2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Distributionally Robust Behavioral Cloning for Robust Imitation Learning

Kishan Panaganti*,[1], Zaiyan Xu*,[1], Dileep Kalathil[1], Mohammad Ghavamzadeh[2]

Abstract— **Robust reinforcement learning (RL) aims to learn a policy that can withstand uncertainties in model parameters, which often arise in practical RL applications due to modeling errors in simulators, variations in real-world system dynamics, and adversarial disturbances. This paper introduces the robust imitation learning (IL) problem in a Markov decision process (MDP) framework where an agent learns to mimic an expert demonstrator that can withstand uncertainties in model parameters without additional online environment interactions. The agent is only provided with a dataset of state-action pairs from the expert on a single (nominal) dynamics, without any information about the true rewards from the environment. Behavioral cloning (BC), a supervised learning method, is a powerful algorithm to address the vanilla IL problem. We propose an algorithm for the robust IL problem that utilizes distributionally robust optimization (DRO) with BC. We call the algorithm DR-BC and show its robust performance against parameter uncertainties both in theory and in practice. We also demonstrate the empirical performance of our approach to addressing model perturbations on several MuJoCo continuous control tasks.**

Index Terms— **Imitation Learning, Reinforcement Learning, Robust Reinforcement Learning**

I. Introduction

A child, a dog, or even a reptile is capable of learning through imitation [1]. Such intuitive way of learning naturally extends from animal's survival instincts to solving potentially complicated control tasks. Hence, it serves as the primary philosophy underlying most, if not all, methods in Imitation Learning (IL), a very fundamental reinforcement learning (RL) setting in which the goal is to learn a control policy exclusively from expert demonstrations. However simple and fundamental the idea of imitation learning may sound, variations in the training (simulators) and testing (real-world) environments can result in significant failures of current RL and IL control policies [2]–[5].

Imitation Learning: Learning through imitation can be traced back to as early as [6]. Imitation learning assumes access to only expert demonstrations. This has given rise to the most natural approach, behavior cloning [7], which is a supervised learning method and learns by simply minimizing differences between the actions of the learners and those of the experts for the states seen by the expert. [8] studied behavior cloning and characterized a tight bound on its sub-optimality gap of order $O(\varepsilon H^2)$, where H is the horizon and ε is the learning and statistical errors of the algorithm. Most of works in imitation learning then try to improve this bound with additional assumptions. [9] proposed DAGGER which required hindsight expert actions for the states visited by the learner. They showed that the above bound on the sub-optimality gap can be improved to $O(\varepsilon u H)$, where u is the cost of taking a different action than the expert's at one step and following the expert's suggestions afterward, by querying the expert and interacting with the environment. In worse cases, u can be as large as H. Another approach, DRIL [10], needs environment interactions but no expert query, and achieves a sub-optimality gap bound that is linear in H. GAIL [11] uses a discriminator network to distinguish between the expert's states from those visited by the learner's policy while interacting with the environment. GAIL also achieves a sub-optimality gap bound that is linear in H [12]. In [13], the authors provide a game-theoretic framework for the IL problem that naturally competes with noisy expert policies. Different from these works, we focus on setting up an IL problem to address the parameter mismatch between the training and testing environments, and provide a practical algorithm to solve it.

Robust RL: The framework of the robust Markov decision process (RMDP) [14], [15] addresses the problem of learning a policy that is robust against mismatches between the training and testing environments. This is the goal of distributionally robust reinforcement learning (DR-RL). Robust RL has application in many real-world evolving systems in which there is always a gap between the true model and the simulator. Deploying naive RL policies [16] can be catastrophic when this gap is large. The RMDP problem is well-studied. [17]–[21] have investigated various types of uncertainty sets and sought tractable methods to solve RMDP. [22]–[24] have studied the sample complexity of model-based robust RL algorithms in a tabular setting using a generative model, which is a strong oracle enabling learners to query arbitrary transitions. [25] developed a model-free online robust RL algorithm with linear function approximation to tackle potentially infinite state spaces and [26] similarly developed a model-free *offline* robust RL algorithm with general function approximation. [27] proposed an online robust Q-learning with an R-contamination uncertainty set. We would like to note that robust RL has a strong connection to distributionally robust optimization (DRO). Many of the optimization techniques and analyses in robust RL were originally developed in the context of supervised learning by the DRO community [28]–[33]. This line of work is closest to ours, but as per our knowledge, ours is the first work to focus on addressing the parameter mismatch between the training and testing environments using

* Equal contributions, [1] Authors are with the Department of Electrical and Computer Engineering at Texas A&M University, College Station, TX, USA. Email: {kpb, zxu43, dileep.kalathil}@tamu.edu , [2] Author is with Google Research, USA. Email: ghavamza@google.com. This work was supported in part by the National Science Foundation (NSF) grants NSF-CAREER-EPCN-2045783 and NSF-CNS 1955696.

DRO techniques in the context of *imitation learning* setting.

Main Contributions: We summarize our contributions in this paper as follows and refer to the relevant sections: **(i)** We introduce the problem of robust imitation learning for mismatch in model parameters. In this work, we consider mismatch in system transition dynamics. Robust learning in RL is studied widely but not in IL. Critical real-world applications, such as power systems, healthcare, self-driving automobiles, have guidance from expert across diverse scenarios [34]–[36]. **(ii)** We propose a novel robust IL algorithm, called Distributionally Robust Behavioral Cloning (DR-BC). We provide theoretical guarantees for DR-BC. The BC method is computationally efficient which makes any other non-robust IL algorithm falls short on, and the DRO methodology addresses the model mismatch. Our proposed method cleverly bridges these two methods to solve the robust IL problem. We discuss this further in Section II. **(iii)** We perform extensive simulations on four notable continuous-action OpenAI [37] Gym MuJoCo [38] environments. We demonstrate that the DR-BC policy is robust against model perturbations in that when BC has catastrophic drop in performance, DR-BC weathers model mismatches for much more sever model perturbations.

