PEAC: Unsupervised Pre-training for Cross-Embodiment Reinforcement Learning

Chengyang Ying[1] **Zhongkai Hao**[1] **Xinning Zhou**[1] **Xuezhou Xu**[1]
Hang Su[1,2*] **Xingxing Zhang**[1] **Jun Zhu**[1,2]
[1]Department of Computer Science & Technology, Institute for AI, BNRist Center,
Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University
[2]Pazhou Lab (Huangpu), Guangzhou, China
ycy21@mails.tsinghua.edu.cn

Abstract

Designing generalizable agents capable of adapting to diverse embodiments has achieved significant attention in Reinforcement Learning (RL), which is critical for deploying RL agents in various real-world applications. Previous Cross-Embodiment RL approaches have focused on transferring knowledge across embodiments within specific tasks. These methods often result in knowledge tightly coupled with those tasks and fail to adequately capture the distinct characteristics of different embodiments. To address this limitation, we introduce the notion of Cross-Embodiment Unsupervised RL (CEURL), which leverages unsupervised learning to enable agents to acquire embodiment-aware and task-agnostic knowledge through online interactions within reward-free environments. We formulate CEURL as a novel Controlled Embodiment Markov Decision Process (CE-MDP) and systematically analyze CEURL's pre-training objectives under CE-MDP. Based on these analyses, we develop a novel algorithm Pre-trained Embodiment-Aware Control (PEAC) for handling CEURL, incorporating an intrinsic reward function specifically designed for cross-embodiment pre-training. PEAC not only provides an intuitive optimization strategy for cross-embodiment pre-training but also can integrate flexibly with existing unsupervised RL methods, facilitating cross-embodiment exploration and skill discovery. Extensive experiments in both simulated (e.g., DMC and Robosuite) and real-world environments (e.g., legged locomotion) demonstrate that PEAC significantly improves adaptation performance and cross-embodiment generalization, demonstrating its effectiveness in overcoming the unique challenges of CEURL. The project page and code are in https://yingchengyang.github.io/ceurl.

1 Introduction

Cross-embodiment reinforcement learning (RL) involves designing algorithms that effectively function across various physical embodiments. The fundamental goal is to enable agents to apply skills and strategies learned from some embodiments to other embodiments, which may own different physical dynamics, action-effectors, shapes, and so on [28, 70, 58, 52, 12, 69, 60]. This capability significantly enhances the generalization of RL agents, reducing the necessity for embodiment-specific training. By adeptly adapting to new and shifting embodiment, cross-embodiment RL ensures that agents maintain reliable performance in unpredictable real-world scenarios, thereby benefiting the deployment process and reducing the need for extensive data collection for each new embodiment.

*Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1731

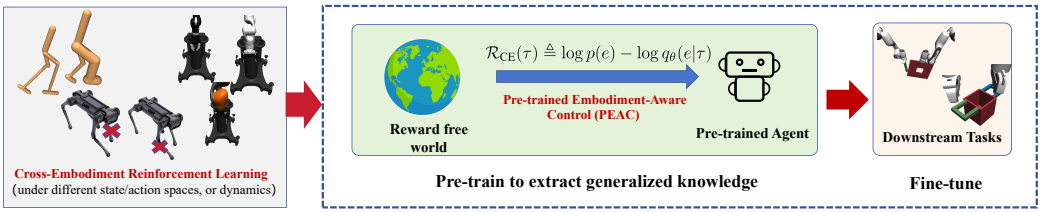


Figure 1: **Overview of Cross-Embodiment Unsupervised Reinforcement Learning (CEURL).** The left subfigure illustrates the cross-embodiment setting with various possible embodiment changes. Directly training RL agents across embodiments under given tasks may result in task-aware rather than embodiment-aware knowledge. CEURL pre-trains agents in reward-free environments to extract embodiment-aware knowledge. The center subfigure shows the Pre-trained Embodiment-Aware Control (PEAC) algorithm, using our cross-embodiment intrinsic reward function $\mathcal{R}_{\text{CE}}(\tau)$. The right subfigure demonstrates the fine-tuning phase, where pre-trained agents fast adapt to different downstream tasks, improving adaptation and generalization.

One of the primary challenges in this area is the transfer of knowledge across embodiments that have vastly different physical dynamics and environmental interactions. This requires the agent to abstract knowledge in a way that is not overly specialized to a single embodiment or some downstream tasks. However, directly training cross-embodiment agents under some given tasks will cause the learned knowledge highly related to these tasks rather than only to embodiments themselves.

Inspired by the transformative effects of unsupervised learning in natural language processing and computer vision [6, 21], which has demonstrated efficiency in extracting generalized knowledge independent of downstream tasks, we propose a natural question: *Can we pre-train cross-embodiment agents in an unsupervised manner, i.e., online cross-embodiment pre-training in reward-free environments, to capture generalized knowledge only related to embodiments?* Existing unsupervised RL techniques, including exploration [45, 38] and skill discovery [10, 26] ones, typically involve pre-training agents by engaging a single embodiment within a controlled Markov Decision Process (MDP) that lacks extrinsic reward signals. These pre-trained agents are then expected to quickly fine-tune to any downstream tasks characterized by extrinsic rewards using this specific embodiment. This approach of unsupervised RL fosters the development of policies that are not overly specialized to specific tasks or reward structures but are rather driven by intrinsic motivations of embodiments, which shows the potential for discovering more generalized knowledge across different embodiments.

In this work, we adapt the unsupervised RL paradigm to the cross-embodiment setting, introducing the concept of Cross-Embodiment Unsupervised RL (CEURL). This setting involves pre-training with a distribution of embodiments in reward-free environments, followed by fine-tuning to handle specific downstream tasks through these embodiments. These embodiments may own similar structures so that we can abstract generalized knowledge from them. To analyze CEURL and design corresponding algorithms, we formulate it as a Controlled Embodiment Markov Decision Process (CE-MDP), which comprises a distribution of controlled MDPs, each defined by its unique embodiment context. Compared to the traditional single-embodiment setting, the CE-MDP framework addresses the additional complexity caused by the inherent variability among embodiments. We then extend the information geometry analyses of the controlled MDP [11] to better explain the complexity of CE-MDP. Our findings indicate that skill vertices within CE-MDP may no longer be simple deterministic policies and the behaviors across different embodiments can display substantial variability.

To address the complexities of CE-MDP, we undertake an in-depth analysis of the pre-training objective in CE-MDP. We aim to enable our pre-trained agent to quickly fine-tune for any downstream tasks denoted as $\mathcal{R}_{\text{ext}}$, especially under the worst-case reward scenarios. Thus, our pre-training objective involves minimizing across $\mathcal{R}_{\text{ext}}$ while maximizing the fine-tuned policy π^*, leading to a complex min-max problem (Eq. 3). We further introduce a novel Pre-trained Embodiment-Aware Control (PEAC) algorithm to optimize this objective and handle CE-MDP, which improves the agent's robustness and adaptability across various embodiments by employing a cross-embodiment intrinsic reward $\mathcal{R}_{\text{CE}}$. This reward is complemented by an embodiment discriminator, which distinguishes between different embodiments. During fine-tuning, the pre-trained policy is further enhanced under the extrinsic reward, $\mathcal{R}_{\text{ext}}$ with limited timesteps. Moreover, PEAC can integrate flexibly with existing single-embodiment unsupervised RL methods to achieve cross-embodiment exploration and skill discovery, resulting in two combination algorithm examples PEAC-LBS and PEAC-DIAYN.

https://doi.org/10.52202/079017-1731

To verify the versatility and effectiveness of our algorithm, we extensively evaluate PEAC in both simulated and real-world environments. In simulations, we choose state-based / image-based DeepMind Control Suite (DMC) environments extending Unsupervised RL Benchmark (URLB) [27] and different robotic arms in Robosuite [73]. Under these settings, PEAC demonstrates superior few-shot learning ability to downstream tasks, and remarkable generalization ability to unseen embodiments, surpassing existing state-of-the-art unsupervised RL models. Besides, we have evaluated PEAC in real-world Aliengo robots by considering practical joint failure settings based on Isaacgym [37], verifying PEAC's strong adaptability on different joint failures and various real-world terrains.

In summary, the main contributions are as follows:

- We propose a novel setting CEURL to enhance agents' adaptability and generalization across diverse embodiments, and then we introduce the Pre-trained Embodiment-Aware Control (PEAC) algorithm for handling CEURL.
- We integrate PEAC with existing exploration and skill discovery techniques, designing practical methods and facilitating efficient cross-embodiment exploration and skill discovery.
- Extensive experiments show that PEAC not only excels in fast fine-tuning but also effectively generalizes across new embodiments, outperforming current SOTA unsupervised RL models.

2 Related Work

Cross-Embodiment RL. Designing generalizable agents simultaneously controlling diverse embodiments has achieved significant attention in RL. A common strategy involves using expert trajectories [70, 49, 5, 66, 60], internet-scale human videos [3, 58, 13], or offline datasets [29, 59, 8, 41] to train a generalist agent that can handle various tasks across different embodiments. However, these methods are often limited by the need for large-scale, costly datasets and the availability of expert trajectories. Additionally, the discrepancy between open-loop training and closed-loop testing may lead to distribution shifts [32], adversely affecting the final performance. An alternative line of research [28, 36, 64, 4, 66, 52, 12, 68] focuses on training general agents through online interaction across diverse environments. However, these methods treat the embodiment and task as a unified training environment and overlook the role of proprioception, i.e., the internal understanding of an agent's embodiment, which has recently proven to be beneficial for representation learning and optimization in RL [23, 15]. Thus these methods may not fully capture the intrinsic properties of different embodiments by linking knowledge to specific tasks. Emerging research suggests the potential of decoupling the training of embodiment characteristics from task execution, aiming to develop a unified cross-embodiment model. This involves unsupervised pre-training across a variety of embodiments, followed by task-aware fine-tuning, enabling a single agent to adeptly manage both roles effectively.

Unsupervised RL. Unsupervised RL leverages interactions with reward-free environments to extract useful knowledge, such as exploratory policies, diverse skills, or world models [17, 18]. These pre-trained models are utilized to fast adapt to downstream tasks within specific embodiments and environments. Unsupervised RL methods can be categorized into two main types: exploration and skill discovery. Exploration methods aim to maximize state coverage, typically through intrinsic rewards that encourage uncertainty [45, 7, 46, 51, 38, 40, 67] or state entropy [30, 47, 35, 34]. The resulting exploratory trajectories benefit pre-training actor-critic or world models, thereby enhancing fine-tuning efficiency [27]. Skill discovery methods focus on learning an array of distinguishable skills, often by maximizing the mutual information between states and acquired skills [10, 54, 20, 55, 25, 26, 72, 24, 61]. This approach benefits from theoretical insights into the information geometry of skill state distributions, emphasizing the importance of maximizing distances between different skills [11, 22, 42, 43, 44]. Recent efforts also explore incremental skill learning in dynamic environments [53, 31]. Unlike these methods generally focus on single embodiments, we aim to develop generalizable models capable of handling downstream tasks across a variety of embodiments.

3 Cross-Embodiment Unsupervised RL

In this section, we analyze the cross-embodiment RL in an unsupervised manner, which is formulated as our Controlled Embodiment MDP. Then we propose a novel algorithm PEAC to optimize CE-MDP.

https://doi.org/10.52202/079017-1731

3.1 Controlled Embodiment Markov Decision Processes

Cross-embodiment RL can be formulated by contextual MDP [19] with a distribution of Markovian decision processes (MDPs) of $\{\mathcal{M}_{\boldsymbol{e}}\}$. Cross-embodiment RL hopes to learn shared knowledge from this distribution of MDPs, which is crucial for enhancing the adaptability or generalization of agents across embodiments. However, directly optimizing agents by online interacting with $\{\mathcal{M}_{\boldsymbol{e}}\}$ or utilizing offline datasets sampled from $\{\mathcal{M}_{\boldsymbol{e}}\}$ may learn knowledge not only related to these embodiments but also highly related to these task reward functions in $\{\mathcal{M}_{\boldsymbol{e}}\}$. This phenomenon may have negative impacts on learning the general knowledge across embodiments or improving the agent's generalization ability. For example, as the agent is required to handle $\{\mathcal{M}_{\boldsymbol{e}}\}$, it will less explore the trajectories with low rewards. These trajectories, although not optimal for the embodiment in this task, might also include embodiment knowledge and be useful for other tasks. Without extrinsic task rewards, the agent is encouraged to learn embodiment-aware and task-agnostic knowledge, which can effectively adapt to any downstream task across embodiments.

In this paper, we propose to pre-train cross-embodiment agents in reward-free environments to ensure that the agent can learn knowledge only specialized in these embodiments themselves. In other words, we introduce unsupervised RL into cross-embodiment RL as a novel setting: *cross-embodiment unsupervised RL* (CEURL). As shown in Fig. 1, in CEURL, we first pre-train a general agent by interacting with the reward-free environment through varying embodiments sampled from an unknown embodiment distribution. Given any downstream task represented by the extrinsic reward $\mathcal{R}_{\text{ext}}$, the pre-trained agent is subsequently fine-tuned to control these embodiments, and other unseen embodiments from the distribution, to complete this task within limited steps (like one-tenth of the pre-training steps). Formally, we formulate CEURL as the following controlled embodiment MDP (CE-MDP):

Definition 3.1 (Controlled Embodiment MDP (CE-MDP)). *A CE-MDP includes a distribution of controlled MDPs defined as $\mathcal{M}_{\boldsymbol{e}}^c = (\mathcal{S}_{\boldsymbol{e}}, \mathcal{A}_{\boldsymbol{e}}, \mathcal{P}_{\boldsymbol{e}}, \gamma)$, where $\boldsymbol{e} \sim \mathcal{E}$ and $\mathcal{E}$ represents the embodiment distribution. Each embodiment may have different state spaces $\mathcal{S}_{\boldsymbol{e}}$ and action spaces $\mathcal{A}_{\boldsymbol{e}}$. $\mathcal{P}_{\boldsymbol{e}} : \mathcal{S}_{\boldsymbol{e}} \times \mathcal{A}_{\boldsymbol{e}} \to \Delta(\mathcal{S}_{\boldsymbol{e}})$ denoting the transition dynamics for embodiment $\boldsymbol{e}$ and γ is the discount factor. We define the state space $\mathcal{S} = \cup_{\boldsymbol{e}} \mathcal{S}_{\boldsymbol{e}}$ and adopt a unified action embedding space $\mathcal{A}$ with corresponding action projectors $\phi_{\boldsymbol{e}} : \mathcal{A} \to \mathcal{A}_{\boldsymbol{e}}$, which can be fixed or learnable.*

Thus we can establish a unified policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ across all embodiments. For any embodiment $\boldsymbol{e}$, we sample an action $\boldsymbol{a}$ from $\pi(\cdot|\boldsymbol{s})$ for a state $\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{e}}$ and execute the projected action $\phi_{\boldsymbol{e}}(\boldsymbol{a})$. Without loss of generality, we assume $\phi_{\boldsymbol{e}}$ is fixed and focus our analysis on the policy π. To explain the complexities of CE-MDP with varying embodiment contexts, we extend the single-embodiment information geometry analyses [11] into our cross-embodiment setting. First, we consider the discount state distribution of π within $\mathcal{M}_{\boldsymbol{e}}^c$ at state $\boldsymbol{s}$ as $d_\pi^{\boldsymbol{e}}(\boldsymbol{s}) = (1-\gamma)\sum_{t=0}^{\infty}[\gamma^t \mathcal{P}_{\boldsymbol{e}}(\boldsymbol{s}_t = \boldsymbol{s})]$. It is well known that the trajectory return of the state-based reward function can be computed as

$$J_{\mathcal{M}_{\boldsymbol{e}}^c, \mathcal{R}_{\text{ext}}}(\pi) \triangleq \mathbb{E}_{\tau \sim \mathcal{M}_{\boldsymbol{e}}^c, \pi}\left[\mathcal{R}_{\text{ext}}(\tau)\right] = \frac{1}{1-\gamma}\mathbb{E}_{s \sim d_\pi^{\boldsymbol{e}}}\left[\mathcal{R}_{\text{ext}}(\boldsymbol{s})\right]. \tag{1}$$

Thus, the properties of $d_\pi^{\boldsymbol{e}}$ are significant in determining useful initializations for downstream tasks. We consider the set $\mathcal{D}^{\boldsymbol{e}} = \{d_\pi^{\boldsymbol{e}} \in \Delta(\mathcal{S}) \mid \forall \pi\}$, which includes all feasible $d_\pi^{\boldsymbol{e}}$ over the probability simplex. As shown in [11], for each $\boldsymbol{e}$, $\mathcal{D}^{\boldsymbol{e}}$ is a convex set, and any useful policy, which can be optimal for certain downstream tasks under embodiment $\boldsymbol{e}$, must be a vertex of $\mathcal{D}^{\boldsymbol{e}}$, typically corresponding to deterministic policies.

