Policy Learning for Off-Dynamics RL with Deficient Support

Linh Le Pham Van
Applied Artificial Intelligence Institute, Deakin University
Geelong, Australia
l.le@deakin.edu.au

Hung The Tran
Applied Artificial Intelligence Institute, Deakin University
Geelong, Australia
hung.tranthe@deakin.edu.au

Sunil Gupta
Applied Artificial Intelligence Institute, Deakin University
Geelong, Australia
sunil.gupta@deakin.edu.au

ABSTRACT

Reinforcement Learning (RL) can effectively learn complex policies. However, learning these policies often demands extensive trial-and-error interactions with the environment. In many real-world scenarios, this approach is not practical due to the high costs of data collection and safety concerns. As a result, a common strategy is to transfer a policy trained in a low-cost, rapid source simulator to a real-world target environment. However, this process poses challenges. Simulators, no matter how advanced, cannot perfectly replicate the intricacies of the real world, leading to dynamics discrepancies between the source and target environments. Past research posited that the source domain must encompass all possible target transitions, a condition we term full support. However, expecting full support is often unrealistic, especially in scenarios where significant dynamics discrepancies arise. In this paper, our emphasis shifts to addressing large dynamics mismatch adaptation. We move away from the stringent full support condition of earlier research, focusing instead on crafting an effective policy for the target domain. Our proposed approach is simple but effective. It is anchored in the central concepts of the skewing and extension of source support towards target support to mitigate support deficiencies. Through comprehensive testing on a varied set of benchmarks, our method's efficacy stands out, showcasing notable improvements over previous techniques.

KEYWORDS

Off-Dynamics; Deficient Support; Transfer Learning; Reinforcement Learning

ACM Reference Format:
Linh Le Pham Van, Hung The Tran, and Sunil Gupta. 2024. Policy Learning for Off-Dynamics RL with Deficient Support. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 8 pages.

1 INTRODUCTION

Reinforcement Learning (RL) has shown its capacity to acquire intricate behaviors in numerous real-world challenges [12, 16, 23]. However, it requires numerous trial-and-error interactions with the environment, which may not be feasible due to the high costs of data collection or safety concerns in many real-world scenarios (e.g. robotics, autonomous driving, medical treatment, etc). Training the policy in an alternative source environment, e.g. a simulator, which is both safer and faster, and using a limited set of data from the real-world target environment has, therefore, become a common approach. This approach is usually known as Off-Dynamics Reinforcement Learning [6].

Previous work on this problem includes [1, 3, 15, 17]. Ljung [15] and Chebotar et al. [3] propose an approach to align the source dynamics to the target dynamics by real-world data using the system identification method. These methods require a detailed understanding of the target domain (e.g. knowing the physics behind the systems). Peng et al. [17] train the policy on a set of randomized simulators to yield a robust policy. Again, the set of randomizations is chosen carefully based on the detailed understanding of the target domain. The requirement of having a detailed understanding of target and source domains limits the applicability of such methods. *Thus new methods that do not rely on such detailed knowledge are required.*

The other recent works such as [4, 5, 9, 10] learn the target policy via learning an action transformation function which maps the actions suggested by the source policy to make them suitable for the target domain. In a related approach, Eysenbach et al. [6], Liu et al. [14] use the dynamics discrepancy term as an additional reward to prevent the policy from exploiting the dynamics mismatch area. However, all these works make a strong assumption that the source domain (e.g. a simulator) encompasses all possible target (real-world) transitions, a condition which we call *Full support*. However, full support condition rarely holds in practice as a simulator no matter how advanced cannot perfectly replicate the intricacies of the real world, and thus can not cover all the transitions in the target domain. For example, an autonomous driving vehicle may face changed conditions such as new kind of places (i.e. highway, city, countryside), weather (i.e. sunny, rainy, hazy), or time (i.e. day, night), resulting in only a fraction of target transitions in the support of the source domain. In Section 5, we show that the existing methods fail drastically when a source domain does not fully support the target domain. *Therefore, the problem of off-dynamics reinforcement learning under deficient support remains an open problem.*

In this paper, we address the aforementioned challenges relaxing both the detailed domain understanding and full support requirements in Off-dynamics RL to deal with source support deficiency. Under this setting, we propose an effective method to reduce source deficiency by creating a *modified source* domain using two operations: (1) by *skewing* the source transitions to support the target domain with higher probability, (2) and *extending* the source transitions towards the target transitions to improve the source support for the target. The skewing is guided by an importance weighting which is learned by solving an optimization problem and the source

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

support extension is done by following the MixUp scheme [32]. Finally, we utilize both the skewed and mixup transitions and adjust the rewards to compensate for the dynamics discrepancy between the modified source domain and the target domain, and use this modified data for target policy learning.

Our main contributions are:

- We are the first to address the policy learning for off-dynamics RL with deficient support, which is a novel problem and is encountered in numerous real-world scenarios.
- We conduct a theoretical analysis of off-policy RL under the setting of deficient support, offering valuable practical insights for the development of effective policy learning algorithm.
- We propose DADS, a practical and effective algorithm for off-dynamics RL with deficient support via source skewing and extension operations.
- Finally, we demonstrate the superior performance of our proposed method over the existing methods through a diverse set of experiments.

2 RELATED WORK

Domain adaptation in RL: Domain adaptation in RL is needed when there is a difference in the observation space, transition dynamics, or reward function. In this paper, we study domain adaptation with dynamics mismatch. System identification method [3, 11, 15, 30] is a direct approach to align the source dynamics with the observed target data. However, these methods typically require a model of the source environment and a large set of target data to adjust the parameters of the source environment to align it with the target domain. Another approach, domain randomization [1, 17, 19, 25], involves training RL policies over a collection of randomized simulated source domains. However, this approach often exhibits sensitivity to the selection of randomized parameters or parameter distributions [6].

