[\pi_k - \eta_k \nabla_\pi J_{P_k}^{\pi_k} \right], \quad (6)$$

where $J_{P_k}^{\pi_k} - J_{\mathcal{U}}^{\pi_k} \leq \epsilon \gamma^k$, finds global solution in $O(\epsilon^{-4})$ iterations [17].

Proposition C.3. [17] For initial tolerance $\epsilon \leq \sqrt{T}$, and learning rate $\eta_k = \frac{\delta}{\sqrt{T}}$ for any $\delta > 0$,

$$J_{\mathcal{U}}^* - J_{\mathcal{U}}^{\pi_k} \leq \epsilon,$$

for $k \geq CS^2A^3(S+A)^2\epsilon^{-4}$ where some constant C .

The gradient $\nabla_\pi J_{\mathcal{U}}^{\pi_k}$ of the robust return may not exist due to non-differentiability. However, sub-differential can be defined using the standard policy gradient theorem [36], as

$$\partial_\pi J_{\mathcal{U}}^\pi = \nabla_\pi J_P^\pi \Big|_{P=P_{\mathcal{U}}^\pi} = \sum_{s,a} d_{\mathcal{U}}^\pi(s) Q_{\mathcal{U}}^\pi(s, a) \nabla \pi(a|s).$$

where $Q_{\mathcal{U}}^\pi := Q_{P_{\mathcal{U}}^\pi}^\pi$ is robust Q-value, $d_{\mathcal{U}}^\pi := d_{P_{\mathcal{U}}^\pi}^\pi$ is robust occupation measure, and $P_{\mathcal{U}}^\pi$ are worst values associated with policy π , defined as

$$P_{\mathcal{U}}^\pi := \arg \inf_{P \in \mathcal{U}} J_P^\pi.$$

To compute the above sub-gradient, we require the access to the worst (adversarial) parameters $P_{\mathcal{U}}^\pi$ that we show can be computed efficiently using nominal parameters \hat{P} in close form, for s-rectangular robust uncertainty sets. Then these worst parameters, can be used to compute robust occupation measure $d_{\mathcal{U}}^\pi$ and robust Q-value $Q_{\mathcal{U}}^\pi$, which in turn can be used to compute the above robust gradient.

However, it may be possible to compute the gradient directly without computing the worst kernel, as :

Proposition C.4. [20] *The policy gradient is given by:*

$$\frac{\partial J_{\mathcal{U}}^\pi}{\partial \pi(a|s)} = \left[d^\pi(s) - c^\pi(s) \right] Q_{\mathcal{U}}^\pi(s, a),$$

where c^π is a correction term. Taking $k = v_{\mathcal{U}}^\pi$, it is

$$c^\pi(s) = \gamma \frac{\langle d^\pi, b \rangle}{1 + \gamma \langle d_k^\pi, b \rangle} d_k^\pi(s),$$

radius $b(s) = \sum_a \pi(a|s) \beta_{sa}, \beta_s \|\pi_s\|_q$ for \mathcal{U}_p^{sa} and \mathcal{U}_p^s respectively.

Intuition. Here, we have the robust Q-value instead of the non-robust one. Note that the correction term c^π , resulting from switching the occupation measure of the worst kernel against the nominal, is distribution of zero, that is

$$\langle c^\pi, \mathbf{1} \rangle = 0,$$

as $\langle d_k^\pi, \mathbf{1} \rangle = k^\top (I - P_0^\pi)^{-1} \mathbf{1} = \frac{k^\top \mathbf{1}}{1-\gamma} = 0$. Observe that the c^π is positive for those states which are on average visited more from good value states compared to bad value states, w.r.t. nominal values. Given the result, we understand that the adversary spends more time on states which are more visited from bad value states, compared to the nominal agent.

The results holds for rectangular robust MDPs. Unfortunately, nothing is known about the nature of the adversary for non-rectangular uncertainty sets.

D Helper Results

Proposition D.1 (Sherman–Morrison Formula [34].). *If $A \in \mathbb{R}^{n \times n}$ invertible matrix, and $u, v \in \mathbb{R}^n$, then the matrix $A + uv^T$ is invertible if and only if $1 + v^T A^{-1} u \neq 0$:*

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

Proposition D.2.

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

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

Proposition D.3. Let $\mathcal{U}_2^{sa}, \mathcal{U}_2^s$ be 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.

Proof. Volume of n -dimension sphere of radius r is $c_n r^n$ where $c_n \leq \frac{8\pi^2}{15}$ [24]. And to cover n -dimension sphere of radius r , we need cube of radius $2r$ whose volume is $(2r)^n$. Hence the first result $\frac{\text{vol}(\mathcal{U}_2)}{\text{vol}(\mathcal{U}_2^{sa})} = O(2^{-SA})$ immediately follows.

Now, volume of set of $X = \times_{s \in \mathcal{S}} X_s$ where X_s is an A -dimension sphere of radius r then the volume of X is $(c_A r)^S$. And the volume of an SA dimensional sphere is $c_{SA} r^{SA}$, where $\lim_{n \rightarrow \infty} c_n \rightarrow 0$ [24]. Hence the ratio of their volume is $O((c_A)^S)$, implying the other result. \square

Proposition D.4. Non-rectangular uncertainty \mathcal{U}_p can be written as an infinite union of sa -rectangular sets \mathcal{U}_p^{sa} , as

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

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} .

Proof. By definition, we have

$$\mathcal{U}_p = \{P \mid \|P - \hat{P}\|_p \leq \beta, \sum_{s'} P(s'|s, a) = 1\} \quad (7)$$

$$= \{P \mid \sum_{s,a} \|P_{sa} - \hat{P}_{sa}\|_p^p \leq \beta^p, \sum_{s'} P(s'|s, a) = 1\} \quad (8)$$

$$= \{P \mid \sum_{s,a} b_{sa}^p \leq \beta^p, \|P_{sa} - \hat{P}_{sa}\|_p^p = b_{sa}^p, \sum_{s'} P(s'|s, a) = 1\} \quad (9)$$

$$= \{P \mid \sum_{s,a} b_{sa}^p \leq \beta^p, \|P_{sa} - \hat{P}_{sa}\|_p^p \leq b_{sa}^p, \sum_{s'} P(s'|s, a) = 1\} \quad (10)$$

$$= \bigcup_{\sum_{s,a} b_{sa}^p \leq \beta^p} \{P \mid \|P_{sa} - \hat{P}_{sa}\|_p^p \leq b_{sa}^p, \sum_{s'} P(s'|s, a) = 1\} \quad (11)$$

$$= \bigcup_{b \in \mathcal{B}} \mathcal{U}_p^{sa}(b). \quad (12)$$

\square

Proposition D.5. For any vector $\|x\| = 1$, we have

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

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

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

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

And $\text{Proj}_{\mathbb{R}_+^n}(x) = x_+$ and $\text{Proj}_{\mathbb{R}_+^n}(-x) = x_-$, the rest follows. \square

Proposition D.6. For $\|k\|_p$ and $k^T \mathbf{1} = 0$, we have

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

for all π , $\|b\|_p \leq \beta$, $b \succeq 0$.

