Unsupervised Domain Adaptation with Dynamics-Aware Rewards in Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Unsupervised reinforcement learning aims to acquire skills without prior goal representations, where an agent automatically explores an open-ended environment to represent goals and learn the goal-conditioned policy. However, this procedure is often time-consuming, limiting the rollout in some potentially expensive target environments. The intuitive approach of training in another interaction-rich environment disrupts the reproducibility of trained skills in the target environment due to the dynamics shifts and thus inhibits direct transferring. Assuming free access to a source environment, we propose an unsupervised domain adaptation method to identify and acquire skills across dynamics. Particularly, we introduce a KL regularized objective to encourage emergence of skills, rewarding the agent for both discovering skills and aligning its behaviors respecting dynamics shifts. This suggests that both dynamics (source and target) shape the reward to facilitate the learning of adaptive skills. We also conduct empirical experiments to demonstrate that our method can effectively learn skills that can be smoothly deployed in target.

1 Introduction

Recently, the machine learning community has devoted attention to unsupervised reinforcement learning (RL) to acquire useful skills, ie, the problem of automatic discovery of a goal-conditioned policy and its corresponding goal space [7]. As shown in Figure 1 (left), the standard training procedure of learning skills in an unsupervised way follows: (1) representing goals, consisting of automatically generating the goal distribution $p(g)$ and the corresponding goal-achievement reward function r_g; (2) learning the goal-conditioned policy π_θ with the acquired $p(g)$ and r_g. Leveraging fully autonomous interaction with the environment, the agent sets up goals, builds the goal-achievement reward function, and extrapolates the goal-conditioned policy in parallel by adopting off-the-shelf RL methods [36, 18]. While we can obtain skills without any prior goal representations ($p(g)$ and r_g) in an unsupervised way, a major drawback of this approach is that it requires a large amount of rollout steps to represent goals and learn the policy itself, together. This procedure is often impractical in some target environments (eg, the robot in real world), where online interactions are time-consuming and potentially expensive.

That said, there often exist environments that resemble in structure (dynamics) yet provide more accessible rollouts (eg, unlimited in simulators). For problems with such source environments available, training the policy in a source environment significantly reduces the cost associated with interaction in the target environment. Critically, we can train a policy in one environment and deploy it in another by utilizing their structural similarity and the excess of interaction. However, it is reasonable to suspect that the learned policy overfits the training environment, the dynamics of which, dictating the goal distribution and reward function, implicitly shape goal representation and guide policy acquisition. Such deployment would then make the policy struggle to adapt to new, unseen environments and produce a large drop in performance in target due to the dynamics shifts, as shown

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

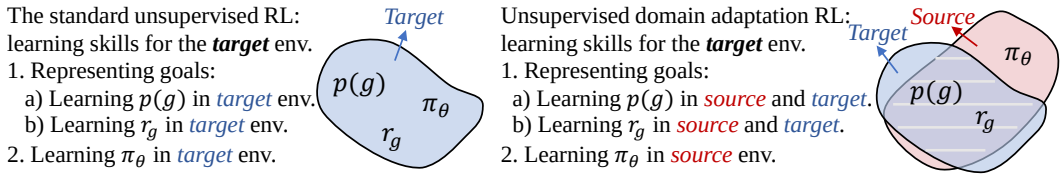


Figure 1: The training procedures of (left) the standard unsupervised RL in a single environment, and (right) the unsupervised domain adaptation RL in a pair of source and target environments. $p(g)$: the goal distribution; r_g: the goal-achievement reward; π_θ: the goal-conditioned policy.

in Figure 2 (top). In this paper, we overcome the limitations (of limited rollout in target and dynamics shifts) associated with the *(source, target)* environments pair through unsupervised domain adaptation.

In practice, while performing a *full* unsupervised RL method in target that represents goals and captures all of them for learning the entire goal-conditioned policy (Figure 1 left) can be extremely challenging with the limited rollout steps, learning a model for *only* (partially) representing goals is much easier. This gives rise to learning policy in source and taking the limited rollout in target into account only for identifying the goal representations, with which to further shape the policy. As shown in Figure 1 (right), we represent goals in both environments while optimizing the policy only in the source environment, alleviating the excessive need for the rollout steps in the target environment.

Furthermore, we introduce a KL regularization to address the challenge of dynamics shifts. This objective allows us to incorporate a reward modification into the goal-achievement reward function in the standard unsupervised RL, aligning the trajectory induced in the target environment against that induced in the source by the same policy. Importantly, it enables useful inductive biases towards the target dynamics: it allows the agent to specifically pursue skills that are competent in the target dynamics, and penalizes the agent for exploration in the source where the dynamics significantly differ. As show in Figure 2 (bottom), the difference in dynamics (a wall in the target while no wall in the source) will pose a penalty when the agent attempts to go through an area in the source wherein the target stands a wall. Thus, skills learned in source with such modification are adaptive to the target.

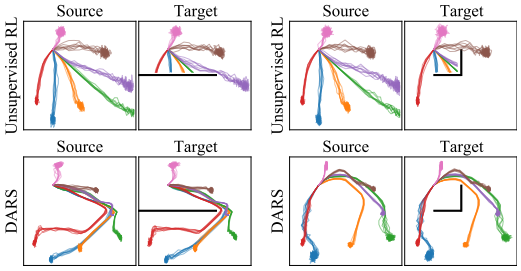


Figure 2: Skills *learned in the source environment*, each represented by a distinct color, are deployed in the source and target respectively. *Top plots* depict states visited by the standard unsupervised RL method, where skills fail to run in the target environment. *Bottom plots* depict trajectories induced by policy π_θ trained with our DARS, resulting in successful deployment in the target environment.

We name our method unsupervised domain adaptation with dynamics-aware rewards (DARS), suggesting that source and target dynamics both shape r_g: (1) we employ a latent-conditioned probing policy in the source to represent goals [29], making the goal-achievement reward source-oriented, and (2) we adopt two classifiers [10] to provide reward modification derived from the KL regularization. This means that the repertoires of skills are well shaped by the dynamics of both the source and target. Formally, we further analyze the conditions under which our DARS produces a near-optimal goal-conditioned policy for the target environment. Empirically, we demonstrate that our objective can obtain dynamics-aware rewards, enabling the goal-conditioned policy learned in a source to perform well in the target environment in various settings (stable and unstable settings, and sim2real).

