Constrained Reinforcement Learning Under Model Mismatch

Zhongchang Sun [1] **Sihong He** [2] **Fei Miao** [2] **Shaofeng Zou** [1 3]

Abstract

Existing studies on constrained reinforcement learning (RL) may obtain a well-performing policy in the training environment. However, when deployed in a real environment, it may easily violate constraints that were originally satisfied during training because there might be model mismatch between the training and real environments. To address this challenge, we formulate the problem as constrained RL under model uncertainty, where the goal is to learn a policy that optimizes the reward and at the same time satisfies the constraint under model mismatch. We develop a Robust Constrained Policy Optimization (RCPO) algorithm, which is the first algorithm that applies to large/continuous state space and has theoretical guarantees on worst-case reward improvement and constraint violation at each iteration during the training. We show the effectiveness of our algorithm on a set of RL tasks with constraints.

1. Introduction

In reinforcement learning (RL), the agent aims to learn a policy that maximizes the expected cumulative reward by interacting with an environment (Sutton & Barto, 2018). However, in real-life applications, e.g., robotics (Levine et al., 2016; Ono et al., 2015), health care (Yu et al., 2019a), autonomous driving (Kiran et al., 2020; Fisac et al., 2018; He et al., 2023a;b) and industry automation (Gasparik et al., 2018), where it is crucial to meet constraints while maximizing reward, application of RL remains limited. For example, an unmanned aerial vehicle performing post-disaster search and rescue needs to return before running out of battery, and communication system needs to maximize throughput while adhering to power consumption constraints.

[1]Department of Electrical Engineering, University at Buffalo, New York, USA [2]School of Computing, University of Connecticut, Storrs, USA [3]Department of Computer Science & Engineering, University at Buffalo, New York, USA. Correspondence to: Shaofeng Zou <szou3@buffalo.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

The framework of constrained Markov Decision Process (CMDP) was developed (Altman, 1999) to tackle the above challenge and the goal is to search for one policy that maximizes the overall reward among the policies that satisfy the constraint, and an optimal policy for CMDP can be found via linear programming.

When a well-performing policy trained using a simulator is deployed in a real environment, it may easily violate constraints that were originally satisfied during training because there might be model mismatch between the training and real environments. This could be due to environment non-stationarity, sim-to-real gap and adversarial attacks. Despite its practical importance, studies on the problem of robust RL under constraints are rather limited in the literature. Several attempts were made in (Russel et al., 2020; Mankowitz et al., 2020), where two heuristic approaches were proposed. Their basic idea is to first evaluate the worst-case performance of the policy over the uncertainty set, and then use that together with classical policy improvement methods, e.g., policy gradient, to update the policy. However, there is no guarantee to obtain an improved robust policy by doing so. A robust primal-dual approach was developed in (Wang et al., 2022), which however cannot guarantee monotonic robust reward improvement or constraint satisfaction during the training. Also, the results in (Wang et al., 2022) are limited to the tabular case with finite state and action spaces.

In this paper, we study the problem of constrained RL under model mismatch. Specifically, we consider an uncertainty set of transition kernels that characterizes the potential model mismatch (see (Siddique et al., 2019) for an example of uncertainty set construction). The goal is to guarantee that for any MDP in the uncertainty set, the constraint is always satisfied. Among these policies, we aim to identify one that maximizes the worst-case accumulated reward over the uncertainty set. Solution to the above problem is robust in that for any MDP in the uncertainty set, the constraint is always satisfied, and at the same time, the overall reward of the obtained policy is also robust to model mismatch.

We develop a robust constrained policy optimization (RCPO) algorithm, and theoretically bound the constraint violation for any transition kernel in the uncertainty set , and the worst-case reward improvement over the uncertainty set for every policy during training. One thing to highlight is

that our algorithm applies to Markov Decision Processes (MDPs) with continuous state space, which allows applications to large scale practical problems.

One essential theoretical result that drives our RCPO algorithm development is a generalization of the performance difference lemma (Kakade & Langford, 2002; Achiam et al., 2017) to robust MDPs. Specifically, we consider the robust value function, which measures the worst-case performance of a policy over the uncertainty set. We bound the difference between robust value functions for two different policies using the difference between the two policies.

Our algorithm consists of two steps for each update: (i) robust policy improvement step and (ii) projection.

Step (i) uses our robust performance difference lemma to develop a local approximation of the robust value function, and design a robust policy improvement step that searches in the neighborhood of the current policy. This step generalizes the trust region method (Schulman et al., 2015a) to the robust setting and guarantees robust reward improvement. Unlike the non-robust setting where there is only one transition kernel which stays the same throughout the training, under model uncertainty the worst-case transition kernel changes with the policy and is different for reward and utility (see Section 3.2). One challenge is that the local approximation implicitly depends on the policy to be optimized, which is in the neighborhood of the current policy, through its worst-case transition kernel, making the optimization intractable. We develop a novel approximation using the current policy as a surrogate, and prove that such an approximation still provides guaranteed robust reward improvement (and later in step (ii) robust constraint satisfaction).

The obtained policy guarantees the reward improvement, but may violate the constraint due to bad initialization and stochastic noise. This leads to a potential problem of the constrained policy optimization (CPO) approach in (Achiam et al., 2017) that there may not exist any feasible solution during the updates (as pointed out in (Yang et al., 2019)). To address this challenge, in step (ii) we further project the obtained policy so that it satisfies the constraint for any transition kernel in the uncertainty set.

One of the key challenges in the analysis lies in that the worst-case transition kernel changes when the policy updates. We address this challenge by leveraging our robust performance difference lemma and a novel integration of the Lipchitz property of robust value function and the change of measure technique.

2. Related Work

Constrained RL. The CMDPs (Altman, 1999) have been an active field of research. The primal-dual method is one of the most commonly used method (Altman, 1999; Auer et al., 2008; Liang et al., 2018; Paternain et al., 2019; Tessler et al., 2018; Yu et al., 2019b; Stooke et al., 2020; Efroni et al., 2020; Zheng & Ratliff, 2020; Zhang et al., 2020), which converts the constrained optimization problem to an unconstrained Lagrangian formulation and alternatively updates the primal and dual variables. Thanks to the strong duality of the non-robust CMDPs (Paternain et al., 2019), the problem can be solved exactly in the dual domain and the primal-dual method is guaranteed to converge to the optimal solution (Ding et al., 2020; 2021; Li et al., 2021; Liu et al., 2021; Ying et al., 2021; Wei et al., 2022). Another widely studied method is the primal method (Chow et al., 2018; Dalal et al., 2018; Liu et al., 2020; Xu et al., 2021; Bura et al., 2022), which takes all the updates in the primal domain without formulating the Lagrangian function. The trust region-based methods have also been proposed to solve the non-robust CMDPs (Achiam et al., 2017; Yang et al., 2019; Kim et al., 2023), which guarantee the reward improvement and constraint satisfaction during the training. Under model mismatch, the worst-case transition kernel is different as the policy updates, and therefore these approaches may not be applied, and the obtained policy may easily violate the constraint when there is a model mismatch.

Robust RL. Robust RL was firstly introduced in (Iyengar, 2005; Nilim & El Ghaoui, 2004) where the goal is to optimize the worst-case performance over the uncertainty set of transition kernels. Algorithms with convergence guarantee have been proposed for both the model-based robust RL with known uncertainty set (Iyengar, 2005; Nilim & El Ghaoui, 2004; Xu & Mannor, 2010; Lim & Autef, 2019; Wang et al., 2023a;b) and the model-free robust RL with unknown uncertainty set (Roy et al., 2017; Zhou et al., 2021; Panaganti & Kalathil, 2021; Yang et al., 2021; Wang & Zou, 2022; Wang et al., 2023c; Wang & Zou, 2021; Wang et al., 2022). Compared with the unconstrained robust RL, the robust constrained RL is more challenging since we also need to guarantee the constraint is satisfied for any transition kernel in the uncertainty set. Directly applying algorithms designed for unconstrained robust RL to robust constrained RL will lead to constraint-violating policies.

Constrained RL under Model Uncertainty. Unlike the non-robust CMDPs, there are rather limited works on constrained RL under model uncertainty. In (Russel et al., 2020), a heuristic approach is proposed. The basic idea is that they first estimate the robust value function, and then update the policy using the non-robust policy gradient (Sutton et al., 1999). Since the worst-case transition kernel is also a function of the policy, the non-robust policy gradient may not update the policy along the direction of the real gradient. Therefore, it cannot guarantee the performance improvement for each update, and thus the convergence of this heuristic approach may not hold. In (Mankowitz et al.,

2020), the robust value function estimate is first performed and a non-robust continuous control algorithm is applied to update the policy. Similar to (Russel et al., 2020), the non-robust policy improvement cannot guarantee the convergence of the algorithm. In (Wang et al., 2022), a robust primal-dual algorithm (RPD) is proposed where the primal and dual variables are updated alternatively, and the robust policy gradient is employed to update the policy. However, the strong duality may not hold when there is model mismatch. Secondly, the constraint may be violated during the training, which is attributed to the nature of the primal-dual approach. In contrast, the update of our algorithm is performed in the primal domain, and we provide performance guarantee on the robust reward improvement and robust constraint violation for each update. Our RCPO ensures constraint satisfaction throughout the training, which is critical for many practical applications.

3. Preliminaries and Problem Formulation

3.1. Constrained MDP

A constrained Markov Decision Process (CMDP) (Altman, 1999) is defined by a tuple $(\mathcal{S}, \mathcal{A}, p, r, c)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p = \{p_s^a \in \Delta(\mathcal{S}), s \in \mathcal{S}, a \in \mathcal{A}\}$[1] is the transition kernel of the CMDP, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, and $c : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the utility function. A stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is defined as the probability distribution of choosing actions in $\mathcal{A}$ at the current state s. After choosing action a at state s, the system transits to the next state s' based on the transition kernel p_s^a. At the same time, the agent receives a reward $r(s, a)$ and a utility $c(s, a)$. For the sake of simplicity in presentation, we consider the case with one constraint, and the results in this paper can be extended to the case with multiple constraints.

