Soft-Robust Algorithms for Batch Reinforcement Learning

Elita A. Lobo , **Mohammad Ghavamzadeh** , **Marek Petrik**
[1]University of New Hampshire
[2]University of New Hampshire
[3]Google
{eal1063@wildcats.unh.edu, ghavamza@google.com, mpetrik@cs.unh.edu

Abstract

In reinforcement learning, robust policies for high-stakes decision-making problems with limited data are usually computed by optimizing the percentile criterion, which minimizes the probability of a catastrophic failure. Unfortunately, such policies are typically overly conservative as the percentile criterion is non-convex, difficult to optimize, and ignores the mean performance. To overcome these shortcomings, we study the soft-robust criterion, which uses risk measures to balance the mean and percentile criterion better. In this paper, we establish the soft-robust criterion's fundamental properties, show that it is NP-hard to optimize, and propose and analyze two algorithms to approximately optimize it. Our theoretical analyses and empirical evaluations demonstrate that our algorithms compute much less conservative solutions than the existing approximate methods for optimizing the percentile criterion.

1 Introduction

Markov Decision Process (MDP) is an established model for optimizing returns in sequential decision-making problems [Puterman, 2005; Sutton and Barto, 2018]. In the batch Reinforcement Learning (RL) setting, MDPs must be estimated from logged data. However, without the ability to explore, the transition probability estimates derived from the logged data are inevitably imprecise. Such errors in the transition model estimate often result in learning policies that fail catastrophically when deployed. In this paper, we aim to compute robust policies from logged data in a way that accounts for the uncertainty in transition models. As is common in prior work, we use parametric Bayesian models to represent the uncertainty in the transition model estimates [Xu and Mannor, 2006, 2009, 2012; Delage and Mannor, 2010; Petrik *et al.*, 2016; Russel and Petrik, 2019]. The most common robust objective in this setting is the *percentile criterion*, which maximizes a given α-quantile of the expected returns [Delage and Mannor, 2010; Tamar *et al.*, 2014; Chow *et al.*, 2018; Russel and Petrik, 2019].

Despite its simplicity and popularity, the percentile criterion objective suffers from three major shortcomings. *First*, it ignores the mean performance even when there are multiple optimal random policies [Iancu and Trichakis, 2014]. This behavior gives rise to policies that are unnecessarily pessimistic. *Second*, the percentile criterion also ignores the tail of the distribution below the $(1-\alpha)$ quantile in addition to ignoring the mean. In problems with heavy tail risks, such as some portfolio optimization settings [Krokhmal *et al.*, 2003], the percentile criterion learns over-optimistic policies that may result in disastrous worst-case outcomes [Rockafellar and Uryasev, 2000]. *Third*, the percentile criterion is non-convex which complicates its analysis and optimization. Optimizing this criterion using robust optimization methods [Ben-Tal *et al.*, 2010] requires constructing a convex uncertainty set that accounts for $\alpha\%$ of the model parameter values [Wiesemann *et al.*, 2013; Russel and Petrik, 2019]. In practice, these sets are constructed using statistical confidence intervals. Recent work [Russel and Petrik, 2019; Gupta, 2019] has shown that such uncertainty sets (that are confidence regions) are often very large and lead to overly conservative policies.

To overcome the limitations of the percentile criterion, we adopt the *soft-robust criterion* [Ben-Tal *et al.*, 2010]. This criterion optimizes a convex combination of the mean and a robust performance and is itself convex. We measure the robust performance in soft-robust criterion using the Conditional Value at Risk (CVaR) measure [Rockafellar and Uryasev, 2000], which represents the mean of the expected return of the worst $(1-\alpha)\%$ of the models. The CVaR measure bounds the percentile criterion from below and takes into account the tail risk. We note that although the soft-robust criterion has been widely studied in finance and risk-averse RL, (see e.g., Prashanth 2014; Chow and Ghavamzadeh 2014; Tamar *et al.* 2015; Tang *et al.* 2019), it is not well-understood in the context of robust RL. We discuss related work in greater depth in Appendix G.

We begin by analyzing a new *static* soft-robust formulation for RL, which differs significantly from earlier formulations [Xu and Mannor, 2012; Derman *et al.*, 2018; Mankowitz *et al.*, 2020]. Since the earlier formulations embed the robustness within the dynamic programming equations, we refer to them as robust objectives with *dynamic uncertainty model*. Dynamic uncertainty model can be interpreted, in certain cases, as an assumption that the uncertain transition model changes randomly in every time step [Xu and Mannor, 2012]. Our *static uncertainty model*, in contrast, assumes that the model is uncertain but it does not change throughout the execution. We show

that, despite being computationally challenging, the static uncertainty model has two important advantages. *First*, the static uncertainty model is less conservative than the dynamic one because it allows the agent to exploit any information about the uncertain parameters to make better decisions. *Second*, because the static uncertainty model accounts for model uncertainty more accurately, it effectively eliminates over-optimism driven by model uncertainty, also known as the *optimizer's curse* [Smith and Winkler, 2006].

In addition to describing the static soft-robust criterion, we derive two new algorithms for optimizing it in Section 4. The first algorithm is a new mixed-integer linear program (MILP) formulation that computes the optimal deterministic soft-robust policy. While this non-convex formulation obviously does not scale beyond small problems, it is unlikely that more tractable optimal algorithms exist, because the soft-robust objective is NP hard. The second algorithm approximates the static objective by a robust MDP and scales to continuous problems using value function approximation. Finally, we derive a new structural approximation error bounds for the robust MDP formulation in Section 5. Our experimental results in Section 6 illustrate the algorithms on two small, but realistic, problem domains.

2 Preliminaries

We model the agent's interaction with the environment as an MDP [Puterman, 2005]. An MDP is a tuple $(\mathcal{S}, \mathcal{A}, P, r, p_0, \gamma)$ that consists of a set of states $\mathcal{S} = \{1, 2, \ldots, S\}$, a set of actions $\mathcal{A} = \{1, 2, \ldots, A\}$, a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, a transition probability function $P : \mathcal{S} \times \mathcal{A} \to \Delta^S$, an initial state distribution $p_0 \in \Delta^S$, and a discount rate $\gamma \in (0, 1)$. The symbol Δ^S denotes the S-dimensional probability simplex. Our objective is to maximize the infinite-horizon discounted return. We also assume that $|r(s, a, s')| \leq r_{\max} \in \mathbb{R}$ for all $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$.

We consider a *batch* RL setting (e.g., Lange *et al.* 2012), where the reward function $r(s, a, s')$ is known but the true transition model $P^\star$ is unknown and must be estimated from the batch of data $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^M$, where $s'_i \sim P^\star(s_i, a_i, \cdot) = p^\star_{s_i,a_i}$. We take a parametric Bayesian approach to model the uncertainty over the true transition model $P^\star$ [Delage and Mannor, 2010; Xu and Mannor, 2012; Russel and Petrik, 2019; Derman *et al.*, 2019]. In this approach, the transition model $P^\star$ is a random variable. Using the batch of data $\mathcal{D}$, one can derive a posterior distribution over $P^\star$ conditional on $\mathcal{D}$, which is denoted by $\hat{P} = P^\star \mid \mathcal{D}$ and distributed according to a measure f. As it is common in methods like sample average approximation (SAA) [Shapiro *et al.*, 2014], we approximate $\hat{P}$ by finite samples $\hat{P}^\omega$, $\omega \in \Omega$ with weights f_ω, $\omega \in \Omega$ and sample size $N = |\Omega|$. The samples $\hat{P}^\omega$ in our experiments come from MCMC and can be computed using tools like Stan or PyMC3 (e.g., Gelman *et al.* 2014).

A policy $\pi : \mathcal{S} \to \Delta^A$ prescribes the probability of taking an action $a \in \mathcal{A}$ when the agent is in a state $s \in \mathcal{S}$. We denote by $\Pi = (\Delta^A)^\mathcal{S}$ and $\Pi_D = \mathcal{A}^\mathcal{S}$, the sets of all randomized and deterministic policies, respectively. For a given realization of transition model P, the initial state distribution p_0, and a policy π, the expected discounted return is defined as

$$\rho(\pi, P) \;=\; \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \cdot r(S_t, A_t, S_{t+1})\right],$$

where $S_0 \sim p_0$, $A_t \sim \pi(S_t)$, and $S_{t+1} \sim P(S_t, A_t, \cdot)$.

Percentile Criterion and Risk Measures. Percentile optimization has been commonly used to derive robust policies and risk-adjusted discounted returns for an MDP under uncertainty [Delage and Mannor, 2010; Russel and Petrik, 2019]. The chance-constrained objective that this criterion optimizes is of the form

$$\max_{\pi \in \Pi,\, y \in \mathbb{R}} \left\{ y \;\mid\; \mathbb{P}_{\hat{P} \sim f}[\rho(\pi, \hat{P}) \geq y] \geq \alpha \right\}, \tag{2.1}$$

where y lower-bounds the true expected discounted return with confidence $\alpha \in [0, 1]$. Increasing the value of α in (2.1) increases the confidence that the return $\rho(\pi, \hat{P})$ is at least y. Alternatively, the percentile criterion in (2.1) can be interpreted using the framework of risk measures as:

$$\max_{\pi \in \Pi} \operatorname{VaR}^\alpha_{\hat{P}} \left[\rho(\pi, \hat{P})\right], \tag{2.2}$$

where VaR is the well-known value at risk measure [Shapiro *et al.*, 2014] that is defined for a random variable Z with PDF g and CDF G as $\operatorname{VaR}^\alpha_g[Z] = \inf\{z \in \mathbb{R} \mid G(z) \geq 1 - \alpha\}$.

Robust MDPs. Robust MDPs (RMDPs) (e.g., Iyengar 2005; Wiesemann *et al.* 2013) are commonly used to optimize the percentile criterion [Delage and Mannor, 2010; Xu and Mannor, 2012; Petrik *et al.*, 2016; Russel and Petrik, 2019]. We will also use them in this paper to approximately optimize the soft-robust criterion. RMDPs generalize MDPs by allowing for ambiguous transition models. An RMDP consists of the same components as an MDP, except the fixed transition function P is replaced by an ambiguity set $\mathcal{P} \subseteq \{P : \mathcal{S} \times \mathcal{A} \to \Delta^S\}$ of plausible transition models. The objective is to compute a policy $\pi \in \Pi$ that maximizes the return for the worst-case realization of $P \in \mathcal{P}$, i.e.,

$$\max_{\pi \in \Pi} \min_{P \in \mathcal{P}} \; \rho(\pi, P)\,. \tag{2.3}$$

Although solving (2.3) is NP-hard [Bagnell, 2004], it is tractable for certain classes of ambiguity set $\mathcal{P}$, including SA-, S-, R-, and K-rectangular [Iyengar, 2005; Tallec, 2007; Mannor *et al.*, 2016; Goyal and Grand-Clement, 2020a]. We focus on S-rectangular sets in this paper because they are both general and tractable [Wiesemann *et al.*, 2013]. All our results can be extended to SA-rectangular sets [Wiesemann *et al.*, 2013] as shown in Appendix A. An ambiguity set $\mathcal{P}$ is S-rectangular if

$$\mathcal{P} = \left\{ p \in (\Delta^S)^{S \times A} \;\mid\; (p_{s,a})_{a \in \mathcal{A}} \in \mathcal{P}_s,\; \forall s \in \mathcal{S} \right\}, \tag{2.4}$$

for some $\mathcal{P}_s \subseteq (\Delta^S)^\mathcal{A}$. Each element $p \in \mathcal{P}_s$ is a function $p : \mathcal{A} \to \Delta^S$ that determines the transition probabilities for all actions $a \in \mathcal{A}$ at the state s. The intuitive interpretation is that the adversary can choose the worst-case transition probability for each state independently.

In S-rectangular RMDPs, the optimal value function $v^\star \in \mathbb{R}^S$ exists, is unique, and is the fixed-point of the S-rectangular

robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}} : \mathbb{R}^S \to \mathbb{R}^S$ that is defined for each $v \in \mathbb{R}^S$ and $s \in \mathcal{S}$ as [Iyengar, 2005; Wiesemann *et al.*, 2013]

$$(\mathfrak{T}_{\mathcal{P}}\, v)(s) = \max_{d \in \Delta^A} \min_{p \in \mathcal{P}_s} \sum_{a \in \mathcal{A}} d_a \cdot \left(r_{s,a} + \gamma \cdot p_a^\top v\right). \quad (2.5)$$

The randomized decision rule d in (2.5) can be used to construct the optimal randomized policy. The optimal value function $v^\star$ can be computed using either value iteration [Wiesemann *et al.*, 2013] or policy iteration [Iyengar, 2005; Kaufman and Schaefer, 2013; Ho *et al.*, 2018] style algorithms. These algorithms can be scaled to infinite-state problems. For instance, Robust Projected Value Iteration (RPVI) [Tamar *et al.*, 2014] does this for robust value iteration by combining it with linear function approximation.

3 Static Soft-Robust Framework

In this section, we describe the *static* soft robust criterion, discuss its relationship to the percentile criterion, show that optimizing it is NP hard, and compare it with its *dynamic* counterpart. We propose to maximize the *static soft-robust* objective $\rho^S : \Pi \to \mathbb{R}$, defined as

$$\max_{\pi \in \Pi} \rho^S(\pi) := \\ := (1-\lambda) \cdot \underbrace{\mathbb{E}\left[\rho(\pi, \hat{P})\right]}_{\text{mean return}} + \lambda \cdot \underbrace{\mathrm{CVaR}^{\alpha}\left[\rho(\pi, \hat{P})\right]}_{\text{robust return}}. \quad (3.1)$$

Here, CVaR^{α} represents the conditional value at risk at level α, which is defined for any random variable $Z \sim g$ as [Rockafellar and Uryasev, 2000]

$$\mathrm{CVaR}_g^{\alpha}[Z] := \max_{b \in \mathbb{R}} \left(b - \frac{\mathbb{E}\left[\max\{b - Z, 0\}\right]}{1-\alpha}\right). \quad (3.2)$$

The robust return term $\mathrm{CVaR}^{\alpha}\left[\rho(\pi, \hat{P})\right]$ in (3.1) represents the average of the expected returns of the worst $1-\alpha$ fraction of the models. The parameters $\alpha \in [0,1]$ and $\lambda \in [0,1]$ are domain specific and give the decision-maker fine-grained control over the policy's robustness. The parameter $\lambda \in [0,1]$ balances the importance of mean and robust returns. The risk-aversion parameter $\alpha \in [0,1]$ controls the robustness of the return of π. For example, when $\alpha = 0.95$, the robust return is computed by averaging the returns over the worst 5% of the models.