In contrast, ground action transformation techniques [4, 5, 9, 10, 31] eliminate the need for a parameterized simulator or manually selected randomized dynamics parameters. These techniques aim to rectify dynamics mismatch by learning action transformations of the source policy using the target data. Such action transformation techniques require the existence of an accurate action transformation policy, which may be infeasible when the dynamics mismatch between the source and target is large e.g. when the source domain lacks transitions seen in the target domain. In the absence of an accurate action transformation policy, such methods exhibit poor performance.

Xu et al. [29] introduce value-guided data filtering that removes the transitions that have high value discrepancies during policy training. Recently, Eysenbach et al. [6], Liu et al. [14] proposed using dynamics discrepancy to correct the reward during training the policy. However, these works rely on a full support condition, that the source domain must contain all possible transitions in the target domain, which rarely holds in real-world scenarios, thus preventing handling the large dynamics gap problems. Our study takes advantage of the reward correction but relaxes the full support condition.

Mixup in RL: MixUp was first introduced in [32] in the supervised learning setting as a novel data augmentation method that improves the generalizability of the deep learning models by training them on convex linear combinations of dataset samples. In the RL problems, [13, 21, 28, 33] have demonstrated that MixUp helps to improve the generalizability and sample efficiency of the learned policy. From a different perspective, we employ MixUp to expand the support of the source domain to cover the support of the target domain, reducing the support deficiency problem.

Deficient support in Off-policy Bandits: The deficient support problem has been explored in off-policy bandit settings, where it refers to the lack of data for certain actions under the logging policy compared to the target policy [7, 18, 20, 27]. However, this existing work focuses primarily on the policy space. In contrast, our work tackles the novel challenge of deficient support in the context of off-dynamics policy learning for reinforcement learning. We address the gap between the source and target transition dynamics distributions, which is a critical issue for effective policy adaptation in RL.

3 PROBLEM SETTING AND PRELIMINARIES

Background: In this section, we introduce our notation and a formal definition of off-dynamics online policy learning. We consider two infinite-horizon Markov Decision Processes (MDPs) $\mathcal{M}_{src} := (\mathcal{S}, \mathcal{A}, P_{src}, r, \gamma, \rho_0)$ and $\mathcal{M}_{tar} := (\mathcal{S}, \mathcal{A}, P_{tar}, r, \gamma, \rho_0)$ representing source domain and target domain, respectively. In our setting, we assume that the two domains share the same state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, discount factor $\gamma \in [0, 1)$, and the initial state distribution $\rho_0 : \mathcal{S} \to [0, 1]$; the only difference between two domains is in their transition dynamics, $P_{src}(s'|s, a)$ and $P_{tar}(s'|s, a)$.

A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is defined as a map from states to a probability distribution over actions. Then, we denote the probability that the policy π encounters state s at the time step t in an MDP $\mathcal{M}$ as $\mathrm{P}^{\pi}_{\mathcal{M},t}(s)$, and the normalized state-action occupancy of state-action pair (s, a) in $\mathcal{M}$ is $\rho^{\pi}_{\mathcal{M}}(s, a) := (1 - \gamma) \sum_{t=0}^{\infty} \mathrm{P}^{\pi}_{\mathcal{M},t}(s)\pi(a|s)$. The performance of a policy π in the MDP $\mathcal{M}$ is defined as $\eta_{\mathcal{M}}(\pi) = \mathbb{E}_{s,a\sim\rho^{\pi}_{\mathcal{M}}}[r(s, a)]$. The value function on the MDP $\mathcal{M}$ and policy π is defined as $V^{\pi}_{\mathcal{M}}(s) := \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s\right]$.

We focus on the Off-Dynamics Online (ODO) policy learning, which is formally defined as follows:

Definition 1 (Off-Dynamics Online Policy Learning). *Given a source domain represented by $\mathcal{M}_{src}$ and a target domain represented by $\mathcal{M}_{tar}$ with distinct dynamics functions, our goal is to leverage source interactions and a small number of target interactions to derive a good policy that achieves high reward in $\mathcal{M}_{tar}$.*

We highlight that the previous methods often require the assumption of a *full support condition*, that implies every possible transition in the target domain $\mathcal{M}_{tar}$ is covered by the source domain $\mathcal{M}_{src}$. We formally define the *full support condition* as follows:

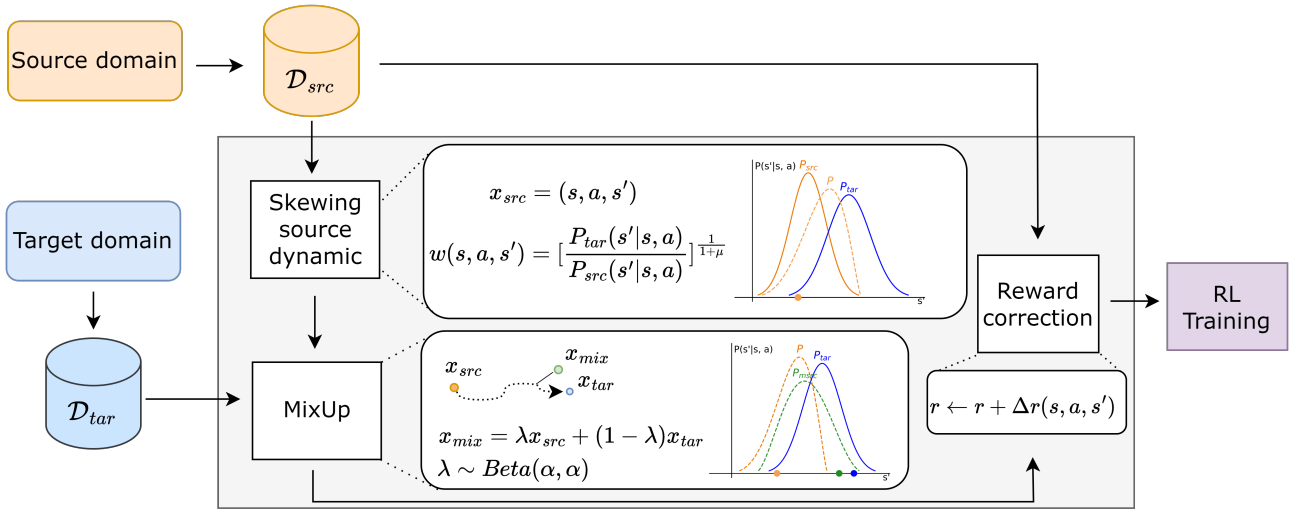

Figure 1: Off-dynamics online policy learning with deficient support. Given the source domain and target domain with limited online interaction, we propose skewing the source dynamics, which enables us to sample the source transitions that are closely aligned with the target dynamics and employ MixUp procedure to expand the source support set towards the target support set. Then we adopt the Reward correction to compensate the policy with an additional reward for encouraging dynamic-consistent behaviors.

Definition 2 (Full support). *We say that a source domain $\mathcal{M}_{src}$ has full support for a target domain $\mathcal{M}_{tar}$ if every transition with non-zero probability in the target domain $\mathcal{M}_{tar}$ also has a non-zero probability in the source domain $\mathcal{M}_{src}$: $P_{tar}(s'|s,a) > 0 \Rightarrow P_{src}(s'|s,a) > 0, \forall s, s' \in \mathcal{S}, a \in \mathcal{A}$.*

In the ODO policy learning with the *full support* condition holds, the target performance can be guaranteed as follows:

Proposition 3 (Performance bound). *Let $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$ are the source domain and target domain with different dynamics P_{src} and P_{tar} respectively. The performance difference of any policy π in $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$ can be bounded as follows:*

$$\begin{aligned} &\left|\eta_{M_{tar}}(\pi) - \eta_{M_{src}}(\pi)\right| \\ &\leq \frac{\gamma r_{max}}{(1-\gamma)^2} \cdot \underbrace{\sqrt{2\mathbb{E}_{\rho^{\pi}_{tar}}\left[D_{KL}(P_{tar}(.|s,a), P_{src}(.|s,a))\right]}}_{(a)}. \end{aligned} \tag{1}$$

The performance bound, as outlined in Proposition 3, depends on the dynamics discrepancy term (a). A recent approach called DARC [6] uses the dynamics discrepancy between source and target domains as an incremental reward, to prevent the policy from exploiting areas in the source that have a high dynamics mismatch with the target domain.

However, the *full support condition* is stringent and might not hold in many real-world scenarios. When this condition is not met, it results in the challenge of *deficient support*, which we formally define as:

Definition 4 (Deficient support). *We say source MDP $\mathcal{M}_{src}$ has support deficiency for target MDP $\mathcal{M}_{tar}$ if there exists a set $\{(s', s, a)\} \neq \emptyset$ such that for each transition (s', s, a) belongs to it, we have $P_{tar}(s'|s,a) > 0$ but $P_{src}(s'|s,a) = 0$.*

The deficient support assumption does not require the source domain to encompass all potential target transitions. Thus, when deficient support happens, it poses a challenge due to the uncovered target areas. In this paper, we relax the *full support* assumption and propose a method for the off-dynamics online (ODO) policy learning with *deficient support*.

Under the deficient support problem, we derive the performance bound as follows:

Proposition 5 (Performance bound under Deficient Support). *Let $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$ are source domain and target domain with different dynamics P_{src} and P_{tar} respectively. For each state-action pair s, a, denote $S^0_{s,a} = \{[s'_{0_i}, s'_{0_j}]\}$ contains intervals where $P_{src}(.|s,a) = 0$, and $S^1_{s,a} = \{[s'_{1_i}, s'_{1_j}]\}$ includes intervals where $P_{src}(.|s,a) > 0$, and $S^0_{s,a} \cup S^1_{s,a} = supp\left(P_{tar}(.|s,a)\right)$. The performance difference of any policy π in $\mathcal{M}_{src}$ and $\mathcal{M}_{tar}$ can be bounded as follows:*

$$\begin{aligned} &\left|\eta_{tar}(\pi) - \eta_{src}(\pi)\right| \\ &\leq \frac{\gamma r_{max}}{(1-\gamma)^2} \cdot \mathbb{E}_{\rho^{\pi}_{tar}(s,a)}\left[\sum_{S^1_{s,a}} \left| \int_{s'_{1_i}}^{s'_{1_j}} P_{tar}(s'|s,a) - P_{src}(s'|s,a) ds' \right|\right] \\ &+ \frac{\gamma}{1-\gamma} \cdot \underbrace{\mathbb{E}_{\rho^{\pi}_{tar}(s,a)}\left[\sum_{S^0_{s,a}} \int_{s'_{0_i}}^{s'_{0_j}} P_{tar}(s'|s,a) \cdot \left|V^{\pi}_{src}(s')\right| ds'\right]}_{\text{support deficiency}}. \end{aligned} \tag{2}$$

Proposition 5 highlights the gap between $\eta_{\mathcal{M}_{tar}}(\pi)$ and $\eta_{\mathcal{M}_{src}}(\pi)$ due to support deficiency. Notably, the In Eq (1) emerges as a special instance of In Eq (2) when full support is assumed. In this special case, the support deficiency term in (2), which quantifies the target value V_{tar} on the unsupported set, vanishes. Based on Proposition 5, guaranteeing performance on the target domain hinges on minimizing both the dynamics discrepancy within the supported region and the support deficiency. With this dual objective in mind, the next section introduces our method aimed at simultaneously reducing both terms to ensure robust performance guarantees.

