Cross-Domain Knowledge Transfer for RL via Preference Consistency

Ting-Hsuan Huang [1] **Ping-Chun Hsieh** [1]

Abstract

We study the cross-domain RL (CDRL) problem from the perspective of preference-based learning. We identify the critical correspondence identifiability issue (CII) in the existing unsupervised CDRL methods and propose to mitigate CII with the weak supervision of preference feedback. Specifically, we propose the principle of cross-domain preference consistency (CDPC), which can serve as additional guidance for learning a proper correspondence between the source and target domains. To substantiate the principle of CDPC, we present an algorithm that integrates a state decoder learned by the preference consistency loss during training and a cross-domain MPC method for action selection during inference. Through extensive experiments in both MuJoCo and Robosuite, we demonstrate that CDPC can achieve effective and data-efficient knowledge transfer across domains than the state-of-the-art CDRL benchmark methods.

1. Introduction

Reinforcement Learning (RL) has shown impressive success on a wide range of tasks, encompassing both discrete and continuous control scenarios, such as game playing (Mnih et al., 2015; Silver et al., 2016; Vinyals et al., 2019) and robot control (Levine et al., 2016; Tobin et al., 2017). However, solving these tasks in a data-efficient manner has remained a significant challenge in RL, mainly due to the need for extensive online trial-and-error interactions and the resulting prolonged training periods. To alleviate the data efficiency issue, one natural and promising approach is to reuse the control policies learned on similar tasks for fast knowledge transfer. Built on this intuition, cross-domain reinforcement learning (CDRL) offers a generic formulation that extends the applicability of transfer learning to RL, where the source domain and the target domain can have different transition dynamics as well as distinct state-action representations. With access to the source domain (either the data samples or the environment) and the pre-trained source-domain models (either policies or value functions), CDRL aims to transfer the knowledge acquired from the source domain to the target domain in a data-efficient manner. This adaptability of CDRL is crucial for overcoming the data inefficiency in conventional RL, offering a more flexible and resource-efficient solution.

Several recent attempts on CDRL (Zhang et al., 2021a; Gui et al., 2023) have demonstrated the possibility of direct policy transfer by learning the state-action correspondence between domains, or essentially a mapping function, from unpaired trajectories in a fully unsupervised manner (i.e., no reward signal available in the target domain). For example, (Gui et al., 2023) proposes to learn the state-action correspondence (i.e., a target-to-source state decoder and a source-to-target action encoder) by minimizing a dynamic cycle consistency loss, which is meant to align the one-step transition of the unpaired trajectories from the two domains. These unsupervised approaches can serve as powerful RL solutions in practice as it is widely known that the reward design can require substantial efforts and hence be rather time-consuming. Despite the progress, we identify that this unsupervised approach can be prone to the *correspondence identifiability issue* (CII). To illustrate this, we provide a toy example of a gridworld as shown in Figure 1. This phenomenon indicates that without any supervision from the target domain, learning the state-action correspondence can be an underdetermined problem. As a result, there is one important research question to be answered: *How to address the correspondence identifiability issue in cross-domain transfer for RL with only weak supervision?*

In this paper, we answer the above question from the perspective of *cross-domain preference-based RL* (CD-PbRL). Specifically, we present a new CDRL setting where the agent in the target domain can receive additional weak supervision signal in the form of *preferences over trajectory pairs*. The primary motivation for proposing the method is the complexity involved in designing a reward function for most RL environments. Particularly, the target domain is often unknown and more complex compared to the source domain. Therefore, we aim to achieve transfer learning without re-

[1] Department of Computer Science, National Yang Ming Chiao Tung University. Correspondence to: Ting-Hsuan Huang <gyxuan0527.11@nycu.edu.tw >, Ping-Chun Hsieh <pinghsieh@nycu.edu.tw>.

