A Conservative Approach for Few-Shot Transfer in Off-Dynamics Reinforcement Learning

Paul Daoudi[1], **Christophe Prieur**[2], **Bogdan Robu**[2], **Merwan Barlier**[1] and **Ludovic Dos Santos**[3]
[1]Huawei Noah's Ark Lab
[2]GIPSA Lab
[3]Criteo AI Lab
{paul.daoudi1, merwan.barlier}@huawei.com, l.dossantos@criteo.com,
{christophe.prieur, bogdan.robu}@gipsa-lab.grenoble-inp.fr

Abstract

Off-dynamics Reinforcement Learning (ODRL) seeks to transfer a policy from a source environment to a target environment characterized by distinct yet similar dynamics. In this context, traditional RL agents depend excessively on the dynamics of the source environment, resulting in the discovery of policies that excel in this environment but fail to provide reasonable performance in the target one. In the few-shot framework, a limited number of transitions from the target environment are introduced to facilitate a more effective transfer. Addressing this challenge, we propose a new approach inspired by recent advancements in Imitation Learning and conservative RL algorithms. The proposed method introduces a penalty to regulate the trajectories generated by the source-trained policy. We evaluate our method across various environments representing diverse off-dynamics conditions, where access to the target environment is extremely limited. Across most tested scenarios, our proposed method demonstrates performance improvements compared to existing baselines.

1 Introduction

Traditional online Reinforcement Learning (RL) is a promising path to obtain a near-optimal policy for complex systems within a general trial-and-error framework. However, this standalone mechanism faces numerous challenges [Dulac-Arnold *et al.*, 2021] when applied to real-world systems, especially in scenarios where interactions can be prohibitively expensive due to safety [Garcıa and Fernández, 2015; Achiam *et al.*, 2017] or time [Weiss *et al.*, 2016] considerations. In such context, one alternative is to leverage a cheap source environment, usually built as a simplification of the target environment. While these environments may also differ w.r.t. their observations [Gamrian and Goldberg, 2019] and rewards [Barreto *et al.*, 2017], we study the off-dynamics setting where the mismatch is between their dynamics [Höfer *et al.*, 2021]. It is a relevant consideration when it is possible to estimate or simplify the physics of the target environment, for instance within a simulator. We also consider that the transition probabilities of both the source and target environments are unavailable to reflect any kind of simulator.

The off-dynamics framework is particularly challenging: direct policy transfer from the source to the target environment usually fails [Muratore *et al.*, 2019; Ju *et al.*, 2022] due to compounding errors. The small discrepancies between the transition probabilities may accumulate over time, leading to increasing deviations between the trajectories of the source and target environments over time. Worse, modern optimization-based agents may exploit these discrepancies to find policies that perform exceptionally well in the source but result in trajectories that are impossible to replicate in the target.

In general, in addition to relying on a source environment, it is still possible to deploy the agent in the target system to collect data. However, this deployment is limited due to the mentioned safety and time considerations: the available data is this often limited to a few narrow trajectories. Two orthogonal approaches are possible to include this data in the derivation of the policy. The first one - well studied [Abbeel *et al.*, 2006; Zhu *et al.*, 2018; Desai *et al.*, 2020; Hanna *et al.*, 2021] - leverages this data to improve the source domain, then learns a traditional RL agent on the upgraded system. The second approach maintains the source environment fixed and biases the learning process to account for the dynamics discrepancies [Koos *et al.*, 2012]. This second line of work is complementary to the first one, as both could be combined to make best use of the limited target samples. To the best of our knowledge, only a few works have taken this purely off-dynamics direction, and even fewer have focused on the low data regime scenario. Currently, the most prominent approach is DARC [Eysenbach *et al.*, 2021] which modifies the reward function to search for parts of the source that behave similarly to the target system. Although this method is effective for a few classes of problems, such as the "broken" environments, we have found that it may fail in others, limiting its application to a restrictive class of discrepancies between the environments.

In this paper, we introduce the **F**ew-sh**O**t **O**ff **D**ynamics (FOOD) algorithm, a conservative method that penalizes the derived policy to be around the trajectories observed in the target environment. We theoretically justify this method, which directs the policy towards feasible trajectories in the target system, and thus mitigates the potential trajectory shifts towards untrustworthy regions of the source system. Our regularization takes the form of a divergence between visitation distributions and can be practically implemented using state-of-the-art techniques from the Imitation Learning (IL) literature [Hussein *et al.*, 2017]. Our method is validated on a set of environments with multiple off-dynamics disparities. We show that, compared to other baselines, our approach is the most successful in exploiting the few available data. Our agent is also shown to be relevant for a wider range of dynamic discrepancies.

2 Related Work

The off-dynamics setting has been studied in two distinct contexts, depending on the accessibility of the agent to transitions from the target environment, referred to as "zero-shot" and "few-shot" off-dynamics RL.

Zero-Shot Off-Dynamics RL. Sampling data from the target environment can be impossible due to strict safety constraints or time-consuming interactions. In such cases, the source environment is used to ensure robustness to guarantee a certain level of performance without sampling from the target system. It can take many forms. One possible choice is domain randomization [Mordatch *et al.*, 2015] where relevant parts of the source system are randomized to make it resilient to changes. Another line of work focuses on addressing the worst-case scenarios under stochastic source dynamics [Abdullah *et al.*, 2019]. Robustness can also be achieved w.r.t. actions [Jakobi *et al.*, 1995; Tessler *et al.*, 2019], that arise when certain controllers become unavailable in the target environment. These techniques are outside the scope of this paper as they do not involve any external data in the learning process.

Few-Shot Off-Dynamics RL. When data can be sampled from the target environment, two orthogonal approaches have been developed to propose efficient agents. The first approach, well established, is to improve the accuracy of the source environment. The parameters of the source physics can be optimized directly if available [Zhu *et al.*, 2018; Tan *et al.*, 2018]. Otherwise, expressive models can be introduced to improve the source dynamics model [Abbeel *et al.*, 2006]. Within this category, a family of methods builds an action transformation mechanism that - when taken in the source system - produces the same transition that would have occurred in the target environment [Hanna *et al.*, 2021]. In particular, GARAT [Desai *et al.*, 2020] leverages recent advances in Imitation Learning from Observations [Torabi *et al.*, 2019] to learn this action transformation and ground the source environment with only a few trajectories. All these algorithms are orthogonal to our work since once the source system has been improved, a new RL agent has to be trained.