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

D.1 Binary Search Approach

Proposition D.7. For $\lambda^* = \max_{x \in C} \frac{g(x)}{h(x)}$, $F(\lambda) := \max_{x \in C} (g(x) - \lambda h(x))$, we have $F(\lambda^*) = 0$ and $f(\lambda) \geq 0 \iff \lambda^* \geq \lambda$.

Proof. • If $F(\lambda) \geq 0$ then

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

• If $F(\lambda) \leq 0$ then

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

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

D.2 NP-Hardness of non-rectangular RMDP

Reduction of Integer Program to Robust MDP

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

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

is a NP-Hard problem [37], [9] which reduces into 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 τ 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 μ .
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 6

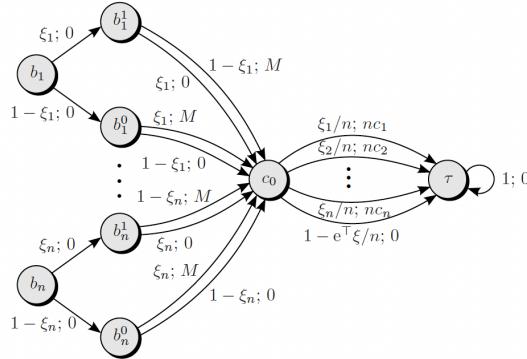


Figure 6: MDP P_ξ , and R (Figure 5 of [9]).

Robust policy evaluation is proven to be NP-hard for general uncertainty sets defined as intersections of finite hyperplanes [9]. Specifically, robust MDPs with uncertainty set $\mathcal{U}_{hard} := \{P_\xi \mid F\xi \leq g, \xi \in [0, 1]^n\}$ where P_ξ is a specially designed kernel with ladder structure with only action (effectively no decision) and a terminal state [9].

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

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

Solution:

- Case 1) If $g < 0$ then **no**.
- Case 2) If $g = 0, \zeta \geq 0$ then **yes** and $g = 0, \zeta < 0$ then **yes**.
- If $g > 0$ then compute the sum of g smallest coordinate of c , and this sum is less/equal than ζ then answer is **yes**, otherwise **no**.

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 uncertainty have one global constraint and a small radius β , which corresponds to a Knapsack Problem with a small budget (IP with one constraint and a small g) which are much easier to solve [38, 37].

We can thus conclude that the hardness result of [9] doesn't apply to our uncertainty case.

E Dual Formulation

s-rectangular uncertainty sets. Now, we turn our attention to the uncertainties coupled across different actions in each state.

Lemma E.1. *For the s-rectangular uncertainty set \mathcal{U}_p^s , the robust return can be written as*

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

where $b \in \mathbb{R}^{S \times A}$, $b_s = b_{s.}$, and $b^\pi(s) = \sum_a \pi(a|s)b_{sa}$.

Proof. The proof follows similarly to the **sa**-rectangular case, and can be found in the appendix. The key additional step is to decompose the **s**-rectangular uncertainty set \mathcal{U}_p^s into a union of **sa**-rectangular uncertainty sets \mathcal{U}_p^{sa} . \square

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 $\mathcal{B} = \{b \in \mathbb{R}^{S \times A} \mid \|b_s\|_p \leq \beta_s\}$.

Comparing the penalty term from the previous results in [19, 20], 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.

Lemma E.2. For the sa-rectangular uncertainty set $\mathcal{U} = \mathcal{U}_p^{sa}(\beta)$ with radius vector $\beta \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$, the robust return can be written as the following optimization problem,

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

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

Proof. From [20], 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 one-perturbation of P . In other words,

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

for some $k \in \mathbb{R}^{\mathcal{S}}$ 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,

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

□

Lemma E.3. 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.

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

□

Lemma E.4. For $\mathcal{U} = \mathcal{U}_p$, the robust return can be written as the following optimization problem

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

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

Proof. Now,

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

□

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

$$\sigma_q(v_{\mathcal{U}}^\pi) = \max_{\|k\|_p = 1, 1^T k = 0} \frac{k^T v_R^\pi}{1 + \gamma k^T v_\beta^\pi}, \quad (13)$$

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

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

Intuition on the adversary. We know that the $\sigma(v_{\mathcal{U}}^\pi)$ is the penalty for robustness, that is

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

Knowing $\sigma(v_{\mathcal{U}}^\pi)$ how it arises, sheds the light on the working of the adversary in (13). Furthermore, recall that if $P = P - \beta k^T$ then

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

Basically, k is the direction the adversary discourages the perturbation of the kernel. And the optimal direction k that the adversary chooses is the one that optimizes (13).

s-rectangular uncertainty sets. Now, we move our attention to the coupled uncertainty case.

Lemma E.5. *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. The proof follows similarly to the **sa**-rectangular case, and can be found in the appendix. The key additional step is to decompose the s-rectangular uncertainty set \mathcal{U}_p^s into a union of **sa**-rectangular uncertainty sets \mathcal{U}_p^{sa} . \square

Comparing the penalty term from the previous results in [19, 20], we get

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

Again, the above relation looks very interesting as it relates the robust term on LHS with non-robust terms on RHS.

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 suggests that the adversary's payoff is maximized by maximizing the expected uncertainty in the trajectories.

Looking at GSTD from two angles From the binary search method at section D.1, we know that $\sigma_q(v_{\mathcal{U}}^\pi)$ is the solution to the following:

$$\max_{\|k\|_p \leq 1, 1^T k = 0} \left[k^T v_R^\pi - x(1 + \gamma k^T v_\beta^\pi) \right] = 0 \quad (14)$$

$$\max_{\|k\|_p \leq 1, 1^T k = 0} k^T (v_R^\pi - \gamma x v_\beta^\pi) = x \quad (15)$$

$$\max_{\|k\|_p \leq 1, 1^T k = 0} k^T D^\pi (R^\pi - \gamma x \beta^\pi) = x \quad (16)$$

$$v_R^\pi = \gamma \frac{k^T v_R^\pi}{1 + \gamma k^T v_\beta^\pi} v_\beta^\pi = \gamma \sigma(v_U^\pi) v_\beta^\pi.$$

Its maybe possible to make this process online, simultaneously updating $x \rightarrow v_U^\pi$, v and k .