Starting from an initial state s, the reward value function of a policy π is defined as

$$V_{r,p}^{\pi}(s) \triangleq \mathbb{E}_p \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi \right],$$

where $\mathbb{E}_p$ denotes the expectation with respect to the transition kernel p and γ is the discount factor. The reward action value function is defined as

$$Q_{r,p}^{\pi}(s, a) \triangleq \mathbb{E}_p \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a, \pi \right].$$

Similarly, the utility value function and the utility action value function are defined as

$$V_{c,p}^{\pi}(s) \triangleq \mathbb{E}_p \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) | s_0 = s, \pi \right]$$

$$Q_{c,p}^{\pi}(s, a) \triangleq \mathbb{E}_p \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) | s_0 = s, a_0 = a, \pi \right].$$

Let $V_{r,p}^{\pi}(\rho) = \mathbb{E}_{s\sim\rho}[V_{r,p}^{\pi}(s)]$ and $V_{c,p}^{\pi}(\rho) = \mathbb{E}_{s\sim\rho}[V_{c,p}^{\pi}(s)]$ be the discounted accumulative reward function and the discounted accumulative utility function, respectively, when the initial state s follows distribution ρ. Let $d_p^{\pi}(s)$ denote the state occupancy measure when the initial state s follows distribution ρ:

$$d_p^{\pi}(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s | s_0 \sim \rho, \pi, p).$$

The goal of the CMDP is to learn a policy π that maximizes the cumulative discounted reward $V_{r,p}^{\pi}(\rho)$ subject to the constraint on the cumulative discounted utility $V_{c,p}^{\pi}(\rho)$, i.e.,

$$\max_{\pi} V_{r,p}^{\pi}(\rho) \text{ s.t. } V_{c,p}^{\pi}(\rho) \geq d,$$

where d is some positive threshold for the constraint.

3.2. Robust MDP

The robust MDP is defined as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$, where $\mathcal{P}$ is the uncertainty set of transition kernels that measures the model uncertainty. In this paper, we consider the uncertainty set with (s, a)-rectangularity (Nilim & El Ghaoui, 2004; Iyengar, 2005). Specifically, the uncertainty set is defined as $\mathcal{P} = \bigotimes_{s,a} \mathcal{P}_s^a$, where $\mathcal{P}_s^a \subseteq \Delta(\mathcal{S})$ are independent over different state-action pairs. The robust value function is defined as

$$V_r^{\pi}(s) \triangleq \min_{p\in\mathcal{P}} \mathbb{E}_p \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi \Big]. \quad (1)$$

Similarly, the robust action value function is defined as

$$Q_r^{\pi}(s, a) \triangleq \min_{p\in\mathcal{P}} \mathbb{E}_p \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a, \pi \Big]. \quad (2)$$

The transition kernel that achieves the min in (1) and (2) is referred to as the worst-case transition kernel. Denote by $V_r^{\pi}(\rho)$ the robust discounted accumulative reward function when the initial state s follows the distribution ρ. For robust RL, the goal is to find an optimal robust policy π^* that optimizes the worst-case performance over the uncertainty set of transition kernels, i.e.

$$\pi^* = \arg\max_{\pi} V_r^{\pi}(\rho). \quad (3)$$

3.3. Problem Formulation

Define the constrained MDP problem under model mismatch as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, c)$, where $\mathcal{P}$ is an uncertainty

[1] $\Delta(\mathcal{S})$ denotes the probability simplex defined on $\mathcal{S}$.

set of transition kernels as defined in Section 3.2 to characterize the potential model mismatch (see (Siddique et al., 2019) for an example of uncertainty set construction). To guarantee that the constraint is always satisfied even under model mismatch, we define the robust utility value function which measures the worst-case accumulated utility over the uncertainty set:

$$V_c^\pi(s) \triangleq \min_{p\in\mathcal{P}} \mathbb{E}_p\Big[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)|s_0=s,\pi\Big]. \quad (4)$$

We are interested in policies that for any transition kernel in the uncertainty set, i.e., under model mismatch, the accumulative utility is still above a prescribed threshold. Furthermore, among those policies, we would like to find one that achieves a good accumulative reward for any transition kernel in the uncertainty set. Formally, we aim to find a policy that maximizes the worst-case cumulative discounted reward subject to the constraint on the worst-case cumulative discounted utility, i.e.,

$$\max_\pi V_r^\pi(\rho), \text{ s.t. } V_c^\pi(\rho)\geq d. \quad (5)$$

The problem in (5) is referred to as robust constrained RL.

4. Robust Constrained Policy Optimization

In this section, we present our algorithm, the robust constrained policy optimization (RCPO), to solve the problem in (5), and theoretically prove that the obtained policy has an improved robust reward value function and also has guarantees for constraint satisfaction at each iteration. Our RCPO algorithm can be applied to large scale problems with a continuous state space. We also generalize the performance difference lemma in (Achiam et al., 2017) to the robust setting, and show the robust value functions of two policies can be bounded using the divergence between them.

In this section, we first present our algorithm and its theoretical performance analyses. In Section 5, we will provide a practical implementation for an efficient computation.

4.1. Algorithm Design

In the following, we will develop our RCPO algorithm. The basic idea is to first find a policy to maximize the robust reward advantage function in a neighborhood of the current policy, which generalizes the trust region policy optimization (Schulman et al., 2015a) to the robust constrained RL problem, and then to project the obtained policy to meet the robust constraint.

To obtain a local approximation of the robust value function, we first present the robust performance difference lemma. Specifically, we need a bound for the performance difference of the robust value functions between two policies. Let p_π^r denote the worst-case transition kernel of π for reward such that $V_{r,p_\pi^r}^\pi(\rho) = \min_{p\in\mathcal{P}} V_{r,p}^\pi(\rho)$. Let $D_{KL}(f_0\|f_1)$ denote the Kullback-Leibler (KL) divergence between two distributions f_0 and f_1. The following robust performance difference lemma generalizes the bound for standard non-robust value functions in (27) (Achiam et al., 2017) to the robust value functions.

Lemma 4.1 (Robust performance difference lemma)**.** *For any two policies π, π', let*

$$\epsilon_{r,p_{\pi'}^r}^{\pi'} = \max_s |\mathbb{E}_{a\sim\pi'}[A_{r,p_{\pi'}^r}^\pi(s,a)]|.$$

We have the following bound:

$$V_r^{\pi'}(\rho) - V_r^\pi(\rho) \geq \quad (6)$$
$$\frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_{\pi'}^r}^\pi \\ a\sim\pi'}}\left[A_{r,p_{\pi'}^r}^\pi(s,a) - \frac{2\gamma\epsilon_{r,p_{\pi'}^r}^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi'\|\pi)(s)}\right].$$

It can be easily verified that the bound in (6) holds with equality when $\pi=\pi'$.

A first idea is to optimize the lower bound in (6) over π' in the neighborhood of π, and to obtain policy π' with an improved performance. However, it's difficult to implement since the lower bound in (6) involves the advantage function and visitation distribution under $p_{\pi'}^r$ which implicitly depends on π'. To address this unique challenge to the robust setting, we propose to approximate $A_{r,p_{\pi'}^r}^\pi(s,a)$ and $d_{p_{\pi'}^r}^\pi$ by $A_{r,p_\pi^r}^\pi(s,a)$ and $d_{p_\pi^r}^\pi$ respectively in the neighborhood of π. The motivation of such an approximation is that π' is in the neighborhood of π, and the robust value function is Lipschitz in the policy (as shown in (Wang et al., 2023a)). We then use

$$\frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_\pi^r}^\pi \\ a\sim\pi'}}\left[A_{r,p_\pi^r}^\pi(s,a) - \frac{2\gamma\epsilon_{r,p_{\pi'}^r}^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi'\|\pi)(s)}\right] \quad (7)$$

as an approximation of the robust performance difference $V_r^{\pi'}(\rho) - V_r^\pi(\rho)$ and further design our RCPO algorithm based on this approximation. In Appendix C, we show that the approximated loss in (7) matches $V_r^{\pi'}(\rho) - V_r^\pi(\rho)$ up to the first order.

As will be shown below, though (7) may not necessarily be a lower bound of $V_r^{\pi'}(\rho) - V_r^\pi(\rho)$ due to the use of the approximation, we are still able to guarantee both the reward improvement and the constraint violation. This actually corresponds to the additional challenge than the non-robust standard CMDP, where there is only one transition kernel for both policies. Here, we are interested in the robust value function, which is essentially the value function under the worst-case transition kernel, and two different policies induce two different worst-case transition kernels.

Let p_k^r denote the worst-case transition kernel of π_k for reward and p_k^c denote the worst-case transition kernel of π_k

for utility. A direct generalization of the CPO algorithm in (Achiam et al., 2017) is to optimize the policy iteratively using the following update:

$$\pi_{k+1} = \arg\max_{\pi} \mathbb{E}_{\substack{s\sim d^{\pi_k}_{p^r_k} \\ a\sim\pi}}\Big[A^{\pi_k}_{r,p^r_k}(s,a)\Big]$$
$$\text{s.t. } V^{\pi_k}_{c,p^c_k}(\rho) + \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi_k}_{p^c_k} \\ a\sim\pi}}\Big[A^{\pi_k}_{c,p^c_k}(s,a)\Big] \geq d,$$
$$\mathbb{E}_{s\sim d^{\pi_k}_{p^r_k}}[D_{KL}(\pi\|\pi_k)(s)] \leq \delta. \tag{8}$$

Here, the first constraint in (8) guarantees the new policy satisfies the robust constraint, and the second constraint in (8) limits the search to be in the neighborhood of π_k. However, this has an issue that there may be no feasible solution to (8) if the current policy π_k violates the constraint.