2 Preliminaries

Multi-goal Reinforcement Learning: We formalize the multi-goal reinforcement learning (RL) as a goal-conditioned Markov Decision Process (MDP) defined by the tuple $\mathcal{M}_\mathcal{G} = \{S, A, \mathcal{P}, \mathcal{R}_G, \gamma, \rho_0\}$, where S denotes the state space and A denotes the action space. $\mathcal{P} : S \times A \times S \to \mathbb{R}_{\geq 0}$ is the transition probability density. $\mathcal{R}_G \triangleq \{G, r_g, p(g)\}$, where G denotes the space of goals, r_g denotes the corresponding goal-achievement reward function: $r_g : G \times S \times A \times S \to \mathbb{R}$, and $p(g)$ denotes the given goal distribution. γ is the discount factor and ρ_0 is the initial state distribution. Given a $g \sim p(g)$, the γ-discounted return $R(g, \tau)$ of a goal-oriented trajectory $\tau = (s_0, a_0, s_1, \ldots, s_T)$ is $\Sigma_{t=0}^{T-1} \gamma^t r_g(s_t, a_t, s_{t+1})$. Building on the universal value function approximators (UVFA, Schaul et al.

[35]), the standard multi-goal RL seeks to learn a unique goal-conditioned policy $\pi_\theta : A \times S \times G \to \mathbb{R}$ to maximize the objective $\mathbb{E}_{\mathcal{P},\rho_0,\pi_\theta,p(g)}[R(g,\tau)]$.

Unsupervised Reinforcement Learning: In unsupervised RL, the agent is set in an open-ended environment without any pre-defined goals or related reward functions. The agent aims to acquire a repertoire of skills. Following Colas et al. [7], we define skills as the association of goals and the goal-conditioned policy to reach them. The unsupervised skill acquisition problem can now be modeled by a goal-free MDP $\mathcal{M} = \{S, A, \mathcal{P}, \gamma, \rho_0\}$ that only characterizes the agent, its environment and their possible interactions. As shown in Figure 1 (left), the agent needs to autonomously interact with the environment and (1) *learn goal representations* (eg, discovering the goal distribution $p(g)$ and learning the corresponding reward r_g), and (2) *learn the goal-conditioned policy* π_θ as in multi-goal RL.

Here we define a universal (information theoretic) objective for learning the goal-conditioned policy π_θ in unsupervised RL, maximizing the mutual information $\mathcal{I}_{\mathcal{P},\rho_0,\pi_\theta}(g;\tau)$ between the goal g and the trajectory τ induced by policy π_θ running in the environment $\mathcal{M}$ (with $\mathcal{P}$ and ρ_0),

$$\max\ \mathcal{I}_{\mathcal{P},\rho_0,\pi_\theta}(g;\tau) = \mathcal{H}(g) - \mathcal{H}(g|\tau) = \mathcal{H}(g) + \mathbb{E}_{\mathcal{P},\rho_0,\pi_\theta,p(g)}[\log p(g|\tau)]. \tag{1}$$

For representing goals, the specific manifold of the goal space could be a set of *latent variables* (eg, one-hot indicators) or *perceptually-specific goals* (eg, the joint torques of ant). In the absence of any prior knowledge about $p(g)$, the maximum of $\mathcal{H}(g)$ will be achieved by fixing the distribution $p(g)$ to be uniform over all $g \in G$. The term $\mathbb{E}_{\mathcal{P},\rho_0,\pi_\theta,p(g)}[\log p(g|\tau)]$ in Equation 1 is analogous to the objective in the standard multi-goal RL, where the return $R(g,\tau)$ can be seen as the embodiment of $\log p(g|\tau)$. The objective specifically for learning $p(g|\tau)$ are normally optimized by lens of generative loss [31], or contrastive loss [38]. With the learned goal distribution $p(g)$ and reward $p(g|\tau)$, it is straightforward to learn the goal-conditioned policy π_θ using standard RL algorithms [36, 18]. In general, optimizations iteratively alternate for both representing skills ($p(g)$ and $p(g|\tau)$) and learning π_θ.

3 Unsupervised Domain Adaptation with Dynamics-Aware Rewards

3.1 Problem Formulation

Our work addresses domain adaptation in unsupervised RL, raising expectations that an agent trained without prior goal representations ($p(g)$ and r_g) in one environment can perform purposeful tasks in another. Following Wulfmeier et al. [48], we also focus on the domain adaptation of the dynamics, as opposed to states. In this work, we consider two environments characterized by MDPs $\mathcal{M}_\mathcal{S}$ (the source environment) and $\mathcal{M}_\mathcal{T}$ (the target environment), the dynamics of which are $\mathcal{P}_\mathcal{S}$ and $\mathcal{P}_\mathcal{T}$ respectively. Both MDPs share the same state and action spaces S, A, discount factor γ and initial state distribution ρ_0, while differing in the transition distributions $\mathcal{P}_\mathcal{S}$, $\mathcal{P}_\mathcal{T}$. Since the agent does not directly receive $\mathcal{R}_\mathcal{G}$ from either environment, we adopt the information theoretic $\mathcal{I}_{\mathcal{P},\rho_0,\pi_\theta}(g;\tau)$ to acquire skills, equivalently learning a policy π_θ that achieves distinguishable trajectory τ given a goal g by maximizing this objective. Note that for brevity, we now omit the ρ_0 term discussed in Section 2.

In our setup, agents can freely interact with the source $\mathcal{M}_\mathcal{S}$. However, it has access to limited rollouts in the target $\mathcal{M}_\mathcal{T}$ with which are insufficient to train a policy. To ensure that all potential trajectories in the target $\mathcal{M}_\mathcal{T}$ can be attempted in the source environment, we make the following assumption:

Assumption 1. *There is no transition that is possible in the target environment $\mathcal{M}_\mathcal{T}$ but impossible in the source environment $\mathcal{M}_\mathcal{S}$: $\mathcal{P}_\mathcal{T}(s_{t+1}|s_t,a_t) > 0 \implies \mathcal{P}_\mathcal{S}(s_{t+1}|s_t,a_t) > 0$.*

3.2 Domain Adaptation in Unsupervised RL

We aim to acquire skills trained in the source environment $\mathcal{M}_\mathcal{S}$, which can be deployed in the target environment $\mathcal{M}_\mathcal{T}$. To facilitate the unsupervised learning of skills in the target environment (with dynamics $\mathcal{P}_\mathcal{T}$), we maximize the mutual information between the trajectory and the goal (Figure 3 a):

$$\mathcal{I}_{\mathcal{P}_\mathcal{T},\pi_\theta}(g;\tau). \tag{2}$$

However, since interaction with the target environment $\mathcal{M}_\mathcal{T}$ is restricted, acquiring the goal-conditioned policy π_θ by optimizing the mutual information above is intractable. We instead maximize the mutual information in the source environment $\mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(g;\tau)$ modified by a KL divergence of trajectories induced by π_θ in both environments (Figure 3 b):

$$\mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(g;\tau) - \beta D_{\text{KL}}\left(p_{\mathcal{P}_\mathcal{S},\pi_\theta}(g,\tau) \| p_{\mathcal{P}_\mathcal{T},\pi_\theta}(g,\tau)\right), \tag{3}$$

