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

Paul Daoudi¹, Christophe Prieur², Bogdan Robu², Merwan Barlier¹ and Ludovic Dos Santos³

¹Huawei Noah’s Ark Lab
²GIPSA Lab
³Criteo 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 [Garcia 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 [Gemrian and Goldberg, 2019] and rewards [Barreto et al., 2017], we study the off-dynamics setting where the mismatch is between their dynamics [Hofer 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 DAR [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 Few-Shot Off Dynamics (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) $(S, A, r, P, \gamma, \rho_0)$, with a state space $S$, an action space $A$, a transition kernel $P : S \times A \to \Delta(S)$, a reward function $r : S \times A \times S \to [R_{\min}, R_{\max}]$, the initial state distribution $\rho_0$ and a discount factor $\gamma \in [0, 1)$. A policy $\pi : S \to \Delta(A)$ is a decision rule mapping a state over a distribution of actions. The value of a policy $\pi$ is measured through the value function $V_{\pi}(s) = \mathbb{E}_{\pi, P}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) | s_0 = s]$. The objective is to find the optimal policy maximizing the expected cumulative rewards $J_{\pi} = \mathbb{E}_{\rho_0}[V_{\pi}(s)]$. We also define the $Q$-value function $Q_{\pi}(s, a) = \mathbb{E}_{\pi, P}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) | s_0 = s, a_0 = a]$ and the advantage value function $A_{\pi}(s, a) = Q_{\pi}(s, a) - V_{\pi}(s)$. Finally, let $d_{\pi}(s) = (1 - \gamma)[\mathbb{E}_{\pi, P}[\sum_{t=0}^{\infty} \gamma^t P(s_t = s)]$ the state visitation distribution, as well as its extension to state-action $\mu_{\pi}(s, a)$ and transition $\nu_{\pi}(s, a, s')$. 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 $M_s$ and the target environment $M_t$. We hypothesize that they are identical except for their transition probabilities $P_s \neq P_t$. 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 $P$ subscript for the value functions as they are always associated with the source environment $M_s$.

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 \( D_t \). 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 \( \mathcal{M}_t \) 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 \( b_{\pi}^\gamma \) 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 \( \pi_\theta \) and \( V_{\pi_\theta}^{\gamma} \). At each iteration \( k \), the policy is improved using the advantage function built from the approximated value function \( V_{\pi_\theta}^{\gamma} \), with a penalization to respect the lower bound:

\[
\max_{\theta \in \Theta} \mathbb{E}_{s \sim D_{\pi_\theta}} \left[ A_{\omega_k}^{\pi_\theta}(s, \alpha) \right] - \alpha_k D(b_{\pi}^\gamma \| b_{\pi}^\gamma),
\]

where \( D \) is any kind of similarity metric and \( \alpha_k \) is a hyperparameter, often set to a constant \( \alpha \). TRPO [Schulman et al., 2015] can be retrieved with \( b_{\pi}^\gamma = \pi \), by setting \( D \) to be the Kullback-Leibler (KL) divergence and enforcing the penalization as a constraint with a step-size \( \epsilon_k \). 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 \( \pi \) and \( P \) by setting \( b_{\pi}^\gamma = d_{\pi}^\gamma \). 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 Few-shot Off-Dynamics (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 \( J_{\pi}^{\gamma} \) 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 off-dynamics setting, and propose the following lower bound using state visitation distributions.

**Proposition 1.** Let \( J_{\pi}^{\gamma} = \mathbb{E}_{\pi_0}[V_{\pi}^{\gamma}(s)] \) the expected cumulative rewards associated with policy \( \pi \), transitions \( P \) and initial state distribution \( \pi_0 \). For any policy \( \pi \) and any transition probabilities \( P_1 \) and \( P_2 \), the following holds:

\[
J_{\pi}^{\gamma} \geq J_{\pi}^{\gamma} - \frac{2R_{\max}}{1-\gamma} \left(D_{TV}(d_{\pi}^{\gamma}, d_{\pi_1}^{\gamma}) + D_{TV}(P_1, P_2)\right),
\]

with \( D_{TV} \) the Total Variation distance and \( D_{TV}(P_1, P_2) = \mathbb{E}_{s \sim d_{\pi_1}^{\gamma}(\cdot), \alpha \sim \pi_{\alpha}(\cdot | s)} [D_{TV}(P_1(\cdot | s, a), P_2(\cdot | s, a))] \).