F Robust Policy Evaluation

Lemma F.1. *The robust return can be expressed as*

$$J_{U_p}^\pi = J^\pi - \lambda^*,$$

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

Proof. We want to evaluate the following

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

This is of the form $\max_x \frac{f(x)}{g(x)}$. Then according to Proposition D.7, we have $f(\lambda^*) = 0$ and $f'(\lambda) > 0$ if and only if $\lambda^* > \lambda$, where

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

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

□

G Robust Policy Improvement

In the previous section, , we found that the worst kernel is a rank-one perturbation of the nominal kernel. Exploiting this, we developed a method to evaluate the robust policy efficiently. This methods also computes the perturbation (βk^T), and consequently the worst kernel.

We can use it directly to compute the gradient w.r.t. the policy for this computed worst kernel. Then, we can apply policy improvement by gradient ascent as in [17]:

$$\pi_{n+1} = \text{proj} \left[\pi_n + \eta_k \nabla_\pi J_{P_n}^\pi |_{\pi=\pi_n} \right], \quad (17)$$

where P_n is the estimate of the worst kernel for π_k . This has global convergence guarantees with iteration complexity of $O(\epsilon^{-4})$ [17].

Alternatively, we can derive the policy gradient for the approximated perturbation, as done in the result below.

Lemma G.1 (Approximate Policy Gradient Theorem). *Given the transition kernel $P = \hat{P} - \beta k^\top$, the return is given as*

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

and the gradient is given as

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

Proof. The expression for the return directly follows from the inverse matrix theorem as proved in [20]. Now, we move our attention towards evaluating the gradient, using the policy gradient theorem [27] in the format used in [39] (see appendix).

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

□

The main advantage of the above policy gradient is that constituents terms like $J_\beta^\pi, v_\beta^\pi, Q_\beta^\pi$, in addition to nominal terms $J_R^\pi, v_R^\pi, Q_R^\pi$ can be computed easily with bootstrapping exploiting Bellman operators.

H Evaluation of $\max_{x,y} x A y$

Algorithm 1 requires an oracle access to

$$\max_{\|b\|_p \leq \beta, \|k\|_p \leq 1, k^\top k = 0} k^\top A b,$$

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

Proposition H.1. [Orthogonality Equivalence] Let $\mathcal{K} = \{k \mid \|k\|_p \leq 1, 1^\top k = 0\}$, and $\mathcal{W} = \{k^T(I - \frac{11^T}{S}) \mid \|k\|_p \leq 1\}$. Then we have,

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

Proof. Now let $k \in \mathcal{K}$, then $k^T(I - \frac{11^T}{S}) = k^\top \in \mathcal{W}$. Now the other direction, let $k \in \mathcal{W}$, then $\langle k^T(I - \frac{11^T}{S}), 1 \rangle = 0$ by construction and $\|k^T(I - \frac{11^T}{S})\|_p \leq \|k\|_p \leq 1$, this implies $k^T(I - \frac{11^T}{S}) \in \mathcal{K}$. \square

The above result implies that

$$\begin{aligned} \max_{\|b\|_p \leq \beta, \|k\|_p \leq 1, 1^\top k = 0} k^T Ab &= \max_{\|b\|_p \leq \beta, k \in \mathcal{K}} k^T Ab \\ &= \max_{\|b\|_p \leq \beta, k \in \mathcal{W}} k^T Ab, \quad (\text{as } \mathcal{K} = \mathcal{W} \text{ from above Proposition H.1}) \\ &= \max_{\|b\|_p \leq \beta, \|k\|_p = 1} k^\top(I - \frac{11^T}{S})Ab, \quad (\text{def. of } \mathcal{W}). \end{aligned}$$

Further, we have equivalence of optimizers

$$\arg \max_{\|k\|_p \leq 1, 1^\top k = 0, \|b\|_p \leq \beta} k^T Ab = \left\{ (b^*, (I - \frac{11^T}{S})k^*) \mid (b^*, k^*) \in \arg \max_{\|k\|_p = 1, \|b\|_p \leq \beta} k^\top(I - \frac{11^T}{S})Ab \right\}.$$

Proposition H.2. The solving of

$$\max_{\|k\|_p \leq 1, 1^\top k = 0, \|b\|_p \leq \beta} k^T Ab, \quad \text{is equivalent to} \quad \max_{\|k\|_p = 1, \|b\|_p \leq \beta} k^\top(I - \frac{11^T}{S})Ab.$$

Proof. Directly follow from the proposition above. \square

H.1 Eigenvalue Approach (Spectral Methods)

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

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

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

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

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

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

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

- Zero Order Solution:

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

- First order solution:

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

- Second order solution:

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

- Third order solution:

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

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

- Zero order upper bound: λ_1
- First order upper bound:

$$\sqrt{\sum_i \lambda_i c_i},$$

where

$$c_i = \begin{cases} \langle v_i, u_i \rangle^2, & \text{if } \sum_{j=1}^i \langle v_j, u_j \rangle^2 \leq 1, \\ 1 - \sum_{j=1}^{i-1} \langle v_j, u_j \rangle^2, & \text{if } \sum_{j=1}^i \langle v_j, u_j \rangle^2 \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

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

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

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

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

Proof. We have $\|v_1^+\| \geq \frac{1}{\sqrt{2}}$ from Proposition D.5. Let $u = \frac{(v_1)_+}{\|(v_1)_+\|} = \sum_i \sigma_i v_i$, where $\sigma_i = \langle u, v_i \rangle$,

we have

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

Rest follows.

□

Proposition H.4 (First Order is Better than the First).

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

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

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

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

Rest follows.

□

Proposition H.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 equal to $\max_{\|x\|_2 \leq 1, x \geq 0} \|Ax\|_2$ when A is rank two.

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

The intuition behind this approach is that the matrix $A^T A$ can be decomposed into its eigenvalues and eigenvectors, representing the principal directions of the transformation applied by A . The eigenvector corresponding to the largest eigenvalue provides the direction of maximum scaling for A . However, since the solution is constrained to the nonnegative orthant ($x \geq 0$), we adjust the eigenvectors by only considering their positive parts. The method identifies an

approximate solution u_j by selecting and normalizing the positive part of the eigenvector that contributes the most to the objective function.

Algorithm 4 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.