The second approach, more closely aligned with our work, is the line of inquiry that modifies the learning process of the RL policy in the source to be efficient in the target environment. One group of approaches creates a policy - or policies - that can quickly adapt to a variety of dynamic conditions [Arndt *et al.*, 2020; Yu *et al.*, 2020; Kumar *et al.*, 2021]. It requires the ability to set the parameters of the source dynamics model which may not always be feasible, e.g., if the model is a black box. A more general algorithm is DARC [Eysenbach *et al.*, 2021]. It learns two classifiers to distinguish transitions between the source and target environments and incorporates them into the reward function to account for the dynamics shift. Learning the classifiers is easier than correcting the dynamics of the source environment, but as we will see in the experiments, this technique seems to work mainly when some regions of the source environment accurately model the target environment and others don't. Another related work is H2O [Niu *et al.*, 2022b] which extends the approach by considering access to a fixed dataset of transitions from a target environment. It combines the regularization of the offline algorithm CQL [Kumar *et al.*, 2020] with the classifiers proposed by DARC. However, the performance of H2O depends on the amount of data available. In fact, it performed similarly, or worse, to the pure offline algorithm when only a small amount of target data was available [Niu *et al.*, 2022b, Appendix C.3].

3 Background

3.1 Preliminaries

Let $\Delta(\cdot)$ be the set of all probability measures on $(\cdot)$. The agent-environment interaction is modeled as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, r, P, \gamma, \rho_0)$, with a state space $\mathcal{S}$, an action space $\mathcal{A}$, a transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [R_{\min}, R_{\max}]$, the initial state distribution ρ_0 and a discount factor $\gamma \in [0, 1)$. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a decision rule mapping a state over a distribution of actions. The value of a policy π is measured through the value function $V_P^\pi(s) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) | s_0 = s\right]$. The objective is to find the optimal policy maximizing the expected cumulative rewards $J_P^\pi = \mathbb{E}_{\rho_0}\left[V_P^\pi(s)\right]$. We also define the Q-value function $Q_P^\pi(s, a) = \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) | s_0 = s, a_0 = a\right]$ and the advantage value function $A_P^\pi(s, a) = Q_P^\pi(s, a) - V_P^\pi(s)$. Finally, let $d_P^\pi(s) = (1 - \gamma)\mathbb{E}_{\rho_0,\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s)\right]$ the state visitation distribution, as well as its extension to state-action $\mu_P^\pi(s, a)$ and transition $\nu_P^\pi(s, a, s')$. All these quantities are expectations w.r.t. both the policy and the transition probabilities.

The off-dynamics setting involves two MDPs: the source system $\mathcal{M}_{\text{s}}$ and the target environment $\mathcal{M}_{\text{t}}$. We hypothesize that they are identical except for their transition probabilities $P_{\text{s}} \neq P_{\text{t}}$. Our hypothesis states that while most of the MDP parameters are known, the underlying physics of the environment are only estimated. It encapsulates many real-world applications: a model of the dynamics may have been previously learned, or practitioners may have created a simulator based on a simplification of the system's physics. We do not assume access to any parameter modifying the transition probabilities P_{s} of the source physics to encompass black-box simulators. For readability purposes, we drop the P subscripts for the value functions as they are always associated with the source environment $\mathcal{M}_{\text{s}}$.

Many few-shot off-dynamics agents [Abbeel *et al.*, 2006;

Desai *et al.*, 2020; Hanna *et al.*, 2021] typically employ the following procedure to handle complex environments. First, the policy and value functions are initialized in the source system. The choice of objective at this stage can vary, although a classical approach is to solely maximize the rewards of the source MDP. At each iteration, the policy is verified by experts. If it is deemed safe, N trajectories are gathered in the target environment and saved in a replay buffer $\mathcal{D}_{\mathrm{t}}$. These trajectories are then used to potentially correct the source dynamics and/or the training objective and induce a new policy. This process is repeated until a satisfactory policy is found. This setup is time-consuming and may be risky even when the policy is verified by experts, hence the need to learn with as few data as possible from the target environment.

This work focuses on how to best modify the objective with few trajectories. If handled properly, this could reduce the number of interactions required by the whole process overcoming the need to build a perfect source environment. For the purpose of our study, we assume that $\mathcal{M}_{\mathrm{t}}$ remains fixed throughout the process.

3.2 Conservative Algorithms and the Visitation Distribution Constraint

Due to their efficiency and stability, conservative algorithms [Schulman *et al.*, 2015; Schulman *et al.*, 2017; Kumar *et al.*, 2020] have shown strong efficiency in various RL settings. Many of them are based on [Kakade and Langford, 2002], where an iteration scheme improves the policy by maximizing a lower bound on the true objective. This process has been extended by refining the lower bound [Terpin *et al.*, 2022; Moskovitz *et al.*, 2021], for example by introducing a behavior b_P^π that encapsulates any additional property of the MDP [Pacchiano *et al.*, 2020; Touati *et al.*, 2020].

This family of algorithms is formalized as follows. A policy and a value function are parametrized with respective weights $\theta \in \Theta$ and $\omega \in \Omega$, that we denote from now on π_θ and $V_\omega^{\pi_\theta}$. At each iteration k, the policy is improved using the advantage function built from the approximated value function $V_{\omega_k}^{\pi_{\theta_k}}$, with a penalization to respect the lower bound:

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\substack{s \sim d_{P_{\mathrm{s}}}^{\pi_{\theta_k}}(\cdot), \\ a \sim \pi_\theta(\cdot|s)}} \left[A_{\omega_k}^{\pi_{\theta_k}}(s,a) \right] - \alpha_k \, D\Big(b_P^{\pi_\theta} \, \Big\| \, b_P^{\pi_{\theta_k}} \Big), \tag{1}$$

where D is any kind of similarity metric and α_k is a hyperparameter, often set to a constant α. TRPO [Schulman *et al.*, 2015] can be retrieved with $b_P^\pi = \pi$, by setting D to be the Kullback-Leibler (KL) divergence and enforcing the penalization as a constraint with a step-size ϵ_k. Alternative behavior options can be found in [Pacchiano *et al.*, 2020; Touati *et al.*, 2020; Moskovitz *et al.*, 2021]. In particular, [Touati *et al.*, 2020] proposed to encapsulate the whole trajectories induced by π and P by setting $b_P^\pi = d_P^\pi$. It resulted in better results both in terms of sample efficiency and final cumulative rewards than most of its counterparts. This is natural as the new constraint between the state visitation distributions takes the whole trajectories induced by the policy into account, providing more information than the policy alone.