However, in the context of CEURL, the unknown embodiment context e introduces partial observability [14] and significant differences in the corresponding points of the same skill across different embodiments. In CE-MDP, we consider the entire embodiment space and define $d_\pi^{\mathcal{E}}(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{e} \sim \mathcal{E}}\left[d_\pi^{\boldsymbol{e}}(\boldsymbol{s})\right]$, with $\mathcal{D}^{\mathcal{E}} = \{d_\pi^{\mathcal{E}} \in \Delta(\mathcal{S}) \mid \forall \pi\}$. The primary challenge lies in the high variability of embodiments, which complicates the process of learning a policy that generalizes well across different embodiments. We demonstrate that the vertices of $\mathcal{D}^{\mathcal{E}}$ may no longer correspond to deterministic policies, as they need to handle all embodiments in the distribution. This significantly heightens the challenge of the pre-training process in CE-MDP, making it more difficult to find useful cross-embodiment skills (proofs and discussion in Appendix A.1).

To solve CEURL under the paradigm of CE-MDP, the agent will collect reward-free trajectories $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, ...)$ with probability $p_{\mathcal{M}_{\boldsymbol{e}}^c, \pi}(\tau) = \mathcal{P}_{\boldsymbol{e}}(\boldsymbol{s}_0)\prod_{t=0}\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)\mathcal{P}_{\boldsymbol{e}}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ via some

sampled embodiments $\boldsymbol{e}$ during the pre-training. These trajectories are then used in CEURL methods to design intrinsic rewards $\mathcal{R}_{\text{int}}$ for pre-training agents. During fine-tuning, we will sample several embodiments $\boldsymbol{e}$ from $\mathcal{E}$ and combine $\mathcal{M}_{\boldsymbol{e}}^{c}$ with a downstream task represented by extrinsic rewards $\mathcal{R}_{\text{ext}}$, and agents are required to maximize the task return over all embodiments, i.e., $\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\left[J_{\mathcal{M}_{\boldsymbol{e}}^{c},\mathcal{R}_{\text{ext}}}(\pi)\right]$, within limited steps (like one-tenth or less of the pre-training steps).

3.2 Pre-trained Embodiment-Aware Control

We primarily focus on the pre-training objective of CEURL, specifically determining the optimal pre-trained policy π for CEURL. In the fine-tuning stage, given any downstream task characterized by extrinsic reward $\mathcal{R}_{\text{ext}}$, the pre-trained policy π will be optimized into the fine-tuned policy π^* with *limited* steps to handle $\mathcal{R}_{\text{ext}}$ via some RL algorithms like PPO [50]. Consequently, it is widely assumed that π^* will remain close to π during fine-tuning due to constraints on limited interactions with the environment [11]. Our *cross-embodiment fine-tuning objective* thus combines *policy improvement* under $\mathcal{R}_{\text{ext}}$ and a *policy constraint* evaluated via KL divergence

$$\mathcal{F}(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) \triangleq \underbrace{\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi^*}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)]}_{\text{Policy Improvement}} \underbrace{- \beta D_{\text{KL}}(p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi^*}(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau))}_{\text{Policy Constraint}}, \tag{2}$$

where $\beta > 0$ is the unknown trade-off parameter related to the fine-tuning steps (when fine-tuning steps tend towards infinity, β tends to 0 and this objective converges to the original RL objective), and $\bar{\mathcal{M}}$ represents the "average embodiment MDP" satisfying that $p_{\bar{\mathcal{M}},\pi}(\tau) = \mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\left[p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi}(\tau)\right]$. During fine-tuning, we hope to optimize π^* by maximizing $\mathcal{F}$, i.e., the fine-tuned result is $\max_{p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi^*}(\tau)} \mathcal{F}(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e})$. As the pre-trained policy π needs to handle *any* downstream task, we consider the worst-case extrinsic reward function across the embodiment distribution, and our *cross-embodiment pre-training objective* can be formally represented as maximizing

$$\mathcal{U}(\pi,\mathcal{E}) \triangleq \mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\left[\min_{\mathcal{R}_{\text{ext}}(\tau)} \max_{p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi^*}(\tau)} \mathcal{F}(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e})\right]. \tag{3}$$

This objective is a min-max problem that is hard to optimize. Fortunately, we can simplify it as below

Theorem 3.2 (Proof in Appendix A.2)**.** *The pre-training objective Eq.* (3) *of* $(\pi,\mathcal{E})$ *satisfies*

$$\mathcal{U}(\pi,\mathcal{E}) = \mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\left[-\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi}(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right] = \beta\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}}^{c},\pi}(\tau)}\left[\log\frac{p(\boldsymbol{e})}{p_\pi(\boldsymbol{e}|\tau)}\right]. \tag{4}$$

Here $p(\boldsymbol{e})$ and $p_\pi(\boldsymbol{e}|\tau)$ are embodiment prior and posterior probabilities, respectively. This result simplifies our pre-trained objective as a form easy to calculate and optimize. Also, although β is an unknown parameter, the optimal pre-trained policy is independent of β. Based on these analyses, we propose a novel algorithm named Pre-trained Embodiment-Aware Control (PEAC). In PEAC, we first train an embodiment discriminator $q_\theta(\boldsymbol{e}|\tau)$ to approximate $p_\pi(\boldsymbol{e}|\tau)$, which can learn the embodiment context via historical trajectories. For cross-embodiment pre-training, PEAC then utilizes our cross-embodiment intrinsic reward, which is defined following Eq. (4) as

$$\mathcal{R}_{\text{CE}}(\tau) \triangleq \log p(\boldsymbol{e}) - \log q_\theta(\boldsymbol{e}|\tau). \tag{5}$$

Assuming the embodiment prior $p(\boldsymbol{e})$ is fixed, $\mathcal{R}_{\text{CE}}$ encourages the agent to explore the region with low $\log q_\theta(\boldsymbol{e}|\tau)$. In these trajectories, the embodiment discriminator is misled, where the agent may not have explored enough or different embodiment posteriors are similar. Thus, the embodiment discriminator can boost itself from these trajectories and learned embodiment-aware contexts that can effectively represent different embodiments, which benefit generalizing to unseen embodiments.

In practice, $\mathcal{R}_{\text{CE}}$ needs to be calculated for each state $\boldsymbol{s}$ rather than the whole trajectory τ, also, the embodiment discriminator needs to classify the embodiment context for every state. For RL backbones that encode historical information as the hidden state $\boldsymbol{h}$ like Dreamer [17, 18, 64], we directly train $q_\theta(\boldsymbol{e}|\boldsymbol{h},\boldsymbol{s})$ as the discriminator and further calculate $\mathcal{R}_{\text{CE}}$. For RL algorithms with Markovian policies like PPO [50], we encode a fixed length historical state-action pair to the hidden state h and also train $q_\theta(\boldsymbol{e}|\boldsymbol{h},\boldsymbol{s})$, following [28]. For a fair comparison, our policy still uses Markovian policy and does not utilize encoded historical messages. PEAC's pseudo-code is in Appendix C.

https://doi.org/10.52202/079017-1731

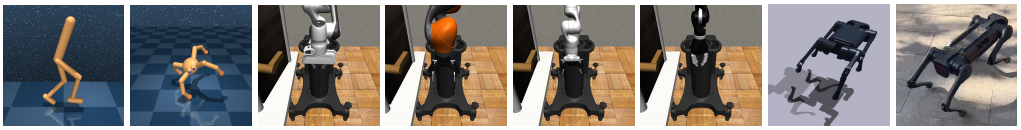


Figure 2: **Benchmark environments,** including DMC [56], Robosuite [73], Isaacgym [37].

4 Cross-Embodiment Exploration and Skill Discovery

As shown above, PEAC pre-trains the agent for the optimal initialization to few-shot handle downstream tasks across embodiments. Besides, although PEAC does not directly explore or discover skills, it is flexible to combine with existing unsupervised RL methods, including exploration and skill discovery ones, to achieve cross-embodiment exploration and skill discovery. Below we will discuss in detail the specific combination between PEAC and these two classes respectively, exporting two practical combination algorithms, PEAC-LBS and PEAC-DIAYN, as examples.

Embodiment-Aware Exploration. Existing exploration methods mainly encourage the agent to explore unseen regions. As PEAC suggests the agent explores the region where the embodiment discriminator is wrong, it is natural to directly combine $\mathcal{R}_{\text{CE}}$ and exploration intrinsic rewards to achieve cross-embodiment exploration, i.e., balancing embodiment representation learning and unseen state exploration. As an example, we take LBS [38], of which the intrinsic reward is the KL divergence between the latent prior and the approximation posterior, as the PEAC-LBS. As $\mathcal{R}_{\text{CE}}$ and $\mathcal{R}_{\text{LBS}}$ are both related to some KL divergence, we can directly add up these two intrinsic rewards with the same weight in PEAC-LBS, of which the detailed pseudo-code is in Appendix C.

Embodiment-Aware Skill Discovery. Single-embodiment skill-discovery mainly maximizes the mutual information between trajectories τ and skills $\boldsymbol{z}$ as $\mathcal{I}(\tau;\boldsymbol{z}) = D_{\text{KL}}(p(\tau,\boldsymbol{z})\|p(\tau)p(\boldsymbol{z}))$ [10], which has been shown as optimal initiation to some skill-based adaptation objective [11]. We combine it and our cross-embodiment fine-tuning objective Eq. (2) to propose a unified *cross-embodiment skill-based adaptation objective* as

$$\begin{aligned}\mathcal{F}_s(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) \triangleq& \mathbb{E}_{p_{\mathcal{M}^c_{\boldsymbol{e}},\pi^*}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \max_{\boldsymbol{z}^*} \mathbb{E}_{p_{\mathcal{M}^c_{\boldsymbol{e}},\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)] \\ &-\beta D_{\text{KL}}(p_{\mathcal{M}^c_{\boldsymbol{e}},\pi^*}(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau)).\end{aligned} \tag{6}$$

Similar to Theorem 3.2, we can define our pre-training objective and simplify it as

$$\begin{aligned}&\mathcal{U}_s(\pi,\mathcal{E}) \triangleq \mathbb{E}_{e\sim\mathcal{E}} \min_{\mathcal{R}_{\text{ext}}(\tau)} \max_{p_{\mathcal{M}^c_{\boldsymbol{e}},\pi^*}(\tau)} \mathcal{F}_s(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) \\ =&-\beta\mathbb{E}_{\boldsymbol{e}} \max_{p(\boldsymbol{z}|\mathcal{M}^c_{\boldsymbol{e}})} \left[\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}},\pi}} \log \frac{p_\pi(\boldsymbol{e}|\tau)}{p_\pi(\boldsymbol{e})} + D_{\text{KL}}(p_\pi(\tau,\boldsymbol{z}|\mathcal{M}^c_{\boldsymbol{e}})\|p_\pi(\boldsymbol{z}|\mathcal{M}^c_{\boldsymbol{e}})p_\pi(\tau|\mathcal{M}^c_{\boldsymbol{e}}))\right].\end{aligned} \tag{7}$$

The proof of Eq. (7) is in Appendix A.3, where we also show it is a general form of Theorem 3.2 and the single-embodiment skill-discovery result [11]. The result of Eq. (7) includes two terms for handling cross-embodiment and discovering skills respectively. In detail, the first term is the same as the objective in Eq. (4), thus we can directly optimize it via PEAC. As the second term is similar to the classical skill-discover objective $\mathcal{I}(\tau;\boldsymbol{z})$ but only embodiment-aware, we can extend existing skill-discovery methods into an embodiment-aware version for handling it.

We take DIAYN [10] as an example, resulting in PEAC-DIAYN. Overall, In the pre-training stage, given a random skill $\boldsymbol{z}$ and an embodiment $\boldsymbol{e}$, we will sample trajectories with the policy $\pi_\theta(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{z},\boldsymbol{e})$ that is conditioned on z and the predicted embodiment context. Then we will train a neural network $p(\boldsymbol{z},\boldsymbol{e}|\tau)$ to jointly predict the current skill and the embodiment. For training the policy, we combine $\mathcal{R}_{\text{CE}}$ and $\mathcal{R}_{\text{DIAYN}}$ as the intrinsic reward. During fine-tuning, we utilize the embodiment discriminator, mapping observed trajectories to infer the embodiment context. We then train an embodiment-aware meta-controller $\pi(\boldsymbol{z}|\boldsymbol{e},\tau)$, which inputs the state and predicted context and then outputs the skill. It extends existing embodiment-agnostic meta-controller [39] and directly chooses from skill spaces rather than complicated action spaces. The pseudo-code of PEAC-DIAYN is in Appendix C.

https://doi.org/10.52202/079017-1731

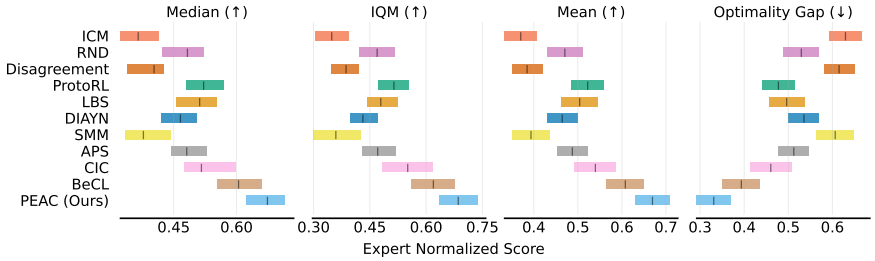


Figure 3: Aggregate metrics [2] in **state-based DMC**. Each statistic for every algorithm has 120 runs (3 embodiment settings $\times$ 4 downstream tasks $\times$ 10 seeds).

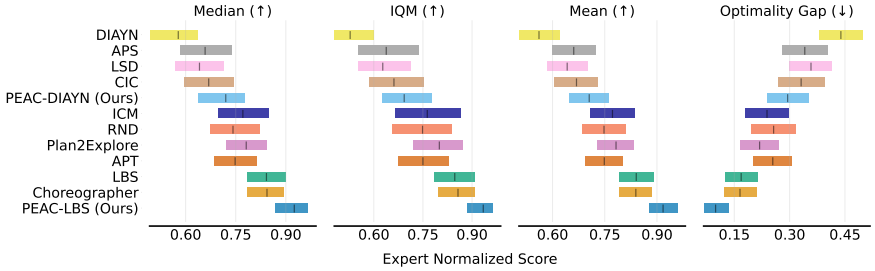


Figure 4: Aggregate metrics [2] in **image-based DMC**. Each statistic for every algorithm has 36 runs (3 embodiment settings $\times$ 4 downstream tasks $\times$ 3 seeds).