To address the above challenge, we design a two-step approach which performs policy improvement followed by a projection step (Yang et al., 2019). Below, we introduce our RCPO algorithm in details, and the pseudocode is provided in Algorithm 1. To handle large-scale MDPs, we consider a parameterized policy class $\Pi_{\boldsymbol{\theta}}$ with parameter $\boldsymbol{\theta}$.

Step 1: Robust Policy Improvement. At the robust policy improvement step, we first estimate the worst-case transition kernel p^r_k for the current policy π_k. This can be done by a gradient-based method (Wang et al., 2023a). We iteratively update $p^{r,t}_k$ using the projected gradient descent as follows,

$$p^{r,t+1}_k = \text{Proj}_{\mathcal{P}}\big(p^{r,t}_k - \beta_t \nabla_p V^{\pi_k}_{r,p^{r,t}_k}(\rho)\big), \tag{9}$$

where β_t is the step size and $\text{Proj}_{\mathcal{P}}$ is the projection operator onto set $\mathcal{P}$: $\text{Proj}_{\mathcal{P}}(p^a_s) = \arg\min_{q\in\mathcal{P}^a_s} D(p^a_s, q)$, where D is some distance measure between two distributions.

Consider the tabular case for an example, an accurate p^r_k can be obtained such that

$$V^{\pi_k}_{r,p^r_k}(\rho) = \min_{p\in\mathcal{P}} V^{\pi_k}_{r,p}(\rho), \tag{10}$$

as shown in Theorem 4.4 in (Wang et al., 2023a). For the large/continuous state space, to estimate the worst-case transition kernel, we parameterize the transition kernel and perform gradient descent to learn the worst-case transition kernel estimate. Consider the case with a large discrete state space as an example, the transition kernel can be parameterized as follows:

$$p^{\boldsymbol{\xi}}_{s,a}(s') = \frac{p^0_{s,a}(s')\cdot\exp(\frac{\boldsymbol{\eta}^\top\boldsymbol{\phi}(s')}{\lambda_{s,a}})}{\sum_x p^0_{s,a}(x)\exp(\frac{\boldsymbol{\eta}^\top\boldsymbol{\phi}(x)}{\lambda_{s,a}})}, \tag{11}$$

where p^0 is the nominal transition kernel of the uncertainty set $\mathcal{P}$, $\boldsymbol{\phi}: \mathcal{S}\to\mathbb{R}^m$ is a m-dimensional feature vector, $\boldsymbol{\xi} = (\boldsymbol{\eta}, \boldsymbol{\lambda})$, $\boldsymbol{\lambda} = \{\lambda_{s,a} > 0, \forall(s,a)\in\mathcal{S}\times\mathcal{A}\}$ and $\boldsymbol{\eta}\in\mathbb{R}^m$ are parameters. We then present another example for the case with a continuous state space, where the transition kernel can be parameterized using the Gaussian mixture model:

$$p^{\boldsymbol{\xi}}_{s,a}(s') = \sum_{i=1}^m \phi_i \mathcal{N}(\mu_i, \sigma_i^2), \tag{12}$$

where $\phi_i: \mathcal{S}\to[0,1]$ and $\sum_{i=1}^m \phi_i = 1$, $\mathcal{N}$ denotes the Gaussian distribution and $\boldsymbol{\mu} = (\mu_1,\cdots,\mu_m): \mathcal{S}\times\mathcal{A}\to\mathbb{R}^m, \boldsymbol{\sigma} = (\sigma_1,\cdots,\sigma_m): \mathcal{S}\times\mathcal{A}\to\mathbb{R}^m$ are the parameters. In this case, let $\boldsymbol{\xi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$.

We then evaluate the advantage function $A^{\pi_k}_{r,p^r_k}$ and the visitation distribution $d^{\pi_k}_{p^r_k}$ by performing policy π_k under the transition kernel p^r_k. The intermediate policy $\pi_{k+\frac{1}{2}}$ is updated by solving the following optimization problem:

$$\max_{\pi\in\Pi_{\boldsymbol{\theta}}} \quad \mathbb{E}_{\substack{s\sim d^{\pi_k}_{p^r_k} \\ a\sim\pi}}[A^{\pi_k}_{r,p^r_k}(s,a)],$$
$$\text{s.t. } \mathbb{E}_{s\sim d^{\pi_k}_{p^r_k}}[D_{KL}(\pi||\pi_k)(s)] \leq \delta. \tag{13}$$

Note that in (13), $A^{\pi_k}_{r,p^r_k}$ and $d^{\pi_k}_{p^r_k}$ are estimated using the sample trajectories from the current policy π_k under the transition kernel p^r_k, which can be easily obtained. We optimize the advantage function over a neighborhood of the current policy π_k. Therefore, the advantage function and visitation distribution under policy π_k are good local approximations for all policies in this neighborhood. For the tabular case, in the policy improvement step, we only need to find a policy π that maximizes the expected value of $A^{\pi_k}_{r,p^r_k}(s,a)$ with $a\sim\pi$, which is linear in π, and satisfies the constraint on the expected $D_{KL}(\pi||\pi_k)(s)$ under the distribution $d^{\pi_k}_{p^r_k}$, which is convex in π. Therefore, (13) is a convex optimization problem and can be solved efficiently.

Step 2: Projection. By solving (13), we obtain a policy $\pi_{k+\frac{1}{2}}$ that maximizes the advantage function in the neighborhood of current policy π_k. However, $\pi_{k+\frac{1}{2}}$ does not necessarily satisfy the constraint. In the projection step, we project the policy $\pi_{k+\frac{1}{2}}$ to the constraint set to obtain a constraint-satisfying policy π_{k+1}. We first estimate the worst-case transition kernel p^c_k for the utility value function under the current policy π_k using the projected gradient descent method:

$$p^{c,t+1}_k = \text{Proj}_{\mathcal{P}}\big(p^{c,t}_k - \beta_t \nabla_p V^{\pi_k}_{c,p^{c,t}_k}(\rho)\big). \tag{14}$$

For the tabular case, p^c_k can be obtained such that

$$V^{\pi_k}_{c,p^c_k}(\rho) = \min_{p\in\mathcal{P}} V^{\pi_k}_{c,p}(\rho). \tag{15}$$

For the large/continuous state space, we parameterize the transition kernel as introduced in Step 1.

We then estimate $A^{\pi_k}_{c,p^c_k}, d^{\pi_k}_{p^c_k}$ using sample trajectories from p^c_k. The projection step is achieved by solving

$$\min_{\pi\in\Pi_{\boldsymbol{\theta}}} \quad \mathbb{E}_{s\sim d^{\pi_k}_{p^r_k}}[D_{KL}(\pi||\pi_{k+\frac{1}{2}})(s)]$$

Algorithm 1 Robust Constrained Policy Optimization

Input: step size $\delta, \{\beta_t\}_{t\geq 0}$, iteration time K, T, initial policy π_0
for $k = 0, 1, \cdots, K-1$ **do**
  Initialize $p_k^{r,0}, p_k^{c,0}$
  for $t = 0, 1, \cdots, T-1$ **do**
    $p_k^{r,t+1} \leftarrow \text{Proj}_{\mathcal{P}}\big(p_k^{r,t} - \beta_t \nabla_p V_{r,p_k^{r,t}}^{\pi_k}(\rho)\big)$
    $p_k^{c,t+1} \leftarrow \text{Proj}_{\mathcal{P}}\big(p_k^{c,t} - \beta_t \nabla_p V_{c,p_k^{c,t}}^{\pi_k}(\rho)\big)$
  end for
  $p_k^r \leftarrow p_k^{r,T}, p_k^c \leftarrow p_k^{c,T}$
  Compute $A_{r,p_k^r}^{\pi_k}$, $A_{c,p_k^c}^{\pi_k}$, $d_{p_k^r}^{\pi_k}$, $d_{p_k^c}^{\pi_k}$
  Update $\pi_{k+\frac{1}{2}}$ according to (13)
  Update π_{k+1} according to (16)
end for
Output: π_K

$$\text{s.t. } V_{c,p_k^c}^{\pi_k}(\rho) + \mathbb{E}_{\substack{s\sim d_{p_k^c}^{\pi_k} \\ a\sim\pi}}[A_{c,p_k^c}^{\pi_k}(s,a)] \geq d. \tag{16}$$

In (16), the constraint $V_{c,p_k^c}^{\pi_k}(\rho) + \mathbb{E}_{s\sim d_{p_k^c}^{\pi_k}, a\sim\pi}[A_{c,p_k^c}^{\pi_k}(s,a)]$ is a local approximation for $V_c^{\pi}(\rho)$. For the tabular case, problem in (16) is a convex optimization since the advantage function and the visitation distribution are obtained from the current policy π_k, and therefore, can be solved efficiently.

Unlike solving (8), which might be infeasible when the current policy π_k doesn't satisfy the constraint, our update rule consists of two convex optimization problems (13) and (16), the feasible set of which are much larger than (8).

4.2. Theoretical Results

We first make the following assumption on the worst-case transition kernel.

Assumption 4.2. We are able to find transition kernels $p_k^{r,\boldsymbol{\xi}}, p_k^{c,\boldsymbol{\xi}}$ such that

$$\begin{aligned} \big|V_{r,p_k^{r,\boldsymbol{\xi}}}^{\pi_k}(\rho) - V_{r,p_k^r}^{\pi_k}(\rho)\big| &\leq \epsilon, \\ \big|V_{c,p_k^{c,\boldsymbol{\xi}}}^{\pi_k}(\rho) - V_{c,p_k^c}^{\pi_k}(\rho)\big| &\leq \epsilon. \end{aligned} \tag{17}$$

This assumption can be satisfied under the tabular case using a direct parameterization of the transition kernel or under the case with a continuous state space if a large enough neural network is used to parameterize the transition kernel.