4 PROPOSED METHOD

In this section, we introduce a novel approach to address the challenge of off-dynamics policy learning in the presence of support deficiency. Our method aims to create a modified source domain that has minimum source deficiency w.r.t to the target. Our method has three primary steps: (1) skewing the source transitions to maximize its support overlap with the target domain; (2) extrapolating the source transitions to extend the source support all the way up to the target domain. This is done using the MixUp procedure by creating new synthetic transitions between the source transitions and the target transitions via their convex combinations; and (3) combining the source and MixUp transitions to form a modified source transition set, adjusting their rewards similarly to [6], and use these modified transitions to train the target policy. Our method is depicted in Figure 1. Our approach aims to minimize the second and third terms in the performance bound in In Eq (2). The second term is minimised by iteratively *skewing* the source support towards the target. The third term in the performance bound is minimized by *extending* the source support toward the target via mix up as it can generate the samples in the unsupported region.

4.1 Skewing Source Dynamics

We present the skewing source dynamics strategy, which enables us to sample the source transitions that are closely aligned with the target dynamics. Specifically, we learn a dynamics distribution $P(s'|s,a)$ that is close to the target dynamics distribution $P_{tar}(s'|s,a)$ but not significantly far from the source dynamics $P_{src}(s'|s,a)$, measured in terms of the KL divergence. This is formulated as the following constrained function optimization problem:

$$\begin{aligned} &\min_{P\in\mathcal{P}} D_{\mathrm{KL}}(P(.|s,a)||P_{tar}(.|s,a))\ \forall s\in\mathcal{S}, a\in\mathcal{A}. \\ &\text{s.t } D_{\mathrm{KL}}(P(.|s,a)||P_{src}(.|s,a))\le\epsilon\ \forall s\in\mathcal{S}, a\in\mathcal{A}. \\ &\int_{s'} P(s'|s,a)ds' = 1\ \forall s\in\mathcal{S}, a\in\mathcal{A}. \end{aligned} \tag{3}$$

where $\mathcal{P}$ is a family of all transition dynamics distributions and we have $P_{tar}(.|s,a)\in\mathcal{P}$ and $P_{src}(.|s,a)\in\mathcal{P}$. The first constraint with KL divergence and the parameter ϵ regularizes the dynamics function $P(s'|s,a)$ to stay close to the source dynamics $P_{src}(s'|s,a)$, while the second constraint is to ensure $P(s'|s,a)$ is a valid probability density function.

To solve the constrained optimization problem, we create a Lagrangian-based objective function and then solve for the optimal transition dynamics $P^*(s'|s,a)$ by taking a derivative and equating it to zero. With a few steps of analysis, we obtain the optimal skewed dynamics function $P^*(s'|s,a)$ as follows:

$$P^*(s'|s,a)\propto P_{src}(s'|s,a)\exp[\frac{1}{1+\mu}\log\frac{P_{tar}(s'|s,a)}{P_{src}(s'|s,a)}]. \tag{4}$$

The parameter μ serves as the Lagrange multiplier linked to the KL constraint in (3), and it essentially dictates the degree of constraint strength or how far the optimal solution can deviate from the source. The full derivation is provided in the *Appendix*.

To effectively sample the skewed distribution P^*, we leverage the existing source domain data P_{src}. We achieve this by re-weighting samples from P_{src} based on the density ratio between the skewed transition dynamics and the original (source) transition dynamics. In particular, we propose a sampling approach where source transitions (s,a,s') are chosen with probabilities proportional to the density ratio $w(s,a,s')$ as follows:

$$\begin{aligned} w(s,a,s') &= \exp\left[\frac{1}{1+\mu}(\log\frac{P_{tar}(s'|s,a)}{P_{src}(s'|s,a)})\right] \\ &\propto P^*(s'|s,a)/P_{src}(s'|s,a). \end{aligned} \tag{5}$$

Concretely, we define the sampling probability of the i-th source transition (s,a,s') as follows:

$$p^i(s,a,s') = \frac{w^i(s,a,s')}{\sum_k w^k(s,a,s')}. \tag{6}$$

where $w^i(s,a,s')$ is the priority weight of source transition i.

Estimating the ratio of source and target transition dynamics: Calculating the sampling probability of each source transition requires estimating the density ratio between the target dynamics and the source dynamics for each source transition. Similar to [6], we adopt the probabilistic classification technique [24] to estimate this density ratio. Specifically, we use a pair of binary classifiers, $q_{\theta_{SAS}}(.|s,a,s')$ and $q_{\theta_{SA}}(.|s,a)$, which distinguish whether a transition (s,a,s') (or a state-action pair (s,a)) comes from the source or target domain. The density ratio is computed as follows:

$$\begin{aligned} \log\frac{P_{tar}(s'|s,a)}{P_{src}(s'|s,a)} &= \log\frac{q_{\theta_{SAS}}(\text{target}|s,a,s')}{q_{\theta_{SAS}}(\text{source}|s,a,s')} \\ &\quad + \log\frac{q_{\theta_{SA}}(\text{source}|s,a)}{q_{\theta_{SA}}(\text{target}|s,a)}. \end{aligned} \tag{7}$$

The two classifiers $q_{\theta_{SAS}}(.|s,a,s')$ and $q_{\theta_{SA}}(.|s,a)$ are learned with the standard cross-entropy loss using the source and target data.

4.2 Extending Source Support

The deficient support presents the existence of uncovered target areas. Our idea is to extend the source support toward the target support by employing the MixUp procedure, thus filling the uncovered target support. Specifically, we utilize the skewing source transitions from the previous step and mix them up with target transitions to create MixUp transitions. While we should ideally be mixing up s' from source and target domains conditioned on the same state-action pair (s,a), since we deal with continuous state and action spaces, it is challenging to find an identical pair. Even a nearest neighbor approach can result in fairly distant (s,a) pairs from the source and target domains. To avoid this problem, we mix up the quadruples (s,a,r,s') of the source with those of the target.