5 Experiments

We now present extensive empirical results to answer the following questions:

- Does PEAC enhance the cross-embodiment unsupervised pre-training for handling different downstream tasks? (Sec. 5.2)
- Can CEURL benefit cross-embodiment RL and effectively generalize to unseen embodiments? (Sec. 5.3)
- Does CEURL advantage to real-world cross-embodiment applications? (Sec. 5.4)

5.1 Experimental Setup

To fully evaluate PEAC in CEURL, we choose extensive benchmarks (Fig. 2), including state-based / image-based Deepmind Control Suite (DMC) [56] in URLB [27], Robosuite [73, 69] for robotic manipulation, and Isaacgym [37] for simulation as well as real-world legged locomotion. Below we will introduce embodiments, tasks, and baselines for these settings, with more details in Appendix B.

State-based DMC & Image-based DMC. These two benchmarks extend URLB [27], classical single-embodiment unsupervised RL settings. Based on basic embodiments, we change the mass or damping to conduct three distinct embodiment distributions: walker-mass, quadruped-mass, and quadruped-damping, following previous work with diverse embodiments [28, 65]. All downstream tasks follow URLB. These two settings take robot states and images as observations respectively.

In state-based DMC, we compare PEAC with 5 exploration and 5 skill-discovery methods: ICM [45], RND [7], Disagreement [46], ProtoRL [62], LBS [38], DIAYN [10], SMM [30], APS [34], CIC [26], and BeCL [61], which are standard and SOTA for this setting. For all methods, we take DDPG [33] as the RL backbone, which is widely used in this benchmark [27]. In image-based DMC, we take 5 exploration baselines: ICM, RND, Plan2Explore [51], APT [35], and LBS; as well as 4 skill-discovery baselines: DIAYN, APS, LSD [42], and CIC. Also, we choose a SOTA baseline Choreographer [39], which combines exploration and skill discovery. For all methods, we take DreamerV2 [18] as the backbone algorithm, which has currently shown leading performance in this benchmark [48].

Robosuite. We further consider embodiment distribution with greater change: different robotic arms for manipulation tasks from Robosuite [73]. We pre-train our agents in robotic arms Panda, IIWA, and Kinova3. Besides, we take robotic arm Jaco for evaluating generalization. Following [69], we take DrQ [63] as the RL backbone and choose standard task settings: Door, Lift, and TwoArmPegInHole.

https://doi.org/10.52202/079017-1731

Domains	Robosuite		A1-disabled				
	Train	Test	run	climb	leap	crawl	tilt
ICM	174.4	178.3	6.7	5.7	4.0	8.5	13.9
RND	171.2	185.0	8.1	3.5	2.2	7.6	6.3
LBS	157.7	166.6	0.4	1.7	1.4	1.1	2.1
PEAC (Ours)	**190.7**	**200.8**	**19.2**	**10.3**	**10.0**	**20.3**	**17.3**

Table 1: Results of **Robosuite** and **Isaacgym**.

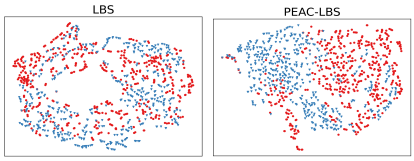


Figure 5: **Generalization Visualization**

Domains	Walker-mass	Quadruped-mass	Quadruped-damping
ICM	391.5 ± 224.9	227.1 ± 163.6	160.7 ± 129.7
RND	364.8 ± 172.9	588.0 ± 164.8	139.5 ± 119.9
Disagreement	321.6 ± 152.6	434.2 ± 176.7	140.8 ± 73.9
ProtoRL	440.1 ± 212.7	471.6 ± 209.0	328.2 ± 195.4
LBS	380.3 ± 227.0	508.2 ± 222.7	350.4 ± 226.2
DIAYN	267.6 ± 155.3	456.6 ± 173.3	397.0 ± 159.9
SMM	451.2 ± 196.6	217.3 ± 145.9	162.5 ± 119.2
APS	393.8 ± 222.0	464.6 ± 206.0	285.5 ± 157.5
CIC	503.9 ± 260.6	602.2 ± 193.8	166.2 ± 126.6
BeCL	**544.5 ± 258.7**	475.6 ± 228.5	421.9 ± 246.9
PEAC (Ours)	491.3 ± 250.1	**631.0 ± 235.7**	**573.7 ± 220.3**

Domains	Walker-mass	Quadruped-mass	Quadruped-damping
DIAYN	463.6 ± 250.8	399.6 ± 183.7	499.9 ± 187.5
APS	555.5 ± 245.2	566.9 ± 158.0	546.8 ± 190.6
LSD	556.6 ± 273.0	510.7 ± 173.2	520.9 ± 163.8
CIC	609.4 ± 260.2	527.4 ± 229.9	558.6 ± 169.5
PEAC-DIAYN (Ours)	621.9 ± 235.1	556.1 ± 179.4	557.2 ± 160.7
ICM	648.1 ± 252.4	695.8 ± 180.1	590.2 ± 168.9
RND	658.2 ± 238.8	625.7 ± 179.5	588.4 ± 175.8
Plan2Explore	677.4 ± 245.2	660.2 ± 162.1	608.2 ± 157.7
APT	643.9 ± 242.6	617.7 ± 160.5	600.2 ± 149.7
LBS	658.2 ± 219.7	730.7 ± 162.3	732.7 ± 142.5
Choreographer	687.8 ± 222.7	682.3 ± 159.4	724.6 ± 116.6
PEAC-LBS (Ours)	**754.8 ± 214.6**	**740.8 ± 171.3**	**742.1 ± 165.2**

Table 2: **Generalization** results of **unseen embodiments** in **state-based DMC** (left) and **image-based DMC** (right). For each domain, we report the average return of each different algorithm and **bold** the best performance.

Isaacgym. To explore CEURL in realistic environments, we design embodiment distributions based on the Unitree A1 robot in Isaacgym simulation [37], which is widely used for the real-world legged robot control [1, 74]. As A1 owns 12 controllable joints, we design A1-disabled, a uniform distribution of 12 embodiments, each with a joint failure, respectively. It is realistic as robots may damage some joints when deploying in the real world, and they are still required to complete tasks to their best. We choose standard RL backbone PPO [50] and five downstream tasks: run, climb, leap, crawl, and tilt, following [74]. We take classical baselines for Robosuite and Isaacgym: ICM, RND, and LBS. Besides, we have deployed Aliengo robots with different failure joints to evaluate the effectiveness of PEAC in real-world applications.

5.2 Evaluation of PEAC

State-based DMC. We first report results in state-based DMC to show that PEAC can facilitate cross-embodiment pre-training. All algorithms, repeated 10 random seeds, are pre-trained 2M timesteps in reward-free environments with different embodiments, followed by fine-tuned downstream tasks for all these embodiments with 100k timesteps. We train DDPG agents for each downstream task 2M steps to get the expert return and calculate the expert normalized score for each method. Following [2], in Fig. 3, we report mean, median, interquartile mean (IQM), and optimality gap (OG) metrics along with stratified bootstrap confidence intervals. Fig. 3 demonstrates that PEAC substantially outperforms other baselines on all metrics, indicating that our cross-embodiment intrinsic reward contributes positively to downstream tasks across different embodiments. Notably, compared with BeCL and CIC which get the second and third scores, PEAC not only has higher performance but also a smaller confidence interval, highlighting its stability. Appendix B.4 reports detailed results of these statistics (Table 8) and individual results for each downstream task (Table 9).

Image-based DMC. As described in Sec. 4, PEAC can flexibly combine with existing unsupervised RL methods. To verify it, we evaluate PEAC-LBS and PEAC-DIAYN in image-based DMC. The pre-training and fine-tuning steps are still 2M and 100k respectively. Also, we present four metrics: Median, IQM, Mean, and OG with stratified bootstrap confidence intervals in Fig. 4. Taking IQM as our primary metric, PEAC-LBS not only has a higher value but also a relatively smaller confidence interval, indicating its better stability. As mentioned in [39], pure skill-discovery methods like DIAYN struggle on this benchmark with a certain gap compared to exploratory method. The phenomenon seems more pronounced in cross-embodiment setting than single-embodiment setting, which might be because of the increased difficulty of finding consistent skills across embodiments. As PEAC-DIAYN discovers skills across-embodiment, it consistently leads in performance compared with all other pure skill discovery methods across all four statistics. In Appendix B.5, we report detailed results of these statistics in Table 10 and detailed results for all downstream tasks in Table 11.

https://doi.org/10.52202/079017-1731

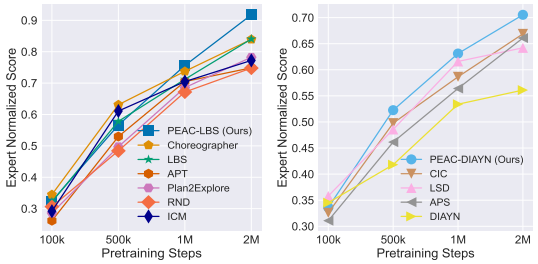


Figure 6: Ablation studies on pre-training timesteps.

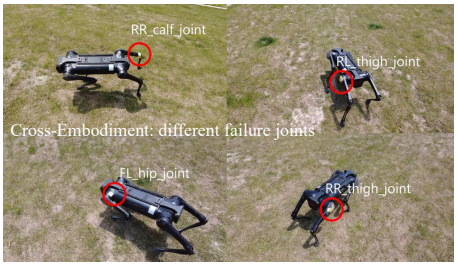


Figure 7: Real-world results.

Robosuite. Besides, we validate PEAC in a more challenging setting Robosuite where different embodiments own different robotic arms (subfigures 3-6 in Fig. 2). As shown in Table 1, PEAC still significantly outperforms all baselines in both training and testing embodiments, demonstrating its powerful cross-embodiment ability and better generalization ability. The detailed results of each robotic arm are in Table 12 of Appendix B.6.

Ablation studies. We do several ablation studies in image-based DMC to clarify the contribution of PEAC better. First, we evaluate the effectiveness of pre-trained steps in fine-tuned performance. We pre-train agents for 100k, 500k, 1M, and 2M steps and then fine-tune them for 100k steps. As shown in Fig. 6, all algorithms improve with pre-training timesteps increasing, indicating that cross-embodiment pre-training effectively benefits fast handling downstream tasks. PEAC-LBS becomes the best-performing method from 1M steps on and PEAC-DIAYN significantly exceeds skill discovery methods. This suggests that PEAC excels at handling cross-embodiment tasks with increased pre-training steps. Additional results are in Appendix B.7. Besides pre-training steps, we also do more ablations studies of different components in PEAC to verify their effectiveness in Appendix B.8. For example, we evaluate the stability of PEAC-LBS in DMC-image under different β (we set it as 1.0 in all main experiments), which is the trade-off parameter for balancing the policy improvement term and the policy constraint term. Moreover, we also do an ablation study on our embodiment discriminator to verify the contribution of each component in our PEAC. More results and analyses are in Appendix B.8.

5.3 Generalization to Unseen Embodiments

To answer the second question, we further assess the generalization ability of PEAC to unseen embodiments. First, we directly leverage pre-trained agents to zero-shot sample trajectories with different unseen embodiments and then visualize results through t-SNE [57] in Fig. 5, where different colored points represent states sampled via different embodiments. As shown in Fig. 5, PEAC-LBS can distinguish different embodiments' states more effectively compared to LBS, which is difficult to distinguish them (more results are in Appendix B.9). Furthermore, we evaluate the generalization ability of fine-tuned agents for all methods by zero-shot evaluating them with unseen embodiments and the same downstream task. In Table 2, we report the detailed generalization results of all 3 domains about state-based DMC and image-based DMC. The results demonstrate that the fine-tuned agents of PEAC can successfully handle the same downstream task with unseen embodiments, which illustrates that PEAC effectively learns cross-embodiment knowledge. Detailed results for each downstream task are in Appendix B.10 (Table 16-17).

5.4 Real-World Applications

To validate CEURL in more realistic settings, we conduct results based on legged locomotion in Isaacgym, which is widely used for real-world applications. First, we present simulation results of A1-disabled in Table 1, with 100M pre-train timesteps and 10M fine-tune timesteps. As shown in Table 1, PEAC effectively establishes a good initialization model across embodiments with different joint failures and quickly adapts to downstream tasks, especially for challenging climb and leap tasks.

Besides, we have deployed PEAC fine-tuned agents in real-world Aliengo-disabled robots, i.e., Aliengo robots with different failure joints. As shown in Fig. 7, due to joint failure, the movement ability of the robot is limited compared to normal settings, but the robot still demonstrates strong adaptability on various terrains not seen in simulators. More images and videos of real-world applications are in Appendix B.12.

https://doi.org/10.52202/079017-1731

5.5 Limitations and Discussion

In terms of limitations, we assume that different embodiments may own similar structures so that we can pre-train a unified agent for them. As a result, it might be challenging for PEAC to handle extremely different embodiments. Also, existing unsupervised RL methods still struggle to handle more challenging downstream tasks. In Appendix B.11, we take the first step to evaluate several more challenging downstream tasks and more different embodiment distributions, of which the results show that PEAC can still perform better than baselines. Designing more efficient cross-embodiment unsupervised algorithms for these more difficult and practical settings are interesting future directions. The Broader Impact os this work is discussed in Appendix D.

6 Conclusion

In this work, we propose to analyze cross-embodiment RL in an unsupervised RL perspective as CEURL, i.e., pre-training in an embodiment distribution. We formulate it as CE-MDP, with some more challenging properties than the single-embodiment setting. By analyzing the optimal cross-embodiment initialization, we propose PEAC with a principled intrinsic reward function and further show that PEAC can flexibly combine with existing unsupervised RL. Experimental results demonstrate that PEAC can effectively handle downstream tasks across embodiments for extensive settings, ranging from image-based observation, state-based observation, and real-world legged locomotion. We hope this work can encourage further research in developing RL agents for both task generalization and embodiment generalization, especially in real-world control.

Acknowledgments and Disclosure of Funding

This work was supported by NSFC Projects (Nos. 92248303, 92370124, 62350080, 62276149, 62061136001), BNRist (BNR2022RC01006), Tsinghua Institute for Guo Qiang, and the High Performance Computing Center, Tsinghua University. J. Zhu was also supported by the XPlorer Prize.

References

[1] Ananye Agarwal, Ashish Kumar, Jitendra Malik, and Deepak Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on robot learning*, pages 403–415. PMLR, 2023.

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in neural information processing systems*, 34:29304–29320, 2021.

[3] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

[4] Michael Beukman, Devon Jarvis, Richard Klein, Steven James, and Benjamin Rosman. Dynamics generalisation in reinforcement learning via adaptive context-aware policies. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

[8] Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, Alexander Herzog, Karl Pertsch, et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, pages 3909–3928. PMLR, 2023.

[9] Thomas M Cover and Joy A Thomas. Elements of information theory. 1991.

[10] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2018.

[11] Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. The information geometry of unsupervised reinforcement learning. In *International Conference on Learning Representations*, 2021.

[12] Gilbert Feng, Hongbo Zhang, Zhongyu Li, Xue Bin Peng, Bhuvan Basireddy, Linzhu Yue, Zhitao Song, Lizhi Yang, Yunhui Liu, Koushil Sreenath, et al. Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning*, pages 1893–1903. PMLR, 2023.

[13] Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pages 11321–11339. PMLR, 2023.

[14] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in Neural Information Processing Systems*, 34:25502–25515, 2021.