4 Few-Shot Off-Dynamics Reinforcement Learning

In this section, we propose a new objective to better transfer a policy learned in the source to the target environment. We extend the conservative objective to the off-dynamics setting. Then, we remind necessary results on Imitation Learning (IL) before deriving our practical algorithm **F**ew-sh**O**t **O**ff **D**ynamics (FOOD) RL.

4.1 A New Conservative Off-Dynamics Objective

Given the discrepancies between the dynamics of the source and the target environment, applying the same policy to both environments may result in different trajectories. This poses a challenge as the agent may make the most of these differences to find policies that produce excellent trajectories in the source environment but are impossible to replicate in the target system.

We analyze the difference between the objectives J_P^π associated with the target and source environments, depending on a metric between visitation distributions. For this, we first apply directly the tools from traditional conservative methods [Schulman *et al.*, 2015; Achiam *et al.*, 2017] to the off-dynamics setting, and propose the following lower bound using state visitation distributions.

Proposition 1. *Let $J_P^\pi = \mathbb{E}_{\rho_0}\left[V_P^\pi(s)\right]$ the expected cumulative rewards associated with policy π, transitions P and initial state distribution ρ_0. For any policy π and any transition probabilities P_t and P_s, the following holds:*

$$J_{P_t}^\pi \geq J_{P_s}^\pi - \frac{2R_{max}}{1-\gamma}\left(D_{TV}\big(d_{P_s}^\pi, d_{P_t}^\pi\big) + D_{TV}^\pi(P_s, P_t) \right), \tag{2}$$

with D_{TV} the Total Variation distance and $D_{TV}^\pi(P_s, P_t) = \mathbb{E}_{\substack{s \sim d_{P_s}^\pi(\cdot), \\ a \sim \pi(\cdot|s)}}\left[D_{TV}(P_s(\cdot|s,a), P_t(\cdot|s,a))\right]$.

The Proposition also holds by replacing $D_{\mathrm{TV}}\big(d_{P_{\mathrm{s}}}^\pi, d_{P_{\mathrm{t}}}^\pi\big)$ with $D_{\mathrm{TV}}\big(\mu_{P_{\mathrm{s}}}^\pi, \mu_{P_{\mathrm{t}}}^\pi\big)$. We defer the proof to Appendix A. It illustrates how the performance of the optimal policy in the target environment may differ from that of the source due to two metrics. The first metric $D_{\mathrm{TV}}\big(d_{P_{\mathrm{s}}}^\pi, d_{P_{\mathrm{t}}}^\pi\big)$ quantifies the difference between the visited states of the rollouts in the source and target environments. The second $D_{\mathrm{TV}}\big(d_{P_{\mathrm{s}}}^\pi, d_{P_{\mathrm{t}}}^\pi\big)$ describes the difference between the transition probabilities associated with the visited states and the actions following the given policy. These terms must be controlled to allow a good transfer, especially given that they are exacerbated by the factor $\frac{2R_{\max}}{1-\gamma}$. However, optimizing the second metric is difficult since the transition probabilities of both $\mathcal{M}_{\mathrm{t}}$ and $\mathcal{M}_{\mathrm{s}}$ are unknown. Hence, we propose the following simpler lower bound that considers transition visitation distributions instead.

Proposition 2. *Let $J_P^\pi = \mathbb{E}_{\rho_0}\left[V_P^\pi(s)\right]$ the expected cumulative rewards associated with policy π, transitions P and initial state distribution ρ_0. For any policy π and any transition probabilities P_t and P_s, the following holds:*

$$J_{P_t}^\pi \geq J_{P_s}^\pi - \frac{2R_{max}}{1-\gamma} D_{TV}\big(\nu_{P_s}^\pi, \nu_{P_t}^\pi\big), \tag{3}$$

with D_{TV} the Total Variation distance.

We also defer the proof to Appendix A. Here, the lower bound depends on the sole metric $D_{\text{TV}}\left(\nu^{\pi}_{P_s}, \nu^{\pi}_{P_t}\right)$ that directly quantifies the difference in trajectories. As we will see, this term is easily minimized, especially given that the Total Variation distance could be replaced by other divergences. For instance, the Kullback-Leibler divergence or the Jensen-Shannon divergence could be used thanks to Pinsker's inequality [Csiszar and Körner, 1981] or the one in [Corander *et al.*, 2021, Proposition 3.2], provided the minimal assumptions of having a finite state-action space and the absolute continuity of the considered measures. Complete details can be found in Appendix A.

Overall, this lower bound highlights a good transfer between the source and target environment is possible when $D_{\text{TV}}\left(\nu^{\pi}_{P_s}, \nu^{\pi}_{P_t}\right)$ is small, as the policy induces similar objectives J^{π}_{P}. Inspired by this insight, we adapt conservative methods to the off-dynamics setting and propose a new regularization between trajectories by setting the behaviors b^{π}_{P} to be the transition visitation distribution respectively associated with the transition probabilities of the source and target environment ν^{π}_{P}:

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\substack{s \sim d^{\pi_{\theta_k}}_{P_s}(\cdot), \\ a \sim \pi_\theta(\cdot|s)}} \left[A^{\pi_{\theta_k}}_{\omega_k}(s,a)\right] - \alpha\, D\left(\nu^{\pi_\theta}_{P_s} \,\middle\|\, \nu^{\pi_{\theta_k}}_{P_t}\right). \tag{4}$$

The new penalization ensures that the policy is optimized for trajectories that are feasible in the target system, thus preventing the RL agent from exploiting any potential hacks that may exist in the source environment. In addition, remaining close to the data sampled from the target environment can be beneficial when the source system has been constructed using that data, as querying out-of-distribution data can yield poor results [Kang *et al.*, 2022].