Given a source transition $x_{src} = (s,a,r,s')_{src}$ (obtained from the skewing step) and a target transition $x_{tar} = (s,a,r,s')_{tar}$, we

use MixUp to generate a synthetic transition by taking convex combination between x_{src} and x_{tar} as follows:

$$x_{mix} = \lambda x_{src} + (1 - \lambda) x_{tar} \tag{8}$$

where λ is sampled from a Beta distribution as $\lambda \sim B(\alpha, \alpha)$, with $\alpha > 0$. As suggested by [32], we set $\alpha = 0.2$. Note that if either source or target transition encounters a terminal state, we do not employ interpolation, and simply use the target transition instead. This is mainly done to avoid non-binary terminal signal [21].

Let $\{x^i_{src} = (s, a, r, s')^i_{src}\}_{i=1}^N$ be a batch of N source transitions sampled from the source data D_{src} with probabilities $p^i(s, a, s')$ as in Eq (6). We sample uniformly N target transitions $\{x^i_{tar} = (s, a, r, s')^i_{tar}\}_{i=1}^N$ from the target data D_{tar}. We then sample a batch $\{\lambda_i\}_{i=1}^N$ from a Beta distribution, and perform MixUp using each pair of source and target transitions.

4.3 Reward modification

While the modified source now supports the target transitions better, the transition probability densities of the modified source and target domain still may be different. Thus, following the scheme in [6], we adjust the reward for each transition by adding an incremental term Δr, as follows:

$$\Delta r(s, a, s') = \log P_{tar}(s'|s, a) - \log P_{msrc}(s'|s, a) \tag{9}$$

where $P_{msrc}(s'|s, a)$ denotes the transition dynamics of the modified source. In practice, to obtain $\Delta r(s, a, s')$ for each transition (s, a, s'), we employ a similar density ratio estimate as in section 4.1, using two binary classifiers $q_{\phi_{SAS}}(.|s, a, s')$ and $q_{\phi_{SA}}(.|s, a)$ as follows:

$$\begin{aligned}\Delta r(s, a, s') = &\log \frac{q_{\phi_{SAS}}(\text{target}|s, a, s')}{q_{\phi_{SAS}}(\text{modified source}|s, a, s')} \\ &+ \log \frac{q_{\phi_{SA}}(\text{modified source}|s, a)}{q_{\phi_{SA}}(\text{target}|s, a)}.\end{aligned} \tag{10}$$

Then each transition in the batch of modified source transitions is modified as $(s, a, r + \Delta r, s')$ and used to train the target policy.

4.4 Algorithm

We summarize the above steps as our proposed method in Algorithm 1. We perform the skew operation in Lines 8 and 9, and the MixUp procedure in Line 11. We perform reward correction in Line 12 and learn two pairs of domain classifiers in Lines 13 and 14. Finally, in Line 15, we use a standard RL algorithm for policy learning. In our experiment, we use Soft Actor-Critic (SAC) [8] algorithm. To simplify the algorithm, we find that simply using a fixed constant for the Lagrange multiplier μ is also effective, rather than adaptively updating μ. In the implementation, to sample efficiently from Eq (6), we employ the *sum-tree* data structure similar to [22], which allows $O(\log N)$ complexity for updates and sampling.

Discussion: We discuss the key signficance of our two operations: *Skew* and *Extension*. Skewing source transitions shifts their probability mass towards the target transitions without changing the source domain's support or generating any new samples. Unlike skewing, extension of source support using MixUp can generate new synthetic samples from outside the source support closer to the target domain. Both schemes have their individual strengths and limitations. In particular, the skewing operation boosts the sampling of source transitions that are close to the target domain. However, these transitions can not fill the uncovered target areas in the source domain. On the other hand, the MixUp operation can generate novel synthetic samples that can expand the source support to cover unseen target transitions and thus can bring significant improvement in target policy learning. However, a downside can be that randomly mixing up a target transition with any source transition may lead to synthetic samples that do not lie in the target transition manifold, which could at times degrade the performance. In our case, when we use skewing and MixUp together, skewing helps to improve the performance of MixUp by rejecting the source transitions that are unlikely to occur in the target domain before mixing them. Thus, both skewing and Mixup have independent and effective roles in our algorithm. We empirically demonstrate their effectiveness in Section 5.

Algorithm 1 Online Dynamics Adaptation with Deficient Support in RL (DADS)