The Proposition also holds by replacing \( D_{TV}(d_{\pi_1}^{\gamma}, d_{\pi_2}^{\gamma}) \) with \( D_{TV}(\mu_{\pi_1}^{\gamma}, \mu_{\pi_2}^{\gamma}) \). 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 \( D_{TV}(d_{\pi_1}^{\gamma}, d_{\pi_2}^{\gamma}) \) quantifies the difference between the visited states of the rollouts in the source and target environments. The second \( D_{TV}(\mu_{\pi_1}^{\gamma}, \mu_{\pi_2}^{\gamma}) \) 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 \( \mathcal{M}_t \) and \( \mathcal{M}_t \) are unknown. Hence, we propose the following simpler lower bound that considers transition visitation distributions instead.

**Proposition 2.** Let \( J_{\pi}^{\gamma} = \mathbb{E}_{\pi_0}[V_{\pi}^{\gamma}(s)] \) the expected cumulative rewards associated with policy \( \pi \), transitions \( P \) and initial state distribution \( \pi_0 \). For any policy \( \pi \) and any transition probabilities \( P_1 \) and \( P_2 \), the following holds:

\[
J_{\pi}^{\gamma} \geq J_{\pi}^{\gamma} - \frac{2R_{\max}}{1-\gamma} D_{TV}(\nu_{\pi_1}^{\gamma}, \nu_{\pi_2}^{\gamma}),
\]
with $\text{TV}$ the Total Variation distance.

We also defer the proof to Appendix A. Here, the lower bound depends on the sole metric $D_{\text{TV}}(\nu_{\pi_{P}^i}, \nu_{\pi_{P}^k})$ 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 $D_{\text{TV}}(\nu_{\pi_{P}^i}, \nu_{\pi_{P}^k})$ is small, as the policy induces similar objectives $J_{opt}$. Inspired by this insight, we adapt conservative methods to the off-dynamics setting and propose a new regularization between trajectories by setting the behaviors $b_{\pi_{P}}$ to be the transition visitation distribution respectively associated with the transition probabilities of the source and target environment $\nu_{\pi_{P}^i}$:

$$\max_{\theta \in \Theta} \mathbb{E}_{s \sim \nu_{\pi_{P}^i}(.), a \sim \pi_{\theta}(s)} \left[ A_{\omega_{\pi_{P}^i}}(s, a) \right] - \alpha D(\nu_{\pi_{P}^i} || \nu_{\pi_{P}^k}).$$

(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 samples 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 $D_f(\nu_{\pi_{P}^i} || \nu_{\pi_{P}^k}) = \mathbb{E}_{(s, a) \sim \pi_{\theta}} \left[ f(\nu_{\pi_{P}^i}(s, a), \nu_{\pi_{P}^k}(s, a)) \right]$ 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 $\frac{\nu_{\pi_{P}^i}}{\nu_{\pi_{P}^k}}$ for any policy $\pi_{\theta}$ sufficiently close to $\pi_{\theta_{opt}}$, 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 $D(\nu_{\pi_{P}^i} || \nu_{\pi_{P}^k})$. 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 $\pi_{e}$ 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 $\mu_{\pi_{P}^i}$ and the expert’s $\mu_{\pi_{e}}$. The divergence minimization problem is transformed into a reward $r_{imit}$ maximization one, resulting in an imitation value function $V_{imit}^\pi = \mathbb{E}_{s, t \rightarrow T} \sum_{t=0}^{T} \gamma^{t} r_{imit}(s_{t}, a_{t}, s_{t+1}) | s_{0} = s$. 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:

$$\arg\max_{\theta \in \Theta} V_{imit}^\pi = \arg\min_{\theta \in \Theta} D(\nu_{\pi_{P}^i} || \nu_{\pi_{e}}).$$