H.1.1 Reformulation

Proposition H.6.

$$\max_{\|x\|_2 \leq 1, x \geq 0} x A^T A x = \max_{V\sigma \succeq 0, \|\sigma\|_2 \leq 1} \sum_i \lambda_i \sigma_i^2, \quad (\text{where } VV^\top = I, \lambda_i \geq 0)$$

Proof. $x = \sum_i \sigma_i v_i$, where $\sigma_i = \langle x, v_i \rangle$, we have

$$\begin{aligned} x^T A^T A x &= (\sum_i \sigma_i v_i) (\sum_i \lambda_i v_i v_i^T) (\sum_i \sigma_i v_i) \\ &= \sum_i \lambda_i (\sigma_i)^2, \quad (\text{as } v_i \text{ are orthogonal}). \end{aligned}$$

Further, $x = V\sigma$. This map is bijection. Rest follows. \square

Since, $\lambda_i \geq 0$, the objective $\langle \lambda, \sigma \rangle$ is convex in σ , further the domain $\{\sigma \mid V\sigma \succeq 0, \|\sigma\|_2 \leq 1\}$ is intersection of a polytope and a sphere, hence convex. This makes the following

$$\max_{V\sigma \succeq 0, \|\sigma\|_2 \leq 1} \sum_i \lambda_i \sigma_i^2$$

as convex objective with convex domain.

Proposition H.7.

$$\max_{\|x\|_2 \leq 1, x \succeq 0} x A^T A x = \max_{b \in B} \sum_i \lambda_i b_i,$$

where $B = \{b \mid b_i = \langle v_i, x \rangle^2, x \succeq 0\}$.

Proof. $x = \sum_i \sigma_i v_i$, where $\sigma_i = \langle x, v_i \rangle$, we have

$$\begin{aligned} x^T A^T A x &= (\sum_i \sigma_i v_i) (\sum_i \lambda_i v_i v_i^T) (\sum_i \sigma_i v_i) \\ &= \sum_i \lambda_i (\sigma_i)^2, \quad (\text{as } v_i \text{ are orthogonal}). \end{aligned}$$

Rest follows. \square

Proposition H.8. *The set*

$$B = \{b \mid b_i = \langle v_i, x \rangle^2, x \succeq 0, \|x\|_2 = 1\},$$

is convex.

Proof. Let $b, b' \in B$ be corresponding point for $x, x' \succeq 0$, and $\|x\|, \|x'\| = 1$ respectively. Now make circle with origin, x and x' . The minor arc containing x and x' lies entirely in the \mathbb{R}_+^n . The x lying on the arc will generate a line connecting b and b' . Hence B is convex. \square

H.2 Experimental Verification

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

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

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

H.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 7

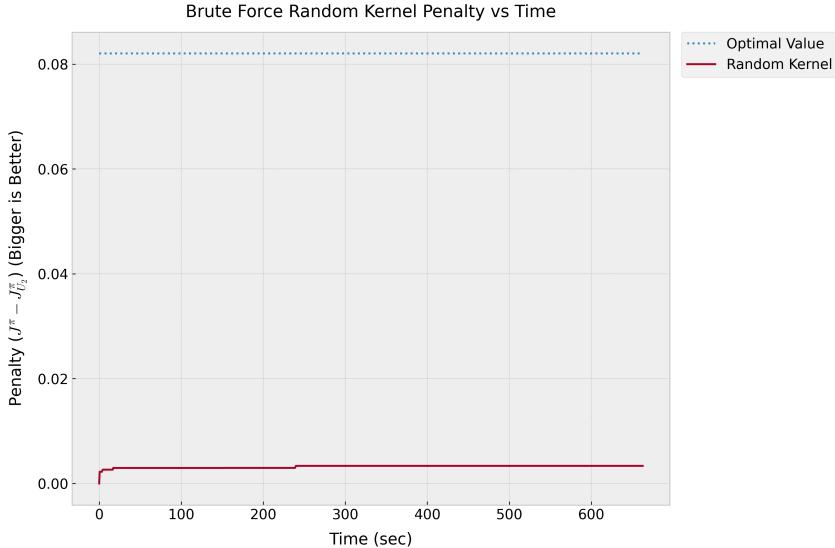


Figure 7: Random Kernel Guess takes exponentially long time to converge. While Algorithm 1 only took 0.14 sec to find the optimal value.

H.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.

H.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.

H.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.

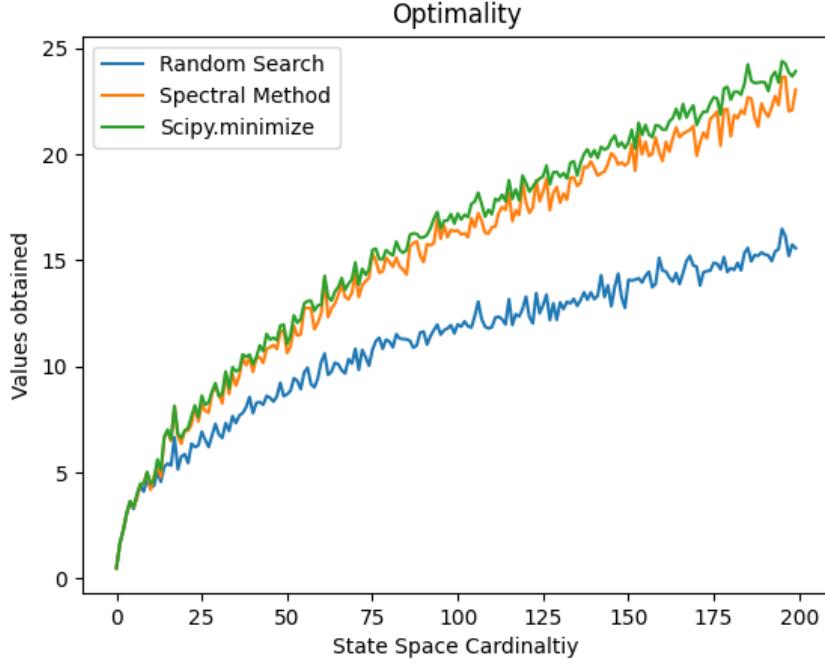


Figure 8: Comparison of optimality across methods.

n	Optimal values attained			Time taken		
	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.