1: **Input:** Source MDP $\mathcal{M}_{src}$ and target $\mathcal{M}_{tar}$; ratio r of experience from source vs. target; the batch size N.
2: **Initialize:** The source data $\mathcal{D}_{src}$ and the target data $\mathcal{D}_{tar}$; policy π; parameters θ for classifiers that distinguish source and target domain $q_{\theta_{SAS}}, q_{\theta_{SA}}$; and parameters ϕ for classifiers that distinguish modified source and target domain $q_{\phi_{SAS}}, q_{\phi_{SA}}$.
3: **for** $t = 1, \ldots,$ num iterations **do**
4: $\quad \mathcal{D}_{src} \leftarrow \mathcal{D}_{src} \cup \text{ROLLOUT}(\pi, \mathcal{M}_{src})$.
5: $\quad$ **if** $t \mod r == 0$ **then**
6: $\quad\quad \mathcal{D}_{tar} \leftarrow \mathcal{D}_{tar} \cup \text{ROLLOUT}(\pi, \mathcal{M}_{tar})$.
7: $\quad$ **end if**
8: $\quad$ Sample N source transitions $\{(s, a, r, s')^i_{src}\}_{i=1}^N$ with each transition's probability $p^i(s, a, s')$ computed via Eq (6) from $\mathcal{D}_{src}$.
9: $\quad$ Update transition priority in D_{source} via Eq (7).
10: $\quad$ Sample N target transitions $\{(s, a, r, s')^i_{tar}\}_{i=1}^N$ uniformly from $\mathcal{D}_{tar}$.
11: $\quad$ Create MixUp transitions $\{(s, a, r, s')^i_{mix}\}_{i=1}^N \leftarrow \text{MixUp}(\{(s, a, r, s')^i_{src}\}_{i=1}^N, \{(s, a, r, s')^i_{tar}\}_{i=1}^N)$ via Eq (8).
12: $\quad$ Modify the reward for each transition in source and mixup batch with Δr via Eq (10).
13: $\quad$ Train the source-target classifiers $\theta \leftarrow \theta - \eta \nabla_\theta \text{Cross-entropyLoss}(\mathcal{D}_{src}, \mathcal{D}_{tar}, \theta)$.
14: $\quad$ Train the modified_source-target classifiers $\phi \leftarrow \phi - \eta \nabla_\phi \text{Cross-entropyLoss}(\mathcal{D}_{src}, \mathcal{D}_{tar}, \phi)$.
15: $\quad$ Train the policy π with modified source and mixup transitions using any standard policy learning algorithm (e.g. SAC).
16: **end for**
17: **return** π.

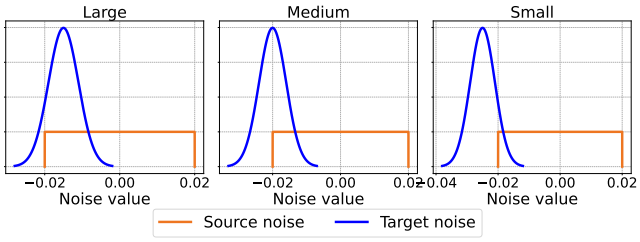

Figure 2: Visualization of the source and target noise distributions in Walker benchmark in three distinct deficient support levels.

5 EXPERIMENTS

In this section, we provide an empirical analysis of our proposed approach across different levels of deficient support: small support overlap (small), medium support overlap (medium), and large support overlap (large). Furthermore, through ablation studies, we delve deeper into the significance of each component in our method.

Environments: We use four simulated robot benchmarks from Mujoco Gym [2, 26]: Ant, HalfCheetah, Hopper and Walker. For each benchmark, we establish three levels of deficient support between the source and target domains: small overlapping, medium overlapping, and large overlapping. More specifically, for each benchmark, we first sample the noise ξ from a pre-defined distribution p_ξ and then add it to s' as follows:

$$s' \leftarrow s' + \xi, \text{ where } \xi \sim p_\xi. \tag{11}$$

We then update the value of the next state s' in the environment. By adding noises from distinct distributions with different support sets, we simulate different levels of support deficiency between the source and the target domain. Specifically, the noise applied to the source domain only partially overlaps with the noise introduced to the target domain. Thus, after adding different noises to the source and target, the source dynamics has the support deficiency w.r.t the target dynamics. Moreover, we adjust this overlapping region to create three different levels of support deficiency. Figure 2 illustrates the noises added to the source and the target domain at three deficient support levels in the Walker environment. Comprehensive details regarding the environments are provided in the *Appendix*.

Baselines: We compare our algorithm with six baselines. *DARC* [6], which uses the additional reward term to encourage the policy to not use source transitions with low likelihood. *GARAT* [4] that learns the grounded source environment obtained via action transformation and then trains the policy on the learned environment. The *Finetune* baseline first trains a policy on the source domain and then finetunes it with the limited transitions from the target domain. The *IW* (Importance Weighting) baseline trains the policy on importance-weighted samples from the source domain. Finally, the *RL on Target* trains the policy only using the target samples and can serve as Oracle. *RL on Source* trains the policy only using source samples. We run all algorithms with the same five random seeds. More details about the baseline settings are in the *Appendix*.

5.1 Off-dynamics Policy Evaluation

In Figure 3, we illustrate the off-dynamics policy performance of all methods across four Mujoco environments for the three support deficiency levels. Across all tasks, the performances of *RL on Source* are considerably lower than *RL on Target* performances, suggesting that directly transferring trained policies from the source to the target domain yields unsatisfactory performance when support deficiency is present.

In cases with substantial support overlap (as seen in large overlap settings), the performance differences between the methods are not too high. However, as support overlap decreases (in medium and small overlap settings), our approach, DADS, consistently excels in most tasks. Specifically, *GARAT* performances are significantly low for all tasks, supporting our intuition that its grounded action environment can be inaccurate and infeasible for policy learning under the dynamics mismatch and deficient support problems. The performances of *DARC* and *IW* drop significantly when the support overlap decreases (from a large level to a small level). Our method outperforms *DARC* and *IW* baselines for most of the tasks. We surpass the *Finetune* baseline in nine out of twelve cases, excluding HalfCheetah. While *Finetune* performs on par with our method on large and medium overlapping levels in HalfCheetah, it outperforms our method in the small overlap cases. We believe that this is because the agent never dies in the HalfCheetah environment, which helps it to transfer any learnings from the source domain to the target domain without any problem. Nonetheless, our method stands out as the only approach that asymptotically matches the performance of RL on Target (i.e. Oracle) and even surpasses *RL on Target* in six out of twelve tasks.

5.2 Ablation studies

In this section, we analyze the impact of each component and hyperparameter in our method. We provide the results for the Walker environment. The results for the other environments in all settings are provided in the *Appendix*.

5.2.1 The impact of Skewing operation: To validate the effect of the skewing operation, we compared our method to a variant that does not use this component. As shown in Figure 4, not including the skewing operation leads to a notable drop in performance and makes target return unstable, indicating the critical role of this component in our method.