II. Robust Imitation Learning

In this section, we formally introduce our imitation learning problem that addresses parameter mismatch between the true and simulated models (transition dynamics).

A. Problem Formulation

The goal of IL in sequential decision-making is to *imitate* an expert's policy by only using demonstrations generated by its interactions with the environment [10], [39]. More formally, consider an infinite-horizon MDP, denoted by the tuple $\{\mathcal{S}, \mathcal{A}, P^o, \gamma, r, \mu\}$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite state and action spaces, $P^o : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition dynamics (*model*) of the environment, γ is a discount factor, μ is the initial state distribution, and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the true reward function (unknown to the learner). A stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states to distributions over actions. For any policy π, the value function of an initial state $s_0 \sim \mu$ is given by $V_\pi(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P^o_{s_t,a_t}]$. We just denote $V_\pi = \mathbb{E}_{s_0 \sim \mu} V_\pi(s_0)$ going forward for notation simplicity.

For any policy π, denote $d^\pi_{P^o} \in \Delta(\mathcal{S})$ as the state distribution of π under the evaluation of model P^o with initial state picked from μ. In this section, we simply denote such state-distributions as d^π and make explicit dependence on P^o where brevity is needed. Formally, let $\Pr_t(s|\pi, s_0 \sim \mu)$ be the probability of visiting state $s \in \mathcal{S}$ at time t following policy π on model P^o starting at initial state $s_0 \sim \mu$. Then, the state distribution of π is $d^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr_t(s|\pi, s_0 \sim \mu, P^o)$. We can now rewrite the value of π as $V_\pi = \mathbb{E}_{s \sim d^\pi, a \sim \pi}[r(s, a)]/(1 - \gamma)$.

In the vanilla IL setting, the true reward function r is unknown to the learner. We instead have the dataset generated by rolling an expert policy (which is unknown to the learner) specified by $\pi_e : \mathcal{S} \to \Delta(\mathcal{A})$. Concisely speaking, we have an expert dataset in the form of i.i.d. tuples $\mathcal{D}_e = \{s_i, a_i\}_{i=1}^N$ sampled from state distribution $d^{\pi_e}_{P^o}$ and an expert policy π_e.

We use the RMDP framework [14], [15] subsuming the MDP framework described above. Consider an RMDP tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r, \mu\}$ where $\gamma \in [0.5, 1)$ and the *uncertainty set*

$$\mathcal{P} = \otimes_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_{s,a} \quad \text{with}$$
$$\mathcal{P}_{s,a} = \{P_{s,a} \in \Delta(\mathcal{S}) : P_{s,a} \ll P^o_{s,a}, D(P_{s,a}, P^o_{s,a}) \leq \rho'_{\mathrm{r}}\}, \quad (1)$$

where $P^o = (P^o_{s,a}, (s, a) \in \mathcal{S} \times \mathcal{A})$ is the simulator model, $D(\cdot, \cdot)$ is a distance measure between two probability distributions (e.g., total variation, chi-square, Kullback-Liebler (f-divergences in general), Wasserstein), and $\rho'_{\mathrm{r}} \in (0, (1-\gamma)/\gamma]$ is the radius of the uncertainty set indicating the level of robustness. We assume the real-world model belongs to this uncertainty set $\mathcal{P}$. We restrict to the total-variation distance D_{TV} for the measure D in this paper and leave other types of measures for future work.

From the RMDP literature [14], [24], [26], [40], we introduce the *robust* value function as $V^{\mathrm{rob}}_\pi(s) = \sum_a \pi(a|s) Q^{\mathrm{rob}}_\pi(s, a)$ and the corresponding *robust* Q-value function as $Q^{\mathrm{rob}}_\pi(s, a) = r(s, a) + \gamma \inf_{P_{s,a} \in \mathcal{P}_{s,a}} P^\top_{s,a} V^{\mathrm{rob}}_\pi$ for policy π. Similar to [8], we let $\pi_e \in \Pi$, where Π is the class of stochastic policies, be a *good robust* policy (under the above RMDP setting). That is, it satisfies $\max_{\pi \in \Pi} V^{\mathrm{rob}}_\pi - V^{\mathrm{rob}}_{\pi_e} \leq o(H)$ (something small compared to time horizon). For notation simplicity, as in [8], [39], we just let $V^{\mathrm{rob}}_{\pi_e} \geq V^{\mathrm{rob}}_\pi$ hold for all $\pi \in \Pi$. **Now, we pose the robust IL problem as follows.** The goal of a robust IL algorithm is to output a policy $\hat{\pi}$ that imitates the expert policy π_e by satisfying $V^{\mathrm{rob}}_{\hat{\pi}} \approx V^{\mathrm{rob}}_{\pi_e}$. We have provided real-world applications that motivate this problem formulation in Section I.

B. Need for Robust Imitation Learning

In this section, we formally show that the vanilla behavioral cloning policy Equation (2) can be arbitrarily bad (as bad as a random policy) compared to an expert policy (*good robust* policy). Consider the following vanilla behavioral cloning [8] optimization problem.

$$\pi_{\mathrm{bc}} = \arg\min_\pi L_{\mathrm{bc}}(\pi) = \mathbb{E}_{s \sim d^{\pi_e}}[l(\pi_e(\cdot \mid s), \pi(\cdot \mid s))]. \quad (2)$$

We assume access to sampling possibly *infinite data* from the state distribution d^{π_e} to calculate the loss $L_{\mathrm{bc}}(\pi_{\mathrm{bc}})$ up to some small error. We consider a simple setting with $\mathcal{P} = \{P^o, P'\}$ where P^o is the simulator model and P' is the perturbed model. Following a similar analysis in [41, Theorem 4] we state the following result.