[15] Kevin Gmelin, Shikhar Bahl, Russell Mendonca, and Deepak Pathak. Efficient rl via disentangled environment and agent representations. In *International Conference on Machine Learning*, pages 11525–11545. PMLR, 2023.

[16] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.

[17] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

[18] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2020.

[19] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

[20] Steven Hansen, Will Dabney, Andre Barreto, David Warde-Farley, Tom Van de Wiele, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*, 2019.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[22] Shuncheng He, Yuhang Jiang, Hongchang Zhang, Jianzhun Shao, and Xiangyang Ji. Wasserstein unsupervised reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6884–6892, 2022.

[23] Edward S Hu, Kun Huang, Oleh Rybkin, and Dinesh Jayaraman. Know thyself: Transferable visual control policies through robot-awareness. In *International Conference on Learning Representations*, 2021.

[24] Zheyuan Jiang, Jingyue Gao, and Jianyu Chen. Unsupervised skill discovery via recurrent skill training. *Advances in Neural Information Processing Systems*, 35:39034–39046, 2022.

[25] Jaekyeom Kim, Seohong Park, and Gunhee Kim. Unsupervised skill discovery with bottleneck option learning. In *International Conference on Machine Learning*, pages 5572–5582. PMLR, 2021.

[26] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. *Advances in Neural Information Processing Systems*, 35:34478–34491, 2022.

[27] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[28] Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 5757–5766. PMLR, 2020.

 https://doi.org/10.52202/079017-1731

[29] Kuang-Huei Lee, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35:27921–27936, 2022.

[30] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[31] Sang-Hyun Lee and Seung-Woo Seo. Unsupervised skill discovery for learning shared structures across changing environments. In *International Conference on Machine Learning*, pages 19185–19199. PMLR, 2023.

[32] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[33] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[34] Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR, 2021.

[35] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021.

[36] Xin Liu, Yaran Chen, Haoran Li, Boyu Li, and Dongbin Zhao. Cross-domain random pre-training with prototypes for reinforcement learning. *arXiv preprint arXiv:2302.05614*, 2023.

[37] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[38] Pietro Mazzaglia, Ozan Catal, Tim Verbelen, and Bart Dhoedt. Curiosity-driven exploration via latent bayesian surprise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7752–7760, 2022.

[39] Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Alexandre Lacoste, and Sai Rajeswar. Choreographer: Learning and adapting skills in imagination. In *The Eleventh International Conference on Learning Representations*, 2022.

[40] Mirco Mutti, Mattia Mancassola, and Marcello Restelli. Unsupervised reinforcement learning in multiple environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7850–7858, 2022.

[41] Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning*, pages 26087–26105. PMLR, 2023.

[42] Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. *arXiv preprint arXiv:2202.00914*, 2022.

[43] Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. *arXiv preprint arXiv:2302.05103*, 2023.

[44] Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.

[45] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[46] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

[47] Vitchyr Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. In *International Conference on Machine Learning*, pages 7783–7792. PMLR, 2020.

[48] Sai Rajeswar, Pietro Mazzaglia, Tim Verbelen, Alexandre Piché, Bart Dhoedt, Aaron Courville, and Alexandre Lacoste. Mastering the unsupervised reinforcement learning benchmark from pixels. In *International Conference on Machine Learning*, pages 28598–28617. PMLR, 2023.

[49] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *Transactions on Machine Learning Research*, 2022.

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

[51] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pages 8583–8592. PMLR, 2020.

[52] Milad Shafiee, Guillaume Bellegarda, and Auke Ijspeert. Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots. *arXiv preprint arXiv:2310.10486*, 2023.

[53] Nur Muhammad Mahi Shafiullah and Lerrel Pinto. One after another: Learning incremental skills for a changing world. In *International Conference on Learning Representations*, 2021.

[54] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. In *International Conference on Learning Representations*, 2019.

[55] DJ Strouse, Kate Baumli, David Warde-Farley, Volodymyr Mnih, and Steven Stenberg Hansen. Learning more skills through optimistic exploration. In *International Conference on Learning Representations*, 2021.

[56] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[58] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *Conference on Robot Learning*, pages 3536–3555. PMLR, 2023.

[59] Yifan Xu, Nicklas Hansen, Zirui Wang, Yung-Chieh Chan, Hao Su, and Zhuowen Tu. On the feasibility of cross-task transfer with model-based reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.

[60] Jonathan Yang, Catherine Glossop, Arjun Bhorkar, Dhruv Shah, Quan Vuong, Chelsea Finn, Dorsa Sadigh, and Sergey Levine. Pushing the limits of cross-embodiment learning for manipulation and navigation. *arXiv preprint arXiv:2402.19432*, 2024.

[61] Rushuai Yang, Chenjia Bai, Hongyi Guo, Siyuan Li, Bin Zhao, Zhen Wang, Peng Liu, and Xuelong Li. Behavior contrastive learning for unsupervised skill discovery. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39183–39204. PMLR, 23–29 Jul 2023.

[62] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021.

[63] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, 2020.

[64] Chengyang Ying, Zhongkai Hao, Xinning Zhou, Hang Su, Songming Liu, Dong Yan, and Jun Zhu. Task aware dreamer for task generalization in reinforcement learning. *arXiv preprint arXiv:2303.05092*, 2023.

[65] Chengyang Ying, Xinning Zhou, Hang Su, Dong Yan, Ning Chen, and Jun Zhu. Towards safe reinforcement learning via constraining conditional value-at-risk. *arXiv preprint arXiv:2206.04436*, 2022.

[66] Chen Yu, Weinan Zhang, Hang Lai, Zheng Tian, Laurent Kneip, and Jun Wang. Multi-embodiment legged robot control as a sequence modeling problem. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7250–7257. IEEE, 2023.

 https://doi.org/10.52202/079017-1731

[67] Mingqi Yuan, Bo Li, Xin Jin, and Wenjun Zeng. Automatic intrinsic reward shaping for exploration in deep reinforcement learning. *arXiv preprint arXiv:2301.10886*, 2023.

[68] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024.

[69] Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. Rl-vigen: A reinforcement learning benchmark for visual generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

[70] Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pages 537–546. PMLR, 2022.

[71] Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning robust state abstractions for hidden-parameter block mdps. *arXiv preprint arXiv:2007.07206*, 2020.

[72] Andrew Zhao, Matthieu Lin, Yangguang Li, Yong-Jin Liu, and Gao Huang. A mixture of surprises for unsupervised reinforcement learning. *Advances in Neural Information Processing Systems*, 35:26078–26090, 2022.

[73] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[74] Ziwen Zhuang, Zipeng Fu, Jianren Wang, Christopher G Atkeson, Sören Schwertfeger, Chelsea Finn, and Hang Zhao. Robot parkour learning. In *7th Annual Conference on Robot Learning*, 2023.

A Proof of Theorems

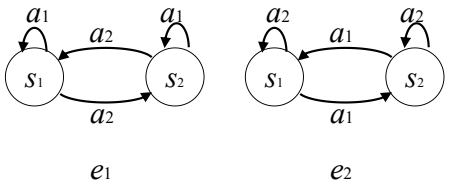
In this section, we will provide detailed proof of theorems in the paper.

A.1 Properties and Challenges of $\mathcal{D}^{\mathcal{E}}$

We first construct an example to show that vertices of $\mathcal{D}^{\mathcal{E}}$ may no longer be deterministic policies.

Considering a simple embodiment distribution with only 2 embodiments $\boldsymbol{e}_1, \boldsymbol{e}_2$ with the embodiment probability $p(\boldsymbol{e}_1) = p(\boldsymbol{e}_2) = \frac{1}{2}$. For each embodiment, there are two states $\boldsymbol{s}_1, \boldsymbol{s}_2$ and two actions $\boldsymbol{a}_1, \boldsymbol{a}_2$ and the dynamic is

$$\begin{aligned}
&p_{\boldsymbol{e}_1}(\boldsymbol{s}_1|\boldsymbol{s}_1,\boldsymbol{a}_1) = 1, p_{\boldsymbol{e}_1}(\boldsymbol{s}_2|\boldsymbol{s}_1,\boldsymbol{a}_1) = 0, p_{\boldsymbol{e}_1}(\boldsymbol{s}_1|\boldsymbol{s}_1,\boldsymbol{a}_2) = 0, p_{\boldsymbol{e}_1}(\boldsymbol{s}_2|\boldsymbol{s}_1,\boldsymbol{a}_2) = 1\\
&p_{\boldsymbol{e}_1}(\boldsymbol{s}_1|\boldsymbol{s}_2,\boldsymbol{a}_1) = 0, p_{\boldsymbol{e}_1}(\boldsymbol{s}_2|\boldsymbol{s}_2,\boldsymbol{a}_1) = 1, p_{\boldsymbol{e}_1}(\boldsymbol{s}_1|\boldsymbol{s}_2,\boldsymbol{a}_2) = 1, p_{\boldsymbol{e}_1}(\boldsymbol{s}_2|\boldsymbol{s}_2,\boldsymbol{a}_2) = 0\\
&p_{\boldsymbol{e}_2}(\boldsymbol{s}_1|\boldsymbol{s}_1,\boldsymbol{a}_1) = 0, p_{\boldsymbol{e}_2}(\boldsymbol{s}_2|\boldsymbol{s}_1,\boldsymbol{a}_1) = 1, p_{\boldsymbol{e}_2}(\boldsymbol{s}_1|\boldsymbol{s}_1,\boldsymbol{a}_2) = 1, p_{\boldsymbol{e}_2}(\boldsymbol{s}_2|\boldsymbol{s}_1,\boldsymbol{a}_2) = 0\\
&p_{\boldsymbol{e}_2}(\boldsymbol{s}_1|\boldsymbol{s}_2,\boldsymbol{a}_1) = 1, p_{\boldsymbol{e}_2}(\boldsymbol{s}_2|\boldsymbol{s}_2,\boldsymbol{a}_1) = 0, p_{\boldsymbol{e}_2}(\boldsymbol{s}_1|\boldsymbol{s}_2,\boldsymbol{a}_2) = 0, p_{\boldsymbol{e}_2}(\boldsymbol{s}_2|\boldsymbol{s}_2,\boldsymbol{a}_2) = 1,
\end{aligned} \tag{8}$$



In this setting, there are four deterministic policies:

$$\begin{aligned}
&\pi_1(\boldsymbol{s}_1) = \boldsymbol{a}_1, \pi_1(\boldsymbol{s}_2) = \boldsymbol{a}_1, \quad \pi_2(\boldsymbol{s}_1) = \boldsymbol{a}_1, \pi_2(\boldsymbol{s}_2) = \boldsymbol{a}_2,\\
&\pi_3(\boldsymbol{s}_1) = \boldsymbol{a}_2, \pi_3(\boldsymbol{s}_2) = \boldsymbol{a}_1, \quad \pi_4(\boldsymbol{s}_1) = \boldsymbol{a}_2, \pi_4(\boldsymbol{s}_2) = \boldsymbol{a}_2.
\end{aligned} \tag{9}$$

For any policy μ, we denote that $\rho_{1,\mu}, \rho_{2,\mu}$ are the state distribution of μ under the environment $\mathcal{P}_{\boldsymbol{e}_1}$ or $\mathcal{P}_{\boldsymbol{e}_2}$ respectively. Then we can calculate that

$$\begin{aligned}
&\rho_{1,\pi_1} = \left(\frac{1}{2}, \frac{1}{2}\right), \rho_{2,\pi_1} = \left(\frac{1}{2}, \frac{1}{2}\right);\\
&\rho_{1,\pi_2} = \left(\frac{1+\gamma}{2}, \frac{1-\gamma}{2}\right), \rho_{2,\pi_2} = \left(\frac{1-\gamma}{2}, \frac{1+\gamma}{2}\right);\\
&\rho_{1,\pi_3} = \left(\frac{1-\gamma}{2}, \frac{1+\gamma}{2}\right), \rho_{2,\pi_3} = \left(\frac{1+\gamma}{2}, \frac{1-\gamma}{2}\right);\\
&\rho_{1,\pi_4} = \left(\frac{1}{2}, \frac{1}{2}\right), \rho_{2,\pi_4} = \left(\frac{1}{2}, \frac{1}{2}\right).
\end{aligned} \tag{10}$$

As the embodiment probability is $p(\boldsymbol{e}_1) = p(\boldsymbol{e}_2) = \frac{1}{2}$, all these four policy share the same state distribution as

$$\rho_{\pi_1} = \rho_{\pi_2} = \rho_{\pi_3} = \rho_{\pi_4} = \left(\frac{1}{2}, \frac{1}{2}\right). \tag{11}$$

Furthermore, we consider a stochastic policy π satisfies that

$$\pi(\boldsymbol{a}_1|\boldsymbol{s}_1) = 1, \pi(\boldsymbol{a}_2|\boldsymbol{s}_1) = 0, \quad \pi(\boldsymbol{a}_1|\boldsymbol{s}_2) = \frac{1}{2}, \pi(\boldsymbol{a}_2|\boldsymbol{s}_2) = \frac{1}{2}. \tag{12}$$

Next we will calculate $\rho_{1,\pi}$ and $\rho_{2,\pi}$. For $\rho_{1,\pi}$, at the timestep 0, we have the initial state distribution as $p_0(\boldsymbol{s}_1) = p_0(\boldsymbol{s}_2) = \frac{1}{2}$, assume that at timestep t we have corresponding $p_t(\boldsymbol{s}_1), p_t(\boldsymbol{s}_2)$, we can naturally get the recurrence relation as

$$p_{t+1}(\boldsymbol{s}_1) = p_t(\boldsymbol{s}_1) + \frac{1}{2}p_t(\boldsymbol{s}_2), \quad p_{t+1}(\boldsymbol{s}_2) = \frac{1}{2}p_t(\boldsymbol{s}_2), \tag{13}$$

Naturally, we have $p_t(\boldsymbol{s}_1) = 1 - \frac{1}{2^t}, p_t(\boldsymbol{s}_2) = \frac{1}{2^t}$ and thus the discount state distribution of $\boldsymbol{s}_2$ is

$$(1-\gamma)\sum_{t=0}^{\infty}\frac{\gamma^t}{2} = \frac{1-\gamma}{2-\gamma}. \tag{14}$$

https://doi.org/10.52202/079017-1731

And we have

$$\rho_{1,\pi} = \left(\frac{1}{2-\gamma}, \frac{1-\gamma}{2-\gamma}\right). \tag{15}$$

Similarly, we can calculate $\rho_{2,\pi}$. At the timestep 0, we have the initial state distribution as $p_0(\boldsymbol{s}_1) = p_0(\boldsymbol{s}_2) = \frac{1}{2}$, assume that at timestep t we have corresponding $p_t(\boldsymbol{s}_1), p_t(\boldsymbol{s}_2)$, we can naturally get the recurrence relation as

$$p_{t+1}(\boldsymbol{s}_1) = \frac{1}{2}p_t(\boldsymbol{s}_2), \quad p_{t+1}(\boldsymbol{s}_2) = p_t(\boldsymbol{s}_1) + \frac{1}{2}p_t(\boldsymbol{s}_2), \tag{16}$$

As $p_t(\boldsymbol{s}_1) + p_t(\boldsymbol{s}_2) = 1$, we can solve this recurrence relation via

$$\begin{aligned} p_{t+1}(\boldsymbol{s}_1) =& \frac{1}{2}p_t(\boldsymbol{s}_2) = \frac{1}{2} - \frac{1}{2}p_t(\boldsymbol{s}_1), \\ (-2)^{t+1}p_{t+1}(\boldsymbol{s}_1) =& (-2)^t p_t(\boldsymbol{s}_1) - (-2)^t \\ =& ... = (-2)^0 p_0(\boldsymbol{s}_1) - \left((-2)^t + (-2)^{t-1} + ... + (-2)^0\right) \\ =& \frac{1}{2} - \frac{1-(-2)^{t+1}}{3} = \frac{1}{6} + \frac{(-2)^{t+1}}{3}, \\ p_{t+1}(\boldsymbol{s}_1) =& \frac{1}{6\times(-2)^{t+1}} + \frac{1}{3} \end{aligned} \tag{17}$$

Thus the discount state distribution of $\boldsymbol{s}_1$ is

$$(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\left(\frac{1}{6\times(-2)^t} + \frac{1}{3}\right) = \frac{1}{3} + \frac{1-\gamma}{6}\sum_{t=0}^{\infty}\left(-\frac{\gamma}{2}\right)^t = \frac{1}{3} + \frac{1-\gamma}{6}\frac{1}{1+\frac{\gamma}{2}} = \frac{1}{2+\gamma}. \tag{18}$$

And we have

$$\rho_{2,\pi} = \left(\frac{1}{2+\gamma}, \frac{1+\gamma}{2+\gamma}\right). \tag{19}$$

As the embodiment probability is $p(\boldsymbol{e}_1) = p(\boldsymbol{e}_2) = \frac{1}{2}$, the state distribution of π is

$$\rho_\pi = \left(\frac{2}{4-\gamma^2}, \frac{2-\gamma^2}{4-\gamma^2}\right). \tag{20}$$

Taking any $\gamma \in (0,1)$, it is obvious that ρ_π is not within the closure composed of $\rho_{\pi_1}, \rho_{\pi_2}, \rho_{\pi_3}, \rho_{\pi_4}$ (actually the point (1/2,1/2)). Thus we have explained that the vertices of $\mathcal{D}^{\mathcal{E}}$ might no longer be simple deterministic policies.