H.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:

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

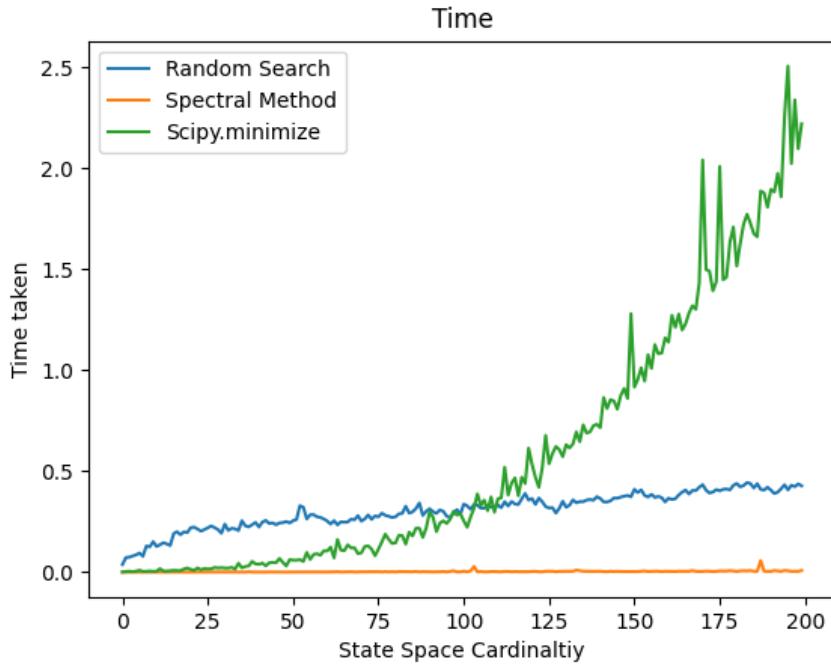


Figure 9: Comparison of computational time across methods

State Space Cardinality and Random matrix Generation

- **State Space Cardinality (n):** The dimension of the problem, denoted by n , represents the state space cardinality. In the experiments, n varied from 1 to 300 to analyze the scalability of the methods.
- **Matrix Generation:** The matrix $A \in \mathbb{R}^{n \times n}$ was generated as a random matrix with entries sampled from a standard normal distribution:

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

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

- 10000 random vectors x were generated for Brute Search Method.

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

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

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

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

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

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

Hardware and Software Specifications The experiments were conducted on the following hardware and software setup:

- **Model Name:** MacBook Pro (2023 model).
- **Model Identifier:** Mac14,7.
- **Chip:** Apple M2 with 8 cores (4 performance and 4 efficiency cores).
- **Memory:** 16 GB Unified Memory.
- **Operating System:** macOS Ventura.
- **Programming Language:** Python 3.9.
- **Libraries Used:**
 - `numpy` for numerical computations.
 - `scipy` for numerical optimization.
 - `matplotlib` for generating plots.
 - `time` for recording computational times.

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

I Convexity of \mathcal{D}

I.1 MDP Configuration

We define an MDP with the following parameters:

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

I.2 Dimensionality Reduction via PCA

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

- We **retain the top 10 components** to capture the dominant variations in the dataset.
- The **explained variance ratio** is visualized to assess how much information each component retains.
- **2D and 3D projections** of the first few principal components are generated for visualization.

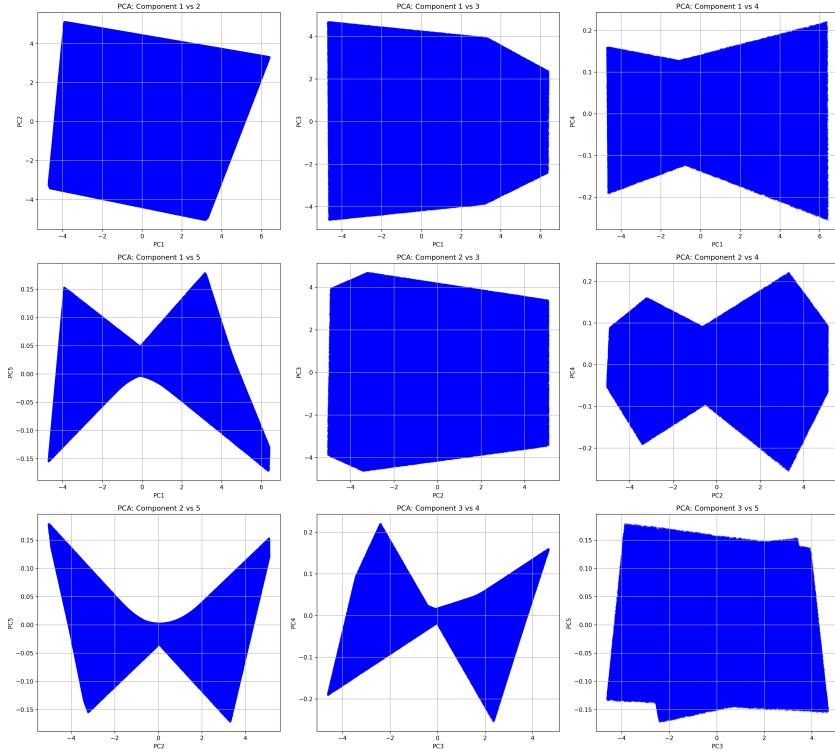


Figure 10: 2D PCA projections of the first 5 components.

I.3 Random Linear Projections

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

- **2D Random Projections:** The data is projected onto **randomly chosen 2D subspaces**.
- **3D Random Projections:** The data is projected onto **randomly chosen 3D spaces**.

J Experiments: Robust Policy Evaluation

Setting: Randomly generate \hat{P} is nominal kernel, reward function R , and π is a policy. Discount factor $\gamma = 0.9$, uncertainty radius $\beta = 0.01$, initial distribution $\mu = \text{uniform distribution}$.

PCA: Components 1 vs 2 vs 3

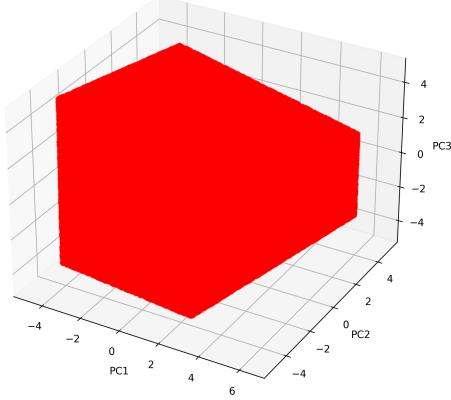


Figure 11: 3D PCA projection of the first three principal components.

Notations: Value function $v^\pi = (I - \gamma \hat{P}^\pi)^{-1} R^\pi$, Occupancy matrix $D^\pi = (I - \gamma \hat{P}^\pi)^{-1} R^\pi$, occupation measure $d^\pi = \mu^T (I - \gamma \hat{P}^\pi)^{-1} R^\pi$, return $J_P^\pi = (I - \gamma P^\pi)^{-1} R^\pi$.