Unfortunately, the difference between the transition probabilities makes the regularization in Equation 4 difficult to compute. The previous work of [Touati *et al.*, 2020] addressed this by restricting D to f-divergences $D_f\left(\mu^{\pi_\theta}_{P} || \mu^{\pi_{\theta_k}}_{P}\right) = \mathbb{E}_{(s,a)\sim\pi_\theta}\left[f\left(\frac{\mu^{\pi_\theta}_{P}}{\mu^{\pi_{\theta_k}}_{P}}\right)\right]$ and by considering state-action visitation distributions. [Touati *et al.*, 2020] used the DualDICE algorithm [Nachum *et al.*, 2019] to directly estimate the relaxed ratio $\frac{\mu^{\pi_\theta}_{P}}{\mu^{\pi_{\theta_k}}_{P}}$ for any policy π_θ sufficiently close to π_{θ_k}, eliminating the need to sample data for each policy. However, this method is not applicable to our setting because DualDICE relies on a modified joined Bellman operator, which assumes that both distributions follow the same transition probabilities. Another solution would be to collect at least one trajectory per update. While this would not pose any safety concerns for the data would be sampled in the source system, it can be time-consuming in practice.

4.2 Practical Algorithm

In order to devise a practical algorithm for addressing Equation 4, we aim to get a surrogate objective for $D\left(\nu^{\pi_\theta}_{P_s} \,\middle\|\, \nu^{\pi_{\theta_k}}_{P_t}\right)$. To construct such proxy, we leverage the recent results from Imitation Learning (IL) [Hussein *et al.*, 2017] that we briefly recall in this paragraph. In this field, the agent aims to reproduce an expert policy π_e using limited data sampled by that expert in the same MDP with generic transition probabilities P. Most current algorithms tackle this problem by minimizing a certain similarity metric D between the learning policy's state-action visitation distribution $\mu^{\pi_\theta}_{P}$ and the expert's $\mu^{\pi_e}_{P}$. The divergence minimization problem is transformed into a reward r_{imit} maximization one, resulting in an imitation value function $V^{\pi}_{\text{imit}} = \mathbb{E}_{\pi,P_s}\left[\sum_{t=0}^{\infty}\gamma^t r_{\text{imit}}(s_t,a_t,s_{t+1})|s_0 = s\right]$. Since these algorithms are based on data, they can be used to minimize the chosen similarity metric D between two transition visitation distributions with different transition probabilities. Applied to our setting where the target trajectories would be the expert ones, this is formalized as:

$$\arg\max_{\theta\in\Theta} V^{\pi_\theta}_{\text{imit}} = \arg\min_{\pi} D\left(\nu^{\pi_\theta}_{P_s} \,\middle\|\, \nu^{\pi_{\theta_k}}_{P_t}\right). \tag{5}$$

The choices for the divergence D are numerous, leading to different IL algorithms [Ho and Ermon, 2016; Fu *et al.*, 2017; Dadashi *et al.*, 2021], some of which are summarized in Table 1. For example, Equation 5 is exactly Theorem 1 in [Desai *et al.*, 2020] in association with GAIL [Ho and Ermon, 2016], and is also straigthforward with PWIL [Dadashi *et al.*, 2021].

GAIL	AIRL	PWIL
$D_{\text{JS}}\left(X^{\pi_\theta}_{P_s} \,\middle\|\, X^{\pi_{\theta_k}}_{P_t}\right)$	$D_{\text{KL}}\left(X^{\pi_\theta}_{P_s} \,\middle\|\, X^{\pi_{\theta_k}}_{P_t}\right)$	$D_{\text{W}}\left(X^{\pi_\theta}_{P_s}, X^{\pi_{\theta_k}}_{P_t}\right)$

Table 1: Objective function for well-known Imitation Learning (IL) algorithms. The variable X can be chosen as either d, μ, or ν. Other IL agents can be found in [Ghasemipour *et al.*, 2020].

These IL techniques enable efficient estimation of this value function using a small number of samples from $\nu^{\pi_{\theta_k}}_{P_t}$ and unlimited access to $\mathcal{M}_s$. Let $\xi \in \Xi$ be the weights of this parametrized value function. The new regularization is $A^{\pi_{\theta_k},\xi_k}_{\text{imit}}(s,a)$, which can be learned with any suitable IL algorithm. It leads to the practical objective:

$$\max_{\theta \in \Theta} \quad \mathbb{E}_{\substack{s \sim d^{\pi_{\theta_k}}_{P_s}(\cdot), \\ a \sim \pi_\theta(\cdot|s)}} \left[A^{\pi_{\theta_k}}_{\omega_k}(s,a) - \alpha\, A^{\pi_{\theta_k},\xi_k}_{\text{imit}}(s,a)\right]. \tag{6}$$

This new agent is quite generic as it could be optimized with different divergences. It takes as input an online RL algorithm [Babaeizadeh *et al.*, 2016; Schulman *et al.*, 2017] denoted $\mathcal{O}$ and an Imitation Learning algorithm denoted $\mathcal{I}$. The whole off-dynamics algorithm process, which we denote Few-shOt Off Dynamics (FOOD) RL, is described as follows. First, the policy and the value weights are initialized in the source environment with $\mathcal{O}$. At each iteration k, the agent samples N new trajectories with π_{θ_k}[1]. Subsequently, the policy, traditional, and imitation value functions are retrained on the source environment with $\mathcal{O}$ and $\mathcal{I}$ according to Equation 4. The whole algorithm is summarized in Algorithm 1.

[1]These trajectories could first be used to improve the source system.

Algorithm 1 Few-shOt Off Dynamics (FOOD).

Input: Algorithms $\mathcal{O}$ and $\mathcal{I}$
Initialize policy and value weights θ_0 and ω_0 with $\mathcal{O}$
Randomly initialize the weights ξ_0
for $k \in (0, \ldots, K-1)$ **do**
  Gather N trajectories $\{\tau_i, \ldots, \tau_N\}$ with π_{θ_k} on the target environment $\mathcal{M}_t$ and add them in $\mathcal{D}_\mathrm{t}$
  Learn the value function weights ω_{k+1} with $\mathcal{O}$ in the source environment $\mathcal{M}_\mathrm{s}$
  Learn the imitation value function weights ξ_{k+1} with $\mathcal{I}$ in $\mathcal{M}_\mathrm{s}$ using $\mathcal{D}_\mathrm{t}$
  Learn the policy maximizing (6) using $\mathcal{D}_\mathrm{t}$ and $\mathcal{M}_\mathrm{s}$ with $\mathcal{O}$
end for

5 Experiments

In this section, we evaluate the performance of the FOOD algorithm in the off-dynamics setting in environments presenting different dynamics discrepancies, treated as black box simulators. The code can be found at https://github.com/PaulDaoudi/FOOD.