Theorem 1 (Robustness Gap)**.** *There exists an uncertainty set $\mathcal{P} = \{P^o, P'\}$, initial state $s_0 \in \mathcal{S}$, expert policy π_e such that $\max_{\pi \in \Pi} V^{\mathrm{rob}}_\pi(s_0) - V^{\mathrm{rob}}_{\pi_e}(s_0) \leq \varepsilon$ for small $\varepsilon > 0$, and discount factor $\gamma \in (\gamma_o, 1]$ such that $V^{\mathrm{rob}}_{\pi_{\mathrm{bc}}}(s_0) \leq V^{\mathrm{rob}}_{\pi_e}(s_0) - c/(1 - \gamma)$, where c is a positive constant.*

Remark 1. The vanilla behavioral cloning policy π_{bc} compared to expert policy (*good robust* policy) is bad with a performance gap $\Omega(1/(1-\gamma))$. Since $|r(s, a)| \leq 1$ uniformly by assumption, $\|V^{\mathrm{rob}}_\pi\|_\infty \leq 1/(1 - \gamma)$ for any policy π. Therefore, the difference between the optimal/expert robust value function and the robust value function of an arbitrary policy cannot be greater than $\mathcal{O}(1/(1-\gamma))$. Thus the performance of π_{bc} can be as bad as an arbitrary policy in an order sense. In the next section, we propose an algorithm to solve the robust imitation learning problem.

C. Robust Against Model Mismatch

We propose a principled adversarial approach by the methodology of distributionally robust optimization (DRO) to solve the robust imitation learning problem. DRO is now a well-established area [28], [32], [42], whose formulation is identical to that in the classical RMDP [14], [15] in DR-RL. The distributionally robust behavioral cloning algorithm solves this optimization problem getting the policy π_{drbc}:

$$\arg\min_{\pi} \max_{P\in\mathcal{P}\,:\,D_{\mathrm{TV}}(d_P^{\pi_e},d_{P^o}^{\pi_e})\le\rho_{\mathrm{r}}} \mathbb{E}_{s\sim d_P^{\pi_e}}[l(\pi(\cdot|s),\pi_e(\cdot|s))], \quad (3)$$

where ρ_{r}, the *robustness radius parameter* which is a problem-dependent constant, is set to $\gamma\rho_{\mathrm{r}}'/(1-\gamma)\in(0,1]$, and $l(\pi(\cdot|s),\pi_e(\cdot|s))$ is a surrogate loss function which measures how far the learner policy π is with respect to the expert action for the states visited by the expert. Examples of the loss function l comprise of 0-1 loss (described by $\mathbb{E}_{a\sim\pi(\cdot|s)}\mathbb{1}(a\neq\pi_e(s))$ for deterministic expert policies), total variation loss (described by $D_{\mathrm{TV}}(\pi_e(\cdot|s),\pi(\cdot|s)) = 0.5\,\|\pi_e(\cdot|s)-\pi(\cdot|s)\|_1$), KL loss (described by $D_{\mathrm{KL}}(\pi(\cdot|s)\|\pi_e(\cdot|s)) = \sum_a \pi(a|s)\log(\pi(a|s)/\pi_e(a|s))$ with π absolutely continuous to π_e), and many more such quantifiers. We simply use the D_{TV} loss function in this paper considering its known connections with other f-divergences [43], [44]. We note that the DR-BC policy π_{drbc} depends on ρ_{r} but we simply choose to make it inherent for notation simplicity. We also remark that we recover the behavioral cloning policy [7] with $\rho_{\mathrm{r}}=0$ in DR-BC policy Equation (3).

We define the uncertainty set parameterized by ρ_{r} as $\mathcal{M}=\{P\in\mathcal{P}: d_P^{\pi_e}\ll d_{P^o}^{\pi_e}, D_{\mathrm{TV}}(d_P^{\pi_e},d_{P^o}^{\pi_e})\le\rho_{\mathrm{r}}\}$. It is straightforward from its definition and Lemma 7 that $\mathcal{M}=\mathcal{P}$. The DR-BC algorithm Equation (3) finds π_{drbc} for the IL problem by minimizing an observed surrogate loss between its actions and the actions of an expert policy under the adversarial state distribution for a model in class $\mathcal{M}$ which acts as a worse-case distribution. We define the model mismatch distributionally robust behavioral cloning loss function as $L_{\mathrm{drbc}}(\pi,\rho_{\mathrm{r}}) = \max_{P\in\mathcal{M}}\mathbb{E}_{s\sim d_P^{\pi_e}}[D_{\mathrm{TV}}(\pi(\cdot|s),\pi_e(\cdot|s))]$ for any policy π and ρ_{r}. But we immediately notice that to solve the inner optimization in Equation (3) we need access to all the state distributions around the expert's state distribution. Even knowing the model P^o, this is computationally intractable. Moreover, we would also need the capability of querying an expert for actions for various states chosen by such state distributions. Also assuming having access to all models in $\mathcal{M}$ is unrealistic. We now discuss circumventing this challenge using the DRO methodology [28], [29].

Motivated from the DR-RL literature [24], [26], [40], we now have the following result that provides a dual reformulation for the inner maximization in Equation (3) as a consequence of the DRO methodology.