In the following theorem, we provide a lower bound on the worst-case reward improvement and an upper bound on the worst-case constraint violation for each iteration of our RCPO algorithm in Algorithm 1.

Theorem 4.3. *Let* $\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}} = \max_s |\mathbb{E}_{a\sim\pi_{k+1}} A_{r,p_{k+1}^r}^{\pi_k}(s,a)|$ *and* $\epsilon_{c,p_{k+1}^c}^{\pi_{k+1}} = \max_s |\mathbb{E}_{a\sim\pi_{k+1}} A_{c,p_{k+1}^c}^{\pi_k}(s,a)|$. *Under Assumption 4.2, when the current policy π_k satisfies the constraint in (5), we have:*

Worst-case reward improvement:

$$V_r^{\pi_{k+1}}(\rho) - V_r^{\pi_k}(\rho) \geq -\frac{1}{1-\gamma} M\Big(2L_\pi + \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\Big)\sqrt{\frac{\delta}{2}};$$

Constraint violation:

$$V_c^{\pi_{k+1}}(\rho) \geq d - \epsilon - \frac{1}{1-\gamma} M\Big(3L_\pi + \frac{2\gamma\epsilon_{c,p_{k+1}^c}^{\pi_{k+1}}}{1-\gamma}\Big)\sqrt{\frac{\delta}{2}},$$

where $M = \sup_{p,p'\in\mathcal{P}} \|d_p^{\pi_k}/d_{p'}^{\pi_k}\|_\infty$ *is finite whenever* $supp(\rho) = \mathcal{S}$ *and* $L_\pi = \frac{\sqrt{|\mathcal{A}|}}{(1-\gamma)^2}$.

Theorem 4.3 shows that we could adjust δ towards improved robust reward value function and smaller constraint violation. Moreover, for the large/continuous state space, our RCPO only incurs an additional degradation ϵ on constraint violation due to the worst-case transition kernel mismatch. For the tabular case, ϵ can be arbitrarily close to zero. On the other hand, throughout the training, the policy π_k may violate the constraint due to the random initialization or estimation errors. Therefore, in the following, we also characterize the performance of our algorithm when the current policy π_k violates the constraint.

Let $b = d - V_c^{\pi_k}(\rho)$. The following theorem provides a lower bound on the worst-case reward improvement and an upper bound on the worst-case constraint violation.

Theorem 4.4. *Under Assumption 4.2, when the current policy π_k violates the constraint in (16), we have:*
Worst-case reward improvement:

$$\begin{aligned} &V_r^{\pi_{k+1}}(\rho) - V_r^{\pi_k}(\rho) \\ &\geq -\frac{1}{1-\gamma} M\Big(2L_\pi + \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\Big)\sqrt{\frac{\delta + b^2\alpha_{KL}}{2}}; \end{aligned} \tag{18}$$

Constraint violation:

$$\begin{aligned} V_c^{\pi_{k+1}}(\rho) &\geq d - \epsilon - \frac{1}{1-\gamma} M\Big(3L_\pi + \frac{2\gamma\epsilon_{c,p_{k+1}^c}^{\pi_{k+1}}}{1-\gamma}\Big) \\ &\times \sqrt{\frac{\delta + b^2\alpha_{KL} + bM'\sqrt{\frac{\alpha_{KL}}{2}}}{2}}, \end{aligned} \tag{19}$$

where $\alpha_{KL} = \frac{1}{2\boldsymbol{h}^\top \boldsymbol{H}^{-1}\boldsymbol{h}}$, $\boldsymbol{h}$ *and* $\boldsymbol{H}$ *are defined in (20) and (21),* $M' < \infty$ *is some constant.*

Theorem 4.4 characterizes the performance of our algorithm when the current policy π_k is infeasible. A small b, i.e., the current policy π_k only violates the constraint slightly, leads to a better robust reward improvement and a smaller constraint violation. If the current policy π_k satisfies the constraint, i.e., $b = 0$, Theorem 4.4 reduces to Theorem 4.3. The misspecified worst-case transition kernel only incurs an additional performance degradation ϵ on the constraint violation for large/continuous state space. For the tabular case, ϵ can be arbitrarily close to zero.

5. Practical Implementation

In this section, we provide a practical implementation of Algorithm 1 to tackle the computational challenge.

To update the policy efficiently, for a small step size δ, we approximate the objective functions and constraints in the optimization problems (13) and (16) using their Taylor expansions. Let

$$\begin{aligned} \boldsymbol{g} &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{s\sim d^{\pi_k}_{p^r_{k,\boldsymbol{\xi}}}, a\sim\pi}[A^{\pi_k}_{r,p^r_{k,\boldsymbol{\xi}}}(s,a)], \\ \boldsymbol{h} &= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{s\sim d^{\pi_k}_{p^c_{k,\boldsymbol{\xi}}}, a\sim\pi}[A^{\pi_k}_{c,p^c_{k,\boldsymbol{\xi}}}(s,a)] \end{aligned} \tag{20}$$

be the gradient of the reward advantage function and the gradient of the utility advantage function, respectively. Let

$$\boldsymbol{H} = \nabla^2_{\boldsymbol{\theta}} \mathbb{E}_{s\sim d^{\pi_k}_{p^r_{k,\boldsymbol{\xi}}}} \big[D_{KL}(\pi||\pi_k)(s)\big] \tag{21}$$

be the Hessian matrix of the KL divergence. We then develop the following practical implementation for our RCPO.

Step 1: Robust Policy Improvement. We first estimate the worst-case transition kernel p^r_k for the current policy π_k. We iteratively update the parameterized transition kernel $p^{r,t}_{k,\boldsymbol{\xi}}$ using the following projected gradient descent method:

$$p^{r,t+1}_{k,\boldsymbol{\xi}} = \text{Proj}_{\mathcal{P}}\big(p^{r,t}_{k,\boldsymbol{\xi}} - \beta_t \nabla_p V^{\pi_k}_{r,p^{r,t}_{k,\boldsymbol{\xi}}}(\rho)\big). \tag{22}$$

We then use the first-order approximation for the objective function and the second-order approximation for the KL divergence constraint at the current policy π_k in (13). Let $\boldsymbol{\theta}_k$ denote the parameter of policy π_k. The parameter of the intermediate policy $\pi_{k+\frac{1}{2}}$ is updated by solving the following practical formulation for (13):

$$\max_{\boldsymbol{\theta}} \boldsymbol{g}^\top(\boldsymbol{\theta}-\boldsymbol{\theta}_k),\ \text{s.t.}\ \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_k)^\top \boldsymbol{H}(\boldsymbol{\theta}-\boldsymbol{\theta}_k) \le \delta. \tag{23}$$

The objective function of (23) is linear in $\boldsymbol{\theta}$ and the constraint is quadratic in $\boldsymbol{\theta}$. Therefore, problem (23) can be easily solved.

Step 2: Projection. For the projection step, we first estimate the worst-case transition kernel p^c_k for utility. Similarly, we iteratively update the parameterized transition kernel $p^{c,t}_{k,\boldsymbol{\xi}}$ using the following projected gradient descent method:

$$p^{c,t+1}_{k,\boldsymbol{\xi}} = \text{Proj}_{\mathcal{P}}\big(p^{c,t}_{k,\boldsymbol{\xi}} - \beta_t \nabla_p V^{\pi_k}_{c,p^{c,t}_{k,\boldsymbol{\xi}}}(\rho)\big). \tag{24}$$

We then approximate the objective function in (16) by its second order expansion and approximate the constraint in (16) by its first order expansion. The parameter of the policy π_{k+1} is then updated by solving the following problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\boldsymbol{k+\frac{1}{2}}})^\top \boldsymbol{H}(\boldsymbol{\theta}-\boldsymbol{\theta}_{\boldsymbol{k+\frac{1}{2}}})$$

$$\text{s.t.}\ \boldsymbol{h}^\top(\boldsymbol{\theta}-\boldsymbol{\theta}_k)+b \le 0. \tag{25}$$

The problems in (23) and (25) can be solved by convex programming (Yang et al., 2019). We have the following update rule for each policy update.

$$\begin{aligned} \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k &+ \sqrt{\frac{2\delta}{\boldsymbol{g}^\top \boldsymbol{H}^{-1}\boldsymbol{g}}}\boldsymbol{H}^{-1}\boldsymbol{g} \\ &- \max\left(\frac{\sqrt{\frac{2\delta}{\boldsymbol{g}^\top \boldsymbol{H}^{-1}\boldsymbol{g}}}\boldsymbol{h}\boldsymbol{H}^{-1}\boldsymbol{g}+b}{\boldsymbol{h}^\top \boldsymbol{H}^{-1}\boldsymbol{h}}, 0\right)\boldsymbol{H}^{-1}\boldsymbol{h}. \end{aligned} \tag{26}$$

In this way, our RCPO algorithm can be implemented efficiently for large-scale problems.

6. Experiments

To validate the proposed algorithm, we compare it with several baseline algorithms (PCPO (Yang et al., 2019), RVI (Iyengar, 2005), CPO (Achiam et al., 2017), R3C (Mankowitz et al., 2020) and CUP (Yang et al., 2022)) in the setting of tabular and deep cases, while using different environments such as the gambler problem (Sutton & Barto, 2018; Zhou et al., 2021; Shi & Chi, 2022), the N-chain problem (Wang et al., 2022), the Frozen-Lake problem (Brockman et al., 2016) and the Point Gather in Mujoco (Achiam et al., 2017; Yang et al., 2019).