A.2 Proof of Theorem 3.2

Proof. Recall that

$$\mathcal{F}(\pi, \pi^*, \mathcal{R}_{\text{ext}}, e) \triangleq \left[\mathbb{E}_{p_{\mathcal{M}^c_{\boldsymbol{e}},\pi^*}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \mathbb{E}_{p_{\mathcal{M}^c_{\boldsymbol{e}},\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \beta D_{\text{KL}}(p_{\mathcal{M}^c_{\boldsymbol{e}},\pi^*}(\tau) \| p_{\bar{\mathcal{M}},\pi}(\tau))\right]. \tag{21}$$

We set a functional f satisfying that

$$f(p(\tau)) = \mathbb{E}_{p(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \mathbb{E}_{p_{\mathcal{M}^c_{\boldsymbol{e}},\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \beta D_{\text{KL}}(p(\tau) \| p_{\bar{\mathcal{M}},\pi}(\tau)). \tag{22}$$

Using the calculus of variations, we can calculate its optimal value at the point p^* satisfying that

$$\mathcal{R}_{\text{ext}}(\tau) = \beta \log \frac{p^*(\tau)}{p_{\bar{\mathcal{M}},\pi}(\tau)} + b\beta, \tag{23}$$

here b is a constant not related to p^*, and we have $p^*(\tau) = p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta} - b}$. As $\int p^*(\tau) = 1$, we can calculate that

$$b = \log \int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}d\tau, \quad p^*(\tau) = \frac{p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}}{\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}d\tau}. \tag{24}$$

Consequently, we have

$$\begin{aligned}
&\max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)} \mathcal{F}(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) = \mathbb{E}_{p^*(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \beta D_{\text{KL}}(p^*(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau)) \\
=&\int p^*(\tau)\mathcal{R}_{\text{ext}}(\tau)d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \beta\int p^*(\tau)\log\frac{p^*(\tau)}{p_{\bar{\mathcal{M}},\pi}(\tau)}d\tau \\
=&\int p^*(\tau)\mathcal{R}_{\text{ext}}(\tau)d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \beta\int p^*(\tau)\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}d\tau + \beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}d\tau \\
=&\beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)].
\end{aligned} \tag{25}$$

Similarly, we set a functional g satisfying that

$$g(r(\tau)) = \beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r(\tau)}{\beta}}d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}[r(\tau)]. \tag{26}$$

Using the calculus of variations, we can calculate its optimal value at the point r^* satisfying that

$$\beta\frac{\frac{1}{\beta}p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r^*(\tau)}{\beta}}}{\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r^*(\tau)}{\beta}}d\tau} = p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau), \quad \frac{r^*(\tau)}{\beta} = \log\frac{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}{p_{\bar{\mathcal{M}},\pi}(\tau)} + \log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r^*(\tau)}{\beta}}d\tau. \tag{27}$$

Thus we can calculate that

$$\begin{aligned}
&\min_{\mathcal{R}_{\text{ext}}(\tau)}\max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)} \mathcal{F}(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) = \beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r^*(\tau)}{\beta}}d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}[r^*(\tau)] \\
=&\beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r^*(\tau)}{\beta}}d\tau - \beta\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}\left[\log\frac{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}{p_{\bar{\mathcal{M}},\pi}(\tau)}\right] - \beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{r^*(\tau)}{\beta}}d\tau \\
=&-\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau)\right),
\end{aligned} \tag{28}$$

i.e.,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\min_{\mathcal{R}_{\text{ext}}(\tau)}\max_{p_{\mathcal{M}_{\boldsymbol{e}},\pi^*}(\tau)} \mathcal{F}(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) = \beta\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\left[-D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right] \\
=&\beta\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}\left[\log\frac{p_\pi(\tau)}{p_\pi(\tau|\mathcal{M}_{\boldsymbol{e}}^c)}\right] = \beta\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}\left[\log\frac{p_\pi(\tau)}{p_\pi(\boldsymbol{e},\tau)/p(\boldsymbol{e})}\right] \\
=&\beta\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}\left[\log\frac{p(\boldsymbol{e})}{p_\pi(\boldsymbol{e},\tau)/p_\pi(\tau)}\right] = \beta\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)}\left[\log\frac{p(\boldsymbol{e})}{p_\pi(\boldsymbol{e}|\tau)}\right].
\end{aligned} \tag{29}$$

Thus we have proven this result. □

A.3 Detailed Discussion and Proof about Embodiment-Aware Skill Discovery

Here we discuss our *cross-embodiment skill-based adaptation objective*.

We begin by proving Eq. (7). First, we show that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}}\min_{\mathcal{R}_{\text{ext}}(\tau)}\max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)} \mathcal{F}_s(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) \\
=&-\mathbb{E}_{\boldsymbol{e}}\max_{p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)}\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)}\left[\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z})\|p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right].
\end{aligned} \tag{30}$$

Our proof is similar to the proof in Appendix A.2. Recall that

$$\begin{aligned}
\mathcal{F}_s(\pi,\pi^*,\mathcal{R}_{\text{ext}},\boldsymbol{e}) \triangleq \Big[&\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)}[\mathcal{R}_{\text{ext}}(\tau)] - \max_{\boldsymbol{z}^*}\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)] \\
&-\beta D_{\text{KL}}(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)\|p_{\bar{\mathcal{M}},\pi}(\tau))\Big].
\end{aligned} \tag{31}$$

Similar to Eq. (22)-Eq. (25), we have

$$\begin{aligned}
&\max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau|\boldsymbol{z})}[\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau|\boldsymbol{z})}[\mathcal{R}_{\text{ext}}(\tau)] - \max_{\boldsymbol{z}^*}\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)] - \beta D_{\text{KL}}(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau|\boldsymbol{z})\|p_{\bar{\mathcal{M}},\pi}(\tau))] \\
=&\beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}d\tau - \max_{\boldsymbol{z}^*}\mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)] \\
=&\min_{\boldsymbol{z}^*}\left[\beta\log\int p_{\bar{\mathcal{M}},\pi}(\tau)e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}}d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)]\right].
\end{aligned} \tag{32}$$

https://doi.org/10.52202/079017-1731

Also, similar to Eq. (26)-Eq. (28), we have

$$\begin{aligned}
&\min_{\mathcal{R}_{\text{ext}}(\tau)} \min_{\boldsymbol{z}^*} \left[\beta \log \int p_{\bar{\mathcal{M}},\pi}(\tau) e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}} d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)]\right] \\
=&\min_{\boldsymbol{z}^*} \min_{\mathcal{R}_{\text{ext}}(\tau)} \left[\beta \log \int p_{\bar{\mathcal{M}},\pi}(\tau) e^{\frac{\mathcal{R}_{\text{ext}}(\tau)}{\beta}} d\tau - \mathbb{E}_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*)}[\mathcal{R}_{\text{ext}}(\tau)]\right] \\
=&\min_{\boldsymbol{z}^*} \left[-\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right].
\end{aligned} \tag{33}$$

Thus we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}} \min_{\mathcal{R}_{\text{ext}}(\tau)} \max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)} \mathcal{F}_s(\pi, \pi^*, \mathcal{R}_{\text{ext}}, \boldsymbol{e}) \\
=&\mathbb{E}_{\boldsymbol{e}} \min_{\boldsymbol{z}^*} \left[-\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right] \\
=&-\mathbb{E}_{\boldsymbol{e}} \max_{\boldsymbol{z}^*} \left[\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}^*) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right] \\
=&-\mathbb{E}_{\boldsymbol{e}} \max_{p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)} \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)} \left[\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right],
\end{aligned} \tag{34}$$

where the last equality holds from the fact that the maximum is achieved when putting all the probability weight on the input $\boldsymbol{z}$ maximizing $D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)$.

Next, we will show that $D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)$ is a general form of our Theorem 3.2 and the results in the single-embodiment setting [11]. Naturally, when we ignore $\boldsymbol{z}$, $\mathcal{F}_s(\pi, \pi^*, \mathcal{R}_{\text{ext}}, \boldsymbol{e})$ will degenerate into $\mathcal{F}(\pi, \pi^*, \mathcal{R}_{\text{ext}}, \boldsymbol{e})$, and Eq. (7) will also degenerate into Eq. (4), i.e., the results in Theorem 3.2. On the other hand, if we change Eq. (7) into the single-embodiment setting, i.e., $\mathcal{E}$ is a Dirac distribution with the probability $p(\boldsymbol{e}) = 1$ for some fixed $\boldsymbol{e}$, then we have

$$\begin{aligned}
&\max_{\pi} \min_{\mathcal{R}_{\text{ext}}(\tau)} \max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)} \mathcal{F}_s(\pi, \pi^*, \mathcal{R}_{\text{ext}}, \boldsymbol{e}) \\
=&\max_{\pi} \left[-\max_{p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)} \mathbb{E}_{z\sim p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)} \left[\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}) \| p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)\right)\right]\right] \\
=&-\min_{\pi} \max_{p(\boldsymbol{z})} \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})} \left[\beta D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}) \| p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau)\right)\right] \\
\approx&-\min_{\rho} \max_{p(\boldsymbol{z})} \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})} \left[\beta D_{\text{KL}}\left(p(\tau|\boldsymbol{z}) \| \rho(\tau)\right)\right],
\end{aligned} \tag{35}$$

the last approximation simplifies the complex coupling relationship between π and $\boldsymbol{z}$, following [11]. Furthermore, by Lemma 6.5 in [11] (proof in Theorem 13.1.1 from [9]), we have

$$\min_{\rho} \max_{p(\boldsymbol{z})} \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})} \left[D_{\text{KL}}\left(p(\tau|\boldsymbol{z}) \| \rho(\tau)\right)\right] = \max_{p(\boldsymbol{z})} \mathcal{I}(\tau; \boldsymbol{z}), \tag{36}$$

which is the objective of existing single-embodiment skill-discovery methods.

Finally, we will Eq. (7), which further indicates that our cross-embodiment skill-based objective can be decomposed into two terms: one for handling cross-embodiment while the other aims at discovering skills. Actually, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{e}\sim\mathcal{E}} \min_{\mathcal{R}_{\text{ext}}(\tau)} \max_{p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi^*}(\tau)} \mathcal{F}_s(\pi, \pi^*, \mathcal{R}_{\text{ext}}, \boldsymbol{e}) \\
=&\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\mathcal{M}_{\boldsymbol{e}}^c)} \left[D_{\text{KL}}\left(p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}(\tau|\boldsymbol{z}) \| p_{\bar{\mathcal{M}},\pi}(\tau)\right)\right] \\
=&\int \frac{p_\pi(\boldsymbol{e}, \tau, \boldsymbol{z})}{p(\boldsymbol{e})} \log \frac{p_\pi(\tau|\boldsymbol{z}, \boldsymbol{e})}{p_\pi(\tau)} d\boldsymbol{z} d\tau = \int p_\pi(\tau, \boldsymbol{z}|\boldsymbol{e}) \log \frac{p_\pi(\boldsymbol{z}, \boldsymbol{e}|\tau)}{p_\pi(\boldsymbol{e}, \boldsymbol{z})} d\boldsymbol{z} d\tau \\
=&\int p_\pi(\tau, \boldsymbol{z}|\boldsymbol{e}) \log \frac{p_\pi(\boldsymbol{e}|\tau) p_\pi(\boldsymbol{z}|\boldsymbol{e}, \tau)}{p_\pi(\boldsymbol{e}) p_\pi(\boldsymbol{z}|\boldsymbol{e})} d\boldsymbol{z} d\tau \\
=&\int p_\pi(\tau|\boldsymbol{e}) \log \frac{p_\pi(\boldsymbol{e}|\tau)}{p_\pi(\boldsymbol{e})} d\tau + \int p_\pi(\tau, \boldsymbol{z}|\boldsymbol{e}) \log \frac{p_\pi(\tau, \boldsymbol{z}|\boldsymbol{e})}{p_\pi(\boldsymbol{z}|\boldsymbol{e}) p_\pi(\tau|\boldsymbol{e})} d\boldsymbol{z} d\tau \\
=&\mathbb{E}_{\tau\sim p_{\mathcal{M}_{\boldsymbol{e}}^c,\pi}} \left[\log \frac{p_\pi(\boldsymbol{e}|\tau)}{p_\pi(\boldsymbol{e})} + D_{\text{KL}}(p_\pi(\tau, \boldsymbol{z}|\boldsymbol{e}) \| p_\pi(\boldsymbol{z}|\boldsymbol{e}) p_\pi(\tau|\boldsymbol{e}))\right].
\end{aligned} \tag{37}$$

B Experimental Details

In this section, we will introduce more detailed information about our experiments. In Sec. B.1, we introduce the detailed environments and tasks used in our experiments. In Sec. B.2, we will illustrate all the baselines compared in experiments. Also, all hyper-parameters of experiments are in Sec. B.3. Moreover, we supplement more detailed experimental results about state-based DMC, image-based DMC, and Robosuite in Sec. B.4, Sec. B.5, and Sec. B.6, respectively. Then we conduct detailed generalization results of pre-trained models and fine-tuned models in Sec. B.9 and Sec. B.10, respectively. Finally, we report more detailed real-world experiments in Sec. B.12.

B.1 Embodiments and Tasks

State-based DMC. This benchmark is based on DMC [56] and URLB [27] with state-based observation. Each domain contains one robot and four downstream tasks. We extend it into the cross-embodiment settings: Walker-mass, Quadruped-mass, and Quadruped-damping. Walker-mass extends the Walker robot in DMC, which is a two-leg robot, and designs a distribution with different mass m, i.e., m times the mass of a standard walker robot. Similarly, Quadruped-mass also considers quadruped robots with different mass m. Quadruped-damping, on the other hand, changes the damping of the standard quadruped robot with l times. The detailed parameters of training embodiments and generalization embodiments are in Table 3.

Image-based DMC. This benchmark is the same with state-based DMC but with image-based observation. Thus we consider similar three embodiment distributions: Walker-mass, Quadruped-mass, and Quadruped-damping.