Proposition 2. *For a fixed expert policy $\pi_e\in\Pi$, we have, for all $\pi\in\Pi$ and $\rho_{\mathrm{r}}\in(0,1]$,*

$$\begin{aligned}&\max_{P\in\mathcal{M}}\mathbb{E}_{s\sim d_P^{\pi_e}}[D_{\mathrm{TV}}(\pi(\cdot|s),\pi_e(\cdot|s))]\\&=\min_{\eta\in\mathbb{R}}\ \mathbb{E}_{s\sim d_{P^o}^{\pi_e}}[(D_{\mathrm{TV}}(\pi(\cdot|s),\pi_e(\cdot|s))-\eta)_+]\\&\quad+(\sup_{s\in\mathcal{S}:d_{P^o}^{\pi_e}(s)>0} D_{\mathrm{TV}}(\pi(\cdot|s),\pi_e(\cdot|s))-\eta)_+\cdot\rho_{\mathrm{r}}+\eta.\end{aligned}$$

Proof. We first rewrite $\max_{P\in\mathcal{M}}\mathbb{E}_{s\sim d_P^{\pi_e}}[D_{\mathrm{TV}}(\pi(\cdot|s),\pi_e(\cdot|s))]$ as $\max_{d_P^{\pi_e}\ll d_{P^o}^{\pi_e}:D_{\mathrm{TV}}(d_P^{\pi_e},d_{P^o}^{\pi_e})\le\rho_{\mathrm{r}}}\mathbb{E}_{s\sim d_P^{\pi_e}}[D_{\mathrm{TV}}(\pi(\cdot|s),\pi_e(\cdot|s))]$ which follows from definition of $\mathcal{M}$. Then the statement immediately follows from Lemma 5. □

We give our DR-BC algorithm that only requires an expert dataset $\mathcal{D}_e$ generated according to model P^o in Algorithm 1 based on Proposition 2. The DRO technique in Proposition 2 transforms the inner maximization in Equation (3) to an unconstrained scalar variables convex optimization problem. We remark that this new optimization problem due to the dual reformulation only depends on the expert's state distribution. This enables us to use the expert dataset to solve the DR-BC objective Equation (3).

Algorithm 1 Distributionally Robust Behavioral Cloning

1: **Input:** Expert dataset $\mathcal{D}_e=(s_i,a_i)_{i=1}^N$ according to model P^o, model mismatch radius parameter ρ_{r}.
2: **Initialize:** Policy π_θ parameterized by θ.
3: Calculate the empirical loss for $L_{\mathrm{drbc}}(\pi_\theta,\rho_{\mathrm{c}})$:

$$\begin{aligned}\min_{\eta\in\mathbb{R}}((1/N)&\sum_{(s,a)\in\mathcal{D}_e}(l(a,\pi_\theta(s))-\eta)_+)\\&+\rho_{\mathrm{r}}\big(\sup_{(s,a)\in\mathcal{D}_e} l(a,\pi_\theta(s))-\eta\big)_+ +\eta.\end{aligned} \quad (4)$$

4: $\theta\leftarrow\arg\min_\theta L_{\mathrm{drbc}}(\pi_\theta,\rho_{\mathrm{r}})$.
5: **Output policy:** $\widehat{\pi}_{\mathrm{drbc}}=\pi_\theta$

We now give the sub-optimality guarantee of model mismatch DR-BC policy. We provide its proof in Section III.

Theorem 3 (Model mismatch DR-BC sub-optimality bound)**.** *Assume small optimization error $L_{\mathrm{drbc}}(\pi_{\mathrm{drbc}},\rho_{\mathrm{r}})=\varepsilon_{\mathrm{drbc}}(\rho_{\mathrm{r}})$. We have $V_{\pi_e}^{\mathrm{rob}}-V_{\pi_{\mathrm{drbc}}}^{\mathrm{rob}}\le 2\varepsilon_{\mathrm{drbc}}(\rho_{\mathrm{r}})/(1-\gamma)^2$.*

Remark 2. We have an $\mathcal{O}(\varepsilon_{\mathrm{drbc}}(\rho_{\mathrm{r}})H^2)$ sub-optimality bound from Theorem 3. With a small optimization error $\varepsilon_{\mathrm{drbc}}(\rho_{\mathrm{r}})$, the sub-optimality guarantee for DR-BC algorithm is superior to the BC policy as discussed in Section II-B. When the robustness parameter $\rho_{\mathrm{r}}=0$, we recover the non-robust BC algorithm and its quadratic horizon dependence [8]. This sub-optimality bound is in fact tight $\Omega(\varepsilon H^2)$ [8], [39].

We also present the approximation result for the sub-optimality of $\hat{\pi}_{\mathrm{drbc}}$ returned by Algorithm 1 that uses the expert dataset $\mathcal{D}_e$. We consider $\mathrm{e}_{\min}$, the minimum non-zero probability value in π_e, as a problem dependent constant. We again consider $L_{\mathrm{drbc}}(\widehat{\pi}_{\mathrm{drbc}},\sqrt{\rho_{\mathrm{r}}})=\varepsilon_{\mathrm{drbc}}(\rho_{\mathrm{r}})>0$ be a small optimization error for all $\rho_{\mathrm{r}}\in(0,1]$.