ICML 2024 Workshop on Models of Human Feedback for AI Alignment.

lying on reward information. Inspired by a series of papers on large language models (LLMs) (Memarian et al., 2021; Liu et al., 2023; Chakraborty et al., 2023; Sun et al., 2023b) and existing preference-based RL (PbRL) methods (Wirth et al., 2017; Busa-Fekete & Hüllermeier, 2014; Kamishima et al., 2010; Wirth & Fürnkranz), we believe that human preference can be used to solve RL problems. Furthermore, by maintaining the consistency of preferences across the two domains, we aim to address the CDRL problem. Accordingly, we present a systematic approach that effectively leverages this preference signal to tackle the identifiability issue. Specifically, we propose the framework of *Cross-Domain Preference Consistency* (CDPC), which can better learn the state-action correspondence by enforcing the trajectory preferences to be consistent across the two domains. The proposed CDPC framework consists of two major components: (i) *Target-to-source state decoder*: To enable the reuse of a source-domain pre-trained policy (denoted by π_{src}), CDPC learns a target-to-source state decoder (denoted by ϕ^{-1}). To learn ϕ^{-1} without suffering from CII, CDPC utilizes a cross-domain pairwise preference loss (or equivalently the negative log-likelihood), which is calculated with respect to the source-domain trajectories induced by ϕ^{-1} with the target-domain preferences as our labels. Compared to the existing unsupervised CDRL, this loss function offers additional constraints for the state decoder such that the identifiability issue can be mitigated. (ii) *Cross-domain model predictive control for inference*: During inference, we propose to leverage the learned state decoder and determine the target-domain actions by *planning* via model-predictive control (MPC). Specifically, at each time step, we generate multiple synthetic target-domain trajectories of finite length (with the help of a learned dynamics model) and choose the first action of the best trajectory. Different from the standard MPC, the proposed cross-domain MPC uses the *source-domain reward* of the source-domain trajectory induced by the state decoder as the selection criterion for MPC. With this design, there is no need to learn the action correspondence between source and target domains. Moreover, this framework is general in the sense that it can be integrated with any enhancements of MPC.

The main contributions can be summarized as follows:

- We identify the correspondence identifiability issue of cross-domain RL and the need for including weak supervision in cross-domain knowledge transfer. To address this issue, we propose a formulation termed cross-domain preference-based RL, where the preferences over trajectories are available as an additional weak supervision signal.
- To solve CD-PbRL, we propose a generic framework based on the concept of cross-domain preference consistency or CDPC. To substantiate CDPC, we learn a target-to-source state decoder by using a pairwise ranking loss during training and determine the target actions by cross-domain MPC during inference.
- Through extensive experiments in both MuJoCo locomotion tasks and the robot arm manipulation tasks in Robosuite, we demonstrate that CDPC can achieve more effective knowledge transfer across domains than the state-of-the-art CDRL benchmark methods. Additionally, we provided an ablation study to verify the importance of preference consistency.

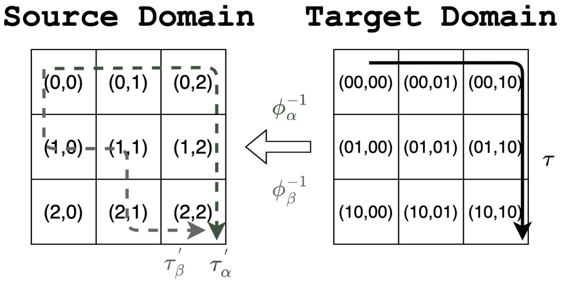


Figure 1. **An illustrative example of the correspondence identifiability issue:** Let the source domain and the target domain be a 3×3 gridworld (with the goal state at the bottom-right corner), represented in decimal numbers and binary numbers, respectively. The two domains share the same action representation. Let (ϕ_α^{-1}) and (ϕ_β^{-1}) be two candidate state decoders that map the trajectory τ into the trajectory τ'_α and τ'_β, respectively. One can verify that both decoders achieve zero dynamics cycle consistency loss. However, it is difficult to distinguish whether τ'_α or τ'_β is better in terms of goal-reaching, indicating an identifiability issue when training the decoder solely based on dynamic cycle consistency loss. The detailed explanation is provided in Appendix A.

2. Related Work

Cross-Domain Knowledge Transfer in RL. Cross-domain transfer in RL (Taylor & Stone, 2009; Zhu et al., 2023; Serrano et al., 2024) is an area of research within reinforcement learning (RL) that specifically addresses the challenge of transferring learned policies or value functions from one domain to another, even when there are disparities in state-action dimensions between the domains. Cross-domain transfer learning can be divided into imitation learning (Kim et al., 2020; Fickinger et al., 2021; Raychaudhuri et al., 2021) and transfer learning. Transfer learning itself can be further categorized into single-source transfer (Ammar & Taylor, 2012) and multiple-source transfer (Ammar et al.; Qian et al., 2020; Talvitie & Singh, 2007; Serrano et al., 2021). From the perspective of what is being transferred, which means the known information, it can be generally divided into demonstrations (Ammar et al., 2015; Shankar et al., 2022; Watahiki et al., 2023), policy (Wang et al., 2022; Yang et al., 2023; Gui et al., 2023; Chen et al., 2024), parameters (Devin et al., 2017; Zhang et al., 2021b), and value function (Torrey et al., 2008; Taylor et al., 2008).