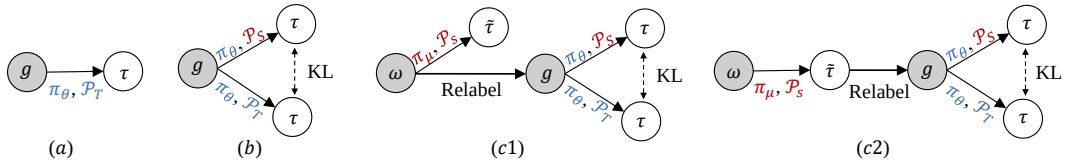


Figure 3: Graphical models of *(a)* the standard unsupervised RL, and DARS with goals *(b)* directly inputted, *(c1)* relabeled with latent variable ω, and *(c2)* relabeled with states induced by probing policy.

where $\beta > 0$ is the regularization coefficient, $p_{\mathcal{P}_\mathcal{S},\pi_\theta}(g,\tau)$ and $p_{\mathcal{P}_\mathcal{T},\pi_\theta}(g,\tau)$ denote the joint distributions of the goal g and the trajectory τ induced by policy π_θ in source $\mathcal{M}_\mathcal{S}$ and target $\mathcal{M}_\mathcal{T}$ respectively.

Intuitively, maximizing the mutual information term rewards distinguishable pairs of trajectories and goals, while minimizing the KL divergence term penalizes producing a trajectory that cannot be followed in the target environment. In other words, the KL term aligns the probability distributions of the mutual-information-maximizing trajectories under the two environment dynamics $\mathcal{P}_\mathcal{S}$ and $\mathcal{P}_\mathcal{T}$. This indicates that the dynamics of both environments ($\mathcal{P}_\mathcal{S}$ and $\mathcal{P}_\mathcal{T}$) shape the goal-conditioned policy π_θ (even though trained in the source $\mathcal{P}_\mathcal{S}$), allowing π_θ to adapt to the shifts in dynamics.

Building on the KL regularized objective in Equation 3, we introduce how to effectively represent goals: generating the goal distribution and acquiring the (partial) reward function. Here we assume the difference between environments in their dynamics negligibly affects the goal distribution[1]. Therefore, we follow GPIM [29] and train a latent-conditioned probing policy π_μ. The probing policy explores the source environment and represents goals for the source to train the goal-conditioned policy π_θ with. Specifically, the probing policy π_μ is conditioned on a latent variable $\omega \sim p(\omega)$[2] and aims to generate diverse trajectories that are further relabeled as goals for π_θ. Such goals can take the form of the latent variable ω itself (Figure 3 c1) or the final state of a trajectory (Figure 3 c2). We jointly optimize the previous objective in Equation 3 with the mutual information between ω and the trajectory induced by π_μ in source, and arrive at the following overall objective:

$$\max \mathcal{J}(\mu,\theta) \triangleq \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\mu}(\omega;\tilde{\tau}) + \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(g;\tau) - \beta D_{\mathrm{KL}}\left[p_{\mathcal{P}_\mathcal{S},\pi_\theta}(g,\tau) \| p_{\mathcal{P}_\mathcal{T},\pi_\theta}(g,\tau)\right], \tag{4}$$

where the context between $p(g)$ and $p(\omega)$ are specified by the graphic model in Figure 3 (c1 or c2). Note that this objective explicitly decouples goal representation (with π_μ) and policy learning (wrt π_θ), providing a foundation for the theoretical guarantee in Section 3.4.

3.3 Optimization with Dynamics-Aware Rewards

Similar to Goyal et al. [15], we take advantage of the data processing inequality (DPI [2]) which implies $\mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(g;\tau) \geq \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(\omega;\tau)$ from the graphical models in Figure 3 (c1, c2). Consequently, maximizing $\mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(g;\tau)$ can be achieved by maximizing the information of ω encoded progressively to π_θ. We therefore obtain the lower bound of Equation 4:

$$\mathcal{J}(\mu,\theta) \geq \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\mu}(\omega;\tilde{\tau}) + \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(\omega;\tau) - \beta D_{\mathrm{KL}}\left(p_{\mathcal{P}_\mathcal{S},\pi_\theta}(g,\tau) \| p_{\mathcal{P}_\mathcal{T},\pi_\theta}(g,\tau)\right). \tag{5}$$

For the first term $\mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\mu}(\omega;\tilde{\tau})$ and the second term $\mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(\omega;\tau)$, we derive the state-conditioned Markovian rewards following Jabri et al. [23]:

$$\mathcal{I}_{\mathcal{P},\pi}(\omega;\tau) \geq \frac{1}{T}\sum_{t=0}^{T-1}\left(\mathcal{H}(\omega) - \mathcal{H}(\omega|s_{t+1})\right) = \mathcal{H}(\omega) + \mathbb{E}_{p_{\mathcal{P},\pi}(\omega,s_{t+1})}\left[\log p(\omega|s_{t+1})\right] \tag{6}$$

$$\geq \mathcal{H}(\omega) + \mathbb{E}_{p_{\mathcal{P},\pi}(\omega,s_{t+1})}\left[\log q_\phi(\omega|s_{t+1})\right], \tag{7}$$

where $p_{\mathcal{P},\pi}(\omega, s_{t+1}) = p(\omega)p_{\mathcal{P},\pi}(s_{t+1}|\omega)$, and $p_{\mathcal{P},\pi}(s_{t+1}|\omega)$ refers to the state distribution (at time step $t+1$) induced by policy π conditioned on ω under the environment dynamics $\mathcal{P}$; the lower bound in Equation 7 derives from training a discriminator network q_ϕ due to the non-negativity of KL divergence, $\mathbb{E}_{p_\pi(s_{t+1})}\left[D_{\mathrm{KL}}(p(\omega|s_{t+1})||q_\phi(\omega|s_{t+1}))\right] \geq 0$. The new bound rewards the discriminator q_ϕ for summarizing agent's behavior with ω as well as encouraging a variety of states.

[1] See Appendix D for the extension when $\mathcal{M}_\mathcal{S}$ and $\mathcal{M}_\mathcal{T}$ have different goal distributions.

[2] Following DIAYN [9] and DADS [39], we set $p(\omega)$ as a fixed prior.

With the bound above, we construct the lower bound of the mutual information terms in Equation 5, taking the same q_ϕ:

$$\mathcal{F}_\mathcal{I} \triangleq \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\mu}(\omega;\tilde{\tau}) + \mathcal{I}_{\mathcal{P}_\mathcal{S},\pi_\theta}(\omega;\tau) \geq 2\mathcal{H}(\omega) + \mathbb{E}_{p_{\text{joint}}}\left[\log q_\phi(\omega|\tilde{s}_{t+1}) + \log q_\phi(\omega|s_{t+1})\right], \quad (8)$$

where p_{joint} denotes the joint distribution of ω, states $\tilde{s}_{t+1}$ and s_{t+1}. The states $\tilde{s}_{t+1}$ and s_{t+1} are induced by the probing policy π_μ conditioned on the latent variable ω and the policy π_θ conditioned on the relabeled goals respectively, both in the source environment (Figure 3 c1, c2).