	Train	Generalization
Walker-mass	$m \in \{0.2, 0.6, 1.0, 1.4, 1.8\}$	$m \in \{0.4, 0.8, 1.2, 1.6\}$
Quadruped-mass	$m \in \{0.4, 0.8, 1.0, 1.4\}$	$m \in \{0.6, 1.2\}$
Quadruped-damping	$l \in \{0.2, 0.6, 1.0, 1.4, 1.8\}$	$l \in \{0.4, 0.8, 1.2, 1.6\}$

Table 3: Environment parameters used for state-based DMC and image-based DMC.

Robosuite. This benchmark utilizes the environment in [73] and follows the experimental setting in RL-Vigen [69], of which the cross-embodiment setting includes Panda, IIWA, and Kinova3. Here different embodiments may own different shapes (observations), and dynamics. Similarly, we pre-train cross-embodiments in all these three embodiments and fast fine-tune the pre-trained agents to downstream tasks. Besides these three embodiments, we also directly fine-tune our pre-trained models in one unseen embodiment: Jaco, to validate the cross-embodiment generalization ability of CEURL. For task sets, we consider three widely used tasks: Door, Lift, and TwoArmPegInHole. Noticing that although these three tasks can be finished by the same robots, their demand for robotic arms varies a lot. For example, TwoArmPegInHole needs two robotic arms but the other two tasks only need one. Consequently, we pre-train cross-embodiment agents for each single task, for all methods.

Isaacgym. We first design a setting in simulation based on Unitree A1 in Isaacgym, which is a challenging legged locomotion task and is widely used for real-world legged locomotion. The action space of A1 is a 12-dimension vector, representing 12 joint torque. Thus we consider our A1-disabled benchmark, including 12 embodiments, each of which owns a joint torque failure, i.e., the torque output of this joint is always 0 in this embodiment. This setting is practical as our robot may experience partial joint failure during use, and we still hope that it can complete the task as much as possible.

Moreover, we deploy PEAC into real-world Aliengo robots with failure joints. Similarly, we consider the embodiment distribution Aliengo-disabled, which owns 12 embodiments, each of which owns a joint torque failure respectively. We first pre-train a unified agent across these 12 embodiment in reward-free environments. During fine-tuning, for each embodiment, we utilize the same pre-trained agent to fine-tune the given moving task through this embodiment. Finally, we deploy the fine-tuned agent into the real-world setting to evaluate its movement ability under different kinds of terrains with joint failure.

https://doi.org/10.52202/079017-1731

B.2 Baselines and Implementations

ICM [45]. Intrinsic Curiosity Module (ICM) designs intrinsic rewards as the divergence between the projected state representations in a feature space and the estimations made by a feature dynamics model.

RND [7]. Random Network Distillation (RND) utilizes a predictor network's error in imitating a randomly initialized target network to generate intrinsic rewards, enhancing exploration in learning environments.

Disagreement [46] / Plan2Explore [51]. The Disagreement algorithm leverages prediction variance across multiple models to estimate state uncertainty, guiding exploration towards less certain states. The Plan2Explore algorithm employs a self-supervised, world-model-based framework, using model disagreement to assess environmental uncertainty and incentivize exploration in sparse-reward scenarios.

ProtoRL [62]. Proto-RL combines representation learning and exploration through a self-supervised learning framework, using prototype representations to pre-train task-independent representations in the environment, effectively improving policy learning in continuous control tasks.

APT [35]. Active Pre-training (APT) estimates entropy for a given state using a particle-based estimator based on the K nearest-neighbors algorithm.

LBS [38]. Latent Bayesian Surprise (LBS) applies Bayesian surprise within a latent space, efficiently facilitating exploration by measuring the disparity between an agent's prior and posterior beliefs about system dynamics.

Choreographer [39]. Choreographer is a model-based approach in unsupervised skill learning that employs a world model for skill acquisition and adaptation, distinguishing exploration from skill learning and leveraging a meta-controller for efficient skill adaptation in simulated scenarios, enhancing adaptability to downstream tasks and environmental exploration.

DIAYN [10]. Diversity is All You Need (DIAYN) autonomously learns a diverse set of skills by maximizing mutual information between states and latent skills, using a maximum entropy policy.

SMM [30]. State Marginal Matching (SMM) develops a task-agnostic exploration strategy by learning a policy to match the state distribution of an agent with a given target state distribution.

APS [34]. Active Pre-training with Successor Feature (APS) maximizes the mutual information between states and task variables by reinterpreting and combining variational successor features with nonparametric entropy maximization.

LSD [42]. Lipschitz-constrained Skill Discovery (LSD) adopts a Lipschitz-constrained state representation function, ensuring that maximizing this objective in the latent space leads to an increase in traveled distances or variations in the state space, thereby enabling the discovery of more diverse, dynamic, and far-reaching skills.

CIC [26]. Contrastive Intrinsic Control (CIC) is an unsupervised reinforcement learning algorithm that leverages contrastive learning to maximize the mutual information between state transitions and latent skill vectors, subsequently maximizing the entropy of these embeddings as intrinsic rewards to foster behavioral diversity.

BeCL [61]. Behavior Contrastive Learning (BeCL) utilizes contrastive learning for unsupervised skill discovery, defining its reward function based on the mutual information between states generated by the same skill.

Next, we will introduce the implementations of baselines for all experimental settings.

For **state-based DMC**, almost all baselines (ICM, RND, Disagreement, ProtoRL, DIAYN, SMM, APS) combined with RL backbone DDPG are directly following the official implementation in urlb (https://github.com/rll-research/url_benchmark). For LBS, we refer the implementation in [48] (https://github.com/mazpie/mastering-urlb) and combine it with the code of urlb. For other more recent baselines, we also follow their official implementations, including CIC (https://github.com/rll-research/cic) and BeCL (https://github.com/Rooshy-yang/BeCL).

For **image-based DMC**, almost all baselines (ICM, RND, Plan2Explore, APT, LBS, DIAYN, APS) combined with RL backbone DreamerV2 are directly following the official implementation in [48] (https://github.com/mazpie/mastering-urlb), which currently achieves the leading performance in image-based DMC of urlb. For CIC, we combine its official code (https://github.com/rll-research/cic), which mainly considers state-based DMC, and the DreamerV2 backbone in [48]. Similarly, for LSD, we refer to its official code (https://github.com/seohongpark/LSD) and combine it with the code of [48]. For Choreographer, of which the backbone is DreamerV2, we directly utilize its official code (https://github.com/mazpie/choreographer).

For **Robosuite**, our code is based on the code of RL-Vigen [69] (https://gemcollector.github.io/RL-ViGen), including the RL backbone DrQ. For **Isaacgym**, our code is based on the official code of [74] (https://github.com/ZiwenZhuang/parkour), which implements five downstream tasks (run, climb, leap, crawl, tilt). For these two settings (Robosuite and Isaacgym), as there are few works considering unsupervised RL in such a challenging setting, we implement classical baselines (ICM, RND, LBS) by referring their implementations in urlb (https://github.com/rll-research/url_benchmark) and [48] (https://github.com/mazpie/mastering-urlb).

B.3 Hyper-parameters

Baseline hyper-parameters are taken from their implementations (see Appendix B.2 above). Here we introduce PEAC's hyper-parameters. For all settings, hyper-parameters of RL backbones (DDPG, DreamerV2, PPO) follow standard settings.

First, for PEAC in state-based DMC with RL backbone DDPG, our code is based on urlb (https://github.com/mazpie/mastering-urlb) and inherits hyper-parameters of DDPG. For completeness, we list all hyper-parameters as

DDPG Hyper-parameter	Value
Replay buffer capacity	10^6
Action repeat	1
Seed frames	4000
n-step returns	3
Mini-batch size	1024
Seed frames	4000
Discount γ	0.99
Optimizer	Adam
Learning rate	1e-4
Agent update frequency	2
Critic target EMA rate τ_Q	0.01
Features dim.	1024
Hidden dim.	1024
Exploration stddev clip	0.3
Exploration stddev value	0.2
Number pre-training frames	2×10^6
Number fine-turning frames	1×10^5
PEAC Hyper-parameter	Value
Historical information encoder	GRU ($\dim(\mathcal{S})+\dim(\mathcal{A})\rightarrow 1024$)
Encoded historical information length	10
Embodiment context model	MLP (1024 $\rightarrow$Embodiment context dim)

Table 4: Details of hyper-parameters used for state-based DMC.

Next, for PEAC-LBS and PEAC-DIAYN in image-based DMC with RL backbone DreamerV2, our code is based on [48] (https://github.com/mazpie/mastering-urlb). Hyper-parameters of

https://doi.org/10.52202/079017-1731

PEAC-LBS and PEAC-DIAYN inherit DreamerV2's hyper-parameters, as well as inherit hyper-parameters of LBS and DIAYN, respectively.

DreamerV2 Hyper-parameter	Value
Environment frames/update	10
MLP number of layers	4
MLP number of units	400
Hidden layers dimension	400
Adam epsilon	1×10^{-5}
Weight decay	1×10^{-6}
Gradient clipping	100
World Model	
Batch size	50
Sequence length	50
Discrete latent state dimension	32
Discrete latent classes	32
GRU cell dimension	200
KL free nats	1
KL balancing	0.8
Adam learning rate	3×10^{-4}
Slow critic update interval	100
Actor-Critic	
Imagination horizon	15
Discount γ	0.99
GAE λ	0.95
Adam learning rate	8×10^{-5}
Actor entropy loss scale	1×10^{-4}
PEAC-LBS Hyper-parameter	Value
Embodiment context model	MLP (DreamerV2 encoder dim $\rightarrow$ 200 $\rightarrow$ 200 $\rightarrow$ Embodiment context dim)
LBS model	MLP (DreamerV2 encoder dim $\rightarrow$ 200 $\rightarrow$ 200 $\rightarrow$ 200 $\rightarrow$ 200 $\rightarrow$ 1)
PEAC-DIAYN Hyper-parameter	Value
Embodiment context model	MLP (DreamerV2 encoder dim $\rightarrow$ 200 $\rightarrow$ 200 $\rightarrow$ Embodiment context dim)
DIAYN model	MLP (DreamerV2 encoder dim $\rightarrow$ 200 $\rightarrow$ 200 $\rightarrow$ skill dim)

Table 5: Details of hyper-parameters used for image-based DMC.

Then, for PEAC in Robosuite, our code follows RL-Vigen [69] (https://gemcollector.github.io/RL-ViGen). PEAC's hyper-parameters, inheriting DrQ's hyperparameters, include

DrQ Hyper-parameter	Value
Discount factor	0.99
Optimizer	Adam
Learning rate	1e-4
Action repeat	1
N-step return	1
Hidden dim	1024
Frame stack	3
Replay Buffer size	1000000
Feature dim	50
PEAC Hyper-parameter	Value
Historical information encoder	GRU (Encoder Feature Dim + $\dim(\mathcal{A}) \rightarrow 50$)
Encoded historical information length	10
Embodiment context model	MLP (50 $\rightarrow$Embodiment context dim)

Table 6: Details of hyper-parameters used for Robosuite.

Finally, for PEAC in A1-disabled of Isaacgym with RL backbone PPO, our code follows [74] (https://github.com/ZiwenZhuang/parkour). PEAC's hyper-parameters, inheriting PPO's hyperparameters, include

PPO Hyper-parameter	Value
PPO clip range	0.2
GAE λ	0.95
Learning rate	1e-4
Reward discount factor	0.99
Minimum policy std	0.2
Number of environments	4096
Number of environment steps per training batch	24
Learning epochs per training batch	5
Number of mini-batches per training batch	4
PEAC Hyper-parameter	Value
Historical information encoder	GRU ($\dim(\mathcal{S}) + \dim(\mathcal{A}) \rightarrow 128$)
Encoded historical information length	24
Embodiment context model	MLP (128 $\rightarrow$Embodiment context dim)

Table 7: Details of hyper-parameters used for Isaacgym.

B.4 Detailed results in state-based DMC

In Table 8, we present detailed results in state-based DMC of all four statistics (medium, IQM, mean, OG) for baselines and our PEAC. The results indicate that PEAC performs the best in all these four metrics, while BeCL and CIC perform second and third respectively. Moreover, we report individual results for each downstream task of state-based DMC in Table 9. PEAC performs comparably to BeCL as well as CIC in the Walker-mass tasks and best on most Quadruped-mass and Quadruped-damping tasks. Especially, in the challenging Quadruped-damping setting, PEAC can complete cross-embodiment downstream tasks and significantly outperforms BeCL and CIC.

Metrics	Median	IQM	Mean	Optimality Gap
ICM	0.37	0.35	0.37	0.63
RND	0.48	0.47	0.47	0.53
Disagreement	0.40	0.39	0.38	0.62
ProtoRL	0.52	0.51	0.52	0.48
LBS	0.51	0.48	0.50	0.50
DIAYN	0.47	0.43	0.46	0.54
SMM	0.38	0.36	0.39	0.61
APS	0.48	0.47	0.49	0.51
CIC	0.52	0.55	0.54	0.46
BeCL	0.60	0.62	0.61	0.39
PEAC (Ours)	**0.67**	**0.69**	**0.67**	**0.33**

Table 8: **Aggregate metrics [2] in state-based DMC**. For every algorithm, there are 3 embodiment settings, each trained with 10 seeds and fine-tuned under 4 downstream tasks, thus each statistic for every method has 120 runs.

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
ICM	665.3	418.0	146.2	246.6	460.2	229.5	215.6	323.5	365.8	182.4	180.2	203.1	0.37
RND	588.9	386.7	176.4	253.8	**820.6**	563.7	409.6	589.5	325.4	166.2	156.0	235.8	0.47
Disagreement	549.3	331.6	139.8	250.0	555.5	372.4	329.8	506.1	274.0	139.1	142.6	217.2	0.38
ProtoRL	731.6	458.0	192.0	325.8	687.0	430.0	348.7	514.3	498.3	336.4	275.2	364.1	0.52
LBS	618.0	370.3	136.8	343.1	740.8	499.1	388.7	517.2	574.0	302.0	258.4	335.8	0.50
DIAYN	502.1	245.2	106.8	212.7	682.7	484.3	371.0	469.1	553.4	386.7	331.8	394.8	0.46
SMM	673.5	509.2	220.7	329.6	357.0	176.4	189.7	277.8	314.2	174.0	183.0	287.5	0.39
APS	629.8	429.8	129.4	291.4	653.1	474.1	325.3	533.7	479.9	254.9	302.4	403.7	0.49
CIC	824.8	536.6	220.7	327.7	762.5	610.9	**442.7**	617.5	335.9	194.1	166.4	267.5	0.54
BeCL	**838.6**	**623.6**	**238.5**	**348.1**	729.8	445.0	349.4	557.1	553.7	485.8	292.0	509.8	0.61
PEAC (Ours)	823.8	499.9	210.6	320.5	786.0	**754.5**	388.3	**645.6**	**712.3**	**644.1**	**393.5**	**541.8**	**0.67**

Table 9: **Detailed results in state-based DMC**. Average cumulative reward (mean of 10 seeds) of the best policy.

https://doi.org/10.52202/079017-1731

B.5 Detailed results in image-based DMC

In Table 10, we present detailed results in state-based DMC of all four statistics (medium, IQM, mean, OG) for baselines and our PEAC-LBS as well as PEAC-DIAYN. Besides these statistics, in Table 11, we further report the detailed results for the 12 downstream tasks, averaged across all embodiments and seeds. Overall, PEAC-LBS's performance is steadily on top, outperforming existing methods, especially in Walker-mass. Also, compared with other pure skill discovery methods, PEAC-DIAYN performs more consistently on all tasks and achieves higher average rewards.