6.1. Tabular Case

In the setting of tabular cases, we evaluate the performance of our algorithm in the environment of gambler problem, N-chain problem and Frozen-Lake problem, where both state and action spaces are finite. We compare our algorithm with four baselines: PCPO (Yang et al., 2019), R3C (Mankowitz et al., 2020), CUP (Yang et al., 2022) and the model-based robust value iteration (RVI) (Iyengar, 2005). The PCPO and CUP learn an optimal policy subjecting to the constraints under the nominal transition kernels without considering the model mismatch. The R3C performs robust value function estimate and non-robust policy improvement for robust constrained RL. The model-based RVI directly optimizes the unconstrained robust reward objective, which serves as an upper bound of the reward value function for the robust constrained problem. We consider the KL divergence uncertainty set. For each problem, we run the algorithms for 5 independent times and plot the mean of the reward and utility along with their standard deviation as a function of the number of iterations. The detailed environments descriptions can be found in Appendix F.

For the gambler problem, the threshold for the constraint is 2.5. It can be seen from Fig. 1(b) that our RCPO always satisfies the constraint during the training while PCPO, RVI,

R3C and CUP violate the constraints. Moreover, the reward of our RCPO in Fig. 1(a) is close to the reward of RVI, which is the best achievable reward for the unconstrained robust RL. Therefore, our algorithm learns a policy that satisfies the worst-case constraint on the utility and achieves optimal reward objective.

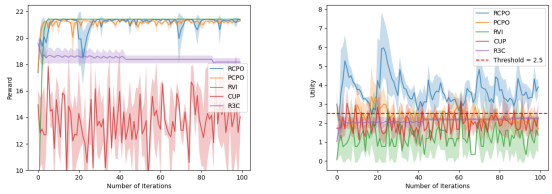

Figure 1: Gambler Problem

For the N-chain problem, the threshold is set to be 6. From Fig. 2(b), it can be seen that all five algorithms satisfy the constraint, indicating that the constraint is easy to satisfy for this problem. However, in Fig. 2(a), the two non-robust algorithms PCPO and CUP doesn't converge to the reward of the unconstrained RVI.

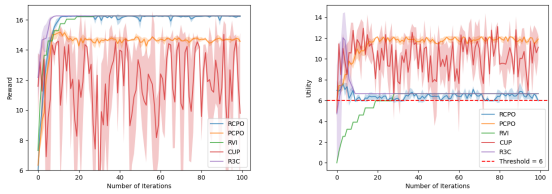

Figure 2: N-chain Problem

For the Frozen-Lake problem, the threshold is set to be 0.7. From Fig. 3(b), it can be seen that only RCPO satisfies the constraint. Moreover, in Fig. 3(a), it can be seen that our RCPO obtain more reward than the two non-robust algorithms PCPO and CUP, which demonstrates the effectiveness and robustness of our algorithm.

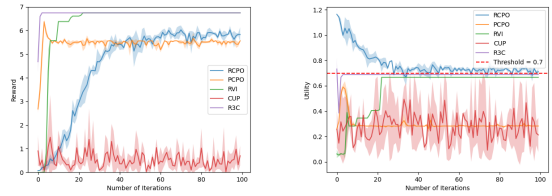

Figure 3: Frozen-Lake Problem

6.2. Deep Case

In the setting of the deep case, we incorporate our algorithm into deep neural networks for tackling high-dimensional spaces (e.g. continuous state spaces). We compare the proposed RCPO with CPO (Achiam et al., 2017), PCPO (Yang et al., 2019) and CUP (Yang et al., 2022). We use the same neural network with two hidden layers of size (64, 32) in all four algorithms. We adopt a Mujoco-based environment, Point Gather task with safety constraints (Achiam et al., 2017), which is a well-recognized constrained MDP environment. We use the following hyper-parameters for training RCPO: discounted factor $= 0.995$, learning step size $= 0.001$, batch size $= 50,000$, and utility-constrained threshold $= 0.1$. To provide fair comparisons, we use the same hyper-parameters for training baseline algorithms. The experiments are implemented in rllab (Duan et al., 2016), a tool for developing and evaluating RL algorithms. To introduce model uncertainties into the environment, we use Gaussian noise to perturb the environment and evaluate the performance of three algorithms under the perturbed environment. We don't report the utility of CUP as it violates the constraint badly. It can be seen from Fig. 4(b) that the rewards of RCPO are much higher than these three non-robust algorithms under model uncertainty, which demonstrates the robustness of our algorithm to model uncertainty when incorporating deep neural networks. Meanwhile, the well-trained RCPO policy satisfies the utility constraint. In summary, RCPO is able to provide efficient, robust, and constraint-satisfied policies in environments with continuous spaces by incorporating deep neural networks.

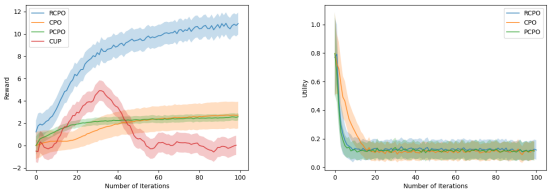

Figure 4: Point Gather

7. Conclusion

In this paper, we study the problem of constrained reinforcement learning under model mismatch. The goal is to maximize the worst-case reward over the uncertainty set subject to a constraint that the utility function for all transition kernels in the uncertainty set shall be above a prescribed threshold. We propose a robust constrained policy optimization (RCPO) algorithm, which consists of several novel technical developments than the CPO algorithm (Achiam et al., 2017) for the non-robust standard CMDP problem. One result that may of independent interest is a robust performance difference lemma that bound the different between the robust value functions of two policies. Our algorithm is applicable to large scale MDPs, and has theoretical guarantees on worst-case reward improvement and constraint violation at each iteration during the training. We further provide an efficient approximation for the purpose of practical implementation of our algorithm. Numerical experiments on demonstrate the effectiveness and robustness of our algorithm under model mismatch.

Impact Statement

This paper presents work whose goal is to advance the field of Reinforcement Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

The work of Zhongchang Sun and Shaofeng Zou is supported by the National Science Foundation under Grants CCF-2007783, CCF-2106560 and ECCS-2337375 (CAREER). The work of Sihong He and Fei Miao is supported by the National Science Foundation under Grant CNS-2047354 (CAREER).

This material is based upon work supported under the AI Research Institutes program by National Science Foundation and the Institute of Education Sciences, U.S. Department of Education through Award # 2229873 - National AI Institute for Exceptional Education. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, the Institute of Education Sciences, or the U.S. Department of Education.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *Proc. International Conference on Machine Learning (ICML)*, pp. 22–31. PMLR, 2017.

Altman, E. *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 21, 2008.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Bura, A., HasanzadeZonuzy, A., Kalathil, D., Shakkottai, S., and Chamberland, J.-F. Dope: Doubly optimistic and pessimistic exploration for safe reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 1047–1059, 2022.

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. A lyapunov-based approach to safe reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Csiszár, I. and Körner, J. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C., and Tassa, Y. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Ding, D., Zhang, K., Basar, T., and Jovanovic, M. Natural policy gradient primal-dual method for constrained Markov decision processes. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 8378–8390, 2020.

Ding, D., Wei, X., Yang, Z., Wang, Z., and Jovanovic, M. Provably efficient safe exploration via primal-dual policy optimization. In *Proc. International Conference on Artifical Intelligence and Statistics (AISTATS)*, pp. 3304–3312. PMLR, 2021.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.

Efroni, Y., Mannor, S., and Pirotta, M. Exploration-exploitation in constrained MDPs. *arXiv preprint arXiv:2003.02189*, 2020.

Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7): 2737–2752, 2018.

Gasparik, A., Gamble, C., and Gao, J. Safety-first AI for autonomous data centre cooling and industrial control. *DeepMind blog*, 2018.

He, S., Han, S., and Miao, F. Robust electric vehicle balancing of autonomous mobility-on-demand system: A multi-agent reinforcement learning approach. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5471–5478. IEEE, 2023a.

He, S., Wang, Y., Han, S., Zou, S., and Miao, F. A robust and constrained multi-agent reinforcement learning electric vehicle rebalancing method in amod systems. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5637–5644. IEEE, 2023b.

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

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

Kim, D., Lee, K., and Oh, S. Trust region-based safe distributional reinforcement learning for multiple constraints. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2002.00444*, 2020.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Li, T., Guan, Z., Zou, S., Xu, T., Liang, Y., and Lan, G. Faster algorithm and sharper analysis for constrained Markov decision process. *arXiv preprint arXiv:2110.10351*, 2021.

Li, Y., Lan, G., and Zhao, T. First-order policy optimization for robust markov decision process. *arXiv preprint arXiv:2209.10579*, 2022.

Liang, Q., Que, F., and Modiano, E. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1802.06480*, 2018.

Lim, S. H. and Autef, A. Kernel-based reinforcement learning in robust Markov decision processes. In *Proc. International Conference on Machine Learning (ICML)*, pp. 3973–3981. PMLR, 2019.

Liu, T., Zhou, R., Kalathil, D., Kumar, P., and Tian, C. Fast global convergence of policy optimization for constrained MDPs. *arXiv preprint arXiv:2111.00552*, 2021.

Liu, Y., Ding, J., and Liu, X. Ipo: Interior-point policy optimization under constraints. In *Proc. Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 4940–4947, 2020.

Mankowitz, D. J., Calian, D. A., Jeong, R., Paduraru, C., Heess, N., Dathathri, S., Riedmiller, M., and Mann, T. Robust constrained reinforcement learning for continuous control with model misspecification. *arXiv preprint arXiv:2010.10644*, 2020.