Comparing the percentile criterion in (2.2) with the soft-robust criterion in (3.1), one can appreciate how the soft-robust criterion addresses the issues that arise with the percentile criterion: The soft-robust criterion explicitly includes the mean performance (weighted by $1-\lambda$) and the robust performance CVaR and is both, sensitive to the tail of the distribution and convex. (e.g., Shapiro *et al.* 2014).

The following results establish fundamental properties of the soft-robust objective $\rho^S(\pi)$. First, the following proposition justifies the need to consider randomized policies when optimizing this objective.

Proposition 3.1. *There may be no stationary* deterministic *policy $\pi \in \Pi_D$ that attains the optimal objective of the soft-robust optimization problem* (3.1)*.*

Proposition 3.1 follows immediately from *Theorem 2* in Buchholz and Scheftelowitsch [2020] by setting $\lambda = 0$ or $\alpha = 0$ in (3.1). Similar argument shows that history-dependent randomized policies may further outperform stationary ones [Steimle *et al.*, 2018]. The following proposition establishes the computational complexity of the optimization problem (3.1).

Proposition 3.2. *Computing the optimal (randomized or deterministic) policy of the soft-robust problem* (3.1) *is NP-hard.*

Proposition 3.2 follows readily from *Theorem 1* in Buchholz and Scheftelowitsch [2019] by setting $\lambda = 0$ or $\alpha = 0$.

In the remainder of the section, we argue that our static formulation handles the uncertainty over $\hat{P}$ more accurately than prior dynamic formulations. Model uncertainty has serious consequences in RL. Increasing uncertainty in $\hat{P}$ causes the value function of a non-robust policy to become unrealistically optimistic. This effect, which is driven by always choosing the maximum over uncertain action value estimates, is known as *optimizer's curse* [Smith and Winkler, 2006]. Such optimistic value-function inherently drives the agent to prefer states with high model uncertainty which is undesirable in robust RL. Double Q-learning [van Hasselt *et al.*, 2015] and other methods [Powell, 2011; Buckman *et al.*, 2021] mitigate the optimizer's curse but do not eliminate it.

We now show that the static soft-robust formulation with $\lambda = 0$ eliminates the optimizer's curse for the mean returns. The dynamic soft-robust formulations [Xu and Mannor, 2012; Derman *et al.*, 2018] and almost all other RL algorithms suffer from this curse.

To formally define the optimizer's curse [Smith and Winkler, 2006], recall that the random variable $P^\star$ represents the true transition probability used to generate the dataset $\mathcal{D}$. The term *post-decision surprise* refers to the difference $\rho(\bar{\pi}(\mathcal{D}), P^\star) - \bar{\rho}(\mathcal{D})$ between the true return of $\bar{\pi}(\mathcal{D}) \in \Pi$ and its estimated return $\bar{\rho}(\mathcal{D}) \in \mathbb{R}$. Note that both the policy $\bar{\pi}(\mathcal{D})$ and its estimated return $\bar{\rho}(\mathcal{D})$ depend on the dataset $\mathcal{D}$. If the average post-decision surprise is negative, $\mathbb{E}_{\mathcal{D},P^\star}\left[\rho(\bar{\pi}(\mathcal{D}), P^\star) - \bar{\rho}(\mathcal{D})\right] < 0$, then the algorithm is said to suffer from the *optimizer's curse*. As described above, consistently optimistic (or biased) return estimates drive the agent to more uncertain states.

We are now ready to show that the static soft-robust formulation is immune to the optimizer's curse. Let $\bar{\pi}_S(\mathcal{D})$ denote an optimal solution to (3.1) for $\hat{P} = P^\star \mid \mathcal{D}$ and, similarly, let $\bar{\rho}_S(\mathcal{D})$ be the optimal objective value of (3.1).

Theorem 3.3. *Optimal solution $\bar{\pi}_S(\mathcal{D})$ with objective $\bar{\rho}_S(\mathcal{D})$ to* (3.1) *with $\lambda = 0$ has no expected post-decision surprise:*

$$\mathbb{E}_{\mathcal{D},P^\star}\left[\rho(\bar{\pi}_S(\mathcal{D}), P^\star) - \bar{\rho}_S(\mathcal{D})\right] = 0\,.$$

Moreover, the expected post-decision surprise is non-negative for any $\lambda \in (0,1]$ and any $\alpha \in [0,1]$.

The proof of the theorem can be found in Appendix C.

To illustrate the implications of Theorem 3.3, Figure 1 compares the post-decision surprise of the static soft-robust model with a dynamic model and an empirical method. We use a small MDP with 5 states and 3 actions with $P^\star$ sampled from the uniform Dirichlet prior and $|\mathcal{D}| = 100$ drawn from a random policy. The dynamic soft-robust criterion with $\lambda = 0$

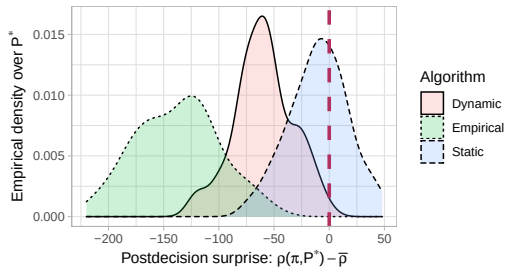

Figure 1: Post-decision surprise of policies computed using static and dynamic soft-robust criteria and the empirical model.

solves $\max_{\pi\in\Pi}\rho(\pi,\mathbb{E}[\hat{P}])$ [Derman *et al.*, 2018]. Note that the expectation is inside of the return rather than outside. The empirical method solves for $\max_{\pi\in\Pi}\rho(\pi,\bar{P})$, where $\bar{P}$ are empirical transition probabilities. The results show that the empirical solution consistently suffers from significant negative average post-decision surprise, which is slightly reduced by the dynamic formulation, and eliminated by the static formulation.

4 Soft-Robust Optimization

In this section, we present two approximate algorithms for maximizing the soft-robust objective. First, we derive a mixed-integer linear program (MILP) formulation in Section 4.1 that can be used to compute an optimal *deterministic* policy. Because MILPs do not scale well, we then show how the soft-robust objective can be represented as an RMDP in Section 4.2, and solved with linear value function approximation in Section 4.3.

We start by stating the following lemma that shows the soft-robust criterion reduces to a worst-case expectation over a certain set of measures over the transition model $\hat{P}$.

Lemma 4.1. *Define a set* $\Xi\subseteq\Delta^{|\Omega|}$ *as*

$$\Xi=\left\{\xi\in\Delta^{|\Omega|}\;\middle|\;(1-\lambda)\cdot f\leq\xi\leq\frac{1-\alpha+\lambda\alpha}{1-\alpha}\cdot f\right\}. \quad (4.1)$$

Then, for each $\pi\in\Pi$*, the objective* $\rho^S(\pi)$ *in* (3.1) *satisfies*

$$\rho^S(\pi)\;=\;\min_{\xi\in\Xi}\mathbb{E}_{\hat{P}\sim\xi}\left[\rho(\pi,\hat{P})\right]. \quad (4.2)$$

The proof of Lemma 4.1, which we report in Appendix D, follows by algebraic manipulation from the robust representation of CVaR in (3.2). This result allows us to rewrite the *static soft-robust* optimization (3.1) as

$$\max_{\pi\in\Pi}\;\rho^S(\pi)=\max_{\pi\in\Pi}\;\min_{\xi\in\Xi}\;\mathbb{E}_{\hat{P}\sim\xi}\left[\rho(\pi,\hat{P})\right]. \quad (4.3)$$

4.1 Soft-Robust Optimization using MILP

Our MILP formulation of the soft-robust optimization, which we call SR-MILP and present it in Figure 2, is based on (4.3). The intuitive explanation for this formulation is as follows. One may compute the soft-robust objective by simultaneously solving a series of MDPs with transition functions given by $\hat{P}^\omega$, one for each $\omega\in\Omega$. The variable $u(s,a,\omega)\in\mathbb{R}_+$ in Figure 2 represents the occupancy frequency for the state s and action a in the MDP given by ω. The second constraint ensures that u is a valid occupancy frequency. The binary variable $\pi(s,a)\in\{0,1\}$ (deterministic policy), is used to guarantee that the policy is consistent across the MDPs $\hat{P}^\omega$ by enforcing the fourth constraint. The fourth constraint ensures that $u(s,a,\omega)>0\Leftrightarrow\pi(s,a)=1$. Finally, the variables b and y and the first constraint are used to represent the CVaR formulation in (3.2). We are now ready to state the correctness of our formulation in Proposition 4.2, whose proof we report in Appendix D.

Proposition 4.2. *Any* $\pi^\star$ *optimal in Figure 2 satisfies that* $\pi^\star\in\arg\max_{\pi\in\Pi_D}\rho^S(\pi)$.

We note that the MILP in Figure 2 returns deterministic policies but the optimal policy for the problem in (4.3) may be stochastic. While this may seem like a limitation, it actually offers some tangible advantages. In practice, deterministic policies are often preferred over randomized ones, when randomizing between different actions is undesirable. In medical domains, for example, it may be unethical to randomize outside of a medical trial. In other domains, randomization hinders reproducibility and may make it very difficult to evaluate and diagnose the policy once it is deployed.

Although the MILP in Figure 2 returns stationary policies, they can still benefit from the static uncertainty assumption in (3.1). To illustrate this point, consider a cancer treatment problem, where the agent has to decide on the amount of chemotherapy to be administered. The fact that a particular state of the patient reveals some information about their response to the treatment can be used to make more informed decisions. The dynamic uncertainty model, on the other hand, assumes that the patient model changes throughout the execution and cannot exploit this information.

4.2 Soft-Robust Optimization using RMDPs

In this section, we describe how to construct an RMDP that approximates the soft-robust criterion. First we note that Xu and Mannor [2012] showed that optimizing any coherent risk measure of the return (including the soft-robust criterion) is equivalent to solving an RMDP. However, there are three main differences between their results and ours: *1)* To show the equivalence to an RMDP problem, Xu and Mannor [2012] assume that each state is visited only once within an episode. This assumption is too strong and does not hold for infinite-horizon MDPs with finite state spaces. Hence, we propose the dynamic soft-robust objective in this section as an approximation to the original soft-robust criterion, and then in Section 5, bound the error due to this approximation. *2)* We derive explicit RMDP ambiguity sets for soft-robust criterion instead of an abstract representation, as in Xu and Mannor [2012]. *3)* We present a scalable algorithm to solve the dynamic soft-robust objective in Section 4.3.

Our soft robust reduction to a RMDP proceeds in two steps.

Step 1: Approximate the soft-robust optimization (4.3) as

$$\max_{\pi\in\Pi}\;\rho^D(\pi)\;=\;\max_{\pi\in\Pi}\;\min_{\xi\in\Xi}\;\rho\big(\pi,\mathbb{E}_{\hat{P}\sim\xi}[\hat{P}]\big). \quad (4.4)$$

where the superscript D indicates that this is the ambiguity set corresponding to the dynamic soft-robust formulation. Note

$$
\begin{array}{lll}
\underset{\substack{\pi\in\{0,1\}^{S\times A},\, b\in\mathbb{R},\\ u\in\mathbb{R}_+^{S\times A\times N},\, y\in\mathbb{R}_+^{N}}}{\text{maximize}} & \lambda\cdot\Big(b-\dfrac{1}{1-\alpha}\displaystyle\sum_{\omega\in\Omega}y(\omega)\Big)+(1-\lambda)\cdot\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{\omega\in\Omega}u(s,a,\omega)\sum_{s'\in\mathcal{S}}r(s,a,s')\cdot P^{\omega}(s,a,s') & \\
\text{subject to} & y(\omega)-b\cdot f_\omega \;\geq\; -\displaystyle\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}u(s,a,\omega)\sum_{s'\in\mathcal{S}}P^{\omega}(s,a,s')\cdot r(s,a,s'), & \omega\in\Omega,\\
& \displaystyle\sum_{a\in\mathcal{A}}u(s,a,\omega) \;=\; \sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}}\gamma\cdot u(s',a',\omega)\cdot P^{\omega}(s',a',s)+f_\omega\cdot p_0(s), & s\in\mathcal{S},\ \omega\in\Omega,\\
& \displaystyle\sum_{a\in\mathcal{A}}\pi(s,a) \;=\; 1, & s\in\mathcal{S},\\
& u(s,a,\omega) \;\leq\; f_\omega\cdot\pi(s,a)/(1-\gamma), & s\in\mathcal{S},\ a\in\mathcal{A},\ \omega\in\Omega.
\end{array}
$$

Figure 2: SR-MILP: Mixed-Integer linear program that solves the soft-robust optimization problem (4.3).

that, in contrast to $\rho^S(\pi)$ in (4.3), the expectation in (4.4) is inside the return function ρ. This approximation is helpful because it can be represented as a non-rectangular RMDP (see (2.3)) with the ambiguity set

$$\mathcal{P}^D \subseteq (\Delta^S)^{\mathcal{S}\times\mathcal{A}},\ \mathcal{P}^D=\Big\{\sum_{\omega\in\Omega}\xi_\omega\cdot\hat{P}^{\omega}\ \mid\ \xi\in\Xi\Big\}. \quad (4.5)$$

Step 2: Although solving a non-rectangular RMDP is NP-hard [Wiesemann *et al.*, 2013], it can be turned into a tractable rectangular one by a process called *rectangularization* in the context of dynamic risk measures [Roorda *et al.*, 2005; Iancu *et al.*, 2011]. Rectangularization constructs the smallest rectangular set that contains the entire non-rectangular one. Since the rectangular set is a superset of the non-rectangular one, the rectangular robust objective lower-bounds its non-rectangular counterpart.

To formalize the rectangularization procedure, assume that the soft-robust ambiguity set, which we denote by $\mathcal{P}^R$, is S-rectangular, i.e.,[1]

$$\mathcal{P}^R=\bigotimes_{s\in\mathcal{S}}\mathcal{P}^R_s,\quad \mathcal{P}^R_s=\Big\{\sum_{\omega\in\Omega}\xi_\omega\cdot\hat{P}^{\omega}_s\ \mid\ \xi\in\Xi\Big\}. \quad (4.6)$$

Here Ξ is defined in (4.1) and $\hat{P}^{\omega}_s\in(\Delta^S)^A$ is a value of the posterior sample $\hat{P}^{\omega}$ in state s. Recall from Section 2 that the S-rectangular RMDPs can be solved efficiently [Ho *et al.*, 2018]. Finally, the following optimization problem defines the S-rectangular objective $\rho^R:\Pi\to\mathbb{R}$:

$$\rho^R(\pi) \;=\; \min_{P\in\mathcal{P}^R}\ \rho\left(\pi,P\right). \quad (4.7)$$

The following proposition (proof in Appendix D) shows how the non-rectangular ambiguity set $\mathcal{P}^D$ in (4.5) and its corresponding return ρ^D, defined in (4.4), are related to the S-rectangular ambiguity set $\mathcal{P}^R$ and return ρ^R.