5.2.2 The impact of MixUp operation: We evaluate the impact of the MixUp operation by removing it from our approach and retraining the policy. As shown in Figure 5, excluding MixUp results in a significant performance drop in target return, especially in the settings where support deficiencies are large (medium and small overlap settings). This verifies the effectiveness of MixUp component in our method.

Notably, omitting either the Skewing or MixUp operations results in a significant reduction in target returns. This observation confirms the effectiveness of both operations.

5.2.3 The impact of μ: Our method only has one hyperparameter μ that controls the strength of the source dynamics regularization in Equation 3. We conduct experiments with different source dynamics regularization μ values $(0, 1/3, 2/3, 1, 2, 4)$, where $\mu = 0$ mean

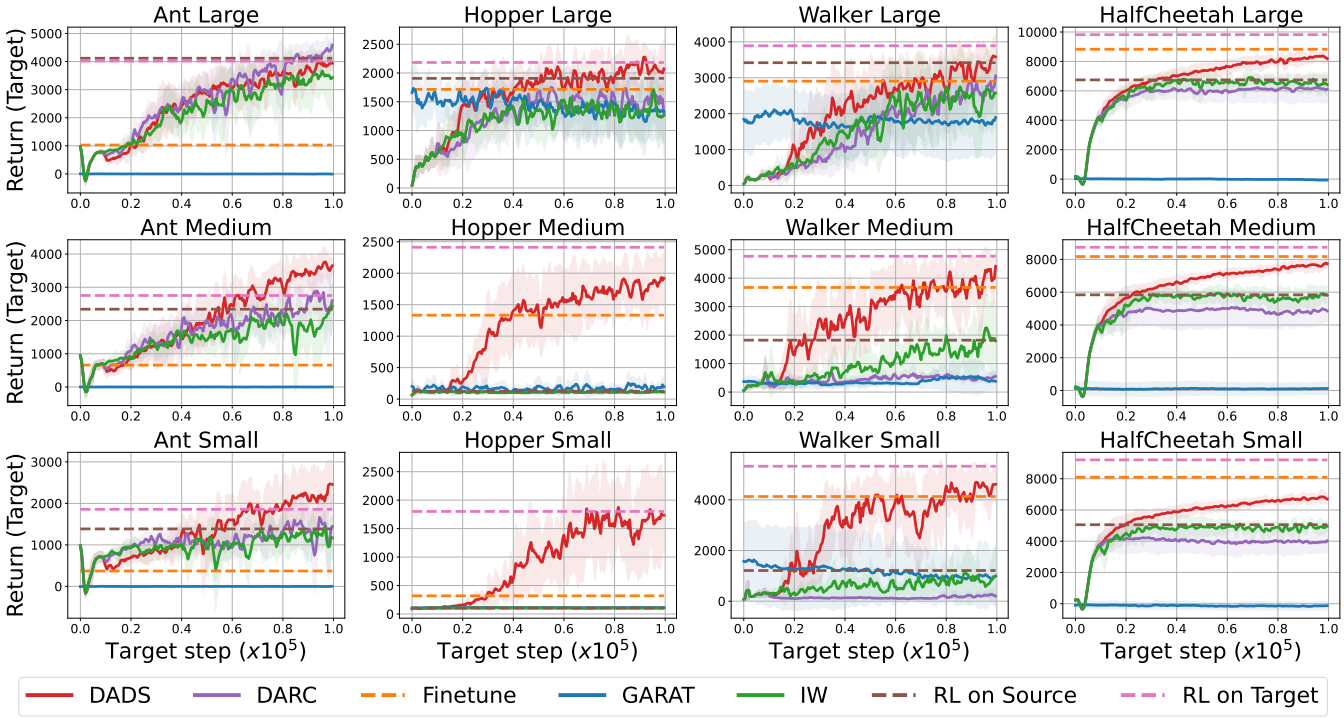

Figure 3: The target return of different methods in four Mujoco benchmarks with different deficient support levels: large overlapping support (Top row), medium overlapping support (Middle row), and small overlapping support (Bottom row). The solid curves are the average target returns over 5 runs with different random seeds, and the shaded areas represent standard deviation.

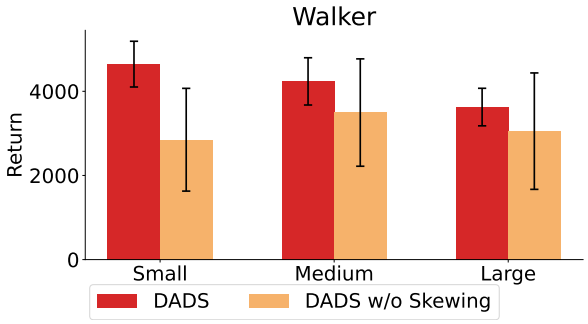

Figure 4: Comparison between our DADS method, and its variant without Skewing operation.

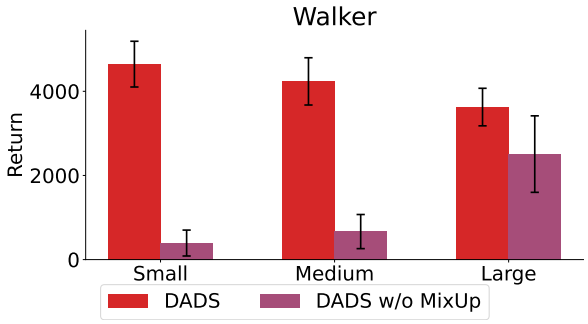

Figure 5: Comparison between our DADS method, and its variant without MixUp.

that we ignore the source dynamics regularization and increasing values of μ indicate higher weight for the source dynamics regularization. The results are shown in Figure 6. We can see that the best range for μ is from 2/3 to 1. If we decrease the value of μ to 0, the policy performance in the target domain drops significantly. The reason is that reducing or ignoring the source dynamics regularization results in sampling transitions that are too close to the target domain, which reduces the diversity of samples for the subsequent MixUp operation. On the other hand, a high value of μ might also result in reduced, unstable performance as there are not enough source transitions that are close to the target domain. This could also adversely affect MixUp operations. Thus source dynamics constraint is important for the effective performance of our algorithm. In our experiments, we used $\mu = 1$ due to the highest target return values.