The environments are based on Open AI Gym [Brockman *et al.*, 2016] and the Minitaur environment [Coumans and Bai, 2016 2021] where the target environment has been modified by various mechanisms. These include gravity, friction, and mass modifications, as well as broken joint(s) systems for which DARC is known to perform well [Eysenbach *et al.*, 2021, Section 6]. We also add the Low Fidelity Minitaur environment, highlighted in previous works [Desai *et al.*, 2020; Yu *et al.*, 2018] as a classical benchmark for evaluating agents in the off-dynamics setting. In this benchmark, the source environment has a linear torque-current relation for the actuator model, and the target environment - proposed by [Tan *et al.*, 2018] - uses accurate non-linearities to model this relation.

All of FOOD experiments are carried out using both GAIL [Ho and Ermon, 2016], a state-of-the-art IL algorithm, as $\mathcal{I}$. We find that GAIL performed similarly, or better than other IL algorithms such as AIRL [Fu *et al.*, 2017] or PWIL [Dadashi *et al.*, 2021]. FOOD is tested with its theoretically motivated metric between transition visitation distributions ν_P^π, as well as with d_P^π and μ_P^π for empirically analyzing the performance associated with the different visitation distributions. The performance of our agent with the different IL algorithms can be found in Appendix C.5. We compare our approach against various baselines modifying the RL objective, detailed below. They cover current domain adaptation, robustness, or offline Reinforcement Learning techniques applicable to our setting. Further details of the experimental protocol can be found in Appendix C. Especially, a study with respect to the number of available target data is done in Appendix C.6.

- **DARC** [Eysenbach *et al.*, 2021] is our main baseline. It is a state-of-the-art off-dynamics algorithm that introduces an importance sampling term in the reward function to cope with the dynamics shift. In practice, this term is computed using two classifiers that distinguish transitions from the source and the target environment. In this agent, an important hyper-parameter is the standard deviation σ_{DARC} of the centered Gaussian noise injected into the training data to stabilize the classifiers [Eysenbach *et al.*, 2021, Figure 7]. We draw inspiration from the open-source code [Niu *et al.*, 2022a].
- **Action Noise Envelope (ANE)** [Jakobi *et al.*, 1995] is a robust algorithm that adds a centered Gaussian noise with standard deviation σ_{ANE} to the agent's actions during training. Although simple, this method outperformed other robustness approaches in recent benchmarks [Desai *et al.*, 2020] when the source environment is a black box.
- **CQL** [Kumar *et al.*, 2020] is a purely offline RL algorithm that learns a policy using target data. It does not leverage the source system in its learning process and thereby serves as a lower bound to beat. This algorithm inserts a regularization into the Q-value functions, with a strength β. We use [Geng, 2021] to run the experiments.
- **H2O** [Niu *et al.*, 2022b] is another off-dynamics algorithm that leverages data from the target system. It combines the classifiers from DARC to the CQL regularization. Similarly to FOOD, the agent is encentisized to stay close to the target data, but does so with a combination of DARC and CQL regularization. We also use [Niu *et al.*, 2022a] to run these experiments. However, this method performed poorly, which was expected considering it was proposed for a setting where a large amount of target data is available. Hence, its results are omitted from Table 2 and deferred in Appendix C.9.
- We also consider two RL agents, $\mathbf{RL_s}$ trained solely on the source system (without access to any target data) and $\mathbf{RL_t}$ trained solely on the target environment. Both algorithms were trained to convergence. Even though the latter baselines do not fit in the off-dynamics setting they give a rough idea of how online RL algorithms would perform in the target environment. The online RL algorithm $\mathcal{O}$ depends on the environment: we use A2C [Babaeizadeh *et al.*, 2016] for Gravity Pendulum and PPO [Schulman *et al.*, 2017] for the other environments.

Experimental Protocol. Our proposed model and corresponding off-dynamics/offline baselines require a batch of target data. To provide such a batch of data, we first train a policy and a value function of the considered RL agent until convergence by maximizing the reward on the source environment. After this initialization phase, 5 trajectories are sampled from the target environment to fit the restricted target data regime. They correspond to 500 data points for Pendulum and 1000 data points for the other environments. If some trajectories perform poorly in the target environment, we remove them for FOOD, DARC, and CQL to avoid having a misguided regularization. FOOD, DARC, and ANE are trained for 5000 epochs in the source environment and were optimized with the same underlying agent. Both $\mathrm{RL_s}$ and $\mathrm{RL_t}$ are trained until convergence. CQL is trained for 100000 gradient updates for Gravity Pendulum and 500000 gradient updates for all other environments. Additional details can be found in Appendix C.

Environment	RL_s	RL_t	CQL	ANE	DARC	FOOD (Ours)		
						μ_P^π	d_P^π	ν_P^π
Gravity Pendulum	-1964 ± 186	-406 ± 22	-1683 ± 142	-2312 ± 11	-3511 ± 865	-2224 ± 43	$\mathbf{-485 \pm 54}$*	-2327 ± 14
Broken Joint Cheetah	1793 ± 1125	5844 ± 319	143 ± 104	3341 ± 132	2501 ± 211	3801 ± 155	$\mathbf{3888 \pm 201}$	$\mathbf{3921 \pm 85}$*
Broken Joint Minitaur	7.4 ± 4.1	20.8 ± 4.8	0.25 ± 0.09	7.8 ± 6	12.6 ± 1.12	$\mathbf{14.9 \pm 3}$	$\mathbf{13.6 \pm 3.8}$	$\mathbf{16.9 \pm 4.7}$*
Heavy Cheetah	3797 ± 703	11233 ± 1274	41 ± 34	$\mathbf{7443 \pm 330}$*	4165 ± 257	4876 ± 181	4828 ± 553	4519 ± 240
Broken Joint Ant	5519 ± 876	6535 ± 352	1042 ± 177	3231 ± 748	5446 ± 162	$\mathbf{6145 \pm 98}$*	5547 ± 204	$\mathbf{6135 \pm 122}$
Friction Cheetah	1427 ± 674	9455 ± 3554	-466.4 ± 13	$\mathbf{6277 \pm 1405}$*	3302 ± 591	3690 ± 1495	3212 ± 2279	3289 ± 236
Low Fidelity Minitaur	8.9 ± 5.8	27.1 ± 8	10.2 ± 1	6.4 ± 3	3.2 ± 1.8	$\mathbf{17 \pm 2}$	15.7 ± 2.8	$\mathbf{17.6 \pm 0.4}$*
Broken Leg Ant	1901 ± 981	6430 ± 451	830 ± 8	2611 ± 220	2305 ± 175	2652 ± 356	2345 ± 806	$\mathbf{2977 \pm 85}$*
Median NAR and std	$0\ ;\ 0.25$	$1\ ;\ 0.26$	$-0.32\ ;\ 0.02$	$0.1\ ;\ 0.11$	$0.04\ ;\ 0.04$	$\mathbf{0.37\ ;\ 0.09}$*	$\mathbf{0.29\ ;\ 0.17}$	$\mathbf{0.36\ ;\ 0.03}$