Common practices to solve CDRL under different state and action representations include leveraging cycle consistency and transition between states and actions across two domains to discover mapping functions (Zhang et al., 2021a; You et al., 2022; Li et al., 2022; Wu et al., 2022; Raychaudhuri et al., 2021; Gui et al., 2023), or employing adversarial training techniques to identify mapping relationships between states and actions in the source and target domains (Gui et al., 2023; Li et al., 2022; Wulfmeier et al., 2017; Mounsif et al., 2020; Raychaudhuri et al., 2021; Watahiki et al., 2022).

Preference-based RL (PbRL). PbRL (Wirth et al., 2017; Busa-Fekete & Hüllermeier, 2014; Kamishima et al., 2010; Wirth & Fürnkranz) is a popular RL setting that focuses on learning policies or value functions from preferences rather than explicit reward signals. One common approach is to model the preference feedback as a binary classification problem (Lee et al.; 2021; Akrour et al., 2011; Pilarski et al., 2011; Akrour et al., 2012; Wilson et al., 2012; Ibarz et al., 2018). PBRL has been applied to various real-world domains, including personalized recommendation systems (Li et al., 2010), interactive learning from human feedback (Knox & Stone, 2009), and robot learning from human preferences (Warnell et al., 2018). Besides, PBRL can also be employed for automatic summarization of articles (Stiennon et al., 2020). This approach enables the model to acquire sophisticated summarization techniques through preference-based learning (Stiennon et al., 2020; Ouyang et al., 2022; Achiam et al., 2023; Lee et al., 2023; Kirk et al., 2023; Sun et al., 2023a). Beyond its application in large language models, preference-based techniques are also commonly utilized in training RL agents (Memarian et al., 2021; Liu et al., 2023; Chakraborty et al., 2023; Sun et al., 2023b). By leveraging human feedback to train reward functions, these techniques enable RL agents to approximate real-world rewards more accurately, guiding the agents towards convergence to an optimal policy.

3. Preliminaries

In this section, we first describe the standard problem formulation of preference-based RL and proceed to present the proposed extended version of preference-based RL for cross-domain transfer. Throughout this paper, for any set $\mathcal{X}$, we use $\Delta(\mathcal{X})$ to denote the set of all probability distributions over $\mathcal{X}$. In this work, as in typical RL, we model each domain as a Markov decision process (MDP) denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \mu, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(S)$ is the transition kernel that maps each state-action pair to a probability distribution over the next state, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function, $\mu \in \Delta\mathcal{S}$ is the initial state distribution, and $\gamma \in (0, 1]$ is the discount factor. Let $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ denote the policy of the RL agent and let $\tau = (s_0, a_0, r_1, \cdots)$ denote a trajectory generated under π in the domain $\mathcal{M}$. Given a trajectory τ, we slightly abuse the notation and use $R(\tau)$ to denote the total expected reward accrued along τ, i.e., $R(\tau) := \sum_{t=0}^{\infty} R(s_t, a_t)$. Let Π denote the set of all stationary Markov policies. We define the expected total discounted reward under π as $V^{\pi}_{\mathcal{M}}(\mu) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)|s_0 \sim \mu, \pi]$. Let $\pi^*_{\mathcal{M}} := \arg\max_{\pi\in\Pi} V^{\pi}_{\mathcal{M}}(\mu)$ be an optimal policy for $\mathcal{M}$ in that it maximizes the expected total discounted reward.