Theorem 4 (Approximate DR-BC sub-optimality bound)**.** *Let $\widehat{\varepsilon}_{\mathrm{drbc}}(\rho_{\mathrm{r}})=\varepsilon_{\mathrm{drbc}}(\rho_{\mathrm{r}})+\widetilde{\mathcal{O}}(\rho_{\mathrm{r}}\sqrt{\log(1/\delta)/(\mathrm{e}_{\min}N)})$. Then,* *for any $\rho_r \in (0,1]$, policy $\widehat{\pi}_{\rm drbc}$ satisfies $V^{\rm rob}_{\pi_e} - V^{\rm rob}_{\widehat{\pi}_{\rm drbc}} \leq 2\widehat{\varepsilon}_{\rm drbc}(\rho_r)/(1-\gamma)^2$, with probability at least $1-\delta$.*

Remark 3. We note that $\widetilde{\mathcal{O}}(\cdot)$ is order optimal up to a logarithmic term on N and its exact form is available in Section III. The approximate sub-optimality guarantee for DR-BC algorithm is still superior to the BC policy as discussed in Section II-B. We indeed showcase empirically as well that DR-BC algorithm is resilient to model perturbations in Section IV.

III. Analysis Details

A. Useful Technical Results

We now state a result from [26] based on DRO methodology which is useful for proving Proposition 2.

Lemma 5 ([26, Lemma 5]). *Let P^o be a non-zero distribution on the space $\mathcal{X}$ and $l : \mathcal{X} \to \mathbb{R}$ be a loss function. Then*

$$\sup_{P \ll P^o, D_{\rm TV}(P,P^o)\leq\rho} \mathbb{E}_{x\sim P}[l(x)] = \inf_{\eta\in\mathbb{R}} \big\{\mathbb{E}_{x\sim P^o}[(l(x)-\eta)_+] + (\sup_{x\in\mathcal{X}} l(x)-\eta)_+\cdot\rho + \eta\big\}. \tag{5}$$

We now state [28, Corollary 2] for $D_{\rm TV}$ using its relation [44, Theorem 5] with D_{χ^2}.

Lemma 6. *Let $\Theta \subseteq \mathbb{R}^d$, $l : \mathcal{X}\times\Theta \mapsto [0,M]$ and fix any $\rho\in(0,1]$. We have*

$$\sup_{D_{\rm TV}(P,P^o)\leq\rho} \mathbb{E}_P[l(X,\widehat{\theta})] \leq \inf_{\theta\in\Theta} \sup_{D_{\rm TV}(P,P^o)\leq\sqrt{\rho}} \mathbb{E}_P[l(X,\theta)] + cM(1+\rho)\sqrt{(\log(1/\delta)+2d\log(N))/N},$$

which holds with probability at least $1-\delta$, where $c > 0$ is some universal constant and $\widehat{\theta} = \arg\min_{\theta\in\Theta}\sup_{D_{\rm TV}(P,\widehat{P^o})\leq\sqrt{\rho}} \mathbb{E}_P[l(X,\theta)]$.

B. Proof of Theorem 3

We present a few results needed for proving Theorem 3. First, we have the following result following the proof of [45, Lemma 14.1], that when two models are close, then their state-distributions are close under the same policy.

Lemma 7. *Consider any policy π and $P\in\mathcal{P}$. We have $D_{\rm TV}(d^\pi_P, d^\pi_{P^o}) \leq \gamma\rho'_r/(1-\gamma)$ with $d^\pi_P \ll d^\pi_{P^o}$.*

Now we state a result which extends the performance difference lemma [45, Lemma 1.16] notion for robust MDPs.

Lemma 8 (Robust Performance Difference Lemma). *For any π',π policies, we get*

$$V^{\rm rob}_\pi - V^{\rm rob}_{\pi'} \leq \frac{1}{1-\gamma}\cdot \max_{P:D(d^\pi_P,d^\pi_{P^o})\leq\rho_r} \mathbb{E}_{s\sim d^\pi_P}\left[\sum_a(\pi(a|s)-\pi'(a|s))Q^{\rm rob}_{\pi'}(s,a)\right].$$

Proof. We first define few useful notations for this proof. The *robust model* $P^{{\rm rob},\pi}$ for every π is as follows: $P^{{\rm rob},\pi}_{s,a} = \arg\min_{P_{s,a}\in\mathcal{P}_{s,a}} P^\top_{s,a}V^{\rm rob}_\pi, \quad (s,a)\in\mathcal{S}\times\mathcal{A}$. We call V^P_π as the value function for policy π under the model P. Now we can write $V^{\rm rob}_{\pi'} = V^{P^{{\rm rob},\pi'}}_{\pi'}$.

Fix $s_0\sim\mu$. For any π',π policies, we have

$$\begin{aligned}
&V^{\rm rob}_\pi(s_0) - V^{\rm rob}_{\pi'}(s_0) \overset{(a)}{\leq} V^{P^{{\rm rob},\pi'}}_\pi(s_0) - V^{\rm rob}_{\pi'}(s_0)\\
&\overset{(b)}{=} V^{P^{{\rm rob},\pi'}}_\pi(s_0) - V^{P^{{\rm rob},\pi'}}_{\pi'}(s_0)\\
&\overset{(c)}{=} \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi,P^{{\rm rob},\pi'}}_{s_0}}\left[\sum_a(\pi(a|s)-\pi'(a|s))Q^{\rm rob}_{\pi'}(s,a)\right],
\end{aligned}$$

where (a) follows since by definition of $V^{\rm rob}_\pi(s_0)$ we have $V^{\rm rob}_\pi(s_0) \leq V^{P^{{\rm rob},\pi'}}_\pi(s_0)$, and (b) follows from definition of $P^{{\rm rob},\pi'}$ yielding $V^{\rm rob}_{\pi'}(s_0) = V^{P^{{\rm rob},\pi'}}_{\pi'}(s_0)$. Observe $P^{{\rm rob},\pi'}\in\mathcal{P}$. Now taking expectation on $s_0\sim\mu$ with Lemma 7 completes the proof of this result. Now it only remains to show (c).