Nilim, A. and El Ghaoui, L. Robustness in Markov decision problems with uncertain transition matrices. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 839–846, 2004.

Ono, M., Pavone, M., Kuwata, Y., and Balaram, J. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Autonomous Robots*, 39(4):555–571, 2015.

Panaganti, K. and Kalathil, D. Sample complexity of robust reinforcement learning with a generative model. *arXiv preprint arXiv:2112.01506*, 2021.

Paternain, S., Chamon, L., Calvo-Fullana, M., and Ribeiro, A. Constrained reinforcement learning has zero duality gap. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Roy, A., Xu, H., and Pokutta, S. Reinforcement learning under model mismatch. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 3046–3055, 2017.

Russel, R. H., Benosman, M., and Van Baar, J. Robust constrained-MDPs: Soft-constrained robust policy optimization under model uncertainty. *arXiv preprint arXiv:2010.04870*, 2020.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *Proc. International Conference on Machine Learning (ICML)*, pp. 1889–1897. PMLR, 2015a.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

Shi, L. and Chi, Y. Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity. *arXiv preprint arXiv:2208.05767*, 2022.

Siddique, T., Hau, J. L., Atallah, S., and Petrik, M. Robust pest management using reinforcement learning. *The Multi-disciplinary Conference on Reinforcement Learning and Decision Making*, 2019.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by pid lagrangian methods. In *Proc. International Conference on Machine Learning (ICML)*, pp. 9133–9143. PMLR, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., Mansour, Y., et al. Policy gradient methods for reinforcement learning with function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 99, pp. 1057–1063. Citeseer, 1999.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Wang, Q., Ho, C. P., and Petrik, M. Policy gradient in robust mdps with global convergence guarantee, 2023a.

Wang, Y. and Zou, S. Online robust reinforcement learning with model uncertainty. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 7193–7206, 2021.

Wang, Y. and Zou, S. Policy gradient method for robust reinforcement learning. In *Proc. International Conference on Machine Learning (ICML)*, volume 162, pp. 23484–23526. PMLR, 2022.

Wang, Y., Miao, F., and Zou, S. Robust constrained reinforcement learning. *arXiv preprint arXiv:2209.06866*, 2022.

Wang, Y., Velasquez, A., Atia, G., Prater-Bennette, A., and Zou, S. Robust average-reward markov decision processes. In *Proc. Conference on Artificial Intelligence (AAAI)*, 2023b.

Wang, Y., Velasquez, A., Atia, G. K., Prater-Bennette, A., and Zou, S. Model-free robust average-reward reinforcement learning. In *International Conference on Machine Learning*, pp. 36431–36469. PMLR, 2023c.

Wei, H., Liu, X., and Ying, L. Triple-q: A model-free algorithm for constrained reinforcement learning with sublinear regret and zero constraint violation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3274–3307. PMLR, 2022.

Xu, H. and Mannor, S. Distributionally robust Markov decision processes. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 2505–2513, 2010.

Xu, T., Liang, Y., and Lan, G. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In *Proc. International Conference on Machine Learning (ICML)*, pp. 11480–11491. PMLR, 2021.

Yang, L., Ji, J., Dai, J., Zhang, L., Zhou, B., Li, P., Yang, Y., and Pan, G. Constrained update projection approach to safe policy optimization. *Advances in Neural Information Processing Systems*, 35:9111–9124, 2022.

Yang, T.-Y., Rosca, J., Narasimhan, K., and Ramadge, P. J. Projection-based constrained policy optimization. In *International Conference on Learning Representations*, 2019.

Yang, W., Zhang, L., and Zhang, Z. Towards theoretical understandings of robust Markov decision processes: Sample complexity and asymptotics. *arXiv preprint arXiv:2105.03863*, 2021.

Ying, D., Ding, Y., and Lavaei, J. A dual approach to constrained Markov decision processes with entropy regularization. *arXiv preprint arXiv:2110.08923*, 2021.

Yu, C., Liu, J., and Nemati, S. Reinforcement learning in healthcare: A survey. *arXiv preprint arXiv:1908.08796*, 2019a.

Yu, M., Yang, Z., Kolar, M., and Wang, Z. Convergent policy optimization for safe reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019b.

Zhang, Y., Vuong, Q., and Ross, K. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33:15338–15349, 2020.

Zheng, L. and Ratliff, L. Constrained upper confidence reinforcement learning. In *Learning for Dynamics and Control*, pp. 620–629. PMLR, 2020.

Zhou, Z., Bai, Q., Zhou, Z., Qiu, L., Blanchet, J., and Glynn, P. Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In *Proc. International Conference on Artifical Intelligence and Statistics (AISTATS)*, pp. 3331–3339. PMLR, 2021.

A. Review of Constrained Policy Optimization

In this section, we provide an overview of the CPO method developed in (Achiam et al., 2017). Recall that in the standard (non-robust) RL setting, the value function difference between two policies π, π' can be written as (Kakade & Langford, 2002):

$$V_{r,p}^{\pi\prime}(\rho) - V_{r,p}^{\pi}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_p^{\pi'} \\ a\sim\pi'}}[A_{r,p}^{\pi}(s,a)], \tag{27}$$

where $A_{r,p}^{\pi}(s,a) = Q_{r,p}^{\pi}(s,a) - V_{r,p}^{\pi}(s)$ is the reward advantage function. In (Achiam et al., 2017), the above result was further extended to the following one:

$$\begin{aligned} &V_{r,p}^{\pi'}(\rho) - V_{r,p}^{\pi}(\rho) \\ &\geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_p^{\pi} \\ a\sim\pi'}}\left[A_{r,p}^{\pi}(s,a) - \frac{2\gamma\epsilon_r^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi'\|\pi)(s)}\right], \end{aligned} \tag{28}$$

where $\epsilon_r^{\pi'} = \max_s |\mathbb{E}_{a\sim\pi'}[A_{r,p}^{\pi}(s,a)]|$.

Equation (28) connects the performance difference between two policies to an average divergence between them. Compared with (27), the expectation is taken with respect to d_p^{π} in (28) instead of $d_p^{\pi'}$. When π' is close to π, $D_{KL}(\pi'\|\pi)(s)$ is small and $d_p^{\pi'}$ is close to d_p^{π}. Therefore, the right hand side of (28) is a good local approximation for the performance difference $V_{r,p}^{\pi'} - V_{r,p}^{\pi}$. The trust region method for unconstrained RL was proposed (Schulman et al., 2015a;b) based on this approximation and provides monotonic improvement for the reward value function.

For the utility value function, we have the following equivalent expression:

$$\begin{aligned} &V_{c,p}^{\pi'}(\rho) - V_{c,p}^{\pi}(\rho) \\ &\geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_p^{\pi} \\ a\sim\pi'}}\left[A_{c,p}^{\pi}(s,a) - \frac{2\gamma\epsilon_c^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi'\|\pi)(s)}\right], \end{aligned} \tag{29}$$

where $\epsilon_c^{\pi'} = \max_s |\mathbb{E}_{a\sim\pi'}[A_{c,p}^{\pi}(s,a)]|$ and $A_{c,p}^{\pi}(s,a) = Q_{c,p}^{\pi}(s,a) - V_{c,p}^{\pi}(s)$ is the utility advantage function. The right hand side of (29) can be used as an approximation for $V_{c,p}^{\pi'}(\rho) - V_{c,p}^{\pi}(\rho)$. By applying the trust region methods to CMDPs, the constrained policy optimization (CPO) was proposed in (Achiam et al., 2017), where the policy is updated by solving the following optimization problem.

$$\begin{aligned} \pi_{k+1} = \arg\max_{\pi}\ & \mathbb{E}_{\substack{s\sim d_p^{\pi_k} \\ a\sim\pi}}\left[A_{r,p}^{\pi_k}(s,a)\right] \\ \text{s.t. } & V_{c,p}^{\pi_k}(\rho) + \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_p^{\pi_k} \\ a\sim\pi}}\left[A_{c,p}^{\pi_k}(s,a)\right] \geq d, \\ & \mathbb{E}_{s\sim d_p^{\pi_k}}[D_{KL}(\pi\|\pi_k)(s)] \leq \delta. \end{aligned} \tag{30}$$

When the current policy π_k satisfies the constraint, this update rule leads to a policy that has performance improvement and approximate satisfaction of constraints (Achiam et al., 2017). Note that the expectation in the optimization problem (30) is taken with respect to $d_p^{\pi_k}$. The optimization problem (30) depends on π only through the distribution of the current action a, which can thus be optimized efficiently.

B. Proof of Lemma 4.1

For any policies π, π', we have that

$$\begin{aligned} V_r^{\pi'}(\rho) - V_r^{\pi}(\rho) &= V_{r,p_{\pi'}^r}^{\pi'}(\rho) - V_{r,p_{\pi}^r}^{\pi}(\rho) \\ &\geq V_{r,p_{\pi'}^r}^{\pi'}(\rho) - V_{r,p_{\pi'}^r}^{\pi}(\rho) \\ &\geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_{\pi'}^r}^{\pi} \\ a\sim\pi'}}\left[A_{r,p_{\pi'}^r}^{\pi}(s,a) - \frac{2\gamma\epsilon_{r,p_{\pi'}^r}^{\pi'}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi'\|\pi)(s)}\right], \end{aligned} \tag{31}$$

where the first inequality is because $V^{\pi}_{r,p^r_\pi}(\rho) = \min_{p\in\mathcal{P}} V^{\pi}_{r,p}(\rho)$ and the second inequality follows from Theorem 1 in (Achiam et al., 2017).