3.1. Problem Formulation of Preference-based RL

In the standard PbRL, the environment is modeled as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \mu, \gamma)$ as usual. Moreover, the goal of PbRL remains the same as the standard reward-based RL, i.e., finding an optimal policy $\pi^*_{\mathcal{M}}$ that maximizes $V^{\pi}_{\mathcal{M}}(\mu)$. Despite the existence of an underlying true reward function (so that the RL objective function is well-defined), in the PbRL setting, the reward function R is hidden and not observable to the learner during training. Nevertheless, given two trajectories τ and τ', the learner can receive the (possibly randomized) preference over τ and τ', which is determined by the total expected reward $R(\tau)$ and $R(\tau')$ along the trajectories. For notional convenience, we use $\tau \succ \tau'$ (or an equivalent expression $\tau' \prec \tau$) to denote the event that τ is preferred over τ'. Note that a probability preference model $\mathcal{P}(\tau, \tau'; R)$ is typically needed to specify the likelihood of the event $\tau \succ \tau'$. For example, under the celebrated Bradley-Terry model (Bradley & Terry, 1952), we have $\mathcal{P}(\tau, \tau'; R) := 1/(1 + \exp(R(\tau') - R(\tau)))$. We assume that under the preference model, for any pair of trajectories τ, τ', either the event $\tau \succ \tau'$ or $\tau' \succ \tau$ would happen at each time.

To solve PbRL, one popular way is to adopt a two-stage approach, where we first learn the underlying true reward function from the preference feedback and then apply an off-the-shelf RL algorithm for policy learning. Under a preference model $\mathcal{P}(\tau, \tau'; R)$, a reward model $\hat{R}$ can be learned by maximizing the log-likelihood, i.e., given a dataset of trajectories $\mathcal{D}$,

$$\hat{R} = \arg\max_{R':\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}} \mathbb{E}_{\tau,\tau' \in \mathcal{D}, \tau \succ \tau'} \left[\log \mathcal{P}(\tau, \tau'; R')\right]. \tag{1}$$

This approach has been widely used in the fine-tuning of large language models with RLHF (Ouyang et al., 2022).

3.2. Problem Formulation of Cross-Domain Preference-based RL

In this section, we formally present the proposed CDPbRL problem. Specifically, we extend the standard (unsupervised) CDRL problem, which aims to achieve knowledge transfer from a source domain to another target domain, to the scenario where the preferences over trajectories are available as weak supervision in the target domain. The source and target domains are modeled as follows:

- **Source domain:** As usual, we model the source domain as an MDP $\mathcal{M}_{\text{src}} = (\mathcal{S}_{\text{src}}, \mathcal{A}_{\text{src}}, \mathcal{T}_{\text{src}}, R_{\text{src}}, \mu_{\text{src}}, \gamma)$[1]. For efficient knowledge transfer, the source-domain is typically an environment that is cheap and easy to access, e.g., a simulator. Accordingly, we presume that the learner has full access to the source-domain environment and hence can collect data samples and obtain a pre-trained source-domain policy π_{src}. This setting has been adopted by most of the existing CDRL literature (Xu et al., 2023).

- **Target domain:** Similar to the source domain, the target domain is modeled as an MDP $\mathcal{M}_{\text{tar}} = (\mathcal{S}_{\text{tar}}, \mathcal{A}_{\text{tar}}, \mathcal{T}_{\text{tar}}, R_{\text{tar}}, \mu_{\text{tar}}, \gamma)$. Notably, the target-domain MDP can differ from source-domain MDP in both transition dynamics and the state-action representations. Here we only assume that the two domains share the same discount factor, which is a fairly mild condition. In the standard unsupervised CDRL setting (Zhang et al., 2021a; Gui et al., 2023), the learner is given a set of target-domain trajectories $\mathcal{D}_{\text{tar}} = \{\tau_i\}_{i=1}^{D}$ collected under some behavior policy. Moreover, due to the unsupervised setting, the reward function R_{tar} is assumed to be unobservable to the learner, and hence $\mathcal{D}_{\text{tar}}$ only contains information about the visited state-action pairs. Notably, this formulation can suffer from the identificability issue by nature as described in Section 1. By contrast, built on the CDRL, the CD-PbRL formulation additionally includes that the learner can further receive preference information about pairs of trajectories in the target domain, despite the unknown true rewards. The goal of CD-PbRL is again to find an optimal policy $\pi^*_{\mathcal{M}_{\text{tar}}} := \arg\max_{\pi\in\Pi_{\text{tar}}} V^{\pi}_{\mathcal{M}_{\text{tar}}}(\mu_{\text{tar}})$ for the target domain.

4. Methodology

In this section, we formally present the proposed algorithm for the CD-PbRL problem. We start by describing the proposed CDPC principle and thereafter provide the implementation of the training and inference procedure of the resulting CDPC algorithm.