Proposition 4.3. *The ambiguity sets $\mathcal{P}^D$ and $\mathcal{P}^R$ satisfy $\mathcal{P}^D\subseteq\mathcal{P}^R$, and their corresponding returns ρ^D and ρ^R satisfy $\rho^R(\pi)\leq\rho^D(\pi)$, for each $\pi\in\Pi$.*

Proposition 4.3 shows two important results. *First*, the S-rectangular ambiguity set $\mathcal{P}^R$ contains the non-rectangular ambiguity set $\mathcal{P}^D$ (rectangularization procedure). *Second*, the S-rectangular objective ρ^R, which can be tractably computed by solving the resulting S-rectangular RMDP, is a lower-bound of the dynamic soft-robust objective ρ^D.

The optimal value function $v^R\in\mathbb{R}^S$ of the S-rectangular RMDP defined by $\mathcal{P}^R$ satisfies the robust Bellman optimality equation $v^R=\mathfrak{T}_{\mathcal{P}^R}v^R$ and can be approximated using the standard value iteration algorithm (see Section 2). For any $v\in\mathbb{R}^S$ and $s\in\mathcal{S}$, the Bellman optimality operator $(\mathfrak{T}_{\mathcal{P}^R}v)(s)$ can be computed by solving the following linear program with $z_{s,a}=r_{s,a}+\gamma\cdot v$:

$$
\begin{aligned}
\max_{\substack{d\in\Delta^A,\, b\in\mathbb{R}\\ y\in\mathbb{R}_+^{|\Omega|}}}\quad & (1-\lambda)\sum_{\substack{a\in\mathcal{A}\\ \omega\in\Omega}}d_a f_\omega(\hat{P}^{\omega}_{s,a})^\top z_{s,a}\\
& \qquad +\lambda\Big(b-\frac{1}{1-\alpha}\sum_{\omega\in\Omega}f_\omega\cdot y_\omega\Big) \qquad (4.8)\\
y_\omega\geq b-\sum_{a\in\mathcal{A}}d_a(\hat{P}^{\omega}_{s,a})^\top z_{s,a},&\quad \omega\in\Omega\,.
\end{aligned}
$$

Proposition 4.4. *For any $v\in\mathbb{R}^S$ and $s\in\mathcal{S}$, the optimal value of the objective function in* (4.8) *is equal to $(\mathfrak{T}_{\mathcal{P}^R}v)(s)$.*

The correctness of Proposition 4.4 follows from algebraic manipulation of (4.6) and is provided in Appendix D.

4.3 Projected Soft-Robust Value Iteration

We now present our soft-robust value iteration (SRVI) algorithm that we use to (approximately) solve the soft-robust S-rectangular RMDP defined in Section 4.2. SRVI, whose pseudo-code is shown in Algorithm 4.1, generalizes the Robust Projected Value Iteration (RPVI) algorithm [Tamar *et al.*, 2014] to soft-robust S-rectangular RMDPs.

We use the linear approximation $v(s)=\phi(s)^\top w$ for the soft-robust value function $v\in\mathbb{R}^S$, where $\phi(\cdot)\in\mathbb{R}^l$, $l\ll S$ is an l-dimensional feature vector and $w\in\mathbb{R}^l$ is a weight vector. Note that to represent a rich class of value-functions, we can use neural networks as feature processors to generates the high-dimensional features ϕ. Let $\Phi\in\mathbb{R}^{M\times l}$ denote the sample feature matrix of ϕ after observing M samples, and by $h\in\Delta^S$ the steady state distribution of any given policy $\pi\in\Pi$ over states $s\in\mathcal{S}$. Further, let $\sigma_{\Phi^\top w}:\mathbb{R}^S\to\mathbb{R}^S$ be the

[1] This ambiguity set decomposition is similar to the one in (2.4). The mnemonic superscript R in $\mathcal{P}^R$ indicates that it is a S-rectangular ambiguity set.

function obtained by applying the S-rectangular soft-robust Bellman optimality operator to a value function $v = \Phi w$, i.e.,

$$\sigma_{\Phi^\top w}(s) = (\mathfrak{T}_{\mathcal{P}^R} v)(s) = (\mathfrak{T}_{\mathcal{P}^R}(\Phi^\top w))(s). \tag{4.9}$$

Then $\Sigma_{\Phi^\top w}$ represents the vector of the soft-robust Bellman optimality values for matrix Φ: $\{\sigma_{\Phi^\top w}(s_t)\}_{t=1}^M$. Finally, we denote by Ψ the projection operator onto the subspace Φ w.r.t. the h-weighted Euclidean norm.

Algorithm 4.1: Soft-Robust Value Iteration (SRVI)

Input: confidence α, risk factor λ, distribution f
Output: soft-robust value function v
Initialize: weight vector w_0; counter $k \leftarrow 1$;
Sample N parametric models $\{\hat{P}_\theta^{\omega_i}\}_{i=1}^N$ from f ;
Compute mean $\bar{P}_\theta = \mathbb{E}[\hat{P}_\theta]$ from $\{\hat{P}_\theta^{\omega_i}\}_{i=1}^N$;
repeat
  Simulate episodes following $\bar{P}_\theta$ and policy from (4.9) to get samples $\mathcal{D}_k$ and Φ_k;
  Compute w_k from (4.10) using $\Phi = \Phi_k$;
  $k \leftarrow k + 1$;
until $\left\|\Phi_k^\top w_k - \Phi_k^\top w_{k-1}\right\|_\infty \leq \epsilon$;
return $v = \Phi_k w_k$

SRVI approximates $\pi_R \in \arg\max_{\pi\in\Pi} \rho^R(\pi)$ (see Eq. 4.7) by iteratively solving the projected soft-robust Bellman optimality equation $v = \Psi\, \mathfrak{T}_{\mathcal{P}^R}\, v$. In each iteration k, we first simulate episodes using the mean transition probability model $\bar{P}_\theta$ and construct the dataset $\mathcal{D}_k$ of size M to approximately represent the stationary state distribution induced by the current policy. Then, we update the weight vector w_k using the reconstructed data as

$$w_k = (\Phi^\top H \Phi)^{-1} (\Phi^\top H P \Sigma_{\Phi^\top w_{k-1}}), \tag{4.10}$$

where $H = \mathrm{diag}(h)$. Since it is impossible to exactly compute the terms in (4.10), we approximate them using the Sample Average Approximation (SAA) as

$$\begin{aligned} \Phi^\top H \Phi &\sim \frac{1}{M} \sum_{t=1}^M \phi(s_t)\phi(s_t)^\top , \\ \Phi^\top H P_\theta \Sigma_{\Phi^\top w} &\sim \frac{1}{M} \sum_{t=1}^M \sigma_{\Phi^\top w}(s_t). \end{aligned} \tag{4.11}$$

The optimization problem in (4.9) can be solved by formulating it as a linear program in (4.8), and using the SAA method to approximate the value function v. We repeatedly update the weight vector w using (4.10) until the soft-robust value function $\Phi^\top w$ converges to the unique projected fixed-point of $\mathfrak{T}_{\mathcal{P}^R}$. Given the optimal weight vector $w^\star$, the optimal policy for any state s can be then computed by solving (4.9).

We note that we mainly focus on linear value-function approximations because of their convergence guarantees which do not exist when using neural networks. However, for the sake of completeness, we show how to extend a deep actor-critic algorithm to optimize the dynamic soft-robust criterion in Appendix F.

5 RMDP Approximation Error

In this section, we derive approximation error bounds on the RMDP formulation described in Section 4.2. These bounds provide insight into the possible directions for further improvement of the formulation. The error introduced by resorting to the RMDP formulation depends on two main factors: 1) how the model uncertainty impacts the occupancy frequency, and 2) whether there exists some ordering of $\omega_1, \ldots, \omega_N \in \Omega$ such that the model P^{ω_i} is approximately better than $P^{\omega_{i+1}}$ consistently across the states.

The proof of the approximation error proceeds in the same two steps as in Section 4.2. The error introduced in the first step depends on the state occupancy frequency $h_\pi^\omega \in \mathbb{R}^S$ for each $\pi \in \Pi$ and $\omega \in \Omega$ defined as

$$h_\pi^\omega \;=\; \left(\boldsymbol{I} - \gamma \cdot \hat{P}_\pi^{\omega\top}\right)^{-1} p_0 \,. \tag{5.1}$$

We get the following bound on the *first step*'s error.

Theorem 5.1. *The difference between static and dynamic returns is bounded for each $\pi \in \Pi$ as*

$$\left|\rho^D(\pi) - \rho^S(\pi)\right| \;\leq\; \frac{\gamma \cdot r_{\max}}{1-\gamma} \cdot \epsilon_1(\pi) \,,$$

where $\epsilon_1(\pi) \;=\; \max_{\omega_1,\omega_2\in\Omega} \left\|h_\pi^{\omega_1} - h_\pi^{\omega_2}\right\|_1$.

We provide the proof of the theorem in Appendix E. Its main idea is to bound the nonlinearity of $c : \xi \mapsto \rho(\pi, \mathbb{E}_{\hat{P}\sim\xi}[\hat{P}])$. In particular, when $\epsilon_1 = 0$ then c is linear and

$$\rho^S(\pi) = \min_{\xi\in\Xi} \mathbb{E}[\rho(\pi, \hat{P})] = \min_{\xi\in\Xi} \rho(\pi, \mathbb{E}[\hat{P}]) = \rho^D(\pi) \,.$$

The following lemma bounds the error that arises due to the rectangularization in the *second step* of the approximation.

Lemma 5.2. *Suppose that $\pi_D^\star \in \arg\max_{\pi\in\Pi} \rho^D(\pi)$ and $\pi_R^\star \in \arg\max_{\pi\in\Pi} \rho^R(\pi)$. Then:*

$$\rho^D(\pi_D^\star) - \rho^D(\pi_R^\star) \;\leq\; \frac{1}{1-\gamma} \cdot \epsilon_2 \,,$$

where $\epsilon_2 \;=\; \max_{s\in\mathcal{S}, a\in\mathcal{A}} \min_{\xi\in\Xi} \delta(s,a,\xi)$, $\delta(s,a,\xi) \;=\; \sum_{\omega\in\Omega} \xi_\omega \cdot (\hat{P}_{s,a}^\omega)^\top (r_{s,a} + \gamma \cdot v_D^\star) - (v_D^\star)_s$, $v_D^\star \in \mathbb{R}^S$ is the value function of $\pi_D^\star$.

The proof of the lemma is reported in Appendix E.

The value $\delta(s,a,\xi)$ in Lemma 5.2 is the difference between the rectangular robust Bellman value in the state-action pair s, a and the non-rectangular RMDP value of s. It can be readily seen that ϵ_2 vanishes when there exists an ordering of elements $\omega_1, \omega_2, \ldots$ in Ω such that $\hat{P}_{s,a}(\omega_i)^\top (r_{s,a} + \gamma \cdot \hat{v}) \geq \hat{P}_{s,a}(\omega_j)^\top (r_{s,a} + \gamma \cdot \hat{v})$, for $i < j$ and for all states and actions. This is because if the condition holds then the set $\arg\min_{\xi\in\Xi} \delta(s,a,\xi)$ is constant across states and actions and equals to $\arg\min_{\xi\in\Xi} \rho(\pi, \mathbb{E}_{\hat{P}\sim\xi}[\hat{P}])$.

We can now bound the overall RMDP approximation error.

Corollary 5.3. *The soft-robust return ρ^S of $\pi_R^\star \in \arg\max_{\pi\in\Pi} \rho^R(\pi)$ computed by Algorithm 4.1 satisfies that*

$$\rho^S(\pi_S^\star) - \rho^S(\pi_R^\star) \;\leq\; \frac{1}{1-\gamma} \left(2 \cdot \gamma \cdot \epsilon_1 \cdot r_{\max} + \epsilon_2\right),$$

where $\epsilon_1 \;=\; \max_{\pi\in\Pi} \epsilon_1(\pi)$, and $\epsilon_1(\pi)$ and ϵ_2 are defined as in Theorem 5.1 and Lemma 5.2 respectively, and $\pi_S^\star \in \arg\max_{\pi\in\Pi} \rho^S(\pi)$.

The proof of the corollary is reported in Appendix E.

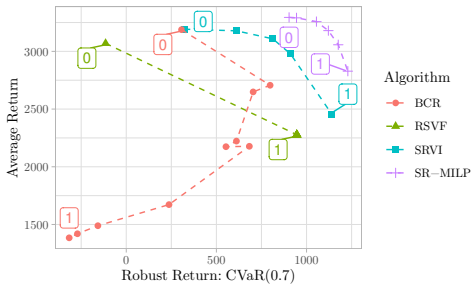

Figure 3: Comparison between the trade-offs of several algorithms as parameterized by λ as indicated by the overlay label.

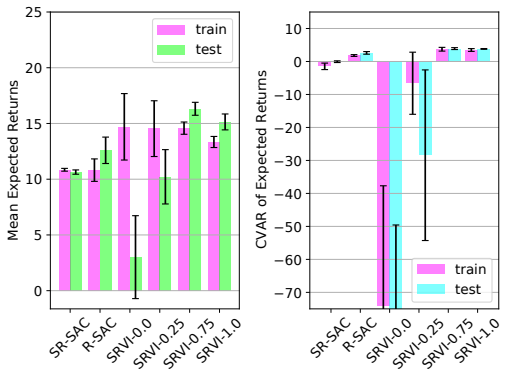

Figure 4: Mean and Robust performance of SRVI, SRD-SAC, and R-SAC in Cancer Environment.

6 Experimental Evaluation

In this section, we present two case studies to demonstrate the performance of the soft-robust criterion. We compare soft-robust algorithms with related baseline algorithms in terms of the mean and robust performance over the posterior distribution f inferred from the fixed dataset $\mathcal{D}$. Please refer to Appendix F for a more detailed detailed description.

6.1 Integrated Pest Control Problem

The domain represents a simplified integrated pest control problem. The decision-maker must decide which, if any, pesticide to use during the growing season. The state represents the pest population, and action determines whether a pesticide is used. Exponential pest growth dynamics drive the transition model and the rewards measure the net profit of the yields less the pesticide costs. The corresponding MDP consists of 51 states, each represents the current pest population as determined by trapping (0 means no pest population). Each one of 5 actions available prescribes the use of an increasingly potent pesticide.

To compute the posterior distribution over $\hat{P}$, we gather 300 state-action transition samples from a single episode. Using these transition samples, we fit an exponential population model [Kery and Schaub, 2012] and sample 100 posterior samples using MCMC. We use these samples to formulate and solve the MILP in Figure 2 and to run Algorithm 4.1. We use confidence $\alpha = 0.7$ for both the percentile criterion and soft-robust objective for the evaluation. We also use $\lambda = 0.5$ for the soft-robust objective.