(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 straightforward with PWIL [Dadashi et al., 2021],

<table>
<thead>
<tr>
<th>GAIL</th>
<th>AIRL</th>
<th>PWIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{\text{KL}}(X_{\pi_{P}^i}</td>
<td></td>
<td>X_{\pi_{e}})$</td>
</tr>
</tbody>
</table>

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

These IL techniques enable efficient estimation of this value function using a small number of samples from $\nu_{\pi_{P}^i}$ and unlimited access to $M_{s}$. Let $\xi \in \Xi$ be the weights of this parametrized value function. The new regularization is $\sum_{(s, a) \sim \pi_{\theta}} \xi(s, a)$, which can be learned with any suitable IL algorithm. It leads to the practical objective:

$$\max_{\theta \in \Theta} \mathbb{E}_{s \sim \nu_{\pi_{P}^i}(.), a \sim \pi_{\theta}(s)} \left[ A_{\omega_{\pi_{P}^i}}(s, a) - \alpha A_{imit}(s, a, \xi) \right].$$

(6)

This new agent is quite generic as it could be optimized with different divergences. It takes as input an online RL algorithm [Babaeezadeh et al., 2016; Schulman et al., 2017] 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 $\pi_{\theta_{opt}}$. Subsequently, the policy, traditional, and imitation value functions are re-trained on the source environment with $\mathcal{O}$ and $\mathcal{I}$ according to Equation 4. The whole algorithm is summarized in Algorithm 1.

1These 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 $\theta_0$ and $\omega_0$ with $\mathcal{O}$
- Randomly initialize the weights $\xi_0$

for $k \in (0, \ldots, K - 1)$ do

- Gather $N$ trajectories $\{\tau_1, \ldots, \tau_N\}$ with $\pi_{\theta_k}$ on the target environment $\mathcal{M}_k$ and add them in $\mathcal{D}_i$
- Learn the value function weights $\omega_{k+1}$ with $\mathcal{O}$ in the source environment $\mathcal{M}_i$
- Learn the imitation value function weights $\xi_{k+1}$ with $\mathcal{I}$ in $\mathcal{M}_i$ using $\mathcal{D}_i$
- Learn the policy maximizing (6) using $\mathcal{D}_i$ and $\mathcal{M}_i$ 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 Bat, 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 in the off-dynamics setting. In this benchmark, the source environment has a linear torque-current relation for the actuator.

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 [Babaeizadeh et al., 2021]. FOOD is tested with its theoretically motivated metric between transition visitation distributions $v_\pi^T$, as well as with $d_\chi^T$ and $\mu_\rho^T$ 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 $\sigma_{\text{DARC}}$ 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 $\sigma_{\text{ANE}}$ 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 $\beta$. 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 incentivized 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 $\mathbf{RL}_s$ and $\mathbf{RL}_t$ 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.
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 RLt and RLs 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 $\alpha$ 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) $\frac{J_{\text{target}}^P - J_{\text{source}}^P}{J_{\text{source}}^P}$ [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 RLs to RLs on all benchmarked environments. For example, the RLs exploits the linear torque-current relation in Low Fidelity Minitaur and fails to learn a near-optimal policy for the target environment. Furthermore, RLs 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 $\nu_P^t$ or $\mu_P^t$ has better results than when it is associated with $d_P^t$. This is expected for $\nu_P^t$ as it optimizes the whole lower bound, and it seems using $\mu_P^t$ implicitly mimics 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 naïve baseline RLs, by a large margin for the Gravity Pendulum and the Low Fidelity Minitaur environments. This may be explained by their reward modification $\Delta_r$ (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 RLs, 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.

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 \( \alpha \) 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