For (c), first denote $\mathcal{T}_{\pi,\pi'}(s) = (s_t,a_t)_{t\geq0}$ trajectory generated from rolling policy π from the initial state s_0 under the robust model $P^{{\rm rob},\pi'}$. Now,

$$\begin{aligned}
&V^{P^{{\rm rob},\pi'}}_\pi(s_0) - V^{P^{{\rm rob},\pi'}}_{\pi'}(s_0)\\
&= \mathbb{E}_{\mathcal{T}_{\pi,\pi'}(s)}[\sum_t\gamma^t r(s_t,a_t)] - V^{P^{{\rm rob},\pi'}}_{\pi'}(s_0)\\
&\overset{(d)}{=} \mathbb{E}_{\mathcal{T}_{\pi,\pi'}(s)}[\sum_t\gamma^t(r(s_t,a_t)+\gamma V^{P^{{\rm rob},\pi'}}_{\pi'}(s_{t+1}) - V^{\rm rob}_{\pi'}(s_t))]\\
&\overset{(e)}{=} \mathbb{E}_{\mathcal{T}_{\pi,\pi'}(s)}[\sum_t\gamma^t(Q^{\rm rob}_{\pi'}(s_t,a_t) - V^{\rm rob}_{\pi'}(s_t))]\\
&\overset{(f)}{=} \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d^{\pi',P^{{\rm rob},\pi'}}_{s_0}}\sum_{a'}\pi(a'|s')(Q^{\rm rob}_{\pi'}(s',a'))\\
&\qquad - \frac{1}{1-\gamma}\mathbb{E}_{s'\sim d^{\pi',P^{{\rm rob},\pi'}}_{s_0}}\sum_{a'}\pi'(a'|s')(Q^{\rm rob}_{\pi'}(s',a')),
\end{aligned}$$

where (d) follows by recursion, (e) follows since $Q^{\rm rob}_{\pi'}(s,a) = r(s,a)+\gamma(P^{{\rm rob},\pi'}_{s,a})^\top V^{P^{{\rm rob},\pi'}}_{\pi'}$, and (f) from $V^{\rm rob}_{\pi'}(s) = \sum_a\pi'(a|s)Q^{\rm rob}_{\pi'}(s,a)$. This proves (c). ☐

Proof of Theorem 3. We start by Lemma 8 with $\pi' = \pi_{\rm drbc}$ and $\pi = \pi_e$. We get

$$\begin{aligned}
&(1-\gamma)(V^{\rm rob}_{\pi_e} - V^{\rm rob}_{\pi'}) \leq\\
&\max_{P:D(d^{\pi_e}_P,d^{\pi_e}_{P^o})\leq\rho_r}\mathbb{E}_{s\sim d^{\pi_e}_P}[\sum_a(\pi_e(a|s)-\pi'(a|s))Q^{\rm rob}_{\pi'}(s,a)]\\
&\overset{(a)}{\leq} \frac{2}{1-\gamma}\max_{P\in\mathcal{M}}\mathbb{E}_{s\sim d^{\pi_e}_P}\left[D_{\rm TV}(\pi'(\cdot|s)\|\pi_e(\cdot|s))\right],
\end{aligned}$$

where (a) follows from Holder's inequality, from $D_{\rm TV}$ definition, and the fact that $\left\|Q^{\rm rob}_{\pi'}(s,\cdot)\right\|_\infty \leq 1/(1-\gamma)$ for any π'. Finally note $L_{\rm drbc}(\pi_{\rm drbc},\rho_r) = \varepsilon_{\rm drbc}(\rho_r)$. ☐

C. Proof of Theorem 4

Proof. Firstly, from Lemma 6, we observe that $L_{\rm drbc}(\hat{\pi}_{\rm drbc},\rho_c) \leq \widehat{\varepsilon}_{\rm drbc}(\rho_c)$ holds with probability at least $1-\delta$ with $\widehat{\varepsilon}_{\rm drbc}(\rho_c) = \varepsilon_{\rm drbc}(\rho_c) + c(1+\rho_c)\sqrt{\log(1/\delta)+2|\mathcal{A}|\log(N)/(\mathrm{e}_{\min}N)}$, where $\mathrm{e}_{\min} = \min_{s,a:\pi_e(a|s)>0}\pi_e(a|s)$ and $c > 0$ is some universal constant. The proof is now complete by following the analysis of Theorem 3 with $\hat{\pi}_{\rm drbc}$. ☐

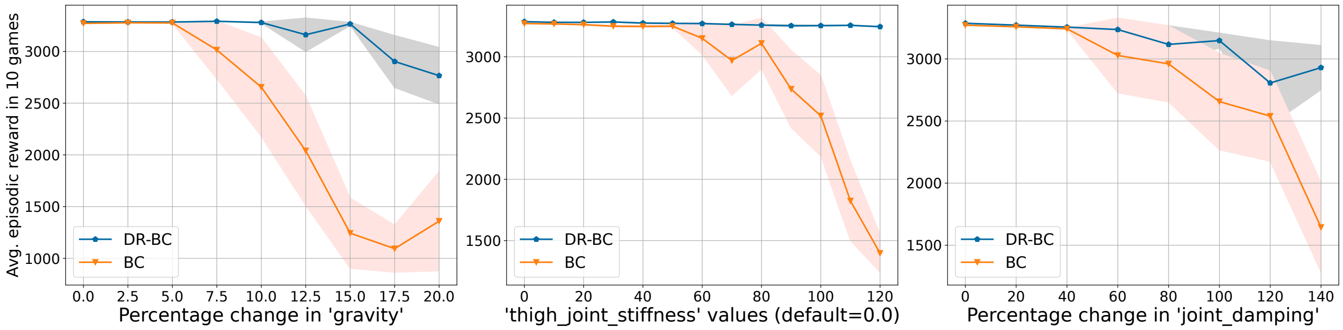

Fig. 1: `Hopper-v3` *perturbation results.* Average episodic reward on 10 differently seeded episodes. From left to right, the perturbations are in: *'gravity'*, *'joint_stiffness'* of the thigh joint, and *'joint_damping'* of all joints.