Figure 3 shows the trade-off between mean and worst-case performance for several robust methods for different choices of $\lambda \in [0, 1]$ as indicated by the floating labels. We compare the optimal MILP policy in Figure 2 (SR-MILP) and Algorithm 4.1 (SRVI) with BCR and RSVF [Russel and Petrik, 2019]. Note that RSVF and BCR optimize the percentile criterion, which has no inherent notion of the trade-off between robust and mean performance. We simulate the effect of λ by simply shrinking the ambiguity sets in the RMDP formulations (multiplying the budget by λ). Our soft-robust algorithms outperform earlier methods and trade-off well between the mean and robust return with changing λ.

6.2 Cancer Growth Simulator

The cancer simulator models the growth of tumors in cancer patients. The state is a 4-dimensional vector that captures the dynamics of the tumor's growth. The monthly binary action determines whether to administer chemotherapy to the patient [Gottesman *et al.*, 2020; Ribba *et al.*, 2012]. The discount factor γ is set to 0.9.

This experiment compares dynamic soft-robust criterion with the dynamic soft-robust objective [Derman *et al.*, 2018] and the robust objective in Mankowitz *et al.* [2020]. We combine these robust objectives with the Soft Actor-Critic (SAC) algorithm [Haarnoja *et al.*, 2018] to obtain two algorithms which we call Soft-Robust (Derman) SAC (SRD-SAC) and Robust SAC (R-SAC). We use SAC instead of robust Q-learning [Hester *et al.*, 2017; Lillicrap *et al.*, 2019] or other actor-critic [Konda and Tsitsiklis, 2000] algorithms because it has been observed to be more stable. For each algorithm, we train 5 separate agents using 100 samples of $\hat{P} \sim f$ and evaluate the mean and robust return of the computed policies using a separate set of 50 samples of $\hat{P} \sim f$. The robust return is computed using CVaR with $\alpha = 0.9$.

Figure 4 compares the mean and robust performance of SRD-SAC and R-SAC with SRVI for $\lambda \in \{0.25, 0.75, 1.0\}$. SRVI outperforms SRD-SAC and R-SAC in mean and robust performance for appropriately chosen λ. This behavior is expected since SRD-SAC ignores the robust return and R-AC ignores the mean return. Focusing on the returns on the training set, SRVI's robust performance improves with an increasing λ, and the mean performance improved with a decreasing λ. This expected trend, however, does not quite hold for the test set because of the generalization error. We leave studying the generalization issue for future work.

7 Conclusion

We proposed a new static soft-robust framework that can balance expected and robust performance in reinforcement learning and handle heavy tail risks. We show that the soft-robust objective can be formulated and solved as a MILP for deterministic policies. To scale to larger problems, we propose a new specific RMDP formulation which we combine with linear value function approximation. Finally, we analyze the approximation error of the RMDP formulation and evaluate the algorithms on two domains.

References

Carlo Acerbi and Dirk Tasche. Expected shortfall: A natural coherent alternative to value at risk. *Economic Notes*, 31(2):379–388, 2002.

J. Bagnell. *Learning decisions: robustness, uncertainty, and approximation*. PhD thesis, CMU, 2004.

Aharon Ben-Tal, Dimitris Bertsimas, and David B. Brown. A soft robust model for optimization under ambiguity. *Operations Research*, 58(4):1220–1234, 2010.

Peter Buchholz and Dimitri Scheftelowitsch. Computation of weighted sums of rewards for concurrent MDPs. *Mathematical Methods of Operations Research*, 89, 2019.

Peter Buchholz and Dimitri Scheftelowitsch. *Concurrent MDPs with Finite Markovian Policies*, pages 37–53. Springer, Cham, 2020.

Jacob Buckman, Carles Gelada, and Marc G. Bellemare. The importance of pessimism in fixed-dataset policy optimization. In *International Conference on Learning Representations*, 2021.

Yinlam Chow and Mohammad Ghavamzadeh. Algorithms for CVaR optimization in MDPs. In *International Conference on Neural Information Processing Systems*, pages 3509–3517. MIT Press, 2014.

Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51, 2018.

E. Delage and S. Mannor. Percentile Optimization for Markov Decision Processes with Parameter Uncertainty. *Operations Research*, 58(1):203–213, 2010.

Esther Derman, Daniel Mankowitz, Timothy A Mann, and Shie Mannor. Soft-Robust Actor-Critic Policy-Gradient. In *Conference on Uncertainty in Artificial Intelligence*, 2018.

Esther Derman, Daniel J. Mankowitz, Timothy A. Mann, and Shie Mannor. A Bayesian approach to robust reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2014.

Omer Gottesman, Joseph Futoma, Yao Liu, Sonali Parbhoo, Leo Anthony Celi, Emma Brunskill, and Finale Doshi-Velez. Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions. In *International Conference on Machine Learning*, 2020.

Vineet Goyal and Julien Grand-Clement. Robust markov decision process: Beyond rectangularity. *Preprint arXiv:1811.00215*, 2020.

Vineet Goyal and Julien Grand-Clement. Robust markov decision process: Beyond rectangularity. *Preprint arXiv:1811.00215*, 2020.

Vishal Gupta. Near-optimal Bayesian ambiguity sets for distributionally robust optimization. *Management Science*, 2019.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1861–1870, 2018.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep Q-learning from demonstrations. *Preprint arXiv:1704.03732*, 2017.

T. Hiraoka, T. Imagawa, T. Mori, T. Onishi, and Yoshimasa Tsuruoka. Learning robust options by Conditional Value at Risk optimization. In *International Conference on Neural Information Processing Systems*, 2019.

Chin Pang Ho, Marek Petrik, and Wolfram Wiesemann. Fast Bellman updates for robust MDPs. In *Proceedings of the 35th International Conference on Machine Learning Conference*, pages 1979–1988, 2018.

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Preprint arXiv:1111.4246*, 2011.

Dan A. Iancu and Nikolaos Trichakis. Pareto Efficiency in Robust Optimization. *Management Science*, 60(1):130–147, 2014.

Dan Iancu, Marek Petrik, and Dharmashankar Subramanian. Tight approximations of dynamic risk measures. *Mathematics of Operations Research*, 40, 2011.

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

Nathan Kallus and Angela Zhou. Confounding-robust policy evaluation in infinite-horizon reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

David L Kaufman and Andrew J Schaefer. Robust modified policy iteration. *INFORMS Journal on Computing*, 25(3):396–410, 2013.

Marc Kery and Michael Schaub. *Bayesian Population Analysis Using WinBUGS*. Elsevier, 2012.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *SIAM Journal on Control and Optimization*, pages 1008–1014. MIT Press, 2000.

Pavlo Krokhmal, Jonas Palmquist, and Stan Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. *Journal of Risk*, 4, 2003.

Sasche Lange, Thomas Gabel, and Martin Riedmiller. *Batch Reinforcement Learning*. Springer, 2012.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *Preprint arXiv:1509.02971*, 2019.

Daniel J. Mankowitz, Nir Levine, Rae Ung Jeong, Abbas Abdolmaleki, Jost Tobias Springenberg, Timothy Mann, Todd Hester, and Martin A. Riedmiller. Robust reinforcement

learning for continuous control with model misspecification. *ArXiv*, abs/1906.07516, 2020.

Shie Mannor, O Mebel, and H Xu. Lightning does not strike twice: Robust MDPs with coupled uncertainty. In *International Conference on Machine Learning (ICML)*, 2012.

Shie Mannor, Ofir Mebel, and Huan Xu. Robust MDPs with K-rectangular uncertainty. *Mathematics of Operations Research*, 41(4):1484–1509, 2016.

Merve Meraklı and Simge Küçükyavuz. Risk aversion to parameter uncertainty in Markov decision processes with an application to slow-onset disaster relief. *IISE Transactions*, pages 1–52, 2019.

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

Marek Petrik and Ronny Luss. Interpretable Policies for Dynamic Product Recommendations. In *Uncertainty in Artificial Intelligence (UAI)*, 2016.

Marek Petrik, Mohammad Ghavamzadeh, and Yinlam Chow. Safe policy improvement by minimizing robust baseline regret. In *International Conference on Neural Information Processing Systems*, pages 2306–2314, 2016.

Marek Petrik. *Optimization-Based Approximate Dynamic Programming*. PhD thesis, University of Massachusetts Amherst, 2010.

Martyn Plummer. Jags: A program for analysis of bayesian graphical models using gibbs sampling. *3rd International Workshop on Distributed Statistical Computing (DSC 2003); Vienna, Austria*, 124, 2003.

Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Princeton University Press, 2011.

L. A. Prashanth. Policy gradients for CVaR-constrained MDPs. In *In Proceedings of the 25th International Conference on Algorithmic Learning Theory*, pages 155–169, 2014.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., 2nd edition, 2005.

B. Ribba, G. Kaloshi, M. Peyre, D. Ricard, V. Calvez, M. Tod, B. Cajavec-Bernard, A. Idbaih, D. Psimaras, L. Dainese, J. Pallud, S. Cartalat-Carel, J. Delattre, J. Honnorat, E. Grenier, and F. Ducray. A tumor growth inhibition model for low-grade glioma treated with chemotherapy or radiotherapy. *Clinical Cancer Research*, 18:5071 – 5080, 2012.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

R. TYRRELL Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Berend Roorda, Johannes Schumacher, and Jacob Engwerda. Coherent acceptability measures in multiperiod models. *Mathematical Finance*, 15:589–612, 2005.

Reazul Hasan Russel and Marek Petrik. Beyond confidence regions: Tight Bayesian ambiguity sets for robust MDPs. In *International Conference on Neural Information Processing Systems*, 2019.

Alexander Schied. Risk measures and robust optimization problems. In *Symposium on Probability and Stochastic Processes*, 2004.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, 2014.

James Smith and Robert Winkler. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52:311–322, 2006.

Lauren Steimle, David Kaufman, and Brian Denton. Multi-model Markov decision processes. *Optimization Online*, 2018.

Alexander L Strehl and Michael L Littman. Exploration via Model-based Interval Estimation. In *Proceedings of the 21st International Machine Learning Conference*, 2004.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, 2018.

Yann Le Tallec. *Robust, risk-sensitive, and data-driven control of Markov decision processes*. PhD thesis, MIT, 2007.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In *Proceedings of the 31st International Conference on Machine Learning*, page 181–189, 2014.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *International Conference on Neural Information Processing Systems*, volume 28, pages 1468–1476, 2015.

Yichuan Charlie Tang, Jian Zhang, and Ruslan Salakhutdinov. Worst cases policy gradients. *Preprint arXiv:1911.03618*, 2019.

Andrea Tirinzoni, Xiangli Chen, Marek Petrik, and Brian D. Ziebart. Policy-conditioned uncertainty sets for robust Markov decision processes. In *International Conference on Neural Information Processing Systems*, page 8953–8963, 2018.

Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning. In *AAAI Conference on Artificial Intelligence*, 2015.

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

Huan Xu and Shie Mannor. The robustness-performance tradeoff in Markov decision processes. In *International Conference on Neural Information Processing Systems*, 2006.

Huan Xu and Shie Mannor. Parametric regret in uncertain Markov decision processes. In *IEEE Conference on Decision and Control*, pages 3606–3613, 2009.

Huan Xu and Shie Mannor. Distributionally robust Markov decision processes. *Mathematics of Operations Research*, 37(2):288–300, 2012.

A SA-rectangular Soft-Robust MDPs

Previously, we described the S-rectangular soft-robust MDPs as a tractable approach for solving the dynamic soft-robust objective. In this section, we extend our results to soft-robust MDPs with SA-rectangular ambiguity set. The SA-rectangular ambiguity set is the simplest class of ambiguity sets and is defined as [Nilim and El Ghaoui, 2005; Wiesemann *et al.*, 2013]

$$\mathcal{P}^{SA} = \left\{ p \in (\Delta^S)^{S \times A} \;\middle|\; p_{s,a} \in \mathcal{P}^{SA}_{s,a},\ \forall s \in \mathcal{S},\ \forall a \in \mathcal{A} \right\},$$

for some $\mathcal{P}^{SA}_{s,a} \subseteq \Delta^S, s \in \mathcal{S}, a \in \mathcal{A}$. The SA-rectangular ambiguity sets assume that the transition models corresponding to each state-action pair are independent. The robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}^{SA}} : \mathbb{R}^S \to \mathbb{R}^S$ for SA-rectangular ambiguity set $\mathcal{P}^{SA}$ is defined as [Iyengar, 2005; Wiesemann *et al.*, 2013]

$$(\mathfrak{T}_{\mathcal{P}^{SA}} v)(s) = \max_{a \in \mathcal{A}} \min_{P \in \mathcal{P}^{SA}_{s,a}} P^\top_{s,a} \left(r_{s,a} + \gamma \cdot v \right). \tag{A.1}$$

In SA-rectangular MDPs, the optimal policies are deterministic [Wiesemann *et al.*, 2013] and the optimal value function $v^\star \in \mathbb{R}^S$ is the unique fixed-point of the robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}^{SA}} : \mathbb{R}^S \to \mathbb{R}^S$

The SA-rectangular MDP corresponding to the dynamic soft-robust objective can be derived following a procedure similar to the 2-steps procedure in Section 4.2. The only difference in the procedure is that instead of assuming that the ambiguity set $\mathcal{P}^R$ is S-rectangular in Step 2, we would assume that the soft-robust ambiguity set $\mathcal{P}^R$ is SA-rectangular, i.e.,

$$\mathcal{P}^R = \bigotimes_{s \in \mathcal{S}} \mathcal{P}^R_{s,a}, \quad \text{where } \mathcal{P}^R_{s,a} = \Big\{ \sum_{\omega \in \Omega} \xi_\omega \cdot \hat{P}^\omega_{s,a} \;\Big|\; \xi \in \Xi \Big\}. \tag{A.2}$$

To differentiate from the S-rectangular soft-robust ambiguity sets denoted by $\mathcal{P}^R$, we denote the soft-robust ambiguity sets with the SA-rectangularity assumption by $\mathcal{P}^{RA}$. The resulting SA-rectangular soft-robust MDP objective which we denote by $\rho^{RA} : \Pi \to \mathbb{R}$ can be expressed as:

$$\rho^{RA}(\pi) \;=\; \min_{P \in \mathcal{P}^{RA}} \rho(\pi, P)\ . \tag{A.3}$$

The optimal value function $v^{RA} \in \mathbb{R}^S$ of the SA-rectangular soft-robust MDP satisfies the robust Bellman optimality equation $v^{RA} = \mathfrak{T}_{\mathcal{P}^{RA}} v^{RA}$ where $\mathfrak{T}_{\mathcal{P}^{RA}}$ is the robust Bellman optimality operator in Equation (A.1) for ambiguity set $\mathcal{P}^{RA}$. We term the operator $\mathfrak{T}_{\mathcal{P}^{RA}}$ as the SA-rectangular soft-robust Bellman optimality operator. Note that like all robust Bellman optimality operators defined for SA-rectangular ambiguity sets, the SA-rectangular soft-robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}^{RA}}$ as well, always results in deterministic optimal policies [Wiesemann *et al.*, 2013]. Hence, contrary to the S-rectangular soft-robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}^R}$, we need not solve the linear program in Equation (4.8) for computing it. Instead, it is sufficient to evaluate the convex combination of CVaR and mean of expected returns for each action independently and then choose the soft-robust returns corresponding to the best action as shown in (A.4). The CVaR measure can be computed in quasi-linear time [Acerbi and Tasche, 2002] and, therefore, the overall complexity of computing $(\mathfrak{T}_{\mathcal{P}^{RA}} v)(s)$ is $\mathcal{O}(SAN \log N)$. The linear program for the S-rectangular soft-robust Bellman optimality operator, on the other hand, takes polynomial time to solve.