Now, we are ready to characterize the KL term in Equation 5. Note that only the transition probabilities terms ($\mathcal{P}_\mathcal{S}$ and $\mathcal{P}_\mathcal{T}$) differ since agents in both environments follow the same policy π_θ. This conveniently leads to the expansion of the KL divergence as a sum of differences in log likelihoods of the transition dynamics: expansion $p_{\mathcal{P},\pi_\theta}(g,\tau) = p(g)\rho_0(s_0)\prod_{t=0}^{T-1}\left[\mathcal{P}(s_{t+1}|s_t,a_t)\pi_\theta(a_t|s_t,g)\right]$, where $\mathcal{P} \in \{\mathcal{P}_\mathcal{S}, \mathcal{P}_\mathcal{T}\}$, gives rise to the following simplification of the KL term in Equation 5:

$$\beta D_{\text{KL}}\left(p_{\mathcal{P}_\mathcal{S},\pi_\theta}(g,\tau)\|p_{\mathcal{P}_\mathcal{T},\pi_\theta}(g,\tau)\right) = \mathbb{E}_{\mathcal{P}_\mathcal{S},\pi_\theta}\left[\beta\Delta r(s_t,a_t,s_{t+1})\right], \quad (9)$$

where $\Delta r(s_t,a_t,s_{t+1}) \triangleq \log \mathcal{P}_\mathcal{S}(s_{t+1}|s_t,a_t) - \log \mathcal{P}_\mathcal{T}(s_{t+1}|s_t,a_t)$.

Combining the lower bound of the mutual information terms (Equation 8) and the KL divergence term pursuing the aligned trajectories in two environments (Equation 9), we optimize $\mathcal{J}(\mu,\theta)$ by maximizing the following bound:

$$2\mathcal{H}(\omega) + \mathbb{E}_{p_{\text{joint}}}\left[\log q_\phi(\omega|\tilde{s}_{t+1}) + \log q_\phi(\omega|s_{t+1})\right] - \mathbb{E}_{\mathcal{P}_\mathcal{S},\pi_\theta}\left[\beta\Delta r(s_t,a_t,s_{t+1})\right]. \quad (10)$$

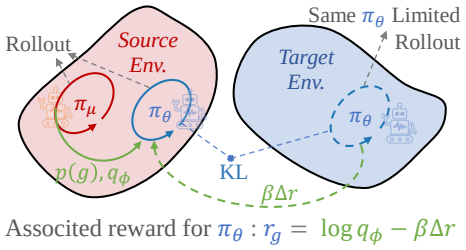


Figure 4: Framework of DARS: probing policy π_μ provides $p(g)$ and q_ϕ for learning π_θ, associated with the modification $\beta\Delta r$.

Overall, as shown in Figure 4, DARS rewards the goal-conditioned policy π_θ with the dynamics-aware rewards (associating $\log q_\phi$ with $\beta\Delta r$), where q_ϕ is shaped by the source dynamics and $\beta\Delta r$ is derived from the difference of the two dynamics. This indicates that the learned policy is shaped by both dynamics, holding the promise of acquiring adaptive skills for the target by training mostly in source.

3.4 Optimality Analysis

Here we discuss the conditions under which our method produces near-optimal skills for the target environment. We first mildly require that the most suitable policy for the target environment does not produce drastically different trajectories in the source environment:

Assumption 2. *Let $\pi^* = \arg\max_\pi \mathcal{I}_{\mathcal{P}_\mathcal{T},\pi}(g;\tau)$ be the policy that maximizes the (non-kl-regularized) objective in the target environment (Equation 2). Then the joint distributions of the goal and its trajectories differ in both environments by no more than a small number $\epsilon/\beta > 0$:*

$$D_{KL}(p_{\mathcal{P}_\mathcal{S},\pi^*}(g,\tau)||p_{\mathcal{P}_\mathcal{T},\pi^*}(g,\tau)) \leq \frac{\epsilon}{\beta}. \quad (11)$$

Given a desired joint distribution $p^*(g,\tau)$ (inferred from a potential goal representation), our problem can be reformulated as finding a closest match [27, 26]. Consequently, we quantify the optimality of a policy π by measuring $D_{\text{KL}}\left(p_{\mathcal{P},\pi}(g,\tau)\|p^*_\mathcal{P}(g,\tau)\right)$, the discrepancy between its joint distribution and the desired one. With a potential goal representation, we prove that its joint distributions with the trajectories induced by our policy and the optimal one satisfy the following theoretical guarantee.

Theorem 1. *Let π^*_{DARS} be the optimal policy that maximizes the KL regularized objective in the source environment (Equation 3), let π^* be the policy that maximizes the (non-regularized) objective in the target environment (Equation 2), let $p^*_{\mathcal{P}_\mathcal{T}}(g,\tau)$ be the desired joint distribution of trajectory and goal in the target (with the potential goal representations), and assume that π^* satisfies Assumption 2. Then the following holds:*

$$D_{KL}\left(p_{\mathcal{P}_\mathcal{T},\pi^*_{DARS}}(g,\tau)\|p^*_{\mathcal{P}_\mathcal{T}}(g,\tau)\right) \leq D_{KL}\left(p_{\mathcal{P}_\mathcal{T},\pi^*}(g,\tau)\|p^*_{\mathcal{P}_\mathcal{T}}(g,\tau)\right) + 2\sqrt{\frac{2\epsilon}{\beta}}L_{max},$$

where L_{max} refers to the worst case absolute difference between log likelihoods of the desired joint distribution and that induced by a policy.

Please see Appendix C for more details and the proof of the theorem. Note that Theorem 1 requires a potential goal representation, which can be precisely provided by the probing policy π_μ in Equation 4.