In this section, we evaluate the performance of Algorithm 1 for robust policy evaluation, focusing on the case $p = 2$. The algorithm requires computing $F(\lambda)$ at each iteration, which involves solving the constrained matrix norm problem $\max_{x \in \mathcal{B}} \|Ax\|_2$. This can be efficiently handled using our spectral Algorithm 2. Figures 4 and 5 compare four methods for computing the robust return, specifically the penalty term $J^\pi - J_{\mathcal{U}_2}^\pi$:

- **SLSQP from scipy [35]** : This is A semi-brute force approach that uses Lemma 3.3. It computes the penalty term via `scipy.optimize.minimize` to directly optimize over (b, k) . This is equivalent to optimize over only rank-one perturbation of nominal kernel as $a(b, k)$ corresponds to selecting a rank-one perturbation of the nominal kernel $P = \hat{P} - bk^\top$. Note that this method is local, hence can sometime be stuck in very bad local solution.
- **Binary search Algorithm 1** : Uses the binary search Algorithm 1, and spectral Algorithm 2 for computing $F(\lambda)$ in each iteration.
- **Random Rank-One Kernel Sampling** : A semi-brute force approach that uses Lemma 3.3 to sample random pairs $(b, k) \in \mathcal{B}, \mathcal{K}$, empirically maximizing the penalty term. Since choosing (b, k) corresponds to selecting a rank-one perturbation of the nominal kernel $P = \hat{P} - bk^\top$, the method is named accordingly.
- **Random Kernel Sampling** : A brute-force approach that samples random kernels directly from the uncertainty set \mathcal{U}_2 , computing the empirical minimum as an estimate of the robust return.

Figure 14 presents the penalty value $(J^\pi - J_{\mathcal{U}_2}^\pi)$ computed by different methods across various state space sizes while keeping the action space fixed, with each method given the same computational time. Our binary search Algorithm 1 performs significantly better in both

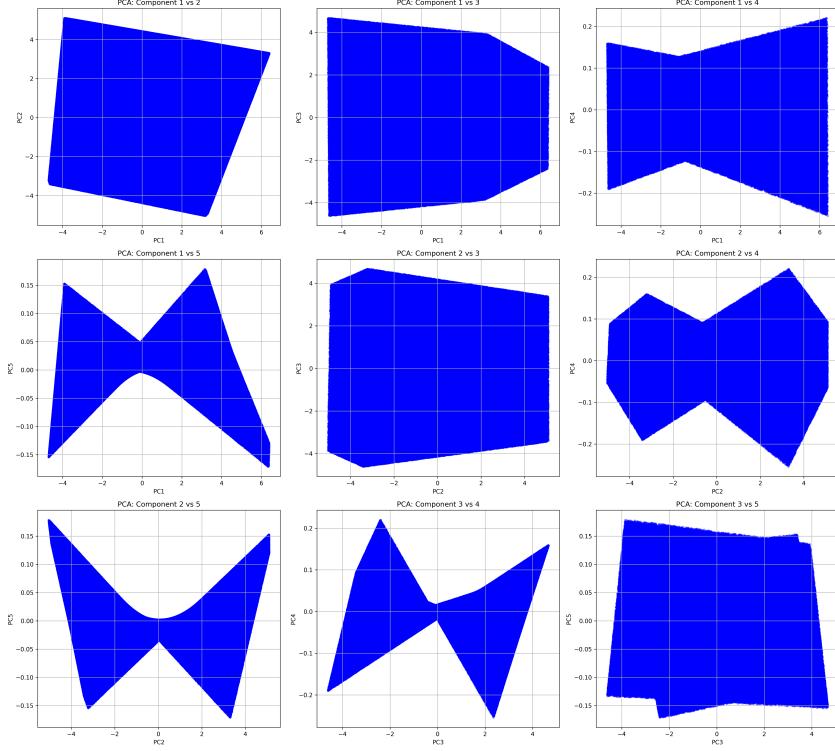


Figure 12: 2D Random Projections of the Data.

variations compared to brute-force (random kernel sampling) and semi-brute (random sampling of rank-one perturbations of the nominal kernel) approaches. Notably, the `scipy SLSQP` variant performs slightly better on average, but the spectral Algorithm 2 is more reliable. This is expected, as the spectral method is global, while `scipy SLSQP` is a local optimizer and thus more prone to getting stuck in suboptimal solutions.

J.1 Penalty Values ($J^\pi - J_{\mathcal{U}_2}^\pi$) vs Sample Size for Different Algorithms

Figure 16, figure 17 and figure 18 show Penalty Values ($J^{pi} - J_{\mathcal{U}_2}^\pi$) for different values of β calculated using different algorithms.

The experiment as shown in figure 18 shows that the SLSQP algorithm is not robust for this problem and can end up in local minima multiple times.

Figure 15 illustrates the convergence of different approaches over time (in seconds) for a fixed state space size ($S = 8$). The spectral variation of Algorithm 1 converges rapidly, while the SLSQP variation gradually approaches the same performance. In contrast, the brute-force method shows slow, logarithmic improvement. This behavior arises because brute-force methods require an exponential number of samples (in the dimensionality of the problem) to adequately explore all directions. In comparison, our binary search Algorithm combined with the spectral Algorithm 2 achieves significantly better efficiency, with a complexity of $O(S^3 A^3 \log \epsilon^{-1})$.

More details of these experiments along with others can be found in the appendix, and codes are available at <https://anonymous.4open.science/r/non-rectangular-rmdp-77B8>.

Our experiments confirm the efficiency of our binary search Algorithm 1 for robust policy evaluation, significantly outperforming brute-force approaches in both accuracy and convergence

Random 3D Projection 1

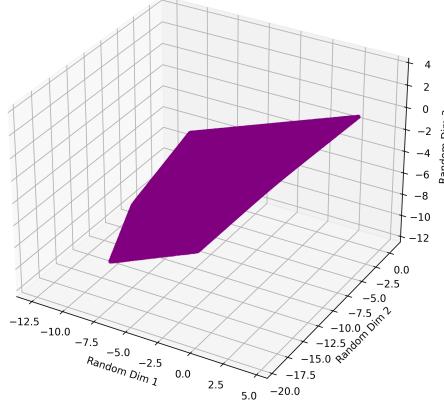


Figure 13: 3D Random Projection Example.

speed.