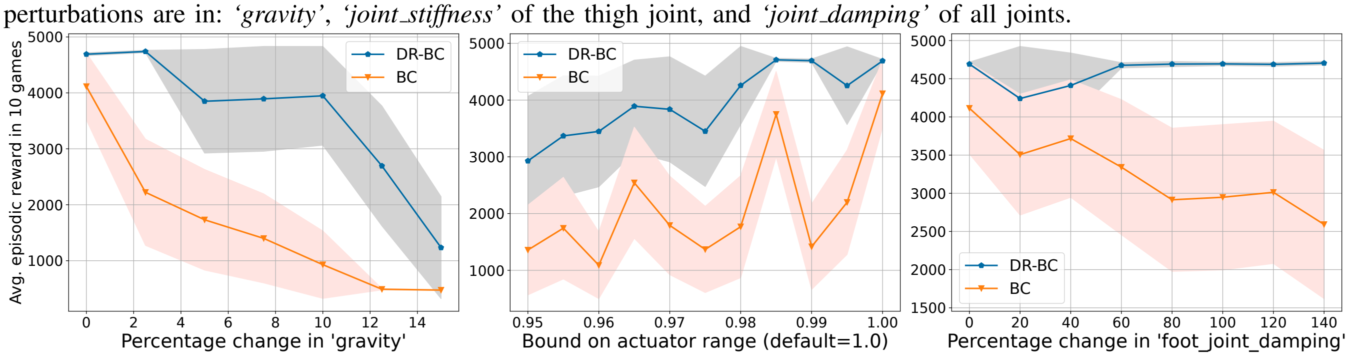

Fig. 2: `Walker2d-v3` *perturbation results.* Average episodic reward on 10 differently seeded episodes. From left to right, the perturbations are in: *'gravity'*, *'actuator_ctrlrange'* of all joints, and *'joint_damping'* of both foot joints.

IV. Experiments

We aim to answer the question: When model mismatches are present, is the DR-BC algorithm robust compared to the non-robust BC algorithm?

A. Experiment Setup and Practical Algorithm

We perform extensive simulations on two OpenAI Gym [37] environments simulated with MuJoCo physics engine [38]: `Hopper-v3` and `Walker2d-v3`. We train both the BC and DR-BC algorithms on the expert data generated by the pre-trained TD3 [46] policies from the RL Baselines3 Zoo repositories [47]. [48] pointed out that BC is very effective at imitating the expert when given large number of samples. Hence, like [10], [48], we give both BC and DR-BC the same but relatively low number of expert trajectories. *For reproducibility, we provide the code and more implementation details in the GitHub repository* `https://github.com/ferocious-cheetah/DRBC`.

Now we explain Algorithm 1 in detail. For simulations we use the loss l as the squared L2 loss, i.e., $l(\pi(s), \pi_e(s)) = \|\pi(s) - \pi_e(s)\|_2^2$. Now we specialize Equation (4) of Algorithm 1 by [26, Proposition 1]. At initialization, we need an expert dataset $\mathcal{D}_e$ of size N and some radius of our uncertainty set ρ_{r}. We also need to initialize a neural network π_θ which is our policy (actor) with random parameters θ. In each iteration, to solve Equation (4), we use the `SLSQP` method [49], a powerful minimization solver under the bounds prescribed in [26, Proposition 1]. In step 4, our policy (actor) is optimized based on the L_{drbc} loss using ADAM [50].

B. Test For Robustness

When the testing environment is perturbed, e.g., change in gravity, perturbed actuator and modified damping coefficient, model mismatches are present. Here we explain the simulation results on each of the four environments in details.

We perturb `Hopper-v3` by changing the model parameter *'gravity'*, *'thigh_joint_stiffness'*, and *'joint_damping'*. Figure 1 shows that DR-BC is tenacious under model perturbations. For example, in the middle figure, when the *'thigh_joint_stiffness'* parameter is positive and increasing, a joint spring is created in the thigh of hopper and becomes stiffer. A non-robust policy such as BC cannot withstand such mismatch between the training and testing environments. Meanwhile, our DR-BC agent refuses to drop in performance. In Figure 2 on `Walker2d-v3`, DR-BC still refuses to lose performance in wide ranges of perturbations on different environments.

V. Conclusion

In this paper, we introduce a novel problem of *robust imitation learning* to incorporate resiliency to changes in the real-world parameters. We present a novel approach to solve this problem, Distributionally Robust Behavioral Cloning (DR-BC) algorithm. Our proposed DR-BC algorithm utilizes the distributionally robust optimization (DRO) technique for BC to efficiently address robustness for the changes in real-world parameters.

While in this paper we only consider the total variation distance for the inner maximization, future work will explore using other types measures such as KL-divergence and Chi-square divergence. The same applies to the loss function considered in this work. We also plan to work on the scenario where the model is not known in large-scale problems using general function approximations.

References

[1] A. Kis, L. Huber, and A. Wilkinson, "Social learning by imitation in a reptile (pogona vitticeps)," *Animal cognition*, vol. 18, 09 2014.