Table 2: Returns over 4 seeds for CQL, RL_s, RL_t and ANE, and 8 seeds for the other agents on the tested environments. The best agent w.r.t. the mean is highlighted with boldface and an asterisk. We perform an unpaired t-test with an asymptotic significance of 0.1 w.r.t. the best performer and highlight with boldface the ones for which the difference is not statistically significant.

Hyper-parameters Optimization. We optimize the hyper-parameters of the evaluated algorithms through a grid search for each different environment. Concerning DARC and ANE, we perform a grid search over their main hyper-parameter $\sigma_{\text{DARC}} \in \{0.0, 0.1, 0.5, 1\}$ and $\sigma_{\text{ANE}} \in \{0.1, 0.2, 0.3, 0.5\}$. The remaining hyper-parameters were set to their default values according to the experiments reported in the open-source code [Niu *et al.*, 2022a]. For CQL, we perform a grid search over the regularization strength $\beta \in \{5, 10\}$, otherwise we keep the original hyper-parameters of [Geng, 2021]. For RL_t and RL_s we used the default parameters specific to each environment according to [Kostrikov, 2018] and trained them over 4 different seeds. We then selected the seed with the best performance in the source environment. For our proposed algorithm FOOD, the regularization strength hyper-parameter α is selected over a grid search depending on the underlying RL agent, $\alpha \in \{0, 1, 5, 10\}$ for A2C and $\alpha \in \{0.5, 1, 2, 5\}$ for PPO. This difference in choice is explained by the fact that the advantages are normalized in PPO, giving a more refined control over the regularization weight.

Results. We monitor the evolution of the agents' performance by evaluating their average return in the target environment during training. We do not gather nor use the data from those evaluations in the learning process since we work under a few-shot framework. The return of all methods is computed and averaged over 4 seeds for CQL, RLs, RLt and ANE, and 8 seeds for FOOD and DARC. For clarity, the standard deviation in Figure 1 is divided by 2 for readable purposes. In all figures, the x-axis represents the number of epochs where each epoch updates the policy and value functions with 8 different trajectories from the source environment.

5.1 Comparison Between the Different Agents

We evaluate the mentioned algorithms on the proposed environments. These experiments provide an overview of the efficiency of the different objectives in finetuning the policy, given reasonably good trajectories. Results are summarized in Table 2, where we also report the median of the normalized average return (NAR) $\frac{J_{P_t}^{\pi_{\text{agent}}} - J_{P_t}^{\pi_{\text{RL}_s}}}{J_{P_t}^{\pi_{\text{RL}_t}} - J_{P_t}^{\pi_{\text{RL}_s}}}$ [Desai *et al.*, 2020] as well as the median of the NAR's standard deviations. The associated learning curves can be found in Appendix C.2.

All the experiments demonstrate the insufficiency of training traditional RL agents solely on the source environment. The optimal policy for the source is far from optimal for the target system as we observe a large drop in performance from RL_t to RL_s on all benchmarked environments. For example, the RL_s exploits the linear torque-current relation in Low Fidelity Minitaur and fails to learn a near-optimal policy for the target environment. Furthermore, RL_s often exhibits a large variance in target environments as it encounters previously unseen situations. This is welcome as relevant trajectories are gathered to guide the agent in the source environment.

Overall, we can see that our algorithm FOOD exhibits the best performances across all considered environments against all other baselines, whether it is constrained by state, state-action or transition visitation distributions. Two exceptions are on Heavy and Friction Cheetah where ANE has very good results. We also note that FOOD associated with its regularization ν_P^π or μ_P^π has better results than when it is associated with d_P^π. This is expected for ν_P^π as it optimizes the whole lower bound, and it seems using μ_P^π implicitely mimizes the second term of Proposition 1.

In addition, we find that the prominent baseline DARC is not efficient in all the use cases. It seems to be particularly good at handling sharp dynamics discrepancies, e.g., when one or two joints are broken or when friction is introduced but struggles for more subtle differences. In fact, it deteriorates over the naive baseline RL_s by a large margin for the Gravity Pendulum and the Low Fidelity Minitaur environments. This may be explained by their reward modification Δ_r (see Appendix B.1) which prevents the agent from entering dissimilar parts of the source environment but seems unable to handle systems with a slight global dynamics mismatch. Even when DARC improves over RL_s, our algorithm FOOD is able to match or exceed its performance. The robust agent ANE is a strong baseline in most environments but may degrade the