Metrics	Median	IQM	Mean	Optimality Gap
DIAYN	0.58	0.53	0.56	0.44
APS	0.66	0.64	0.66	0.34
LSD	0.64	0.63	0.64	0.36
CIC	0.67	0.66	0.67	0.33
PEAC-DIAYN (Ours)	0.72	0.69	0.71	0.29
ICM	0.77	0.76	0.77	0.24
RND	0.74	0.75	0.75	0.26
Plan2Explore	0.78	0.80	0.78	0.22
APT	0.75	0.75	0.75	0.25
LBS	0.84	0.85	0.84	0.17
Choreographer	0.84	0.86	0.84	0.17
PEAC-LBS (Ours)	**0.93**	**0.93**	**0.92**	**0.10**

Table 10: **Aggregate metrics [2] in image-based DMC**. For every algorithm, there are 3 embodiment settings, each trained with 3 seeds and fine-tuned under 4 downstream tasks, thus each statistic for every method has 36 runs.

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
DIAYN	772.6	515.1	193.8	365.7	583.9	425.9	311.9	431.8	791.8	410.8	367.4	536.1	0.56
APS	906.2	554.1	228.9	473.2	814.9	414.8	413.5	677.2	850.5	417.0	379.1	560.7	0.66
LSD	912.8	644.1	227.9	401.9	769.0	409.2	401.3	555.5	634.9	447.9	481.4	608.5	0.64
CIC	930.5	725.7	289.8	423.6	850.3	410.4	341.8	488.2	883.3	457.0	416.5	572.3	0.67
PEAC-DIAYN (Ours)	954.5	731.8	305.9	491.1	720.5	420.1	446.6	548.8	867.3	503.6	440.8	671.6	0.71
ICM	946.5	797.0	304.6	493.8	**937.4**	610.4	461.0	809.7	834.9	458.0	438.7	683.3	0.77
RND	950.1	749.2	326.7	510.6	903.9	509.5	444.4	733.5	814.3	444.5	405.5	708.6	0.75
Plan2Explore	956.5	836.0	342.2	518.8	895.7	652.4	470.9	634.5	890.9	583.8	421.2	689.7	0.78
APT	914.2	781.1	332.8	485.5	833.6	513.4	489.2	718.3	863.8	494.4	450.1	639.1	0.75
LBS	937.9	754.4	365.1	531.2	900.1	732.3	535.1	777.4	883.4	731.7	511.1	758.5	0.84
Choreographer	957.8	819.4	368.3	551.6	913.2	686.1	459.8	757.1	888.0	715.6	590.1	706.8	0.84
PEAC-LBS (Ours)	**964.5**	**892.1**	**418.3**	**673.5**	917.7	**744.5**	**607.7**	**814.9**	**908.3**	**775.9**	**648.2**	**784.1**	**0.92**

Table 11: **Detailed results in image-based DMC**. Average cumulative reward (mean of 3 seeds) of the best policy trained by different algorithms. We **bold** the best performance of each task. The six baselines above are exploration-based methods (Choreographer utilizes both exploration and skill-discovery techniques), while the following four baselines are skill-discovery methods.

B.6 Detailed results in Robosuite

In Table 12, we report detailed results in Robosuite with all tasks and robotic arms. Overall, PEAC performs better in more tasks and owns better generalization ability to unseen robot Jaco.

Domains	Door				Lift				TwoArmPegInHole			
	Panda	IIWA	Kinova3	Jaco	Panda	IIWA	Kinova3	Jaco r	Panda	IIWA	Kinova3	Jaco
ICM	156.2	134.4	32.2	107.7	**134.1**	**151.6**	85.9	89.5	**288.4**	**282.8**	304.1	337.8
RND	128.4	150.5	**148.0**	127.2	74.0	92.7	84.4	64.2	272.7	277.6	**312.8**	**363.5**
LBS	120.4	128.7	79.6	104.0	89.7	80.2	66.7	87.9	268.0	271.5	**314.9**	308.0
PEAC (Ours)	**225.4**	**158.1**	112.4	**161.9**	109.8	140.1	**92.2**	**118.5**	**285.5**	**281.3**	**311.4**	321.9

Table 12: **Detailed results in Robosuite**.

https://doi.org/10.52202/079017-1731

B.7 Ablation of timesteps in image-based DMC

In Figure 8, we show additional results about the performance in three domains of image-based DMC for different algorithms and pre-training timesteps. Overall, PEAC-LBS outperforms all methods, while Choreographer and LBS are still competitive on the Quadruped-mass. Also, PEAC-DIAYN outperforms all other pure skill discovery methods.

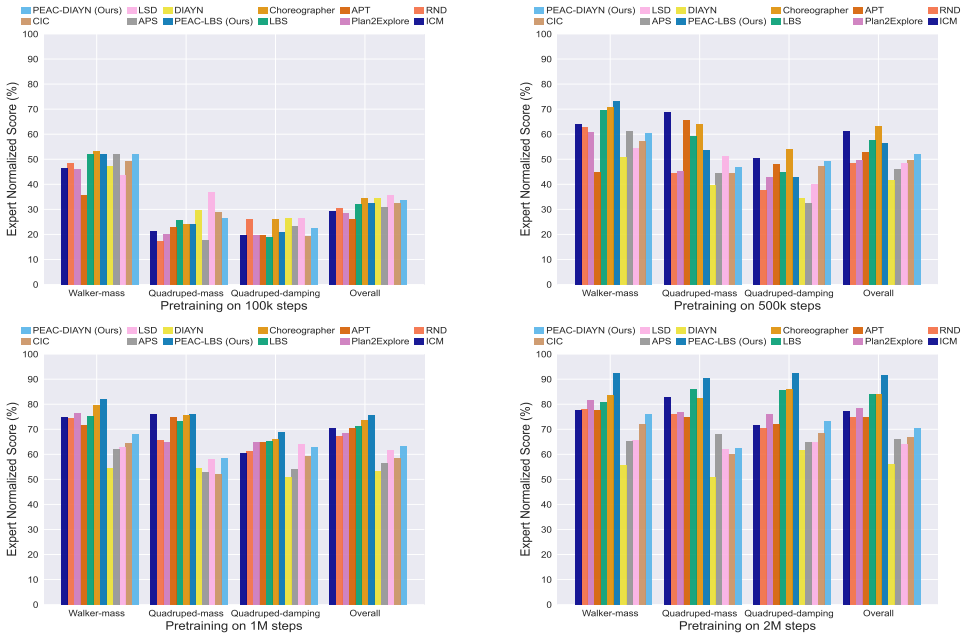


Figure 8: **Ablation study of pre-training steps in image-based DMC**.

B.8 More Ablation Studies

In this part, we conclude more ablation studies to better clarify the contribution of each component in PEAC. First, we supplement ablation studies of the hyperparameter β in Eq. 2 (β is set to 1.0 in all our experiments). As discussed in the paper, β is a parameter that is negatively related to the fine-tuning timesteps and is for balancing the policy improvement term and the policy constraint term. When the fine-tuning timestep tends to the infinity, β tends to 0. Unfortunately, the relationship between β and the fine-tuning timesteps is complicated. Thus we evaluate PEAC-LBS under different β as below

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
$\beta = 1.0$	964.5	892.1	418.3	673.5	917.7	744.5	607.7	814.9	908.3	775.9	648.2	784.1	0.92
$\beta = 0.1$	963.8	877.6	404.8	604.4	905.3	820.0	477.5	797.0	903.0	757.6	648.3	807.6	0.90
$\beta = 0.5$	958.6	896.4	416.5	640.8	929.6	794.3	593.9	806.7	921.2	746.3	540.5	794.7	0.90
$\beta = 2.0$	967.6	891.5	433.8	650.2	945.2	542.2	499.9	780.0	885.9	773.3	499.6	742.2	0.86
$\beta = 3.0$	961.7	901.6	399.6	634.9	892.7	681.0	440.1	728.1	906.0	486.9	466.9	644.5	0.81

Table 13: **Ablation for β of PEAC-LBS in image-based DMC**. Average cumulative reward (mean of 3 seeds) of the best policy trained by different algorithms.

As shown in Table 13, when β is large, the performance of PEAC-LBS decreases more than β is small, but PEAC-LBS is overall stable with different β.

Moreover, to clarify the effectiveness of our embodiment discriminator, we supplement LBS-Context and DIAYN-Context, i.e., combining LBS and DIAYN with the embodiment discriminator in PEAC, which utilizes embodiment information during the pre-training stage. Our results in state-based DMC and Image-based DMC are in Table 14 and Table 15, respectively.

https://doi.org/10.52202/079017-1731

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
LBS	618.0	370.3	136.8	343.1	740.8	499.1	388.7	517.2	574.0	302.0	258.4	335.8	0.50
LBS-Context	784.3	584.7	207.6	389.0	610.1	273.7	308.1	423.1	478.3	355.8	300.4	372.2	0.52
DIAYN	502.1	245.2	106.8	212.7	682.7	484.3	371.0	469.1	553.4	386.7	331.8	394.8	0.46
DIAYN-Context	657.0	341.1	153.1	301.0	735.4	495.2	415.5	581.5	688.4	525.5	290.1	477.0	0.56
PEAC (Ours)	823.8	499.9	210.6	320.5	786.0	754.5	388.3	645.6	712.3	644.1	393.5	541.8	0.67

Table 14: **Ablation study for baselines w/ our embodiment discriminator in state-based DMC**.

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
DIAYN	772.6	515.1	193.8	365.7	583.9	425.9	311.9	431.8	791.8	410.8	367.4	536.1	0.56
DIAYN-Context	946.9	821.9	357.9	465.2	733.7	248.8	251.1	423.1	899.0	350.2	399.7	544.9	0.64
PEAC-DIAYN (Ours)	954.5	731.8	305.9	491.1	720.5	420.1	446.6	548.8	867.3	503.6	440.8	671.6	0.71
LBS	937.9	754.4	365.1	531.2	900.1	732.3	535.1	777.4	883.4	731.7	511.1	758.5	0.84
LBS-Context	933.3	792.4	305.4	530.7	907.7	604.9	477.7	776.7	879.2	797.7	627.3	808.8	0.84
PEAC-LBS (Ours)	964.5	892.1	418.3	673.5	917.7	744.5	607.7	814.9	908.3	775.9	648.2	784.1	0.92

Table 15: **Ablation study for baselines w/ our embodiment discriminator in image-based DMC**.

As shown in these two tables, LBS-Context and DIAYN-Context own comparable or superior performance compared with LBS and DIAYN respectively, and PEAC still significantly outperforms them. Consequently, this ablation study highlights that both the embodiment discriminator and cross-embodiment intrinsic rewards $\mathcal{R}_{\text{CE}}$ are effective for handling CEURL.

B.9 Generalization results of pre-trained models

In Fig. 9, we evaluate the generalization ability of pre-trained models to unseen embodiments of all exploration methods in Walker-mass of image-based DMC. After pre-training on several embodiments, we zero-shot utilize these agents to sample trajectories via two different unseen embodiments. Given the trajectories, we reduce the dimension of the hidden states calculated by the world model via t-SNE [57], where points with different colors represent data generated by different embodiments. As shown in Fig. 9, all the baselines can not distinguish different embodiments, while our PEAC-LBS can roughly divide them into two regions, indicating the pre-trained model of PEAC-LBS own strong generalization ability to unseen embodiments.

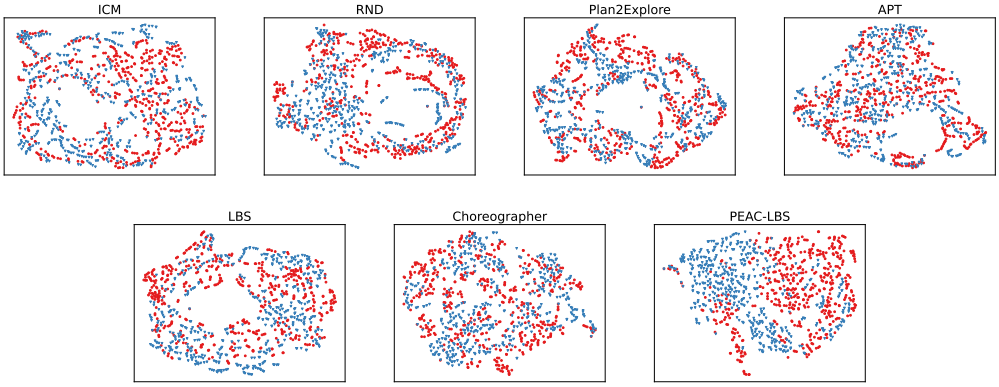


Figure 9: **Visualization of the pre-trained model generalization to unseen embodiments.**

B.10 Generalization results of fine-tuned models

In this part, we evaluate the generalization ability of the fine-tuned agents to unseen embodiments of state-based DMC and image-based DMC. In these two settings, we pre-train and fine-tune the agent with the sampled training embodiments (Train column in Table 3) and zero-shot evaluate the performance of the fine-tuned agents in the same task but with unseen in-distribution embodiments (Generalization column in Table 3). The detailed generalization results of all downstream tasks in state-based DMC and image-based DMC are in Table 16-17, respectively. As shown in Table 16,

https://doi.org/10.52202/079017-1731

PEAC still significantly outperforms all baselines in normalized average return and there is even a greater leading advantage than baselines, compared with the trained embodiments. This indicates that PEAC can effectively generalize to unseen embodiments and effectively handle downstream tasks.

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
ICM	702.0	467.7	146.2	246.6	321.3	165.7	158.8	258.9	259.8	112.3	135.8	134.8	0.32
RND	609.6	421.2	183.5	244.9	810.2	563.2	413.3	583.1	220.5	110.7	87.0	218.3	0.45
Disagreement	537.6	331.6	139.8	250.0	555.7	354.7	323.4	503.2	200.3	118.4	110.0	131.4	0.36
ProtoRL	742.1	494.0	203.9	320.5	626.5	420.9	343.2	495.7	545.4	299.1	236.6	293.1	0.51
LBS	628.0	412.0	142.0	339.4	747.7	462.1	370.7	452.4	553.8	290.4	245.6	312.0	0.49
DIAYN	497.4	257.8	107.5	207.5	677.9	402.0	366.3	451.1	547.9	361.4	328.1	387.1	0.45
SMM	680.8	561.4	232.6	315.6	309.4	144.5	171.2	244.0	278.1	116.8	115.2	211.0	0.36
APS	663.7	481.6	138.0	291.9	605.7	464.0	285.9	502.7	388.2	199.5	246.5	329.4	0.46
CIC	859.5	607.8	235.9	312.4	763.7	601.6	**432.8**	**630.9**	224.3	139.8	112.1	179.4	0.52
BeCL	**874.2**	**693.4**	**255.9**	**354.5**	683.0	369.7	349.6	517.5	522.0	425.1	285.7	491.4	0.59
PEAC (Ours)	860.0	554.3	225.3	324.8	**776.3**	**741.7**	381.5	624.4	**734.6**	**641.9**	**385.7**	**537.0**	**0.68**

Table 16: **Detailed results in state-based DMC in evaluation embodiments**. Average cumulative reward (mean of 10 seeds) of the best policy trained by different algorithms.

Similarly, Table 17 shows that PEAC-LBS not only outperforms baselines but also owns a greater leading advantage than baselines, compared with the trained embodiments. Moreover, PEAC-DIAYN exceeds other pure-exploration methods and demonstrates strong generalization ability.