To solve the SA-rectangular soft-robust MDP, we can use the Soft-Robust Value Iteration algorithm discussed in Section 4.3 with one key difference. Instead of using the S-rectangular soft-robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}^R}$ in (4.9) to derive the policy and construct $\Sigma_{\Phi^\top w}$, we would use the SA-rectangular soft-robust Bellman optimality operator $\mathfrak{T}_{\mathcal{P}^{RA}} v^{RA}$, i.e.,

$$\sigma_{\Phi^\top w}(s) = (\mathfrak{T}_{\mathcal{P}^{RA}} v)(s) = (\mathfrak{T}_{\mathcal{P}^{RA}}(\Phi^\top w))(s) = \max_{a \in \mathcal{A}} \min_{P_{s,a} \in \mathcal{P}^{RA}_{s,a}} P^\top_{s,a} \left(r_{s,a} + \gamma \cdot v \right) \tag{A.4}$$

In this setting, $\Sigma_{\Phi^\top w}$ is constructed as a vector of the SA-rectangular soft-robust Bellman optimality values for matrix Φ: $\{\sigma_{\Phi^\top w}(s_t)\}_{t=1}^M$.

B Auxiliary Results

The following lemma will be useful when bounding the RMDP approximation of the soft-robust objective.

Lemma B.1. *The vector-induced norms for a stochastic matrix $P \in \mathbb{R}^{S \times S}$ satisfy that*

$$\|P\|_\infty = \left\|P^\top\right\|_1 = 1\ .$$

Proof. Let $\mathcal{L}_1 = \{x \in \mathbb{R}^S \mid \|x\|_1 = 1\}$ be the L_1 ball and let $\mathcal{L}_\infty = \{x \in \mathbb{R}^S \mid \|x\|_\infty = 1\}$ be the L_∞ ball. Then, using basic linear algebra, definition of induced matrix norms in steps (a), and the duality of the vector L_1 and L_∞ norm in step (b), we can establish the desired result as follows:

$$\left\|P^\top\right\|_1 \overset{(a)}{=} \max_{x \in \mathcal{L}_1} \left\|P^\top x\right\|_1 = \max_{x \in \mathcal{L}_1} \max_{y \in \mathcal{L}_\infty} y^\top P^\top x \overset{(a)}{=} \max_{y \in \mathcal{L}_\infty} \|Py\|_\infty = \|P\|_\infty\ .$$

The result follows because, as it is well-known, $\|P\|_\infty = 1$ for any stochastic matrix P. □

The following generic lemma establishes the bounds on the error between a maximizer of a function and a maximizer of an approximation of that function.

Lemma B.2. *Let $x^\star \in \arg\max_{x\in\mathcal{X}} f(x)$ and $\tilde{x}^\star \in \arg\max_{x\in\mathcal{X}} \tilde{f}(x)$ be the maximizers of some function $f : \mathcal{X} \to \mathbb{R}$ and its approximation $\tilde{f} : \mathcal{X} \to \mathbb{R}$ respectively. Then the optimality gap of $\tilde{x}^\star$ in non-negative and bounded by:*

$$f(x^\star) - f(\tilde{x}^\star) \;\leq\; |f(x^\star) - \tilde{f}(x^\star)| + |f(\tilde{x}^\star) - \tilde{f}(\tilde{x}^\star)| \leq 2 \cdot \max_{x\in\mathcal{X}} |f(x) - \tilde{f}(x)| \,. \tag{B.1}$$

Moreover, when $\tilde{f}(x) \leq f(x)$ for all $x \in \mathcal{X}$, then the optimality gap of $\tilde{x}^\star$ reduces to:

$$f(x^\star) - f(\tilde{x}^\star) \;\leq\; f(x^\star) - \tilde{f}(x^\star) \,. \tag{B.2}$$

Proof. First, the following basic inequality follows by algebraic manipulation as:

$$\begin{aligned}
f(x^\star) - f(\tilde{x}^\star) &= f(x^\star) - f(\tilde{x}^\star) + \tilde{f}(\tilde{x}^\star) - \tilde{f}(\tilde{x}^\star) && \text{Add } 0 \\
&= f(x^\star) - \tilde{f}(\tilde{x}^\star) + \left(\tilde{f}(\tilde{x}^\star) - f(\tilde{x}^\star)\right) && \text{Rearrange} \\
&\leq f(x^\star) - \tilde{f}(x^\star) + \left(\tilde{f}(\tilde{x}^\star) - f(\tilde{x}^\star)\right) && \text{Optimality of } \tilde{x}^\star \qquad \text{(B.3a)} \\
&\leq \left|f(x^\star) - \tilde{f}(x^\star)\right| + \left|\tilde{f}(\tilde{x}^\star) - f(\tilde{x}^\star)\right| \,. && \qquad \text{(B.3b)}
\end{aligned}$$

Then, the inequality (B.1) follows from (B.3b) because $x^\star \in \mathcal{X}$ and $\tilde{x}^\star \in \mathcal{X}$ and therefore

$$\begin{aligned}
\left|f(x^\star) - \tilde{f}(x^\star)\right| &\leq \max_{x\in\mathcal{X}} |f(x) - \tilde{f}(x)| \\
\left|\tilde{f}(\tilde{x}^\star) - f(\tilde{x}^\star)\right| &\leq \max_{x\in\mathcal{X}} |f(x) - \tilde{f}(x)| \,.
\end{aligned}$$

The inequality (B.2) follows from (B.3a) because $\tilde{f}(x) \leq f(x)$ for all $x \in \mathcal{X}$ and therefore

$$\begin{aligned}
f(x^\star) - f(\tilde{x}^\star) &\leq f(x^\star) - \tilde{f}(x^\star) + \left(\tilde{f}(\tilde{x}^\star) - f(\tilde{x}^\star)\right) \\
&\leq f(x^\star) - \tilde{f}(x^\star) && \text{Because } \tilde{f}(\tilde{x}^\star) - f(\tilde{x}^\star) \leq 0 \,.
\end{aligned}$$

□

C Proofs: Section 3

Proof of Theorem 3.3. We omit the dependence of $\bar{\pi}_S$ and $\bar{\rho}_S$ on $\mathcal{D}$ to reduce clutter. No post-decision regret for $\lambda = 0$ follows by conditioning on the dataset $\mathcal{D}$ as:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D},P^\star}\left[\rho(\bar{\pi}_S, P^\star) - \bar{\rho}_S\right] &= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{P^\star}\left[\rho(\bar{\pi}_S, P^\star) - \bar{\rho}_S \;\middle|\; \mathcal{D}\right]\right] && \text{Property of } \mathbb{E} \\
&= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{P^\star}\left[\mathbb{E}\left[\rho(\bar{\pi}_S, P^\star) \;\middle|\; \mathcal{D}\right] - \bar{\rho}_S \;\middle|\; D\right]\right] && \text{Property of } \mathbb{E} \\
&= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{P^\star}\left[\rho(\bar{\pi}_S, P^\star) \;\middle|\; \mathcal{D}\right] - \bar{\rho}_S\right] && \bar{\rho}_S \text{ constant for } P^\star \\
&= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{P^\star}\left[\rho\left(\bar{\pi}_S, P^\star\right) \;\middle|\; \mathcal{D}\right] - \mathbb{E}_{P^\star}\left[\rho(\bar{\pi}_S, P^\star) \;\middle|\; \mathcal{D}\right]\right] = 0 && \text{from (3.1) and } \lambda = 0
\end{aligned}$$

The result for $\lambda > 0$ follows using the same steps and the fact that $\mathrm{CVaR}[X] \leq \mathbb{E}[X]$ for any random variable and therefore:

$$\mathbb{E}_{P^\star}\left[\rho(\bar{\pi}_S, P^\star) \;\middle|\; \mathcal{D}\right] \;\geq\; \bar{\rho}_S \,.$$

□

D Proofs: Section 4

Proof of Lemma 4.1. First, we show that the negations of the terms in the soft-robust objective (3.1) are support functions [Rockafellar and Uryasev, 2000] of convex sets. For any random variable $X : \Omega \to \mathbb{R}$ with probability measure function f, the robust representation of CVaR takes the following form (e.g., [Schied, 2004; Rockafellar and Uryasev, 2000]):

$$\mathrm{CVaR}^{\alpha}[X] = \min_{\xi\in\Delta^\Omega} \left\{ \sum_{\omega\in\Omega} \xi_\omega \cdot X(\omega) \;\middle|\; \xi_\omega \leq \frac{1}{1-\alpha} f_\omega, \forall \omega \in \Omega \right\} \,, \tag{D.1}$$

and, therefore, the CVaR term in (3.1) becomes

$$\mathrm{CVaR}^{\alpha}_{\hat{P}\sim f}\left[\rho(\pi,\hat{P})\right] = \min_{\xi\in\mathcal{Q}^{\mathrm{CVaR}}} \sum_{\omega=1}^{D} \xi_\omega \cdot \rho(\pi,\hat{P}^\omega) \tag{D.2}$$

where the set $\mathcal{Q}^{\mathrm{CVaR}}$ is defined as

$$\mathcal{Q}^{\mathrm{CVaR}} = \left\{ \xi \in \Delta^D \mid \xi_\omega \le \frac{1}{1-\alpha} f_\omega,\ \omega \in \Omega \right\} .$$

As a result of (D.2), the function $X \mapsto -\mathrm{CVaR}^{\alpha}_{\hat{P}\sim f}[-X]$ for $X:\Omega\to\mathbb{R}$ is the support function of set $\mathcal{Q}^{\mathrm{CVaR}}$. Note that we are interpreting the random variable X as a vector over $\mathbb{R}^\Omega$. Similarly, the mean term in (3.1) trivially equals to

$$\mathbb{E}_{\hat{P}\sim f}\left[\rho(\pi,\hat{P})\right] = \min_{\xi\in\mathcal{Q}^{\mathbb{E}}} \sum_{\omega\in\Omega} \xi_\omega \cdot \rho(\pi,\hat{P}^\omega), \tag{D.3}$$

where $\mathcal{Q}^{\mathbb{E}} = \{f\}$ is a singleton set. As with the CVaR above, it can be seen readily that the function $X \mapsto -\mathbb{E}_{\hat{P}\sim f}[-X]$ for $X:\Omega\to\mathbb{R}$ is the support function of $\mathcal{Q}^{\mathbb{E}}$.

Next, any two support functions $f_1(z) = \max_{q\in\mathcal{Q}_1} z^\top q$ and $f_2(z) = \max_{q\in\mathcal{Q}_2} z^\top q$ over convex sets $\mathcal{Q}_1, \mathcal{Q}_2$ satisfy for $\lambda\in[0,1]$ (see for example Chapter 13 of [Rockafellar, 1970]),

$$\lambda\cdot f_1(z) + (1-\lambda)\cdot f_2(z) = \max_q \left\{ q^\top z \mid q \in (\lambda\cdot\mathcal{Q}_1 + (1-\lambda)\cdot\mathcal{Q}_2) \right\} . \tag{D.4}$$

Multiplying the equality in (D.4) by -1, and using $-z$ as the parameter, we get:

$$\begin{aligned} -\lambda\cdot f_1(-z) - (1-\lambda)\cdot f_2(-z) &= -\max_q \left\{ -q^\top z \mid q \in (\lambda\cdot\mathcal{Q}_1 + (1-\lambda)\cdot\mathcal{Q}_2) \right\} \\ &= \min_q \left\{ q^\top z \mid q \in (\lambda\cdot\mathcal{Q}_1 + (1-\lambda)\cdot\mathcal{Q}_2) \right\} . \end{aligned} \tag{D.5}$$

Consider the sets $\mathcal{Q}_1 = \mathcal{Q}^{\mathrm{CVaR}}$, $\mathcal{Q}_2 = \mathcal{Q}^{\mathbb{E}}$ and support functions $f_1(X) = -\mathrm{CVaR}^{\alpha}[-X]$ and $f_2(X) = -\mathbb{E}[-X]$ in (D.5). Then, we can reformulate (3.1) as:

$$\begin{aligned} \rho^S(\pi) &= (1-\lambda)\cdot\mathbb{E}\left[\rho(\pi,\hat{P})\right] + \lambda\cdot\mathrm{CVaR}^{\alpha}\left[\rho(\pi,\hat{P})\right] \\ &= \min_{\xi\in\Delta^N} \left\{ \sum_{\omega\in\Omega} \xi_\omega\cdot\rho(\pi,\hat{P}^\omega) \mid \xi = \lambda\cdot\xi_1 + (1-\lambda)\cdot\xi_2,\ \xi_1\in\mathcal{Q}^{\mathrm{CVaR}},\ \xi_2\in\mathcal{Q}^{\mathbb{E}} \right\}. \end{aligned}$$

The the feasible set in the equation above in terms in $\xi, \xi_1 \in \mathbb{R}^N$ (note that $\xi_2 = f$) is represented by these inequalities,

$$\begin{aligned} \xi &= \lambda\cdot\xi_1 + (1-\lambda)\cdot f & \xi &\ge \mathbf{0} & \mathbf{1}^\top\xi &= 1 \\ \xi_1 &\le \frac{1}{1-\alpha}\cdot f & \xi_1 &\ge \mathbf{0} & \mathbf{1}^\top\xi_1 &= 1 \end{aligned}$$

Now, substituting $\xi_1 = {}^1\!/\!_\lambda\cdot(\xi - (1-\lambda)\cdot f)$ to the inequalities above, we get

$$\begin{aligned} 0 &= 0 & \xi &\ge \mathbf{0} & \mathbf{1}^\top\xi &= 1 \\ {}^1\!/\!_\lambda\cdot(\xi-(1-\lambda)\cdot f) &\le \frac{1}{1-\alpha}\cdot f & {}^1\!/\!_\lambda\cdot(\xi-(1-\lambda)\cdot f) &\ge \mathbf{0} & \mathbf{1}^\top({}^1\!/\!_\lambda\cdot(\xi-(1-\lambda)\cdot f)) &= 1 , \end{aligned}$$

which, using $\lambda\in[0,1]$ and $f\in\Delta^N$, reduces to:

$$\begin{aligned} 0 &= 0 & \xi &\ge \mathbf{0} & \mathbf{1}^\top\xi &= 1 \\ \xi &\le \frac{\lambda}{1-\alpha}\cdot f + (1-\lambda)\cdot f & \xi &\ge (1-\lambda)\cdot f & 0 &= 0 , \end{aligned}$$