Algorithm 1 DARS

Input: source and target MDPs $\mathcal{M}_\mathcal{S}$ and $\mathcal{M}_\mathcal{T}$; ratio R of experience from source vs. target.
Output: goal-reaching policy π_θ.
Initialize parameters μ, θ, ϕ and ψ.
Initialize buffers $\tilde{\mathcal{B}}_\mathcal{S}$, $\mathcal{B}_\mathcal{S}$ and $\mathcal{B}_\mathcal{T}$.
for $iter = 0, \ldots,$ MAX_ITER **do**
  Sample latent variable: $\omega \sim p(\omega)$.
  Collect probing data in source: $\tilde{\mathcal{B}}_\mathcal{S} \leftarrow \tilde{\mathcal{B}}_\mathcal{S} \cup \text{ROLLOUT}(\pi_\mu, \mathcal{M}_\mathcal{S}, \omega)$.
  Update discriminator q_ϕ: $\phi \leftarrow \text{Update}(\phi, \tilde{\mathcal{B}}_\mathcal{S})$.
  Set reward function for the probing policy π_μ: $\tilde{r} = \log q_\phi(\omega|\tilde{s}_{t+1})$.
  Train probing policy π_μ: $\mu \leftarrow \text{SAC}(\mu, \tilde{\mathcal{B}}_\mathcal{S}, \tilde{r})$.
  Relabel goals: # According to Figure 3 (c1, c2) $g \leftarrow \text{Relabel}(\omega, \tilde{\tau})$.
  Collect source data: $\mathcal{B}_\mathcal{S} \leftarrow \mathcal{B}_\mathcal{S} \cup \text{ROLLOUT}(\pi_\theta, \mathcal{M}_\mathcal{S}, g, \omega)$.
  if $iter$ mod $R = 0$ **then**
    Collect target data: $\mathcal{B}_\mathcal{T} \leftarrow \mathcal{B}_\mathcal{T} \cup \text{ROLLOUT}(\pi_\theta, \mathcal{M}_\mathcal{T}, g)$.
  end if
  Update classifiers q_ψ for computing Δr: $\psi \leftarrow \text{Update}(\psi, \mathcal{B}_\mathcal{S}, \mathcal{B}_\mathcal{T})$. # See [10]
  Set reward function for π_θ: $r_g \leftarrow \log q_\phi(\omega|\tilde{s}_{t+1}) - \beta\Delta r(s_t, a_t, s_{t+1})$.
  Train policy π_θ: $\theta \leftarrow \text{SAC}(\theta, \mathcal{B}_\mathcal{S}, r_g)$.
end for

3.5 Implementation

As shown in Algorithm 1, we alternately train the probing policy π_μ and the goal-conditioned policy π_θ by optimizing the objective in Equation 10 with respect to μ, ϕ, θ and Δr. In the first phase, we update π_μ with reward $\tilde{r} = \log q_\phi(\omega|\tilde{s}_{t+1})$. This is compatible with most RL methods and we refer to SAC here. We additionally optimize discriminator q_ϕ with SGD to maximizing $\mathbb{E}_{\omega,\tilde{s}_{t+1}}\left[q_\phi(\omega|\tilde{s}_{t+1})\right]$ at the same time. Similarly, π_θ is updated with $r_g = \log q_\phi(\omega|s_{t+1}) - \beta\Delta r$ by SAC in the second phase, where π_θ also collects data in both environments to approximate Δr by training the same classifiers q_ψ (wrt state-action and state-action-next-state) as in [10] according to Bayes' rule.

3.6 Connections to Prior Work

Unsupervised RL: Two representative unsupervised RL approaches acquire (diverse) skills by maximizing empowerment [9, 39] or minimizing surprise [3]. Liu et al. [29] also employs a latent-conditioned policy to explore the environment and relabels goals along with the corresponding reward, which can be considered as a special case of DARS with identical source and target environments. However, none of these methods can produce skills tailored to a new environment with dynamics shifts.

Off-Dynamics RL: Eysenbach et al. [10] proposes domain adaptation with rewards from classifiers (DARC), adopting the control as inference framework [27] to maximize $-D_{\text{KL}}(p_{\mathcal{P}_\mathcal{S},\pi_\theta}(\tau)\|p^*_{\mathcal{P}_\mathcal{T}}(\tau))$, but this objective cannot be directly applied to the unsupervised setting. While we adopt the same classifier to provide the reward modification, one major distinction of our work is that we do not require a given goal distribution $p(g)$ or a prior reward function r_g. However, assuming an extrinsic goal-reaching reward in the source environment (ie, the potential $p^*_{\mathcal{P}_\mathcal{S}}(\tau)$), our proposed DARS can be simplified to a *decoupled objective*: maximizing $-D_{\text{KL}}(p_{\mathcal{P}_\mathcal{S},\pi_\theta}(\tau)\|p^*_{\mathcal{P}_\mathcal{S}}(\tau)) - \beta D_{\text{KL}}(p_{\mathcal{P}_\mathcal{S},\pi_\theta}(\tau)\|p_{\mathcal{P}_\mathcal{T},\pi_\theta}(\tau))$. Particularly, DARC can be considered as a special case of our decoupled objective with the restriction — a prior goal specified by its corresponding reward and $\beta = 1$. In Appendix E, we show that the stronger pressure ($\beta > 1$) for the KL term to align the trajectories puts extra reward signals for the policy π_θ to be Δr oriented while still being sufficient to acquire skills.

4 Experiments

In this section, we aim to experimentally answer the following questions: (1) Can our method DARS learn diverse skills, in the source environment, that can be executed in the target environment and keep the same embodiment in the two environments? Specifically, can our proposed associated dynamics-aware rewards ($\log q_\phi - \beta\Delta r$) reveal the perceptible dynamics of the two environments? (2) Does DARS lead to better transferring in the presence of dynamics mismatch, compared to other related approaches, in both stable and unstable environments? (3) Can DARS contribute to acquiring behavioral skills under the sim2real circumstances, with limited interaction in the real world?

We adopt tuples *(source, target)* to denote the source and target environment pairs, with details of the corresponding MDPs in Appendix F.2. Illustrations of the environments are shown in Figure 5.

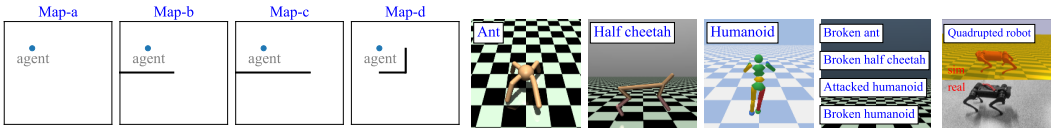


Figure 5: We evaluate our method in 10 *(source, target)* transition tasks, where the shifts in dynamics are either external (the map pairs and the attacked series) or internal (the broken series) to the robot.

For all tuples, we set $\beta = 10$ and the ratio of experience from the source environment vs. the target environment $R = 10$ (Line 13 in Algorithm 1). See Appendix F.3 for the other hyperparameters.

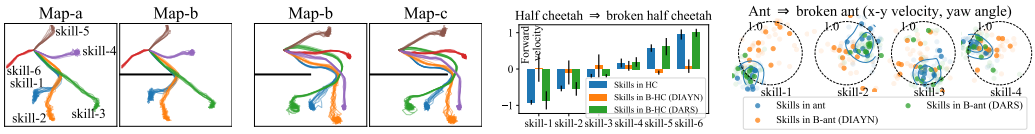
Map. We consider the maze environments: *Map-a*, *Map-b*, *Map-c* and *Map-d*, where the wall can block the agent (a point), which can move around to explore the environment. For the domain adaptation tasks, we consider the following five *(source, target)* pairs: *(Map-a, Map-b)*, *(Map-a, Map-c)*, *(Map-a, Map-d)*, *(Map-b, Map-c)* and *(Map-b, Map-d)*.