[2] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, *et al.*, "The limits and potentials of deep learning for robotics," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[3] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, 2017.

[4] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810, IEEE, 2018.

[5] D. C. Guastella and G. Muscato, "Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review," *Sensors*, vol. 21, no. 1, p. 73, 2020.

[6] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems*, Morgan-Kaufmann, 1988.

[7] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Machine Intelligence 15*, 1995.

[8] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *13th international conference on artificial intelligence and statistics*, 2010.

[9] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *14th international conference on artificial intelligence and statistics*, 2011.

[10] K. Brantley, W. Sun, and M. Henaff, "Disagreement-regularized imitation learning," in *International Conference on Learning Representations*, 2019.

[11] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[12] T. Xu, Z. Li, and Y. Yu, "Error bounds of imitating policies and environments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15737–15749, 2020.

[13] M. A. Bashiri, B. Ziebart, and X. Zhang, "Distributionally robust imitation learning," *Advances in neural information processing systems*, vol. 34, pp. 24404–24417, 2021.

[14] G. N. Iyengar, "Robust dynamic programming," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 257–280, 2005.

[15] A. Nilim and L. El Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[16] N. Corporation, "Closing the sim2real gap with nvidia isaac sim and nvidia isaac replicator," 2021.

[17] H. Xu and S. Mannor, "Distributionally robust Markov decision processes," in *Advances in Neural Information Processing Systems*, pp. 2505–2513, 2010.

[18] W. Wiesemann, D. Kuhn, and B. Rustem, "Robust Markov decision processes," *Mathematics of Operations Research*, vol. 38, no. 1, pp. 153–183, 2013.

[19] P. Yu and H. Xu, "Distributionally robust counterpart in Markov decision processes," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2538–2543, 2015.

[20] S. Mannor, O. Mebel, and H. Xu, "Robust mdps with k-rectangular uncertainty," *Mathematics of Operations Research*, vol. 41, no. 4, pp. 1484–1509, 2016.

[21] R. H. Russel and M. Petrik, "Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps," *Advances in Neural Information Processing Systems*, 2019.

[22] W. Yang, L. Zhang, and Z. Zhang, "Towards theoretical understandings of robust markov decision processes: Sample complexity and asymptotics," 2021.

[23] K. Panaganti and D. Kalathil, "Sample complexity of robust reinforcement learning with a generative model," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

[24] Z. Xu*, K. Panaganti*, and D. Kalathil, "Improved sample complexity bounds for distributionally robust reinforcement learning," Conference on Artificial Intelligence and Statistics, 2023.

[25] K. Panaganti and D. Kalathil, "Robust reinforcement learning using least squares policy iteration with provable performance guarantees," in *International Conference on Machine Learning (ICML)*, pp. 511–520, 2021.

[26] K. Panaganti, Z. Xu, D. Kalathil, and M. Ghavamzadeh, "Robust reinforcement learning using offline data," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[27] Y. Wang and S. Zou, "Online robust reinforcement learning with model uncertainty," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7193–7206, 2021.

[28] J. Duchi and H. Namkoong, "Learning models with uniform performance via distributionally robust optimization," *arXiv preprint arXiv:1810.08750*, 2018.

[29] A. Shapiro, "Distributionally robust stochastic programming," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2258–2275, 2017.

[30] R. Gao and A. Kleywegt, "Distributionally robust stochastic optimization with wasserstein distance," *Mathematics of Operations Research*, 2022.

[31] D. Bertsimas, V. Gupta, and N. Kallus, "Data-driven robust optimization," *Math. Program.*, vol. 167, p. 235–292, feb 2018.

[32] H. Namkoong and J. C. Duchi, "Stochastic gradient methods for distributionally robust optimization with f-divergences," *Advances in neural information processing systems*, vol. 29, 2016.

[33] J. Blanchet, Y. Kang, and K. Murthy, "Robust wasserstein profile inference and applications to machine learning," *Journal of Applied Probability*, vol. 56, no. 3, p. 830–857, 2019.

[34] S. Meinecke, L. Thurner, and M. Braun, "Review of steady-state electric power distribution system datasets," *Energies*, vol. 13, no. 18, 2020.

[35] M. Travers, "10 best healthcare data sets & examples," 2021.

[36] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset," *arXiv*, 2021.

[37] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[38] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

[39] N. Rajaraman, L. Yang, J. Jiao, and K. Ramchandran, "Toward the fundamental limits of imitation learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2914–2924, 2020.

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

[41] K. Panaganti and D. Kalathil, "Sample complexity of robust reinforcement learning with a generative model," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 9582–9602, 2022.

[42] R. Chen, I. C. Paschalidis, *et al.*, "Distributionally robust learning," *Foundations and Trends® in Optimization*, vol. 4, no. 1-2, pp. 1–243, 2020.

[43] T. M. Cover and J. A. Thomas, "Information theory and the stock market," *Elements of Information Theory. Wiley Inc., New York*, pp. 543–556, 1991.

[44] A. L. Gibbs and F. E. Su, "On choosing and bounding probability metrics," *International statistical review*, vol. 70, no. 3, pp. 419–435, 2002.

[45] A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun, "Reinforcement learning: Theory and algorithms," *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 2019.

[46] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*, pp. 1582–1591, 2018.

[47] A. Raffin, "Rl baselines3 zoo." `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020.

[48] J. Chang, M. Uehara, D. Sreenivas, R. Kidambi, and W. Sun, "Mitigating covariate shift in imitation learning via offline data with partial coverage," *Advances in Neural Information Processing Systems*, vol. 34, pp. 965–979, 2021.

[49] D. Kraft, "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[50] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.