C. First-Order Approximation of (7)

We show that the approximated loss in (7) matches the original one up to first order. For the first-order approximation, we have that

$$V_r^{\pi'}(\rho) - V_r^{\pi}(\rho) = \langle \pi' - \pi, \nabla_\pi V_r^{\pi}(\rho)\rangle + \mathcal{O}(\|\pi' - \pi\|_1^2). \tag{32}$$

The policy gradient of the robust MDPs with the (s,a)-entry has the following form (Li et al., 2022):

$$\nabla_\pi V_r^{\pi}(\rho)(s,a) = \frac{1}{1-\gamma} d^{\pi}_{p^r_\pi}(s) Q^{\pi}_{r,p^r_\pi}(s,a). \tag{33}$$

We further have that

$$\begin{aligned}
V_r^{\pi'}(\rho) - V_r^{\pi}(\rho) &= \langle \pi' - \pi, \nabla_\pi V_r^{\pi}(\rho)\rangle + \mathcal{O}(\|\pi' - \pi\|_1^2) \\
&= \frac{1}{1-\gamma}\sum_{s,a}\Big(\pi'(a|s) - \pi(a|s)\Big) d^{\pi}_{p^r_\pi}(s) Q^{\pi}_{r,p^r_\pi}(s,a) + \mathcal{O}(\|\pi' - \pi\|_1^2) \\
&\overset{(a)}{=} \frac{1}{1-\gamma}\sum_{s,a}\big(\pi'(a|s) - \pi(a|s)\big) d^{\pi}_{p^r_\pi}(s)\big(Q^{\pi}_{r,p^r_\pi}(s,a) - V^{\pi}_{r,p^r_\pi}(s)\big) + \mathcal{O}(\|\pi' - \pi\|_1^2) \\
&\overset{(b)}{=} \frac{1}{1-\gamma}\sum_{s,a}\pi'(a|s) d^{\pi}_{p^r_\pi}(s)\big(Q^{\pi}_{r,p^r_\pi}(s,a) - V^{\pi}_{r,p^r_\pi}(s)\big) + \mathcal{O}(\|\pi' - \pi\|_1^2) \\
&= \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi}_{p^r_\pi} \\ a\sim\pi'}}\Big[A^{\pi}_{r,p^r_\pi}(s,a)\Big] + \mathcal{O}(\|\pi' - \pi\|_1^2),
\end{aligned} \tag{34}$$

which matches with the first term in (7), where equality (a) is due to the fact that $\sum_{s,a}\big(\pi'(a|s) - \pi(a|s)\big) d^{\pi}_{p^r_\pi}(s) V^{\pi}_{r,p^r_\pi}(s) = \sum_s d^{\pi}_{p^r_\pi}(s) V^{\pi}_{r,p^r_\pi}(s)\sum_a\big(\pi'(a|s) - \pi(a|s)\big) = 0$ and equality (b) is due to the fact that $\sum_{s,a}\pi(a|s) d^{\pi}_{p^r_\pi}(s)(Q^{\pi}_{r,p^r_\pi}(s,a) - V^{\pi}_{r,p^r_\pi}(s)) = \sum_s d^{\pi}_{p^r_\pi}(s)(V^{\pi}_{r,p^r_\pi}(s) - V^{\pi}_{r,p^r_\pi}(s)) = 0$. Therefore, the approximated loss in (7) matches the original one up to first order.

D. Proof of Theorem 4.3

We first prove the follow Lemma.

Lemma D.1. *If the current policy π_k satisfies the constraint and the constraint set is closed and convex under the policy parameterization, then under the KL divergence projection, we have that*

$$\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_{k+1}||\pi_k)(s)] \leq \delta. \tag{35}$$

Proof. Note that the constraint in (16) is linear in π. Therefore, the constraint set is closed and convex. Since π_k lies in the constraint set and π_{k+1} is the projection of $\pi_{k+\frac{1}{2}}$ onto the constraint set, from the Bregmann divergence projection inequality, we have that

$$\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_k||\pi_{k+\frac{1}{2}})(s)] \geq \mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_k||\pi_{k+1})(s)] + \mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_{k+1}||\pi_{k+\frac{1}{2}})(s)]. \tag{36}$$

Since the KL divergence is non-negative, we have that

$$\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_k||\pi_{k+\frac{1}{2}})(s)] \geq \mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_k||\pi_{k+1})(s)]. \tag{37}$$

When δ is small, the KL divergence is asymptotically symmetric. Therefore, we have that

$$\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_{k+1}\|\pi_k)(s)] \leq \mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\xi}_k}}[D_{KL}(\pi_{k+\frac{1}{2}}\|\pi_k)(s)] \leq \delta. \tag{38}$$

□

With Lemma D.1, we are ready to prove Theorem 4.3.

Proof. From Lemma 4.1, we have that for the reward improvement,

$$V_r^{\pi_{k+1}}(\rho) - V_r^{\pi_k}(\rho) \geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_{k+1}^r}^{\pi_k} \\ a\sim\pi_{k+1}}}\Big[A_{r,p_{k+1}^r}^{\pi_k}(s,a) - \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big]. \tag{39}$$

Note that $A_{r,p_{k+1}^r}^{\pi_k}(s,a) = Q_{r,p_{k+1}^r}^{\pi_k}(s,a) - V_{r,p_{k+1}^r}^{\pi_k}(s)$ is Lipschitz in π_k (Wang et al., 2023a). We have that there exists L_π such that

$$|A_{r,p_{k+1}^r}^{\pi_k}(s,a) - A_{r,p_{k+1}^r}^{k+1}(s,a)| \leq L_\pi\|\pi_{k+1}(s) - \pi_k(s)\|_1. \tag{40}$$

We then have that

$$\begin{aligned}
&\frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_{k+1}^r}^{\pi_k} \\ a\sim\pi_{k+1}}}\Big[A_{r,p_{k+1}^r}^{\pi_k}(s,a) - \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big] \\
&\geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_{k+1}^r}^{\pi_k} \\ a\sim\pi_{k+1}}}\Big[A_{r,p_{k+1}^r}^{\pi_{k+1}}(s,a) - L_\pi\|\pi_{k+1}(s) - \pi_k(s)\|_1 - \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big] \\
&= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{p_{k+1}^r}^{\pi_k}}\Big[-L_\pi\|\pi_{k+1}(s) - \pi_k(s)\|_1 - \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big],
\end{aligned} \tag{41}$$

where the equality is due to the fact that $\mathbb{E}_{a\sim\pi_{k+1}}\big[A_{r,p_{k+1}^r}^{\pi_{k+1}}(s,a)\big] = \mathbb{E}_{a\sim\pi_{k+1}}\big[Q_{r,p_{k+1}^r}^{\pi_{k+1}}(s,a) - V_{r,p_{k+1}^r}^{\pi_{k+1}}(s)\big] = 0$. Since $\|\pi_{k+1}(s) - \pi_k(s)\|_1 = 2D_{TV}(\pi_{k+1}\|\pi_k)(s) \leq \sqrt{2D_{KL}(\pi_{k+1}\|\pi_k)(s)}$ (Csiszár & Körner, 2011), we have that

$$\begin{aligned}
&\frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{p_{k+1}^r}^{\pi_k}}\Big[-L_\pi\|\pi_{k+1}(s) - \pi_k(s)\|_1 - \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big] \\
&\geq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{p_{k+1}^r}^{\pi_k}}\Big[-\big(2L_\pi + \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\big)\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big] \\
&\overset{(a)}{\geq} \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{p_k^{r,\boldsymbol{\xi}}}^{\pi_k}}\Big[-M\big(2L_\pi + \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\big)\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big] \\
&\overset{(b)}{\geq} -\frac{1}{1-\gamma}M\big(2L_\pi + \frac{2\gamma\epsilon_{r,p_{k+1}^r}^{\pi_{k+1}}}{1-\gamma}\big)\sqrt{\frac{\delta}{2}},
\end{aligned} \tag{42}$$

where (a) is due to the fact that $M = \sup_{p,p'\in\mathcal{P}}\|d_p^{\pi_k}/d_{p'}^{\pi_k}\|_\infty$ is finite and (b) is from Lemma D.1 and Jensen's inequality. To characterize the constraint violation, we first have that

$$V_{c,p_k^{c,\boldsymbol{\xi}}}^{\pi_k}(\rho) + \mathbb{E}_{\substack{s\sim d_{p_k^{c,\boldsymbol{\xi}}}^{\pi_k} \\ a\sim\pi_{k+1}}}[A_{c,p_k^{c,\boldsymbol{\xi}}}^{\pi_k}(s,a)] \geq d. \tag{43}$$

and

$$\begin{aligned}
V_c^{\pi_{k+1}}(\rho) - V_c^{\pi_k}(\rho) &= V_{c,p_{k+1}^c}^{\pi_{k+1}}(\rho) - V_{c,p_{\pi_k}^c}^{\pi_k}(\rho) \\
&\geq V_{c,p_{k+1}^c}^{\pi_{k+1}}(\rho) - V_{c,p_{k+1}^c}^{\pi_k}(\rho) \\
&\geq \frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d_{p_{k+1}^c}^{\pi_k} \\ a\sim\pi_{k+1}}}\Big[A_{c,p_{k+1}^c}^{\pi_k}(s,a) - \frac{2\gamma\epsilon_{c,p_{k+1}^c}^{\pi_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big].
\end{aligned} \tag{44}$$