Mujoco. We use two simulated robots from OpenAI Gym [4]: half cheetah (*HC*) and ant. We define two new environments by crippling one of the joints of each robot (*B-HC* and *B-ant*) as described in [10], where *B-* is short for broken. The *(source, target)* pairs include: *(HC, B-HC)* and *(ant, B-ant)*.

Humanoid. In this environment, a (source) simulated humanoid (*H*) agent must avoid falling in the face of the gravity disturbances. Two target environments each contain a humanoid *attacked* by blocks from a fixed direction (*A-H*) and a humaniod with a part of *broken* joints (*B-H*).

Quadruped robot. We also consider the sim2real setting for transferring the simulated quadruped robot to a real quadruped robot. For more evident comparison, we break the left hind leg of the real-world robot (see Appendix F.2). We adopt *(sim-robot, real-robot)* to denote this sim2real transition.

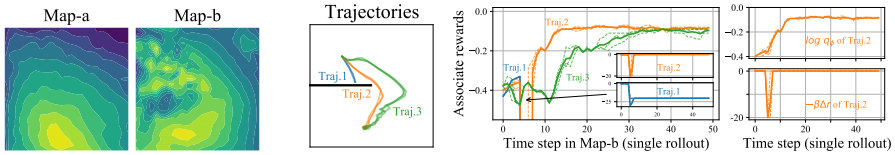
4.1 Emergent Behaviors with DARS



(a) *(Map-a, Map-b)* (b) *(Map-b, Map-c)* (c) *(HC, B-HC)* (d) *(ant, B-ant)*

Figure 6: Visualization of skills. *(a, b)*: colored trajectories in *map* pairs depict the skills, learned with DARS, deployed in source (left) and target (right). *(d, e)*: colored bars and dots depict the velocity of each skill wrt different environments of *mujoco* and models. The variation (blue) across velocities for *HC* and *ant* confirms the diversity of skills. DARS demonstrates its better adaptability by performing similarly on the broken agents (green) to the original ones while DIAYN (orange) fails to do so.

Visualization of the learned skills. We first apply DARS to the *map* pairs and the *mujoco* pairs, where we learn the goal-conditioned policy π_θ in the source environments with our dynamics-aware rewards ($\log q_\phi - \beta\Delta r$). Here, we relabel the latent random variable ω as the goal g for the goal-conditioned policy π_θ: $g \triangleq \text{Relabel}(\pi_\mu, \omega, \tilde{\tau}) = \omega$ (Figure 3 c1). The learned skills are shown in Figures 2, 6 and Appendix E. We can see that the skills learned by our method keep the same embodiment when they are deployed in the source and target environments. If we directly apply the skills learned in the source environment (without $\beta\Delta r$), the dynamics mismatch is likely to disrupt the skills (see Figure 2 *top*, and the deployment of DIAYN in *half cheetah* and *ant* pairs in Figure 6).



(a) Heatmaps of $\log q_\phi$. (b) Three trajectories and and the associated rewards.

Figure 7: *(a)*: The value of $\log q_\phi$ in *Map-a* for *(Map-a, Map-b)* and $\log q_\phi$ in *Map-b* for *(Map-b, Map-c)*. *(b)*: Three trajectories in *Map-b* for the *(Map-b, Map-c)* task, and the recorded rewards.

The dynamics-aware rewards. To gain more intuition for the proposed dynamics-aware rewards capturing the perceptible dynamics of both the source and target environments, we visualize the

learned probing reward $\log q_\phi$ and the modification $\beta\Delta r$ throughout the training for *(Map-a, Map-c)* and *(Map-b, Map-c)* pairs in Figure 7.

The probing policy learns q_ϕ by summarizing the behaviors with the latent random variable ω in source environments. Setting *Map-a* as the source (Figure 7 (a) *left*), we can see that $\log q_\phi$ resembles the usual L2-norm-based punishment. Further, in the pair *(Map-b, Map-c)*, we can find that the learned $\log q_\phi$ is well shaped by the dynamics of the source environment *Map-b* (Figure 7 (a) *right*): even if the agent simply moves in the direction of reward increase, it almost always sidesteps the wall and avoids the entrapment in a local optimal solution produced by the usual L2-norm based reward.

To see how the modification $\beta\Delta r$ guides the policy, we track three trajectories (with the same goal) and the associated rewards ($\log q_\phi - \beta\Delta r$) in the *(Map-b, Map-c)* task, as shown in Figure 7 (b). We see that *Traj.2* receives an incremental $\log q_\phi$ along the whole trajectory while a severe punishment from $\beta\Delta r$ around step 6. This indicates that *Traj.2* is inapplicable to the target dynamics (*Map-c*), even if it is feasible in the source (*Map-b*). With this modification, we indeed obtain the adaptive skills (eg. *Traj.3*) by training in the source. This answers our first question, where both dynamics (source and target) explicitly shape the associated rewards, guiding the skills to be domain adaptive.

4.2 Comparison with Baselines

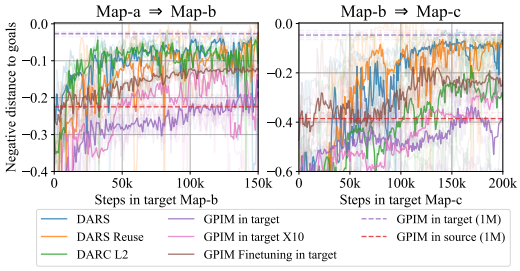


Figure 8: Comparison (training process) with alternative methods for learning skills for target environments. We plot each random seed as a transparent line; each solid line corresponds to the average across four random seeds; the dashed lines denote the performance of trained policies.

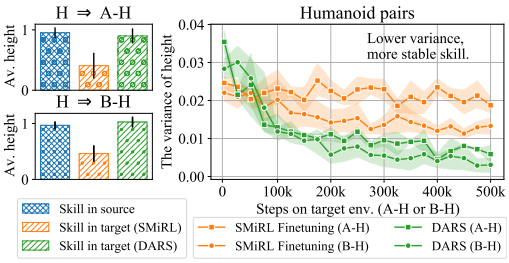


Figure 9: (left): The visualization of skills for humanoid (avoid falling) and the comparisons with *SMiRL Finetuning*, where the stable skills for humanoid keep the average height around 1. (right): Training process. The decrease in the variance of the height implies the emergence of a stable skill.