The result then follows by simple algebraic manipulation. □

Proof of Proposition 4.2. The proof proceeds by first formulating the soft-robust optimization as a nonconvex quadratic optimization problem and then using the McCormick inequality to turn it to a MILP. Recall that the soft-robust objective in (3.1) is defined as:

$$\rho^S(\pi) \;=\; (1-\lambda)\cdot\mathbb{E}\left[\rho(\pi,\hat{P})\right] + \lambda\cdot\mathrm{CVaR}^{\alpha}\left[\rho(\pi,\hat{P})\right] . \tag{D.6}$$

From the standard definition of CVaR, the objective $\rho^S(\pi)$ becomes

$$\rho^S(\pi) = (1-\lambda)\sum_{\omega\in\Omega} f_\omega \cdot \rho(\pi, \hat{P}^\omega) + \lambda \cdot \max_{b\in\mathbb{R}} \left(b - {}^1\!/_{(1-\alpha)} \sum_{\omega\in\Omega} f_\omega \cdot \max\left\{0, b - \rho(\pi, \hat{P}^\omega)\right\}\right) ,$$

which can be formulated as the following linear program by introducing a variable $y_\omega = \max\{0, b - \rho(\pi, \hat{P}^\omega)\}$ as:

$$\begin{array}{ll} \underset{y\in\mathbb{R}^N, b\in\mathbb{R}}{\text{maximize}} & (1-\lambda)\sum_{\omega\in\Omega} f_\omega \cdot \rho(\pi, \hat{P}^\omega) + \lambda \cdot \max_{b\in\mathbb{R}} \left(b - {}^1\!/_{(1-\alpha)} \sum_{\omega\in\Omega} y_\omega \right) \\ \text{subject to} & y_\omega \geq f_\omega \cdot b - f_\omega \cdot \rho(\pi, \hat{P}^\omega), \qquad \forall \omega \in \Omega \\ & y \geq \mathbf{0}\,. \end{array} \tag{D.7}$$

Next, we express $\rho(\pi, \hat{P}^\omega)$ for each $\omega \in \Omega$ as the following optimization problem based on occupancy frequencies u as follows:

$$\begin{array}{rll} \rho(\pi, \hat{P}^\omega) = \underset{u\in\mathbb{R}^{S\times A}}{\text{maximize}} & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} u(s,a) \cdot \sum_{s'\in\mathcal{S}} P(s,a,s') r(s,a,s') & \\ \text{subject to} & \sum_{a\in\mathcal{A}} u(s,a) = \sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}} \gamma \cdot u(s',a') \cdot P^\omega(s',a',s) + p_0(s), & s \in \mathcal{S} \\ & u \geq \mathbf{0} & \\ & u(s,a) = \pi(s,a) \cdot \sum_{a'\in\mathcal{A}} u(s,a'), \qquad & \forall s \in \mathcal{S}, a \in \mathcal{A} \end{array} \tag{D.8}$$

The linear formulation in (D.8) is based on the dual linear program formulation of an MDP as described in (6.9.2) of Puterman [2005]. The last constraint ensures that only occupancy frequencies for π are considered and its correctness follows from Theorem 6.9.4 in Puterman [2005]. Further, one can scale the constants in (D.8) to get:

$$\begin{array}{rll} f_\omega \cdot \rho(\pi, \hat{P}^\omega) = \underset{u\in\mathbb{R}^{S\times A}}{\text{maximize}} & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} u(s,a) \cdot \sum_{s'\in\mathcal{S}} P(s,a,s') \cdot r(s,a,s') & \\ \text{subject to} & \sum_{a\in\mathcal{A}} u(s,a) = \sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}} \gamma \cdot u(s',a') \cdot P^\omega(s',a',s) + f_\omega \cdot p_0(s), & s \in \mathcal{S} \\ & u \geq \mathbf{0}\,, & \\ & u(s,a) = \pi(s,a) \cdot \sum_{a'\in\mathcal{A}} u(s,a'), \qquad & \forall s \in \mathcal{S}, a \in \mathcal{A} \end{array} \tag{D.9}$$

Finally, combining (D.7) with (D.9) we can formulate the optimization problem $\max_{\pi\in\Pi}$ as follows:

$$\begin{array}{lll} \underset{\substack{\pi\in[0,1]^{S\times A}, b\in\mathbb{R}, \\ u\in\mathbb{R}_+^{S\times A\times N}, y\in\mathbb{R}_+^N}}{\text{maximize}} & \lambda \cdot \left(b - \dfrac{1}{1-\alpha}\sum_{\omega\in\Omega} y(\omega)\right) + (1-\lambda) \cdot \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{\omega\in\Omega} u(s,a,\omega) \sum_{s'\in\mathcal{S}} r(s,a,s') \cdot P^\omega(s,a,s') & \\ \text{subject to} & y(\omega) - b \cdot f_\omega \;\geq\; -\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} u(s,a,\omega) \sum_{s'\in\mathcal{S}} P^\omega(s,a,s') \cdot r(s,a,s'), & \omega \in \Omega, \\ & \sum_{a\in\mathcal{A}} u(s,a,\omega) \;=\; \sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}} \gamma \cdot u(s',a',\omega) \cdot P^\omega(s',a',s) + f_\omega \cdot p_0(s), & s \in \mathcal{S},\ \omega \in \Omega, \\ & \sum_{a\in\mathcal{A}} \pi(s,a) \;=\; 1, & s \in \mathcal{S}, \\ & u(s,a,\omega) \;=\; \pi(s,a) \cdot \sum_{a'\in\mathcal{A}} u(s,a',\omega), & s \in \mathcal{S},\ a \in \mathcal{A},\ \omega \in \Omega. \end{array}$$

The MILP formulation then follows by upper-bounding the nonlinear constraint

$$u(s,a,\omega) = \pi(s,a) \cdot \sum_{a'\in\mathcal{A}} u(s,a',\omega)$$

by replacing the right-hand side using the McCormick relaxation (see, for example, Lemma 4.2 and the argument in Petrik and Luss [2016])

$$u(s,a,\omega) \leq \pi(s,a) \cdot \frac{f_\omega}{1-\gamma}$$

and the fact that $\pi(s,a) \in [0,1]$ and $u(s,a,\omega) \in [0, f_\omega/(1-\gamma)]$ (e.g. Lemma C.10 in Petrik [2010]). The optimality of the MILP formulation for deterministic policies holds because the McCormick relaxation is precise for the extreme values of the interval when $\pi(s,a) \in \{0,1\}$. □

Proof of Proposition 4.3. Without loss of generality, consider a transition model $P \in \mathcal{P}^D$ such that P can be represented using a distribution $\zeta \in \Xi$, that is $P = \sum_{\omega\in\Omega} \zeta_\omega \hat{P}^\omega$. Notice that $\forall s \in \mathcal{S},\ P_s$ can be written as $\sum_{\omega\in\Omega} \zeta_\omega \hat{P}^\omega_s$. By construction of $\mathcal{P}^R$ and since $\zeta \in \Xi$, it follows that $\forall s \in \mathcal{S},\ P_s \in \mathcal{P}^R_s$. Since $\mathcal{P}^R$ is a state-wise Cartesian product of $\mathcal{P}^R_s$, we can conclude that $\mathcal{P}^D \subseteq \mathcal{P}^R$.

The second part of the proposition thus immediately follows as $\forall \pi \in \Pi$, $\min_{P\in\mathcal{P}^R} \rho(\pi, P) \leq \min_{P\in\mathcal{P}^D} \rho(\pi, P)$ implies $\rho^R(\pi) \leq \rho^D(\pi)\, \forall \pi \in \Pi$. □

Proof of Proposition 4.4. Recall the s-rectangular soft-robust MDP objective.

$$\rho^R(\pi) \;=\; \min_{P\in\mathcal{P}^R} \; \rho\left(\pi, P\right) . \tag{D.10}$$

From (2.5), the S-rectangular soft-robust Bellman optimality operator can be written as

$$(\mathfrak{T}_{\mathcal{P}^R}\, v)(s) = \max_{d\in\Delta^A} \min_{P_s\in\mathcal{P}^R_s} \sum_{a\in\mathcal{A}} d_a P^\top_{s,a} \left(r_{s,a} + \gamma \cdot v\right). \tag{D.11}$$

Using simple algebraic manipulations, we can show that the dual of the right-hand-side of (D.11) can be formulated as the proposed linear program.

$$\begin{aligned} \max_{\substack{d\in\Delta^A,\, b\in\mathbb{R} \\ y\in\mathbb{R}^{|\Omega|}_+}} \quad & (1-\lambda) \sum_{\substack{a\in\mathcal{A} \\ \omega\in\Omega}} d_a f_\omega (\hat{P}^\omega_{s,a})^\top z_{s,a} \\ & \qquad + \lambda\Big(b - \frac{1}{1-\alpha} \sum_{\omega\in\Omega} f_\omega \cdot y_\omega\Big) \\ y_\omega \geq b - \sum_{a\in\mathcal{A}} & d_a (\hat{P}^\omega_{s,a})^\top z_{s,a}, \quad \omega \in \Omega\,. \end{aligned} \tag{D.12}$$

where $z_{s,a} = r_{s,a} + \gamma \cdot v$. □

E Proofs: Section 5

In this section, we describe the technical results that underlie the proof of Corollary 5.3. The following lemma bounds the difference between a convex combination of occupancy frequencies and the occupancy frequency of the convex combination of transition functions. This serves as the main technical tool when bounding the difference between dynamic and static objectives.

Lemma E.1. *Consider stochastic matrices $P_i \in (\Delta^S)^S, i = 1, \ldots, N$ with occupancy frequencies $h_i = (\boldsymbol{I} - \gamma \cdot P_i^\top)^{-1} p_0$. Let $P_\beta = \sum_{i=1}^N \beta_i \cdot P_i$ be the convex combination of P_i for a given $\beta \in \Delta^N$ and let $h_\beta = (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} p_0$ be its occupancy frequency. The convex combination of individual occupancy frequencies is denoted by $e_\beta = \sum_{i=1}^N \beta_i \cdot h_i$. Then:*

$$\left\| h_\beta - e_\beta \right\|_1 \leq \frac{\gamma}{1-\gamma} \cdot \epsilon_1\,,$$

when $\left\| h_i - h_j \right\|_1 \leq \epsilon_1$ for each $i = 1, \ldots, N$ and $j = 1, \ldots, N$.

Proof. Recall that the following identities hold for the occupancy frequencies [Puterman, 2005]:

$$h_i = \gamma \cdot P_i^\top h_i + p_0, \quad i = 1, \ldots, N, \qquad\qquad h_\beta = \gamma \cdot P_\beta^\top h_\beta + p_0\,. \tag{E.1}$$

Using the identity above and the fact that $\beta \in \Delta^S$, we obtain a similar expression for e_β:

$$e_\beta = \gamma \sum_{i=1}^N \beta_i \cdot P_i^\top h_i + p_0\,. \tag{E.2}$$

Because h_β need not be a convex combination of h_i we use the following representation of the difference between h_β and the

convex combination e_β of h_i:

$$\begin{aligned}
h_\beta - e_\beta &= \gamma \cdot P_\beta^\top h_\beta - \gamma \sum_{i=1}^N \beta_i \cdot P_i^\top h_i && \text{from (E.1) and (E.2)} \\
&= \gamma \cdot P_\beta^\top h_\beta - \gamma \cdot P_\beta^\top e_\beta + \gamma \cdot P_\beta^\top e_\beta - \gamma \sum_{i=1}^N \beta_i \cdot P_i^\top h_i && \text{add } 0 \\
&= \gamma \cdot P_\beta^\top h_\beta - \gamma \cdot P_\beta^\top e_\beta + \gamma \cdot \sum_{i=1}^N \beta_i \cdot P_i^\top e_\beta - \gamma \sum_{i=1}^N \beta_i \cdot P_i^\top h_i && \text{definition of } P_\beta \\
&= \gamma \cdot P_\beta^\top (h_\beta - e_\beta) + \gamma \cdot \sum_{i=1}^N \beta_i \cdot P_i^\top (e_\beta - h_i) \, . && \text{simplify}
\end{aligned}$$

Next, subtracting $\gamma \cdot P_\beta^\top (h_\beta - e_\beta)$ from both sides of the equality above, and multiplying by the appropriate matrix inverse leads to:

$$h_\beta - e_\beta \;=\; \gamma \sum_{i=1}^N \beta_i \cdot (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} P_i^\top (e_\beta - h_i) \, . \tag{E.3}$$

Applying the L_1 norm to both sides of (E.3) we get that:

$$\begin{aligned}
\|h_\beta - e_\beta\|_1 &= \left\| \gamma \sum_{i=1}^N \beta_i \cdot (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} P_i^\top (e_\beta - h_i) \right\|_1 \\
&\leq \gamma \sum_{i=1}^N \beta_i \cdot \left\| (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} P_i^\top (e_\beta - h_i) \right\|_1 && \text{from triangle inequality} \\
&\leq \gamma \sum_{i=1}^N \beta_i \cdot \left\| (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} P_i^\top \right\|_1 \|e_\beta - h_i\|_1 && \text{from } \|Ax\|_1 \leq \|A\|_1 \|x\|_1 \\
&\leq \gamma \sum_{i=1}^N \beta_i \cdot \left\| (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} \right\|_1 \left\| P_i^\top \right\|_1 \|e_\beta - h_i\|_1 && \text{from } \|AB\| \leq \|A\| \cdot \|B\| \\
&\leq \gamma \sum_{i=1}^N \beta_i \cdot \left\| (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} \right\|_1 \|e_\beta - h_i\|_1 \, .
\end{aligned}$$

Then Lemma B.1 combined with the Neumann series representation of matrix inverse implies that $\left\| (\boldsymbol{I} - \gamma \cdot P_\beta^\top)^{-1} \right\|_1 \leq 1/(1-\gamma)$. It can also be shown readily by basic algebra that $\|e_\beta - h_i\|_1 \leq \epsilon_1$. The desired result then follows because $\beta \in \Delta^S$. □

Proof of Theorem 5.1. Before proving the result, we recall several necessary definitions and identities. The static and dynamic returns are defined as

$$\rho^S(\pi) = \min_{\xi \in \Xi} \; \mathbb{E}_{\hat{P} \sim \xi} \left[\rho\big(\pi, \hat{P}\big) \right] \tag{E.4a}$$

$$\rho^D(\pi) = \min_{\xi \in \Xi} \; \rho\left(\pi, \mathbb{E}_{\hat{P} \sim \xi} \left[\hat{P} \right] \right) \, . \tag{E.4b}$$

Recall also that the return of a policy π in an MDP with the transition matrix $P_\pi \in (\Delta^S)^S$ can be expressed in terms of the occupancy frequency h_π, defined in (5.1), as

$$\rho(\pi, P) = p_0^\top v_\pi = p_0^\top (\boldsymbol{I} - \gamma \cdot P_\pi)^{-1} r_\pi = h_\pi^\top r_\pi \, . \tag{E.5}$$

Now, let $\xi^S \in \Delta^{|\Omega|}$ and $\xi^D \in \Delta^{|\Omega|}$ be optimal in (E.4a) and (E.4b). Then the soft-robust returns in (E.4) can be expressed in terms of their occupancy frequencies using (E.5) as

$$\begin{aligned}
\rho^S(\pi) &= \mathbb{E}_{\hat{P} \sim \xi^S} \left[\rho\big(\pi, \hat{P}\big) \right] = \sum_{\omega \in \Omega} \xi_\omega^S \cdot p_0^\top (\boldsymbol{I} - \gamma \cdot \hat{P}_\pi^\omega)^{-1} r_\pi = \sum_{\omega \in \Omega} \xi_\omega^S \cdot (h_\pi^\omega)^\top r_\pi \\
\rho^D(\pi) &= \rho\left(\pi, \mathbb{E}_{\hat{P} \sim \xi^D} \left[\hat{P} \right] \right) = p_0^\top \left(\boldsymbol{I} - \gamma \sum_{\omega \in \Omega} \xi_\omega^D \cdot \hat{P}_\pi^\omega \right)^{-1} r_\pi = (h_\pi^{\xi^D})^\top r_\pi \, .
\end{aligned} \tag{E.6}$$

where $h_\pi^{\xi^S} = \left(\boldsymbol{I} - \gamma \sum_{\omega\in\Omega} \xi_\omega^S \cdot \hat{P}_\pi^\omega\right)^{-1}$ and $h_\pi^\omega = (\boldsymbol{I} - \gamma \cdot \hat{P}_\pi^\omega)^{-1}$.