Domains	Walker-mass				Quadruped-mass				Quadruped-damping				Normilized
Tasks	stand	walk	run	flip	stand	walk	run	jump	stand	walk	run	jump	Average
DIAYN	793.5	537.7	198.1	370.5	565.8	380.5	333.2	365.3	748.8	401.7	365.1	499.2	0.54
APS	927.8	601.8	238.6	473.2	781.6	442.2	430.2	706.3	849.9	409.8	377.1	550.4	0.67
LSD	921.4	706.9	239.4	362.4	737.2	401.8	369.2	534.6	620.6	444.2	487.3	601.6	0.63
CIC	961.5	756.1	308.9	421.4	865.3	397.1	355.1	502.8	857.1	453.1	403.6	562.3	0.67
PEAC-DIAYN (Ours)	964.1	779.1	340.3	485.7	693.5	412.1	422.3	510.8	849.0	540.3	436.8	656.6	0.70
ICM	958.3	793.8	335.6	487.9	907.5	597.0	450.2	786.2	860.4	467.9	407.9	668.1	0.77
RND	963.6	825.5	360.7	506.8	843.8	483.1	429.3	743.6	841.1	449.2	407.3	714.7	0.76
Plan2Explore	967.8	862.4	366.2	517.8	906.0	648.6	487.5	653.2	837.2	550.9	419.8	671.1	0.78
APT	938.0	811.1	357.5	467.0	820.1	485.7	484.8	689.2	777.0	526.0	431.3	645.7	0.74
LBS	944.6	789.7	387.3	529.1	898.8	696.4	542.6	765.8	875.7	**770.4**	524.1	761.5	0.85
Choreographer	956.8	849.9	408.4	542.0	**921.1**	648.4	446.4	748.5	884.3	723.8	592.8	727.5	0.84
PEAC-LBS (Ours)	**967.1**	**902.1**	**444.9**	**695.6**	901.9	**750.6**	**598.8**	**799.9**	**897.2**	748.5	**659.4**	**798.7**	**0.92**

Table 17: **Detailed results in image-based DMC in evaluation embodiments**. Average cumulative reward (mean of 3 seeds) of the best policy trained by different algorithms.

B.11 More challenging tasks and varying embodiments

In this section, we will consider CEURL in much more challenging tasks and more varying embodiment distributions, which are significant future directions for unsupervised cross-embodiment agents in more challenging real-world applications.

We first consider more complicated tasks including locomotion in complicated terrain. Following previous work [16], we design locomotion tasks in incline terrains and the results are below.

Domains	Walker-mass-incline			
Task	stand	walk	run	flip
LBS	489.1	156.0	748.4	493.0
PEAC-LBS (Ours)	**557.9**	**245.0**	**748.8**	**681.7**

Table 18: **Detailed results of Walker-mass-incline in image-based DMC**.

As shown in Table 18, the performance of LBS and PEAC-LBS decreases when locomoting in the incline terrain due to its complexity. PEAC-LBS still significantly outperforms LBS, expressing that our method, especially the cross-embodiment intrinsic rewards, benefits cross-embodiment unsupervised pre-training for handling more complicated tasks.

Besides more complicated tasks, one possible future direction is to consider more different, or even exactly different embodiments. We take the first step by designing several settings with more varying and challenging embodiment distributions:

https://doi.org/10.52202/079017-1731

- Walker-Cheetah: includes two Walker robots with a mass of 0.4 and 1.6 times the normal mass, as well as two Cheetah robots with a mass of 0.4 and 1.6 times the normal mass.
- Walker-Humanoid: includes one Walker robot and one Humanoid robot. Their robot properties, robot shapes, and action spaces are all different.
- Walker-length and Cheetah-torsolength [71]: The former includes walker robots with different foot lengths while the second one includes cheetah robots with different torso lengths. Thus robots' properties and morphologies are different. The figures of these embodiments are in Fig. 10.

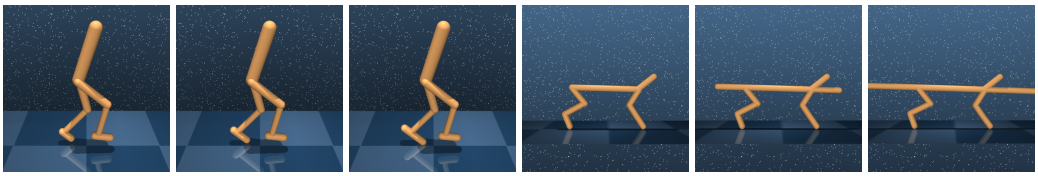


Figure 10: **Benchmark environments** of Walker-length and Cheetah-torsolength. In Walker-length, the **length of the left foot sole** of different robots is different. In Cheetah-torsolength, the **length of the torso** is different.

We mainly compare our PEAC-DIAYN and PEAC-LBS with DIAYN, LBS, and Choreographer in embodiment distributions: Walker-Cheetah, Walker-Humanoid, Walker-length, and Cheetah-torsolength. of which the results are in Table 19, Table 20, and Table 21 respectively.

Domains	Walker-Cheetah			
Task	Walker-stand & Cheetah-run	Walker-run & Cheetah-run	Walker-flip & Cheetah-run	Walker-flip & Cheetah-flip
DIAYN	414.3	246.2	346.6	448.3
PEAC-DIAYN (Ours)	632.6	297.5	442.2	527.5
LBS	604.8	311.7	401.0	646.2
Choreographer	**681.4**	374.2	446.9	624.4
PEAC-LBS (Ours)	**671.2**	**390.7**	**452.3**	**679.2**

Table 19: **Detailed results of Walker-Cheetah in image-based DMC**.

Domains	Walker-Humanoid								
Task	stand-stand	stand-walk	stand-run	walk-stand	walk-walk	walk-run	run-stand	run-walk	run-run
DIAYN	445.1	437.7	423.7	331.5	335.9	339.3	115.3	139.3	127.0
PEAC-DIAYN (Ours)	470.4	447.2	476.3	409.6	355.6	363.1	135.6	126.9	135.9
LBS	**478.9**	**485.2**	476.3	**463.6**	461.0	455.3	179.9	205.6	186.2
Choreographer	471.0	**479.9**	**483.7**	409.8	413.6	403.0	**216.4**	**233.1**	160.3
PEAC-LBS (Ours)	468.4	**480.3**	**482.3**	**460.8**	**470.5**	**466.1**	196.8	**234.4**	**242.8**

Table 20: **Detailed results of Walker-Humanoid in image-based DMC**.

Domains	Walker-length				Cheetah-torso_length			
Task	stand	walk	run	flip	run	run_backward	flip	flip_backward
DIAYN	748.5	764.0	328.7	532.3	721.1	**723.6**	634.2	502.4
PEAC-DIAYN (Ours)	**962.7**	**955.9**	**564.3**	**900.6**	695.4	664.5	689.2	507.8
LBS	**965.7**	**951.9**	525.7	863.4	718.4	685.4	680.6	499.5
Choreo	**961.4**	**958.5**	556.3	**918.0**	708.1	700.3	649.0	459.7
PEAC-LBS (Ours)	**966.1**	**956.6**	**573.4**	**899.3**	**731.8**	704.4	**747.5**	**515.4**

Table 21: **Detailed results of Walker-length and Cheetah-torso_length in image-based DMC**.

As shown in these tables, PEAC can achieve much greater performance compared to baselines. These experiments indicate that PEAC has powerful abilities to handle various kinds of embodiment differences, including different morphologies. Unfortunately, when the embodiments vary a lot (like Walker-Humanoid), the performance of PEAC is still limited, thus designing more effective methods for handling complicated embodiments like Humanoid is a promising future direction for further considering cross-embodiment settings.

B.12 Real-World Applications

As a supplement of Sec. 5.4, we provide more detailed images of real-world robot deployments. As shown in Fig. 11, our method can fast fine-tune to different embodiments and handle different terrains, which are unseen in the simulation. A detailed video is provided on the paper homepage.

Figure 11: Real-world results for Aliengo robot with different joint failure in different terrains.

B.13 Computing Resource

In experiments, all the agents are trained by GeForce RTX 2080 Ti with Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz. In Image-based DMC / state-based DMC / Robosuite / Isaacgym, pre-training each algorithm (each seed, domain) takes around 2 / 0.5 / 1.5 / 2 days respectively.

C Pseduo-codes of Algorithms

Algorithm 1 Pre-trained Embodiment-Aware Control (PEAC)

Require: M training embodiments $\{e_m\}_{m=1}^{M}$, M replay buffers $\{\mathcal{D}_m\}_{m=1}^{M}$, N testing embodiments $\{e_{M+n}\}_{n=1}^{N}$, initialize neural network parameters of the policy

// Pre-Training
while *is unsupervised phase* **do**
  // Data Collection
  for $m = 1, 2, ..., M$ **do**
    Sample state-action pairs $\{(s_t^m, a_t^m)\}_t$ with the policy by controlling the embodiment e_m and store them into $\mathcal{D}_m$.
  end for
  // Model Training
  for update step $= 1, 2, ..., U$ **do**
    Sample state-action pairs form each replay buffer $\{(s_t^i, a_t^i)_{t=1}^{T}\} \sim \mathcal{D}_i, i = 1, 2, ..., M$
    Update the embodiment discriminator via these data.
    Compute the cross-embodiment intrinsic reward $\mathcal{R}_{\text{CE}}$ for each state-action pair and concatenate them together.
    Update the policy by RL backbones (like PPO, DDPG, DreamerV2, and so on) with these data and $\mathcal{R}_{\text{CE}}$.
  end for
end while
// Fine-Tuning
while *is supervised phase* **do**
  Sample state-action-reward pairs with extrinsic rewards $\mathcal{R}_{\text{ext}}$ via embodiment e_m and store them into $\mathcal{D}_m$.
  Update the policy by jointly training data from different replay buffers via RL backbones.
end while
// Evaluation
Evaluate fine-tuned policy with downstream task $\mathcal{R}_{\text{ext}}$ via $\{e_m\}_{m=1}^{M}$ and unseen embodiments $\{e_{M+n}\}_{n=1}^{N}$.

https://doi.org/10.52202/079017-1731

Algorithm 2 PEAC-LBS

Require: M training embodiments $\{e_m\}_{m=1}^{M}$, M replay buffers $\{\mathcal{D}_m\}_{m=1}^{M}$, N testing embodiments $\{e_{M+n}\}_{n=1}^{N}$, initialize neural network parameters of the policy

// Pre-Training
while *is unsupervised phase* **do**
  // Data Collection (the same as PEAC)
  ...
  // Model Training
  for update step $= 1, 2, ..., U$ **do**
    Sample state-action pairs form each replay buffer $\{(s_t^i, a_t^i)_{t=1}^{T}\} \sim \mathcal{D}_i, i = 1, 2, ..., M$
    Update the embodiment discriminator via these data.
    Update the components of LBS, including the Latent Prior model, the Latent Posterior model, and the Reconstruction model (In DreamerV2 backbone, we can directly utilize its prior model and posterior model).
    Compute the intrinsic reward $\mathcal{R}_{\mathrm{CE}} + \mathcal{R}_{\mathrm{LBS}}$ for each state-action pair and concatenate them together.
    Update the policy by RL backbones (like PPO, DDPG, DreamerV2, and so on) with these data and $\mathcal{R}_{\mathrm{CE}} + \mathcal{R}_{\mathrm{LBS}}$.
  end for
end while
// Fine-Tuning(the same as PEAC)
...
// Evaluation
Evaluate fine-tuned policy with downstream task $\mathcal{R}_{\mathrm{ext}}$ via $\{e_m\}_{m=1}^{M}$ and unseen embodiments $\{e_{M+n}\}_{n=1}^{N}$.

Algorithm 3 PEAC-DIAYN

Require: M training embodiments $\{e_m\}_{m=1}^{M}$, M replay buffers $\{\mathcal{D}_m\}_{m=1}^{M}$, N testing embodiments $\{e_{M+n}\}_{n=1}^{N}$, initialize neural network parameters of the behavior policy conditioned on skill and embodiment context $\pi(\cdot|s, z, e)$, initialize neural network parameters of the embodiment-aware skill policy $\pi(z|e, \tau)$

// Pre-Training
while *is unsupervised phase* **do**
  // Data Collection (the same as PEAC)
  ...
  // Model Training
  for update step $= 1, 2, ..., U$ **do**
    Sample state-action pairs form each replay buffer $\{(s_t^i, a_t^i)_{t=1}^{T}\} \sim \mathcal{D}_i, i = 1, 2, ..., M$
    Update the embodiment discriminator via these data.
    Update the skill discriminator of DIAYN via these data.
    Compute the intrinsic reward $\mathcal{R}_{\mathrm{CE}} + \mathcal{R}_{\mathrm{DIAYN}}$ for each state-action pair and concatenate them together.
    Update the behavior policy conditioned on skill and embodiment context by RL backbones (like PPO, DDPG, DreamerV2, and so on) with these data and $\mathcal{R}_{\mathrm{CE}} + \mathcal{R}_{\mathrm{DIAYN}}$.
  end for
end while
// Fine-Tuning
while *is supervised phase* **do**
  Sample state-action-reward pairs with extrinsic rewards $\mathcal{R}_{\mathrm{ext}}$ via embodiment e_m and store them into $\mathcal{D}_m$.
  Update the embodiment-aware skill policy by jointly training data from different replay buffers via RL backbones.
end while
// Evaluation
Evaluate fine-tuned agents with downstream task $\mathcal{R}_{\mathrm{ext}}$ via $\{e_m\}_{m=1}^{M}$ and unseen embodiments $\{e_{M+n}\}_{n=1}^{N}$.

D Broader Impact

Designing generalizable agents for varying tasks and embodiments is a major concern in reinforcement learning. This work focuses on cross-embodiment unsupervised reinforcement learning and proposes a novel algorithm PEAC, which leverages trajectories from different embodiments for pre-training, subsequently broadly enhancing performance on downstream tasks. Such advancements provide the potential for future real-world cross-embodiment control. One of the potential negative impacts is that algorithms using deep neural networks, which lack interoperability and face security and robustness issues. There are no serious ethical issues as this is basic research.

https://doi.org/10.52202/079017-1731

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As mentioned in the abstract and introduction, this work mainly integrates the unsupervised RL paradigm into cross-embodiment RL, as a novel concept CEURL (Sec. 3). Then we theoretically derive a novel algorithm PEAC (Sec. 3-4) and widely evaluate PEAC in a large number of environments (Sec. 5). All these contributions are described in detail in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, we have discussed the limitations of this work in Sec. 5.5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Yes, we have included the detailed proofs of all results in the Appendix A, including properties of $\mathcal{D}^{\mathcal{E}}$ (Appendix A.1), proof of Theorem 3.2 (Appendix A.2), and proof about Embodiment-Aware Skill Discovery (Appendix A.3).

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided the code in the paper homepage. Also, we have provided detailed information on our experiments, including hyper-parameters, codebase, and pseudo-code of our algorithms in Appendix B and Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

https://doi.org/10.52202/079017-1731

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided the source code in our paper homepage.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided details for our experiments in Appendix B, including our environments, hyper-parameters, optimizers, and so on, as well as our source code in the paper homepage.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes] ,

   Justification: To mitigate the effectiveness of the randomness, we repeat several random seeds for all experiments (10 for state-based DMC and 3 for the others, following previous works' settings). When reporting the results, besides reporting mean$\pm$std performance, we

also report IQM and OG in state-based DMC and image-based DMC for better evaluating different algorithms.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we have reported the computing source in Appendix B.13, including the type of computing CPU/GPU, computing time, and so on.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research does not involve human subjects or participants. And we have discussed the potential societal impact in Appendix. D. As this is basic research, there are no serious social issues.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

https://doi.org/10.52202/079017-1731

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we have discussed the Border Impact of this work in Appendix. D.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no such risk in this paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code/environment package and included the URL of these codebases in Appendix B.2.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we have provided documentation of our source code in the supplementary materials.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

https://doi.org/10.52202/079017-1731

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.