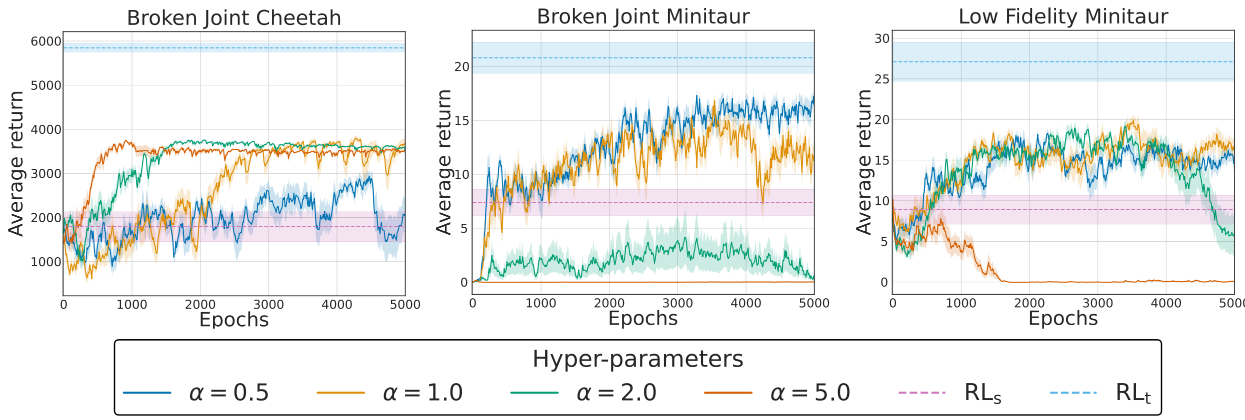

Figure 1: Hyper-parameter sensibility analysis for FOOD on three environments.

performance of traditional RL agents, as seen in Low Fidelity Minitaur, Broken Joint Ant, and Gravity Pendulum. CQL did not provide any good results, except on Gravity Pendulum and Low Fidelity Minitaur, but this was to be expected given the few target trajectories the agent has access to. Finally, we note that the three agents FOOD, DARC, and ANE often reduce the variance originally presented in RL_s.

We attribute FOOD's success to its ability to force the agent to improve the rewards of the source environment along trajectories from the target environment. Its regularization seems to be efficient even in the low data regime.

5.2 Hyper-parameter Sensitivity Analysis

We previously reported the results of the best hyper-parameters of the different methods. In practice, it is important to have a robust range of hyper-parameters in which the considered method performs well. Indeed, to the best of our knowledge, there currently exists no accurate algorithm for selecting such hyper-parameters in a high dimensional environment when the agent has access to limited data gathered with a different policy [Fu *et al.*, 2020]. In this section, we detail the sensitivity of FOOD associated with PPO and GAIL to its hyper-parameter α in 3 environments. They were specifically chosen to illustrate the relevant range for α. FOOD's complete hyper-parameter sensitivity analysis, as well as the one of DARC and ANE, can respectively be found in Appendix C.4, Appendix C.7 and Appendix C.8. They attest that selecting the right hyper-parameters for the baselines DARC and ANE does not have an intuitive pattern.

The hyper-parameter α controls the strength of the regularization in FOOD. If it is too low, the agent will focus mainly on maximizing the rewards of the source environment and becomes very close to the naive baseline RL_s. This can be seen in Broken Joint Cheetah. On the other hand, setting α to a high value may induce the agent to solely replicate the trajectories from the target environment, in which case the agent may also be close to RL_s. Even worse, a hard regularization may degrade over RL_s, as shown in Low Fidelity Minitaur for $\alpha = 5$. However, 5 is an extremely high value as advantages are normalized in PPO, and this may increase the gradient too much and corrupt learning.

In any case, we have found that FOOD provides the best results when the regularization has the same scale as the traditional objective. This is also verified for the environments not displayed in this sub-section. We conclude that FOOD is relatively robust to this range of hyper-parameter, and recommend using PPO with α close to 1 as the underlying RL agent. It is a natural choice given that PPO normalizes its advantage functions.

6 Conclusion

In this work, we investigated different objectives to optimize a policy in different few-shot off-dynamics scenarios, including the state-of-the-art method DARC. We found that these objectives are either too simplistic or unable to cope with complex dynamics discrepancies, thereby limiting their application to real-world systems. To address this challenge, based on theoretical insights, we introduced a novel conservative objective along with a practical algorithm leveraging imitation learning techniques. Through experimentations in different off-dynamics use cases, we have shown that our approach often outperforms the existing methods and seems to be more robust to dynamics changes.

Our agent could also benefit from new advances in the Imitation Learning literature to gain control in building its penalization. Finally, this penalization can be useful when the source environment has been improved using the available target trajectories as it avoids querying the source environment for Out-of-Distribution samples. This will be the primary focus of our future work.

Contribution Statement

Paul Daoudi wrote the main parts of the article and conducted the experiments. Merwan Barlier and Ludovic Dos Santos participated in the development of the main ideas and extensively revised the article. Christophe Prieur and Bogdan Robu oversaw the project and provided additional revisions to the article.

References

[Abbeel *et al.*, 2006] Pieter Abbeel, Morgan Quigley, and Andrew Y Ng. Using inaccurate models in reinforcement learning. In *Proceedings of ICML*, pages 1–8, 2006.

[Abdullah *et al.*, 2019] Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196*, 2019.

[Achiam *et al.*, 2017] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of ICML*, pages 22–31. PMLR, 2017.

[Arndt *et al.*, 2020] Karol Arndt, Murtaza Hazara, Ali Ghadirzadeh, and Ville Kyrki. Meta reinforcement learning for sim-to-real domain adaptation. In *Proceedings of ICRA*, pages 2725–2731. IEEE, 2020.

[Babaeizadeh *et al.*, 2016] Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, and Jan Kautz. Reinforcement learning through asynchronous advantage actor-critic on a gpu. In *Proceedings of ICLR*, 2016.

[Barreto *et al.*, 2017] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Proceedings of NeurIPS*, 30, 2017.

[Barth-Maron *et al.*, 2018] Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, TB Dhruva, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. In *Proceedings of ICLR*, 2018.

[Brockman *et al.*, 2016] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[Corander *et al.*, 2021] Jukka Corander, Ulpu Remes, and Timo Koski. On the jensen-shannon divergence and the variation distance for categorical probability distributions. *Kybernetika*, 2021.

[Coumans and Bai, 2016 2021] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.

[Csiszar and Körner, 1981] Imre Csiszar and J Körner. Coding theorems for discrete memoryless systems. In *Information Theory*. Akadémiai Kiadó (Publishing House of the Hungarian Academy of Sciences), 1981.

[Dadashi *et al.*, 2021] Robert Dadashi, Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. *Proceedings of ICLR*, 2021.

[Desai *et al.*, 2020] Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, and Peter Stone. An imitation from observation approach to transfer learning with dynamics mismatch. *Proceedings of NeurIPS*, 33:3917–3929, 2020.