Next, we get for each $\pi \in \Pi$ that

$$\begin{aligned}
\rho^D(\pi) - \rho^S(\pi) &\leq \mathbb{E}_{\hat{P}\sim\xi^D}\left[\rho\big(\pi,\hat{P}\big)\right] - \rho\left(\pi, \mathbb{E}_{\hat{P}\sim\xi^D}\left[\hat{P}\right]\right) && \text{From (E.6) and } \xi^D \in \Xi \\
&= (h_\pi^{\xi^D})^\top r_\pi - \sum_{\omega\in\Omega} \xi_\omega^h \cdot (h_\pi^\omega)^\top r_\pi && \text{From (E.6)} \\
&\leq \left\| h_\pi^{\xi^D} - \sum_{\omega\in\Omega} \xi_\omega^D \cdot h_\pi^\omega \right\|_1 \cdot \|r_\pi\|_\infty && \text{Holder's inequality} \\
&\leq \frac{\gamma\cdot\epsilon_1}{1-\gamma}\cdot r_{\max} && \text{From Lemma E.1 .}
\end{aligned}$$

Similarly, the reverse inequality follows as

$$\begin{aligned}
\rho^S(\pi) - \rho^D(\pi) &\leq \mathbb{E}_{\hat{P}\sim\xi^S}\left[\rho\big(\pi,\hat{P}\big)\right] - \rho\left(\pi, \mathbb{E}_{\hat{P}\sim\xi^S}\left[\hat{P}\right]\right) && \text{From (E.6) and } \xi^S \in \Xi \\
&= \sum_{\omega\in\Omega} \xi_\omega^S \cdot (h_\pi^\omega)^\top r_\pi - (h_\pi^{\xi^S})^\top r_\pi && \text{From (E.6)} \\
&\leq \left\| \sum_{\omega\in\Omega} \xi_\omega^S \cdot h_\pi^\omega - h_\pi^{\xi^S} \right\|_1 \cdot \|r_\pi\|_\infty && \text{Holder's inequality} \\
&\leq \frac{\gamma\cdot\epsilon_1(\pi)}{1-\gamma}\cdot r_{\max} && \text{From Lemma E.1 .}
\end{aligned}$$

Combining the two inequalities above, we obtain that

$$|\rho^S(\pi) - \rho^D(\pi)| \;\leq\; \frac{\gamma\cdot\epsilon_1(\pi)}{1-\gamma}\cdot r_{\max}\,,$$

which proves the result. □

Proof of Lemma 5.2. To establish this bound, define a robust Bellman value operator $\mathfrak{T}^{\pi,\xi} : \mathbb{R}^S \to \mathbb{R}^S$ for any policy $\pi \in \Pi$, nature's response $\xi \in \Xi$, value function $v \in \mathbb{R}^S$, and state $s \in \mathcal{S}$ as

$$\left(\mathfrak{T}^{\pi,\xi} v\right)_s \;=\; \sum_{a\in\mathcal{A}} \sum_{\omega\in\Omega} \xi_\omega \cdot \pi_{s,a} \cdot (\hat{P}_{s,a}^\omega)^\top (r_{s,a} + \gamma\cdot v)\,.$$

The operator $\mathfrak{T}^{\pi,\xi}$ is linear and has a unique fixed point $v^{\pi,\xi} \in \mathbb{R}^S$ which satisfies $\mathfrak{T}^{\pi,\xi} v^{\pi,\xi} = v^{\pi,\xi}$ [Ho *et al.*, 2018]. Similarly, we define a robust S-rectangular Bellman value operator $\mathfrak{T}^\pi : \mathbb{R}^S \to \mathbb{R}^S$ defined for any policy $\pi \in \Pi$, value function $v \in \mathbb{R}^S$, and state $s \in \mathcal{S}$ as

$$\left(\mathfrak{T}^\pi v\right)_s \;=\; \min_{\xi\in\Xi} \left(\mathfrak{T}^{\pi,\xi} v\right)_s\,.$$

Note that for a fixed policy $\pi \in \Pi$, the operator $\mathfrak{T}^\pi$ is equivalent to the Bellman operator in MDPs and satisfies the same properties. Let π_D^* be the optimal policy that optimizes $\rho^D(\pi)$. Equipped with the definitions above, we proceed to bound the error $\rho^D(\pi_D^\star) - \rho^R(\pi_D^\star)$. Let $\xi_D^\star$ be the minimizer for $\rho^D(\pi_D^\star)$ in (4.2) and therefore

$$\rho^D(\pi_D^\star) \;=\; p_0^\top v^{\pi_D^\star,\xi_D^\star}\,.$$

Similarly, let $\xi_R^\star$ be the minimizer to $\rho^R(\pi_D^\star)$ in (4.7) and therefore

$$\rho^R(\pi_D^\star) \;=\; p_0^\top v^{\pi_D^\star,\xi_R^\star}\,.$$

Exploiting the fact that $\mathfrak{T}^\pi$ is an MDP Bellman operator and using standard arguments for MDP value functions (for example, Corollary 4 in [Ho *et al.*, 2018]) we get that:

$$\begin{aligned}
\rho^D(\pi_D^\star) - \rho^R(\pi_D^\star) = p_0^\top v^{\pi_D^\star,\xi_D^\star} - p_0^\top v^{\pi_D^\star,\xi_R^\star} &\leq \|p_0\|_1 \cdot \left\| v^{\pi_D^\star,\xi_D^\star} - v^{\pi_D^\star,\xi_R^\star} \right\|_\infty = \left\| v^{\pi_D^\star,\xi_R^\star} - v^{\pi_D^\star,\xi_R^\star} \right\|_\infty \\
&\leq \frac{1}{1-\gamma}\cdot \left\| \mathfrak{T}^{\pi_D^\star} v^{\pi_D^\star,\xi_R^\star} - v^{\pi_D^\star,\xi_R^\star} \right\|_\infty \leq \frac{1}{1-\gamma}\cdot\epsilon_2\,,
\end{aligned}$$

for the ϵ_2 stated in the theorem. Finally, we employ Lemma B.2 combined with Proposition 4.3 to show that

$$0 \leq \rho^D(\pi_D^\star) - \rho^R(\pi_R^\star) \leq \rho^D(\pi_D^\star) - \rho^R(\pi_D^\star) \leq \frac{1}{1-\gamma}\cdot\epsilon_2\,,$$

which shows the desired result. □

Proof of Corollary 5.3. The result follows by algebraic manipulation as

$$\begin{aligned}
\rho^S(\pi_S^\star) - \rho^S(\pi_R^\star) &= \rho^S(\pi_S^\star) \overbrace{-\rho^D(\pi_D^\star) + \rho^D(\pi_D^\star)}^{=0} \overbrace{-\rho^D(\pi_R^\star) + \rho^D(\pi_R^\star)}^{=0} - \rho^S(\pi_R^\star) \\
&= \underbrace{\rho^S(\pi_S^\star) - \rho^D(\pi_D^\star)}_{\text{Lemma B.2 \& Theorem 5.1}} + \rho^D(\pi_D^\star) - \rho^D(\pi_R^\star) + \underbrace{\rho^D(\pi_R^\star) - \rho^S(\pi_R^\star)}_{\text{Theorem 5.1}} \\
&\leq \frac{2\gamma \cdot r_{\max} \cdot \epsilon_1}{1-\gamma} + \rho^D(\pi_D^\star) - \rho^D(\pi_R^\star) \\
&= \frac{2\gamma \cdot r_{\max} \cdot \epsilon_1}{1-\gamma} + \underbrace{\rho^D(\pi_D^\star) - \rho^D(\pi_R^\star)}_{\text{Lemma 5.2}} \\
&\leq \frac{2\gamma \cdot r_{\max} \cdot \epsilon_1}{1-\gamma} + \frac{\epsilon_2}{1-\gamma} \,.
\end{aligned}$$

□

F Experimental Details

F.1 Baselines

We describe below the two custom baselines algorithms Soft-Robust (Derman) Soft Actor-Critic (SRD-SAC) and Robust Soft Actor-Critic (R-SAC) algorithms that we use for comparing the performance of the dynamic soft-robust objective. The SRD-SAC algorithm extends the Soft Actor-Critic (SAC) [Haarnoja *et al.*, 2018] to use soft-robust updates [Derman *et al.*, 2018]. Similarly, the R-SAC algorithm extends the SAC algorithm to use robust updates [Mankowitz *et al.*, 2020; Iyengar, 2005]. The SAC algorithm is a variant of the standard policy iteration algorithm that learns maximum-entropy optimal policies. We note that the soft-robust and robust updates only affect the policy-evaluation step of the SAC algorithm. Hence, we will only describe the change in the policy evaluation step. In the policy evaluation step, the SAC algorithm estimates the value function and action-value function of a policy according to an objective that maximizes the future expected returns and entropy of the optimal policy.

Let $V : \mathbb{R}^S \to \mathbb{R}^S$ and $Q : \mathbb{R}^{S\times A} \to \mathbb{R}^{S\times A}$ denote the value function and action-value function of policies respectively. We will refer to the function approximators used by the SAC algorithm to represent the action-value function and the value function as the Q-network and V-network respectively.

The SAC algorithm optimizes the Q-network and V-network in an off-policy manner to minimize the soft-Bellman residual error.

$$J(Q) = \mathbb{E}_{s_t,a_t\sim D}[(Q(s_t,a_t) - \bar{Q}(s_t,a_t))^2] \tag{F.1}$$

$$\text{where } \bar{Q}(s_t,a_t) = \mathbb{E}_{s_{t+1}\sim D}[r_{s_t,a_t} + \gamma \bar{V}(s_{t+1})] \tag{F.2}$$

$$J(V) = \mathbb{E}_{s_t\sim D}[(V(s_t) - \mathbb{E}_{a_t\sim\pi(s_t)}[Q(s_t,a_t) - \log(\pi(s_t,a_t))]^2] \tag{F.3}$$

where D is data collected using some behavior policy $\pi \in \Pi$ and $\bar{V}(s_t)$ is the value function estimated at state s_t as predicted by a target value function network, used for stabilizing training [Haarnoja *et al.*, 2018]. The weights of the target value function are updated as exponentially moving weighted average of the weights of the V-network.

In the SRD-SAC algorithm, we optimize the Q-network to minimize the soft-robust Bellman residual error.

$$J_{SRD}(Q) = \mathbb{E}_{s_t,a_t\sim \bar{P}}[(Q(s_t,a_t) - \bar{Q}(s_t,a_t))^2] \tag{F.4}$$

$$\text{where } \bar{Q}(s_t,a_t) = \sum_{\omega\in\Omega} f_\omega \cdot \mathbb{E}_{s_{t+1}\sim \hat{P}^\omega(s_t,a_t)}[r_{s_t,a_t} + \gamma \bar{V}(s_{t+1})] \tag{F.5}$$

$$J_{SRD}(V) = \mathbb{E}_{s_t\sim \bar{P}}[(V(s_t) - \mathbb{E}_{a_t\sim\pi(s_t)}[Q(s_t,a_t) - \log(\pi(s_t,a_t))]^2] \tag{F.6}$$

Notice that, in this case, the data samples for the updates are collected by simulating the nominal model $\bar{P}$.