Algorithm 5 Binary Search for Robust Policy Evaluation for Uncertainty set \mathcal{U}_p

J.2 Our Method

- 1: **Input:** Tolerance $\epsilon = 0.001$, $\beta = 0.01$
- 2: **Initialize:** $\lambda = \frac{0.5}{1-\gamma}$, $\lambda_{max} = \frac{1}{1-\gamma}$, $\lambda_{min} = 0$
- 3: **while** Tolerance is not met: $f(\lambda) > \epsilon$ **do**
- 4: Compute:

$$f(\lambda) = \max_{\|b\|_p \leq 1, b \succeq 0} \beta \gamma \|\Phi(v^\pi d^{\pi^\top} - \lambda D^\pi) H^\pi b\|_q,$$

where $\Phi = I - \frac{\mathbf{1}\mathbf{1}^\top}{S}$ is projection matrix and $(H^\pi b)(s) = \sum_a \pi(a|s)b(s, a)$ is policy averaging operator.

- 5: Bisection: If $f(\lambda) > \epsilon$, set $\lambda_{min} = \lambda$, if $f(\lambda) < \epsilon$, set $\lambda_{max} = \lambda$
 - 6: **end while**
 - 7: **Output:** Robust return: $J_{\mathcal{U}_p}^\pi = J^\pi - \lambda$.
-

J.3 Robust Penalty Function

We have defined robust penalty function in 3.2 as

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

Figure 19 and Figure 20 show graph of $F(\lambda)$ vs λ for different value of β for fixed value of $S=100$ and $A=10$.

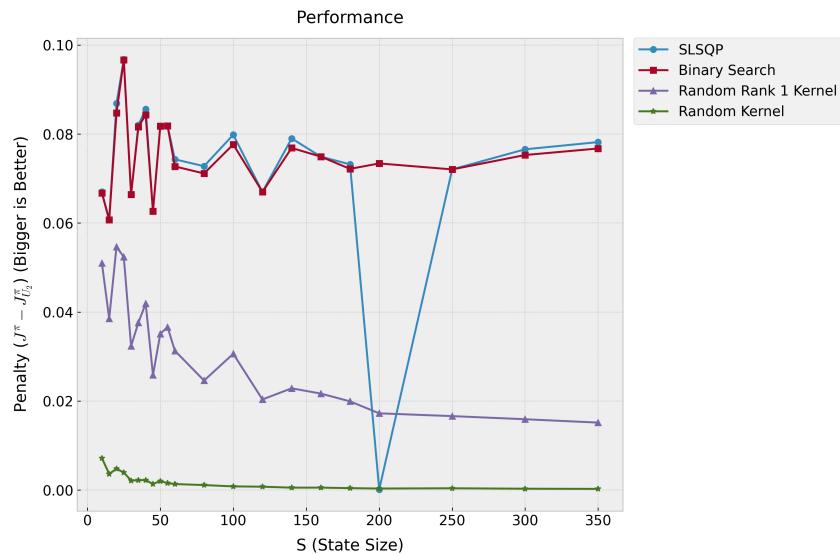


Figure 14: Performance of Robust Policy Evaluation methods with equal amount of time.