Following the proof of (42), we have that

$$V_c^{\pi_{k+1}}(\rho) \geq d - (V_{c,p_k^{c,\boldsymbol{\xi}}}^{\pi_k}(\rho) - V_c^{\pi_k}(\rho)) - \mathbb{E}_{\substack{s\sim d_{p_k^{c,\boldsymbol{\xi}}}^{\pi_k} \\ a\sim\pi_{k+1}}}[A_{c,p_k^{c,\boldsymbol{\xi}}}^{\pi_k}(s,a)]$$

$$
\begin{aligned}
&+\frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi_k}_{p^c_{k+1}}\\ a\sim\pi_{k+1}}}\Big[A^{\pi_k}_{c,p^c_{k+1}}(s,a)-\frac{2\gamma\epsilon^{\pi_{k+1}}_{c,p^c_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big]\\
\geq\ & d-\epsilon-\mathbb{E}_{\substack{s\sim d^{\pi_k}_{p^{c,\boldsymbol{\xi}}_k}\\ a\sim\pi_{k+1}}}\big[A^{\pi_{k+1}}_{c,p^c_k}(s,a)+L_\pi\|\pi_{k+1}(s)-\pi_k(s)\|_1\big]\\
&+\frac{1}{1-\gamma}\mathbb{E}_{\substack{s\sim d^{\pi_k}_{p^c_{k+1}}\\ a\sim\pi_{k+1}}}\Big[A^{\pi_{k+1}}_{c,p^c_{k+1}}(s,a)-L_\pi\|\pi_{k+1}(s)-\pi_k(s)\|_1-\frac{2\gamma\epsilon^{\pi_{k+1}}_{c,p^c_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big]\\
=\ & d-\epsilon-\mathbb{E}_{s\sim d^{\pi_k}_{p^{c,\boldsymbol{\xi}}_k}}\big[L_\pi\|\pi_{k+1}(s)-\pi_k(s)\|_1\big]\\
&+\frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_k}_{p^c_{k+1}}}\Big[-L_\pi\|\pi_{k+1}(s)-\pi_k(s)\|_1-\frac{2\gamma\epsilon^{\pi_{k+1}}_{c,p^c_{k+1}}}{1-\gamma}\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big]\\
\geq\ & d-\epsilon+\frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_k}_{p^{c,\boldsymbol{\xi}}_k}}\Big[-M\big(3L_\pi+\frac{2\gamma\epsilon^{\pi_{k+1}}_{c,p^c_{k+1}}}{1-\gamma}\big)\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi_k)(s)}\Big]\\
\geq\ & d-\epsilon-\frac{1}{1-\gamma}M\big(3L_\pi+\frac{2\gamma\epsilon^{\pi_{k+1}}_{c,p^c_{k+1}}}{1-\gamma}\big)\sqrt{\frac{\delta}{2}}.
\end{aligned}
\tag{45}
$$

This completes the proof. □

E. Proof of Theorem 4.4

We first provide an upper bound on the KL divergence between π_k and π_{k+1} in the following lemma. We then follow the proof of Theorem 4.3 to prove Theorem 4.4.

Lemma E.1. *If the current policy π_k violates the constraint and the constraint set is convex and closed under the policy parameterization, let $b = V^\pi_c(\rho) - d$, then under the KL divergence projection, we have that*

$$
\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}[D_{KL}(\pi_{k+1}||\pi_k)(s)] \leq \delta + b^2\alpha_{KL} + bM'\sqrt{\frac{\alpha_{KL}}{2}},
\tag{46}
$$

where $\alpha_{KL} = \frac{1}{2\boldsymbol{h}^\top\boldsymbol{H}^{-1}\boldsymbol{h}}$, $\boldsymbol{h}$ is the gradient of the utility advantage function, $\boldsymbol{H}$ is the Hessian matrix of the KL divergence constraint, $M' \leq \infty$ is some constant.

Proof. Define the following set:

$$
Z_{\pi_k} = \big\{\pi\big|V^{\pi_k}_{c,p^{c,\boldsymbol{\xi}}_k}(\rho) + \mathbb{E}_{s\sim d^{\pi_k}_{p^{c,\boldsymbol{\xi}}_k},a\sim\pi}[A^{\pi_k}_{c,p^{c,\boldsymbol{\xi}}_k}(s,a)] \geq V^{\pi_k}_{c,p^{c,\boldsymbol{\xi}}_k}(\rho)\big\}.
\tag{47}
$$

Note that the current policy π_k lies in Z_{π_k}. Define the policy π^l_{k+1} as the projection of $\pi_{k+\frac{1}{2}}$ onto Z_{π_k}. We have that

$$
D_{KL}(\pi_{k+1}\|\pi_k)(s) = D_{KL}(\pi^l_{k+1}\|\pi_k)(s) + D_{KL}(\pi_{k+1}\|\pi^l_{k+1})(s) + \big(\pi_{k+1}(s)-\pi^l_{k+1}(s)\big)^\top\log\frac{\pi^l_{k+1}(s)}{\pi_k(s)}.
\tag{48}
$$

From D.1, we have that $\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}[D_{KL}(\pi^l_{k+1}\|\pi_k)(s)] \leq \delta$. For small b, $\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}[D_{KL}(\pi_{k+1}\|\pi^l_{k+1})(s)]$ can be approximated by the second order expansion. We have that

$$
\begin{aligned}
\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}[D_{KL}(\pi_{k+1}\|\pi^l_{k+1})(s)] &\approx \frac{1}{2}(\boldsymbol{\theta}_{k+1}-\boldsymbol{\theta}^l_{k+1})^\top\boldsymbol{H}(\boldsymbol{\theta}_{k+1}-\boldsymbol{\theta}^l_{k+1})\\
&= \frac{1}{2}\Big(\frac{b}{\boldsymbol{h}^\top\boldsymbol{H}^{-1}\boldsymbol{h}}\boldsymbol{H}^{-1}\boldsymbol{h}\Big)^\top\boldsymbol{H}\Big(\frac{b}{\boldsymbol{h}^\top\boldsymbol{H}^{-1}\boldsymbol{h}}\boldsymbol{H}^{-1}\boldsymbol{h}\Big)\\
&= \frac{b^2}{2\boldsymbol{h}^\top\boldsymbol{H}^{-1}\boldsymbol{h}}\\
&= b^2\alpha_{KL},
\end{aligned}
\tag{49}
$$

where $\alpha_{KL} = \frac{1}{2\boldsymbol{h}^\top \boldsymbol{H}^{-1}\boldsymbol{h}}$ and the first equality is from the update rule in (26). For $\left(\pi_{k+1}(s) - \pi^l_{k+1}(s)\right)^\top \log \frac{\pi^l_{k+1}(s)}{\pi_k(s)}$, we have that

$$\left(\pi_{k+1}(s) - \pi^l_{k+1}(s)\right)^\top \log \frac{\pi^l_{k+1}(s)}{\pi_k(s)} \leq \|\pi_{k+1}(s) - \pi^l_{k+1}(s)\|_1 \left\| \log \frac{\pi^l_{k+1}(s)}{\pi_k(s)} \right\|_\infty. \tag{50}$$

Since $\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}[D_{KL}(\pi^l_{k+1}\|\pi_k)(s)] \leq \delta$, there exists M' such that $\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}\left[\left\| \log \frac{\pi^l_{k+1}(s)}{\pi_k(s)} \right\|_\infty\right] \leq M'$. Moreover, we have that $\|\pi_{k+1}(s) - \pi^l_{k+1}(s)\|_1 \leq \sqrt{2D_{KL}(\pi_{k+1}\|\pi^l_{k+1})(s)}$. We then have that

$$\begin{aligned}
&\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}\left[\left(\pi_{k+1}(s) - \pi^l_{k+1}(s)\right)^\top \log \frac{\pi^l_{k+1}(s)}{\pi_k(s)}\right] \\
&\leq \mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}\left[M'\sqrt{\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi^l_{k+1})(s)}\right] \\
&\overset{(a)}{\leq} M'\sqrt{\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}\left[\frac{1}{2}D_{KL}(\pi_{k+1}\|\pi^l_{k+1})(s)\right]} \\
&\overset{(b)}{\approx} bM'\sqrt{\frac{\alpha_{KL}}{2}},
\end{aligned} \tag{51}$$

where (a) is from Jensen's inequality and (b) is from (49).

By combining (48), (49) and (51), we have that

$$\mathbb{E}_{s\sim d^{\pi_k}_{p^{r,\boldsymbol{\xi}}_k}}[D_{KL}(\pi_{k+1}||\pi_k)(s)] \leq \delta + b^2\alpha_{KL} + bM'\sqrt{\frac{\alpha_{KL}}{2}}. \tag{52}$$

□

With Lemma E.1, Theorem 4.4 can be proved similarly as Theorem 4.3.

F. Experiments

The detailed environments descriptions are in the following:

Gambler Problem in a game in which a gambler bets on a sequence of coin tosses, wining the stake when the outcome is head and losing when it's tail. Starting from an initial balance, the game ends once the gambler's balance reaches 16 or 0. For different state-action pairs, the gambler receives different utilities. The reward is 10 when the balance reaches 16 and 0 otherwise. The probability of head for each coin toss is $p = 0.6$. The radius of the uncertainty set is 0.1.

N-chain problem involves a chain with N nodes. At each node, the agent can choose to move to its left or its right. Upon moving to its left, it receives a reward-utility signal of $(1, 0)$, while moving to its right yields a reward-utility signal of $(0, 2)$. When reaching the N-th node, the agent receives a bonus reward of 10. With probability 0.1, the agent may slip to the different direction of its action. We let $N = 40$ and the radius of the uncertainty set be 0.15.

Frozen-Lake problem is about training an agent to cross a 4×4 frozen lake from the starting point to the end point without falling into any holes. Upon falling into the holes, the agent will get trapped and receive zero reward and utility. Reaching the end point yields a reward of $r = 200$, otherwise $r = 0$. At some states, the agent will receive a utility of $c = 1$. The agent may slip to the different direction of its action. The radius of the uncertainty set is 0.1.

Point Gather is a benchmark Mujoco task for constrained MDP, in which an agent is rewarded for gathering green apples but is constrained to collect a limited number of red fruit (Achiam et al., 2017).