6 CONCLUSION

In this paper, we have addressed the problem of off-dynamics RL under deficient support, which is widely encountered in many real-world applications. To the best of our knowledge, ours is the first

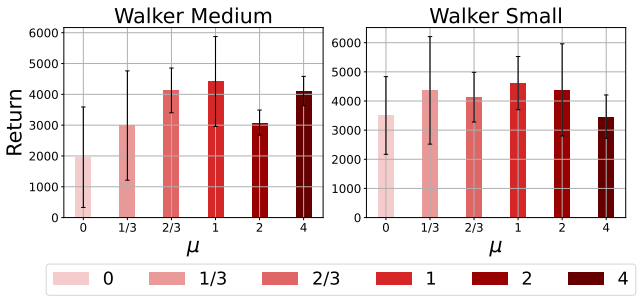

Figure 6: The adaptation performance of DADS with different values of μ.

work on this problem. We proposed DADS, a simple, yet effective method, that reduces the support deficiency of the source domain by modifying it through two operations: skewing and extension. The skewing is learned by solving an optimization problem and the extension is performed by using a Mixup operation between source and target transitions. Extensive experiments have demonstrated the effectiveness of our method compared to the existing state-of-the-art approaches for off-dynamics RL.

REFERENCES

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39, 1 (2020), 3–20.

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

[3] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. 2019. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 8973–8979.

[4] Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, and Peter Stone. 2020. An imitation from observation approach to transfer learning with dynamics mismatch. *Advances in Neural Information Processing Systems* 33 (2020), 3917–3929.

[5] Siddharth Desai, Haresh Karnan, Josiah P Hanna, Garrett Warnell, and Peter Stone. 2020. Stochastic grounded action transformation for robot learning in simulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 6106–6111.

[6] Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. 2020. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. *arXiv preprint arXiv:2006.13916* (2020).

[7] Nicolò Felicioni, Maurizio Ferrari Dacrema, Marcello Restelli, and Paolo Cremonesi. 2022. Off-policy evaluation with deficient support using side information. *Advances in Neural Information Processing Systems* 35 (2022), 30250–30264.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[9] Josiah Hanna and Peter Stone. 2017. Grounded action transformation for robot learning in simulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[10] Haresh Karnan, Siddharth Desai, Josiah P Hanna, Garrett Warnell, and Peter Stone. 2020. Reinforced grounded action transformation for sim-to-real transfer. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4397–4402.

[11] Svetoslav Kolev and Emanuel Todorov. 2015. Physically consistent state estimation and system identification for contacts. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 1036–1043.

[12] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* 17, 1 (2016), 1334–1373.

[13] Junfan Lin, Zhongzhan Huang, Keze Wang, Xiaodan Liang, Weiwei Chen, and Liang Lin. 2021. Continuous Transition: Improving Sample Efficiency for Continuous Control Problems via MixUp. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9490–9497.

[14] Jinxin Liu, Hongyin Zhang, and Donglin Wang. 2022. Dara: Dynamics-aware reward augmentation in offline reinforcement learning. *arXiv preprint arXiv:2203.06662* (2022).

[15] Lennart Ljung. 1998. System identification. In *Signal analysis and prediction*. Springer, 163–173.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[17] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. 2018. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 3803–3810.

[18] Noveen Sachdeva, Yi Su, and Thorsten Joachims. 2020. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 965–975.

[19] Fereshteh Sadeghi and Sergey Levine. 2016. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201* (2016).

[20] Yuta Saito, Qingyang Ren, and Thorsten Joachims. 2023. Off-Policy Evaluation for Large Action Spaces via Conjunct Effect Modeling. *arXiv preprint arXiv:2305.08062* (2023).

[21] Ryan Sander, Wilko Schwarting, Tim Seyde, Igor Gilitschenski, Sertac Karaman, and Daniela Rus. 2022. Neighborhood Mixup Experience Replay: Local Convex Interpolation for Improved Sample Efficiency in Continuous Control Tasks. In *Learning for Dynamics and Control Conference*. PMLR, 954–967.

[22] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (2015).

[23] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.

[24] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. *Density ratio estimation in machine learning*. Cambridge University Press.

[25] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. 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)*. IEEE, 23–30.

[26] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 5026–5033.

[27] Hung Tran-The, Sunil Gupta, Thanh Nguyen-Tang, Santu Rana, and Svetha Venkatesh. 2021. Combining Online Learning and Offline Learning for Contextual Bandits with Deficient Support. *arXiv preprint arXiv:2107.11533* (2021).

[28] Kaixin Wang, Bingyi Kang, Jie Shao, and Jiashi Feng. 2020. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems* 33 (2020), 7968–7978.

[29] Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. 2023. Cross-Domain Policy Adaptation via Value-Guided Data Filtering. *arXiv preprint arXiv:2305.17625* (2023).

[30] Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. 2017. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453* (2017).

[31] Grace Zhang, Linghan Zhong, Youngwoon Lee, and Joseph J Lim. 2021. Policy transfer across visual and dynamics domain gaps via iterative grounding. *arXiv preprint arXiv:2107.00339* (2021).

[32] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[33] Hanping Zhang and Yuhong Guo. 2022. Generalization of Reinforcement Learning with Policy-Aware Adversarial Data Augmentation. In *Decision Awareness in Reinforcement Learning Workshop at ICML 2022*.