Similarly, in the R-SAC algorithm, we optimize the Q-network to minimize the robust Bellman residual error.

$$J_R(Q) = \mathbb{E}_{s_t,a_t\sim \bar{P}}[(Q(s_t,a_t) - \bar{Q}(s_t,a_t))^2] \tag{F.7}$$

$$\text{where } \bar{Q}(s_t,a_t) = \min_{\omega\in\Omega} \mathbb{E}_{s_{t+1}\sim \hat{P}^\omega(s_t,a_t)}[r_{s_t,a_t} + \gamma \bar{V}(s_{t+1})] \tag{F.8}$$

$$J_R(V) = \mathbb{E}_{s_t\sim \bar{P}}[(V(s_t) - \mathbb{E}_{a_t\sim\pi(s_t)}[Q(s_t,a_t) - \log(\pi(s_t,a_t))]^2] \tag{F.9}$$

Parameter	*Value*
Features	2nd order polynomial
Policy learning rate	3e-4
Q-value network learning rate	3e-4
V-network learning rate	3e-4
Train iterations	2000
Episodes per iteration	30
Test episodes per transition model	100
Train transition models	50
Test transition models	100
States sampled per model and update	100
Batch size	150
Target update rate	0.01
Discount factor	0.9
Hidden layers	(400,300)
Activation	Relu

Table 1: Cancer Simulator: SRD-SAC and R-SAC

F.2 Soft-Robust Soft Actor-Critic (SR-SAC)

Taking inspiration from SRD-SAC [Derman *et al.*, 2018], we can easily extend the Soft Actor-Critic algorithm to optimize the dynamic soft-robust criterion in 4.7. The only change required is in the SAC update for optimizing the Q-network, i.e., instead of optimizing the Q-network to minimize the Bellman residual error, we optimize it to minimize the soft-robust Bellman error.

$$J_{SR}(Q) = \mathbb{E}_{s_t,a_t\sim\bar{P}}[(Q(s_t,a_t) - \bar{Q}(s_t,a_t))^2] \tag{F.10}$$

$$\text{where } \bar{Q}(s_t,a_t) = \lambda \operatorname{CVaR}_{\hat{P}^\omega}[\mathbb{E}_{s_{t+1}\sim\hat{P}^\omega(s_t,a_t)}[r_{s_t,a_t} + \gamma\bar{V}(s_{t+1})]] + (1-\lambda)\mathbb{E}_{\hat{P}^\omega}\left[\mathbb{E}_{s_{t+1}\sim\hat{P}^\omega(s_t,a_t)}[r_{s_t,a_t} + \gamma\bar{V}(s_{t+1})]\right] \tag{F.11}$$

$$J_{SR}(V) = \mathbb{E}_{s_t\sim\bar{P}}[(V(s_t) - \mathbb{E}_{a_t\sim\pi(s_t)}[Q(s_t,a_t) - \log(\pi(s_t,a_t))]^2] \tag{F.12}$$

F.3 Population Domain

This MDP consists of 51 states, each represents the current pest population as determined by trapping (0 means no pest population). There are 5 actions available, with each action representing the use of an increasingly potent pesticide. The true transition probabilities are based on a logistic model of population growth as described in [Tirinzoni *et al.*, 2018]. The discount factor is $\gamma = 0.9$.

To compute the posterior distribution over $\hat{P}$, we gather 300 state-action transition samples from a single episode. Using these transition samples, we fit an exponential population model [Kery and Schaub, 2012] using the JAGS modeling language [Plummer, 2003] and sample 100 posterior samples using MCMC. We use these samples to formulate and solve the MILP in Figure 2 and to run Algorithm 4.1. We use confidence $\alpha = 0.7$ for both the percentile criterion and soft-robust objective for the evaluation. We also use $\lambda = 0.5$ for the soft-robust objective. We use 100 samples from the posterior distribution both to compute and evaluate the methods' returns.

F.4 Cancer Simulator

The cancer simulator models the growth of tumors in cancer patients. The state is a 4-dimensional vector that captures the dynamics of the tumor's growth. The monthly binary action determines whether to administer chemotherapy to the patient [Gottesman *et al.*, 2020; Ribba *et al.*, 2012]. The discount factor γ is set to 0.9.

We model the true transition probability model P_{s_t,a_t} as a multivariate Normal random variable with mean $w \in \mathbb{R}^l$ and diagonal covariance matrix $\Sigma \in \mathbb{R}^{l\times l}$. The mean and variance are linearly weighted functions of state features. We sample a batch of data consisting of 600 samples (20 trajectories) using the cancer simulator with transition noise=0.03 and the ϵ-greedy behavior policy provided by [Gottesman *et al.*, 2020] with $\epsilon = 0.1$. Using the sampled data, we train a multivariate Bayesian linear regression model to predict the posterior distribution of weights w and the covariance matrix Σ. We assume a Normal prior $\mathcal{N}(0,1)$ for each element of the weight vector w and a $\text{HalfNormal}(0.001)$ prior for the elements of the covariance matrix Σ. We construct the train uncertainty set as shown in Algorithm 4.1 by sampling 50 weight vectors and covariance matrices from the posterior distribution using the MCMC algorithm [Hoffman and Gelman, 2011]. We similarly construct the test uncertainty set by sampling 100 weight vectors and covariance matrices from the posterior distribution. We keep the test and the train sets consistent across all the experiments on the cancer simulator. Tables 1 and 2 summarizes the parameters of the methods we compare.

Parameter	*Value*
Features	2nd order polynomial
Train iterations	150
Episodes per iteration	30
Test episodes per transition model	100
Train transition models	50
Test transition models	100
Batch size	150
States sampled per model and update	100
Discount factor	0.9

Table 2: Cancer Simulator: SRVI

F.5 Additional Experiments

Inventory Management Problem

This domain represents a common dilemma encountered by retailers while procuring goods for future sales. We assume an infinite horizon time period during which a retailer procures and sells only 1 type of item. The procured goods are stored in an inventory of limited capacity. Goods stored in the inventory have a fixed holding cost-per-unit per time step. The demands received by the retailer at time t are served at time $t+1$ with goods available in the inventory. Any demand that is not satisfied is backlogged with a fixed backlog cost-per-unit per time step. At every time step, the retailer attempts clearance of as many backlogged demands as possible with the available inventory. The states in this context represent the current quantity of goods in the inventory and actions represent the orders placed by the retailer. In our setup, the maximum order quantity is 40 units and the inventory has a fixed capacity of 50 units. The minimum demand is 0 and the maximum demand is 50. For the sake of simplicity, we disable backlogging in our experiments. The variable cost, per-unit purchase price, holding cost, backlog cost, and sales price are 2.49,3.99, 0.1, 0.15, and 4.99 respectively. We set risk level $\alpha = 0.95$ and discount factor $\gamma = 0.9$. The reward at any time step is the profit incurred from the sales. The demand for goods per time step is stochastic, which in turn makes the transitions stochastic. We assume that the demand comes from the Poisson distribution with an unknown rate λ. We assume that the true value of λ is 10.

To generate a batch of data, we use a sampling policy that always purchases the maximum available goods. This enables us to sample uncensored demands which in turn allows us to compute the posterior distribution of the demands analytically. We model the posterior distribution of demands as a Gamma distribution and assume a Gamma prior with shape=4 and scale=6. We fit the posterior distribution using a batch of 50 transitions obtained using the sampling policy and true demand distribution. We sample 100 demand models from the posterior distribution, for training and testing the SRVI RL agent against other baseline RL agents. We assume that the initial distribution is uniform across all the states.

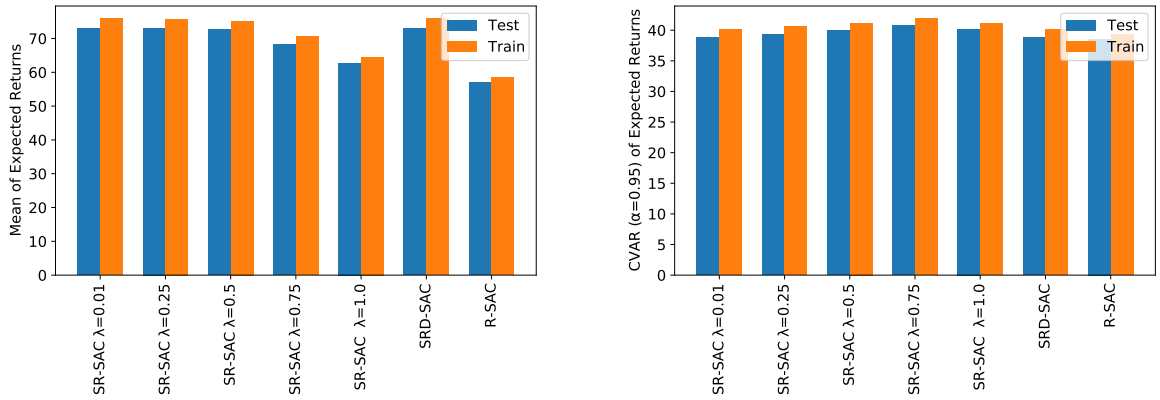

Figure 5: Mean and Robust performance of SR-SAC, SRD-SAC and R-SAC in Inventory Management domain. SR-SAC outperforms R-SAC in mean performance and SRD-SAC in robust performance.

Riverswim

The Riverswim domain is inspired from the Riverswim domain in [Strehl and Littman, 2004]. This domain is modeled as 20-states 2-actions MDP with discount factor $\gamma = 0.9$. The states $s_1, \ldots, s_{20}$ represent the current position of the agent in the river, and the actions a_1 and a_2 represent the act of swimming 1 unit in the direction of the river's current and 1 unit against

the direction of the river's current respectively. The direction of the river's current is from $s_{20} \rightarrow s_1$. Choosing action a_1 in s_i results in transitioning to the state s_{i-1} with probability 1. On the other hand, if the agent chooses a_2, it will transition to the state s_{i+1} with probability 0.2, or to the state s_{i-1} with probability 0.5 or stay in s_i itself with probability 0.3. In states where s_{i-1} or s_{i+1} is undefined, the agent will continue to stay in the current state with the respective probability. The reward received on reaching state s_{20} is +100. The agent also received a reward of +5 each time it moves 1 step closer to state s_{20}. Hence to maximize its returns, the agent has to swim towards state s_{20} i.e., against the direction of the river's current. We assume that the initial distribution is uniform across states.

The posterior distribution over $\bar{P}$ is modeled as a Dirichlet Distribution while assuming a uniform Dirichlet prior. We sample 15 state-action transition samples from a single episode and use them for analytically computing the concentration parameters of the posterior Dirichlet distribution. We sample 100 transition models from the posterior distribution for training and 700 transition models for testing purposes. We set risk level $\alpha = 0.95$.

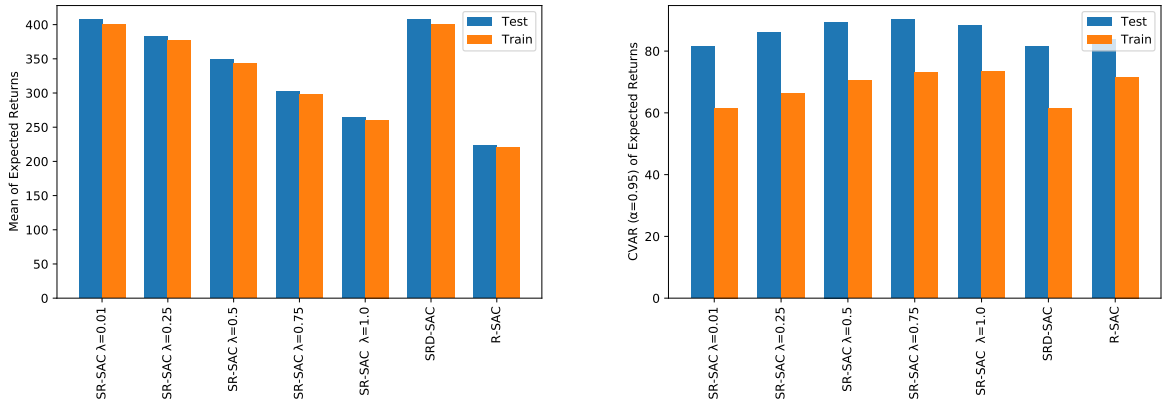

Figure 6: Mean and Robust performance of SR-SAC, SRD-SAC and R-SAC in Riverswim domain. SR-SAC achieves the best mean performance at $\lambda = 0.01$ and the best robust performance at $\lambda = 0.75, 1.0$.

F.6 Code Details

Since our code is heavily dependent on one of our in-house libraries which cannot be easily de-anonymized, we will release the code after the paper is published.

G Related Work

Numerous robust objectives for mitigating model uncertainty have been proposed in the literature. We discuss a number of them in more detail in this section.

Dynamic robust objectives. A vast majority of work in Robust RL has studied objectives that assume a dynamic uncertainty model for achieving tractability. Mankowitz *et al.* [2020] proposed robust algorithms that optimize entropy-regularized policies against the worst model in the uncertainty set. While these algorithms scale to continuous state and action spaces, they do not provide any kind of probabilistic guarantees on the expected returns like our framework and compute overly conservative policies. On the other hand, [Derman *et al.*, 2018] proposed a soft-robust actor-critic that optimizes only the mean of the expected returns computed for a fixed distribution over models in the uncertainty set.

In contrast to the prior work, our dynamic soft-robust algorithm dynamically computes the distribution over uncertain models that provide guarantees on the user-specified quantile of the expected returns for the optimal policy. Derman *et al.* [2019] introduced scalable algorithms that optimize an RMDP objective while accounting for changing dynamics. This framework also suffers from the shortcomings of the percentile criterion. Another related work [Xu and Mannor, 2012] constructs a plausible framework to incorporate any probabilistic information about the uncertain models in RMDPs and shows a connection between coherent risk measures and distributionally-robust MDPs. However, their main objective is different from ours as they do not aim to address the shortcomings of the percentile criterion. Finally, in the same vein as our work, policy gradient methods for optimizing CVaR of expected returns have been studied by Hiraoka *et al.* [2019]. Nonetheless, these methods [Hiraoka *et al.*, 2019] do not exploit the coherent properties of this measure and only tend to find local optimal policies.

Static robust objectives. Few works have focused on optimizing robust objectives while retaining the static uncertainty model assumption [Buchholz and Scheftelowitsch, 2020; Steimle *et al.*, 2018; Buchholz and Scheftelowitsch, 2019; Meraklı and Küçükyavuz, 2019]. However, we note that the robust objectives used in these works are quite different than ours. Steimle *et al.* [2018] proposed a mixed-integer linear program and a fast heuristic algorithm to optimize the weighted expected returns across different models in a finite-horizon setting, whereas our objective optimizes the policy for the worst distribution over models in the ambiguity set. Buchholz and Scheftelowitsch [2020] uses the same objective as in Steimle *et al.* [2018], but considers both finite and infinite-horizon settings. The authors of Steimle *et al.* [2018] proposed a MILP for calculating the exact deterministic policy in the finite-horizon setting, and other approximation algorithms that optimize a finite class of randomized Markovian policies for the infinite-horizon case. In another similar work, Meraklı and Küçükyavuz [2019] proposed an approximate MILP for optimizing the percentile-criterion. However, since the original objective is non-convex, the approximation may not generate optimal deterministic solutions.

Ambiguity set optimization. Some related work has considered partial correlations between uncertain model parameters to mitigate the conservativeness of learned policies [Derman *et al.*, 2019; Mannor *et al.*, 2016; Goyal and Grand-Clement, 2020b; Mannor *et al.*, 2012]. Examples of such works are k-rectangular [Mannor *et al.*, 2016, 2012] and r-rectangular [Goyal and Grand-Clement, 2020b] ambiguity sets. These approaches mitigate the conservativeness of S- and SA-rectangular ambiguity sets by capturing correlations between the uncertainty and by limiting the number of times the uncertain parameters deviate from the mean parameters. Despite this progress, most of this works still relies on weak statistical concentration bounds for the construction of ambiguity sets, which can make the ambiguity sets unnecessarily large and result in conservative policies. In contrast, the soft-robust ambiguity sets are convex and can be precisely constructed without using concentration bounds. Therefore, the soft-robust ambiguity sets are relatively tighter and result in learning less conservative solutions.