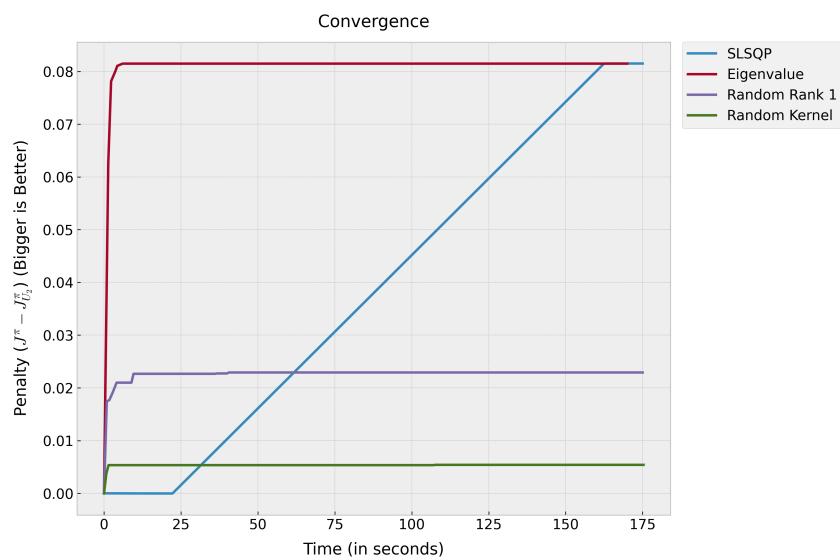


Figure 15: Convergence of Robust Policy Evaluation Methods

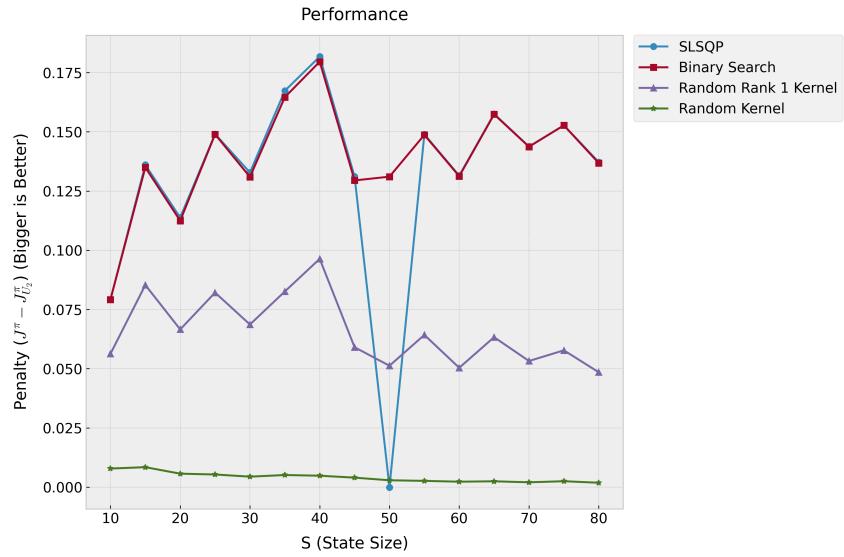


Figure 16: $\beta=0.1$, $A=8$

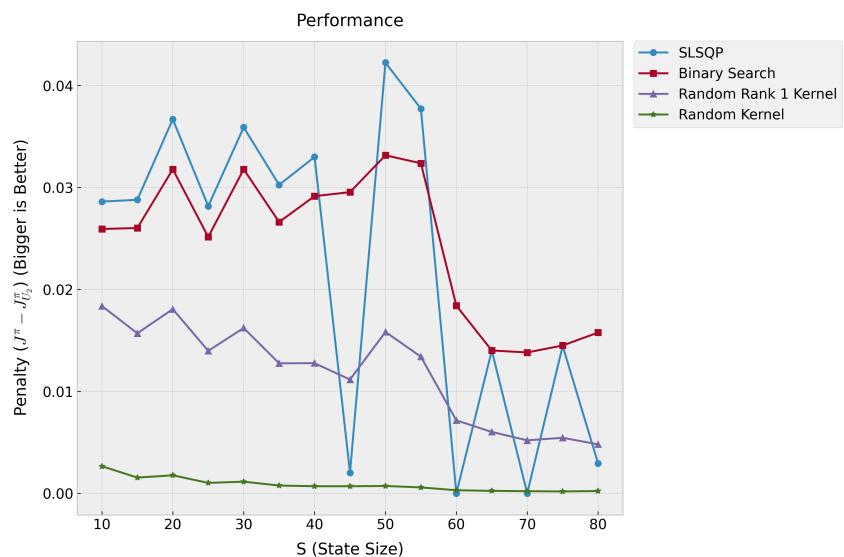


Figure 17: $\beta=0.02$, $A=8$

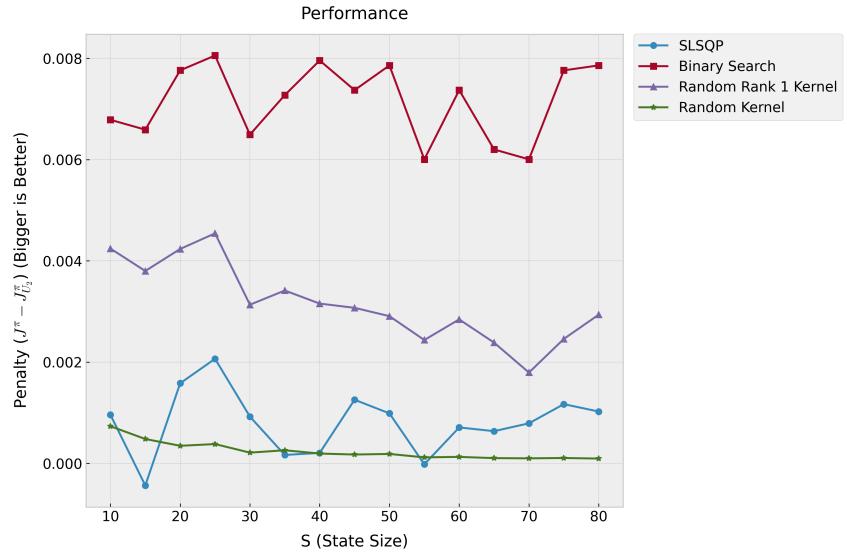


Figure 18: $\beta=0.005$, $A=8$

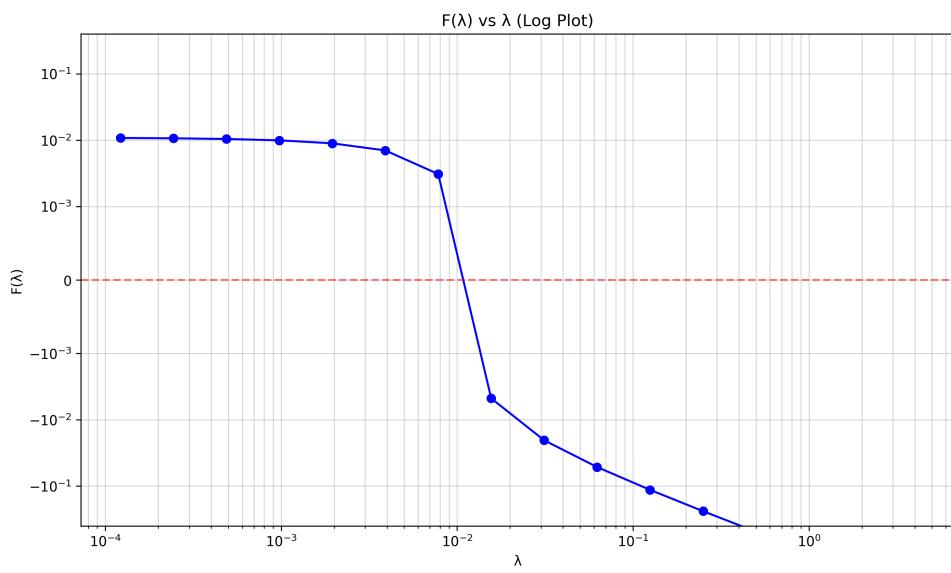


Figure 19: Robust Penalty Function $F(\lambda)$ vs λ for $S = 100$, $a = 10$, $\beta = 1/S$

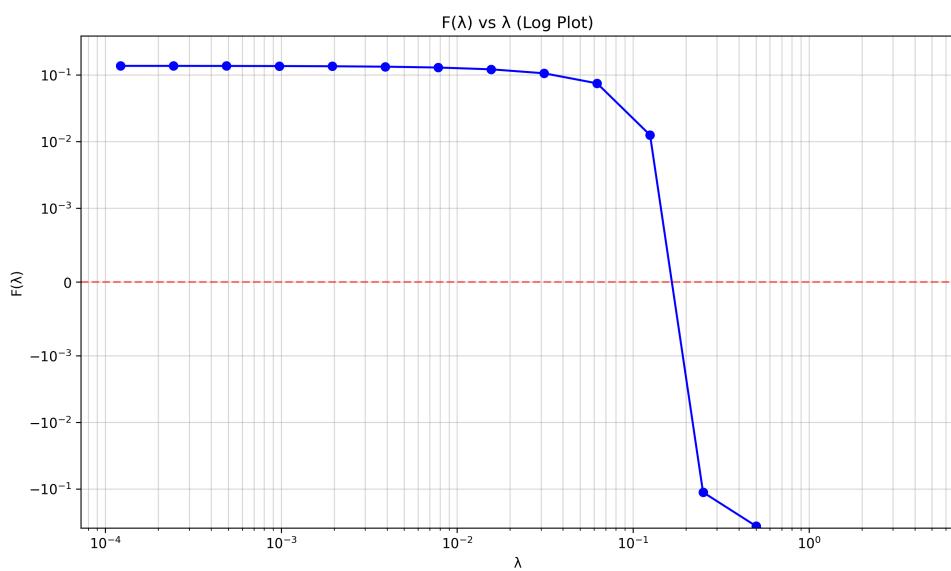


Figure 20: Robust Penalty Function $F(\lambda)$ vs λ for $S = 100$, $a = 10$, $\beta = 0.1$