Behaviors in stable environments. For the second question, we apply our method to state-reaching tasks: $g \triangleq \text{Relabel}(\pi_\mu, \omega, \tilde{\tau}) = \tilde{s}_T$ (Figure 3 c2). We adopt the negative L2 norm (between the goal and the final state in each episode) as the distance metric. We compare our method (*DARS*) against six alternative goal-reaching strategies[3]: (1) additionally updating π_θ with data $\mathcal{B}_\mathcal{T}$ collected in the target (*DARS Reuse*); (2) employing DARC with a negative L2-norm-based reward (DARC L2); training skills with GPIM in the source and target respectively (3) *GPIM in source* and (4) *GPIM in target*); (5) updating GPIM in the target 10 times more (*GPIM in target X10*; $R = 10$ and see more interpretation in [10]); (6) finetuning *GPIM in source* in the target (*GPIM Finetuning in target*).

We report the results in Figure 8. *GPIM in source* performs much worse than *DARS* due to the dynamics shifts as we show in Section 4.1. With the same amount of rollout steps in the target, *DARS* achieves better performance than *GPIM in target X10* and *GPIM Finetuning in target*, and approximates *GPIM in target* within 1M steps in effectiveness, suggesting that the modification $\beta\Delta r$ provides sufficient information regarding the target dynamics. Further, reusing the buffer $\mathcal{B}_\mathcal{T}$ (*DARS Resue*) does not significantly improve the performance. Despite not requiring a prior reward function, our unsupervised DARS reaches comparable performance to (supervised) *DARC L2* in *(Map-a, Map-b)* pair. The more exploratory task *(Map-a, Map-b)* further reinforces the advantage of our dynamics-aware rewards, where the probing policy π_μ boosts the representational potential of q_ϕ.

Behaviors in unstable environments. Further, when we set $p(\omega)$ as the Dirac distribution: $p(\omega) = \delta(\omega)$, the discriminator q_ϕ will degrade to a density estimator: $q_\phi(s_{t+1})$, which keeps the same form

[3]We do not compare with other unsupervised RL methods (eg. Warde-Farley et al. [47]) because they generally study the rewards wrt the high-dimensional states. DARS does not focus on high-dimensional states. Domain randomization [32, 44] and system (dynamics) identification [11, 6, 1] are also not compared because they requires the access of the physical parameters of source environment, while we do not assume this access.

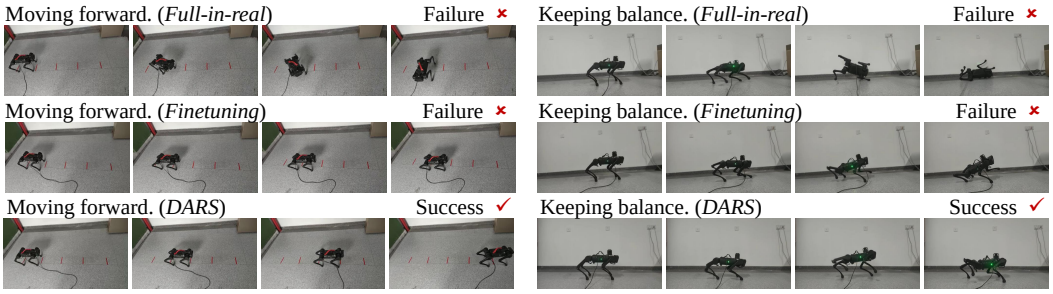


Figure 10: Deploying the learned skills into the real quadruped robot, where all models are trained with limited interaction (three hours for moving forward and one hour for keeping balance) in real.

as in SMiRL [3]. Assuming the environment will pose unexpected events to the agent, SMiRL seeks out stable and repeatable situations that counteract the environment's prevailing sources of entropy.

With such properties, we evaluate DARS in unstable environment pairs, where the sources and the targets are both unstable and exhibit dynamics mismatch. Figure 9 (left) charts the emergence of a stable skill with DARS, while SMiRL suffers from the failure of domain adaptation for both *(H, A-H)* and *(H, B-H)*. Figure 9 (right) shows the comparisons with *SMiRL Finetuning*, denoting training in the source and then finetuning in the target with SMiRL. With the same amount of rollout steps, we can find that *DARS* can learn a more stable skill for the target than *SMiRL Finetuning*, revealing the competence of our regularization term for learning adaptive skills even in the unstable environments.

4.3 Sim2real Transfer on Quadruped Robot

We now deploy our DARS on pair *(sim-robot, real-robot)* to learn diverse skills (moving forward and moving backward) and keeping balance skill in stable and unstable setting respectively. We compare DARS with two baselines: (1) training directly in the real world (*Full-in-real*), (2) finetuning the model, pre-trained in simulator, in real (*Finetuning*). As shown in Figure 10, after three hours (or one hour) of real-world interaction, our DARS demonstrates the emergence of moving skills (or the keeping balance skill), while baselines are unable to do so. As shown in Table 1, *Finetuning* takes significantly more time (four hours vs. one hour) to discovery keeping balance skill in the unstable setting, and the other three comparisons are unable to acquire valid skills given six hours interaction in the real world. Supplementary material contains videos from this sim2real deployment.

Table 1: Time (hours) spent for valid skill emergence in real-world interaction (covering the manual reset time).

	forward & backward	keeping balance
Full-in-real	> 6 h	> 6 h
Finetuning	> 6 h	4 h
DARS	3 h	1 h

5 Related Work

The proposed DARS has interesting connections with unsupervised learning and transfer learning [49] in RL. Most approaches in this field consider learning features [17, 37]of high-dimensional (eg, image-based) states in MDP, then design rewards [22, 31, 38, 47, 40, 30] or enable transfer [22, 15, 16, 12, 21] over the learned features. These can be seen as a procedure on the perception level [19], while we focus on the action level wrt the dynamics of environments, and consider both cases (learning reward and transfer). Previous works on the action level have either focused on learning dynamics-oriented rewards [20, 45, 43, 29] or only considered the transition-oriented modification [10, 48, 24, 13, 8, 46, 28]. Thus, the desirability of our approach is that the acquired reward function uncovers *both* the source's dynamics (q_ϕ) and the difference of transfer across dynamics ($\beta\Delta r$). Complementary to our work, several other works also encourage the emergence of a state-covering goal distribution [34, 5, 25] or transferring policies instead of dynamics [41, 14, 42, 33].

6 Conclusion

In this paper, we propose DARS to acquire adaptive skills for a target environment by training mostly in a source environment especially in the presence of dynamics shifts. We employ a probing policy in source to represent goals and introduce a KL regularization to further identify consistent behaviors for both environments. We show that our method obtains a near-optimal policy for target, as long as a mild assumption is met. Experiments on a range of tasks confirm the effectiveness of our approach.