[Dulac-Arnold *et al.*, 2021] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

[Eysenbach *et al.*, 2021] Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. *Proceedings of ICLR*, 2021.

[Fu *et al.*, 2017] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *Proceedings of ICLR*, 2017.

[Fu *et al.*, 2020] Justin Fu, Mohammad Norouzi, Ofir Nachum, George Tucker, Alexander Novikov, Mengjiao Yang, Michael R Zhang, Yutian Chen, Aviral Kumar, Cosmin Paduraru, et al. Benchmarks for deep off-policy evaluation. In *Proceedings of ICLR*, 2020.

[Gamrian and Goldberg, 2019] Shani Gamrian and Yoav Goldberg. Transfer learning for related reinforcement learning tasks via image-to-image translation. In *Proceedings of ICML*, pages 2063–2072. PMLR, 2019.

[Gangwani, 2021] Tanmay Gangwani. Airl code. https://github.com/tgangwani/RL-Indirect-imitation/tree/master, 2021.

[Garcıa and Fernández, 2015] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *JMLR*, 16(1):1437–1480, 2015.

[Geng, 2021] Xinyang (Young) Geng. Code for conservative q learning for offline reinforcement learning. https://github.com/young-geng/CQL, 2021.

[Ghasemipour *et al.*, 2020] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Proceedings of CoRL*, pages 1259–1277. PMLR, 2020.

[Hanna *et al.*, 2021] Josiah P Hanna, Siddharth Desai, Haresh Karnan, Garrett Warnell, and Peter Stone. Grounded action transformation for sim-to-real reinforcement learning. *Machine Learning*, 110(9):2469–2499, 2021.

[Ho and Ermon, 2016] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Proceedings of NeurIPS*, 29, 2016.

[Höfer *et al.*, 2021] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. Sim2real in robotics and automation: Applications and challenges. *IEEE transactions on automation science and engineering*, 18(2):398–400, 2021.

[Hussein *et al.*, 2017] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[Jakobi *et al.*, 1995] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Life: Third European Conference on Artificial Life*, pages 704–720. Springer, 1995.

[Ju *et al.*, 2022] Hao Ju, Rongshun Juan, Randy Gomez, Keisuke Nakamura, and Guangliang Li. Transferring policy of deep reinforcement learning from simulation to reality for robotics. *Nature Machine Intelligence*, pages 1–11, 2022.

[Kakade and Langford, 2002] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of ICML*, pages 267–274, 2002.

[Kang *et al.*, 2022] Katie Kang, Paula Gradu, Jason J Choi, Michael Janner, Claire Tomlin, and Sergey Levine. Lyapunov density models: Constraining distribution shift in learning-based control. In *Proceedings of ICML*, pages 10708–10733. PMLR, 2022.

[Koos *et al.*, 2012] Sylvain Koos, Jean-Baptiste Mouret, and Stéphane Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2012.

[Kostrikov, 2018] Ilya Kostrikov. Pytorch rl agents. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Proceedings of NeurIPS*, 33:1179–1191, 2020.

[Kumar *et al.*, 2021] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021.

[Mordatch *et al.*, 2015] Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids. In *Proceedings of IROS*, pages 5307–5314. IEEE, 2015.

[Moskovitz *et al.*, 2021] Ted Moskovitz, Michael Arbel, Ferenc Huszar, and Arthur Gretton. Efficient wasserstein natural gradients for reinforcement learning. *Proceedings of ICLR*, 2021.

[Muratore *et al.*, 2019] Fabio Muratore, Michael Gienger, and Jan Peters. Assessing transferability from simulation to reality for reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1172–1183, 2019.

[Nachum *et al.*, 2019] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Proceedings of NeurIPS*, 32, 2019.

[Niu *et al.*, 2022a] Haoyi Niu, Shubham Sharma, Yiwen Qiu, Ming Li, Guyue Zhou, Jianming HU, and Xianyuan Zhan. H2o code. https://github.com/t6-thu/H2O, 2022.

[Niu *et al.*, 2022b] Haoyi Niu, Shubham Sharma, Yiwen Qiu, Ming Li, Guyue Zhou, Jianming Hu, and Xianyuan Zhan. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. *Proceedings of NeurIPS*, 2022.

[Pacchiano *et al.*, 2020] Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Krzysztof Choromanski, Anna Choromanska, and Michael Jordan. Learning to score behaviors for guided policy optimization. In *Proceedings of ICML*, pages 7445–7454. PMLR, 2020.

[Raffin, 2020] Antonin Raffin. Rl baselines3 zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

[Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of ICML*, pages 1889–1897. PMLR, 2015.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Tan *et al.*, 2018] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *Robotics: Science and Systems XIV*, 2018.

[Terpin *et al.*, 2022] Antonio Terpin, Nicolas Lanzetti, Batuhan Yardim, Florian Dorfler, and Giorgia Ramponi. Trust region policy optimization with optimal transport discrepancies: Duality and algorithm for continuous actions. *Proceedings of NeurIPS*, 35:19786–19797, 2022.

[Tessler *et al.*, 2019] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *Proceedings of ICML*, pages 6215–6224. PMLR, 2019.

[Torabi *et al.*, 2019] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. In *Proceedings of IJCAI*, pages 6325–6331. ijcai.org, 2019.

[Touati *et al.*, 2020] Ahmed Touati, Amy Zhang, Joelle Pineau, and Pascal Vincent. Stable policy optimization via off-policy divergence regularization. In *Proceedings of UAI*, pages 1328–1337. PMLR, 2020.

[Weiss *et al.*, 2016] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[Yu *et al.*, 2018] Wenhao Yu, C Karen Liu, and Greg Turk. Policy transfer with strategy optimization. In *Proceedings of ICLR*, 2018.

[Yu *et al.*, 2020] Wenhao Yu, Jie Tan, Yunfei Bai, Erwin Coumans, and Sehoon Ha. Learning fast adaptation with meta strategy optimization. *IEEE Robotics and Automation Letters*, 5(2):2950–2957, 2020.

[Zhu *et al.*, 2018] Shaojun Zhu, Andrew Kimmel, Kostas E. Bekris, and Abdeslam Boularias. Fast model identification via physics engines for data-efficient policy search. In *Proceedings of IJCAI*, pages 3249–3256. ijcai.org, 2018.