References

[1] Adam Allevato, Elaine Schaertl Short, Mitch Pryor, and Andrea Thomaz. Tunenet: One-shot residual tuning for system identification and sim-to-real robot task transfer. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 445–455. PMLR, 2019. URL `http://proceedings.mlr.press/v100/allevato20a.html`.

[2] Normand J. Beaudry and Renato Renner. An intuitive proof of the data processing inequality, 2012.

[3] Glen Berseth, Daniel Geng, Coline Devin, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Smirl: Surprise minimizing RL in dynamic environments. *CoRR*, abs/1912.05510, 2019. URL `http://arxiv.org/abs/1912.05510`.

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[5] Victor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró-i-Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. 119: 1317–1327, 2020. URL `http://proceedings.mlr.press/v119/campos20a.html`.

[6] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan D. Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 8973–8979. IEEE, 2019. doi: 10.1109/ICRA.2019.8793789. URL `https://doi.org/10.1109/ICRA.2019.8793789`.

[7] Cédric Colas, Tristan Karch, Olivier Sigaud, and Pierre-Yves Oudeyer. Intrinsically motivated goal-conditioned reinforcement learning: a short survey. *CoRR*, abs/2012.09830, 2020. URL `https://arxiv.org/abs/2012.09830`.

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

[9] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

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

[11] Alon Farchy, Samuel Barrett, Patrick MacAlpine, and Peter Stone. Humanoid robots learning to walk faster: From the real world to simulation and back. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 39–46, 2013.

[12] Alexandre Galashov, Siddhant M. Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M. Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in kl-regularized rl, 2019.

[13] Tanmay Gangwani and J. Peng. State-only imitation with transition dynamics mismatch. *ArXiv*, abs/2002.11879, 2020.

[14] Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-conquer reinforcement learning, 2018.

[15] Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. Infobot: Transfer and exploration via the information bottleneck, 2019.

[16] Anirudh Goyal, Shagun Sodhani, Jonathan Binas, Xue Bin Peng, Sergey Levine, and Yoshua Bengio. Reinforcement learning with competitive ensembles of information-constrained primitives. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryxgJTEYDr.

[17] Zhaohan Daniel Guo, Bernardo Ávila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3875–3886. PMLR, 2020. URL http://proceedings.mlr.press/v119/guo20g.html.

[18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[19] Danijar Hafner, Pedro A. Ortega, Jimmy Ba, Thomas Parr, Karl J. Friston, and Nicolas Heess. Action and perception as divergence minimization. *CoRR*, abs/2009.01791, 2020. URL https://arxiv.org/abs/2009.01791.

[20] Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. *arXiv preprint arXiv:1907.08225*, 2019.

[21] Leonard Hasenclever, Fabio Pardo, Raia Hadsell, Nicolas Heess, and Josh Merel. Comic: Complementary task learning & mimicry for reusable skills. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 4105–4115. PMLR, 2020. URL http://proceedings.mlr.press/v119/hasenclever20a.html.

[22] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1480–1490. JMLR. org, 2017.

[23] Allan Jabri, Kyle Hsu, Abhishek Gupta, Ben Eysenbach, Sergey Levine, and Chelsea Finn. Unsupervised curricula for visual meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 10519–10531, 2019.

[24] Kuno Kim, Yihong Gu, Jiaming Song, Shengjia Zhao, and S. Ermon. Domain adaptive imitation learning. In *ICML*, 2020.

[25] G. Kovač, A. Laversanne-Finot, and Pierre-Yves Oudeyer. Grimgep: Learning progress for robust goal sampling in visual deep reinforcement learning. *arXiv: Learning*, 2020.

[26] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching, 2020.

[27] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018. URL http://arxiv.org/abs/1805.00909.

[28] Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. *arXiv preprint arXiv:1911.10947*, 2019.

[29] Jinxin Liu, Donglin Wang, Qiangxing Tian, and Zhengyu Chen. Learn goal-conditioned policy with intrinsic motivation for deep reinforcement learning, 2021. URL https://openreview.net/forum?id=MmcywoW7PbJ.

[30] Ashvin Nair, Shikhar Bahl, Alexander Khazatsky, Vitchyr Pong, Glen Berseth, and Sergey Levine. Contextual imagined goals for self-supervised robotic learning. In *Conference on Robot Learning*, pages 530–539. PMLR, 2020.

[31] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pages 9191–9200, 2018.

[32] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018. doi: 10.1109/icra.2018.8460528. URL http://dx.doi.org/10.1109/ICRA.2018.8460528.

[33] Janith C. Petangoda, Sergio Pascual-Diaz, Vincent Adam, Peter Vrancx, and Jordi Grau-Moya. Disentangled skill embeddings for reinforcement learning. *CoRR*, abs/1906.09223, 2019. URL http://arxiv.org/abs/1906.09223.

[34] Vitchyr H. Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning, 2020.

[35] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320, 2015.

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

[37] Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. 2021.

[38] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.

[39] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

[40] Avi Singh, Larry Yang, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. In Antonio Bicchi, Hadas Kress-Gazit, and Seth Hutchinson, editors, *Robotics: Science and Systems XV, University of Freiburg, Freiburg im Breisgau, Germany, June 22-26, 2019*, 2019. doi: 10.15607/RSS.2019.XV.073. URL https://doi.org/10.15607/RSS.2019.XV.073.

[41] Daniel Strouse, Max Kleiman-Weiner, Josh Tenenbaum, Matthew Botvinick, and David J. Schwab. Learning to share and hide intentions using information regularization, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/1ef03ed0cd5863c550128836b28ec3e9-Abstract.html.

[42] Yee Whye Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/0abdc563a06105aee3c6136871c9f4d1-Abstract.html.

[43] Stephen Tian, Suraj Nair, Frederik Ebert, Sudeep Dasari, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Model-based visual planning with self-supervised functional distances. *CoRR*, abs/2012.15373, 2020. URL https://arxiv.org/abs/2012.15373.

[44] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world, 2017.

[45] Srinivas Venkattaramanujam, Eric Crawford, Thang Doan, and Doina Precup. Self-supervised learning of distance functions for goal-conditioned reinforcement learning. *CoRR*, abs/1907.02998, 2019. URL http://arxiv.org/abs/1907.02998.

[46] Luca Viano, Y. Huang, P. Kamalaruban, and V. Cevher. Robust inverse reinforcement learning under transition dynamics mismatch. *ArXiv*, abs/2007.01174, 2020.

[47] David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards, 2018.

[48] Markus Wulfmeier, Ingmar Posner, and Pieter Abbeel. Mutual alignment transfer learning. In *Conference on Robot Learning*, pages 281–290. PMLR, 2017.

[49] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020*, pages 737–744. IEEE, 2020. doi: 10.1109/SSCI47803.2020.9308468. URL `https://doi.org/10.1109/SSCI47803.2020.9308468`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See footnote 3 and Appendix B.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix B.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Apppendix C.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]