4.1. Cross-Domain Preference Consistency

To mitigate the correspondence identifiability issue, we propose to constrain the learning of state correspondence by *preference consistency*, which is meant to ensure that the preference ordering of the corresponding trajectories in the two domains remains consistent. An illustration of the CDPC principle is provided in Figure 2. To better motivate this, we can think of an analogy in language modeling: We can interpret τ_i and τ_j as two sentences written in German. The state decoder acts like a translator, converting a German sentence into one in English. If τ_i is more aligned with natural human language in German than τ_j, then after translation by the decoder, τ_i' is expected to be also more natural and fluent than τ_j' in English expression. The above characteristic can be used as an additional requirement to identify the inter-domain state correspondence.

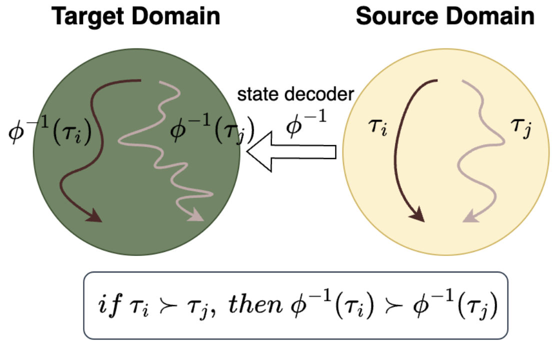


Figure 2. **The principle of cross-domain preference consistency**: Let τ_i and τ_j be two target-domain trajectories. If τ_i is preferred over τ_j, which means it has a higher total return, then the trajectories transformed through a state decoder ϕ^{-1} shall maintain the same preference, i.e., $\phi^{-1}(\tau_i)$ shall be preferred over $\phi^{-1}(\tau_j)$.

Based on the concept of CDPC, here we provide an overview of the proposed algorithm, which consists of the following two major building blocks:

- **(Training)** ***Learning a target-to-source state decoder by preference consistency***: As in typical CDRL methods, our CDPC framework also learns a state decoder $\phi^{-1} : \mathcal{S}_{\text{tar}} \to \mathcal{S}_{\text{src}}$ such that actions taken in $\mathcal{M}_{\text{tar}}$ can be determined through knowledge transfer from a source-domain policy. Recall from Section 1 that fully unsupervised CDRL methods, where the state decoder is learned solely based on dynamics alignment (Gui et al., 2023) or reconstruction (Zhang et al., 2021a), can suffer from the identifiability issue. As a result, we propose to learn the state decoder based on the CDPC principle, which serves as an additional criterion for learning the state correspondence across domains. Specifically, to learn the state decoder[2] $\phi_\theta^{-1} : \mathcal{S}_{\text{tar}} \to \mathcal{S}_{\text{src}}$ (parameterized by θ), we construct a cross-domain loss function based on the pairwise ranking idea in PbRL as follows:

$$\mathcal{L}_{\text{pref}}(\theta) := \mathbb{E}_{\tau_i,\tau_j\sim\mathcal{D}_{\text{tar}}}\left[\log\left(1+e^{R_{\text{src}}\left(\phi_\theta^{-1}(\tau_j)\right)-R_{\text{src}}\left(\phi_\theta^{-1}(\tau_i)\right)}\right)\right]. \tag{2}$$

[1] Throughout this paper, we use the subscripts "src" and "tar" to denote the objects of the source and the target domain, respectively.

[2] Here we use the term "decoder" as this mapping function is typically learned based on an autoencoder network architecture.

The preference loss function in (2) resembles (1) of PbRL but with one major difference: the preference consistency is captured through the state decoder ϕ_θ^{-1}. This preference loss function can be used in conjunction with any other off-the-shelf loss function for unsupervised CDRL, such as dynamics cycle consistency or reconstruction loss (Zhang et al., 2021a). More implementation details of the state decoder are described in Section 4.2.

- **(Inference)** ***Selecting target-domain actions by MPC in target domain with cross-domain trajectory optimization***: With a properly learned state decoder, the next step is to transfer the pre-trained source-domain policy π_{src} to the target domain. Notably, one naive approach is to simply learn an additional action encoder $\psi : \mathcal{A}_{\text{src}} \rightarrow \mathcal{A}_{\text{tar}}$ (e.g., similarly by preference consistency) such that given any state $s \in \mathcal{S}_{\text{tar}}$, a target-domain action can be induced by $\psi(a_{\text{src}})$ with $a_{\text{src}} \sim \pi_{\text{src}}(\phi^{-1}(s))$, as also adopted by (Gui et al., 2023). However, this approach can suffer from inaccurate preference correspondence. The details about this naive approach are provided in Appendix B.

  To better leverage the CDPC principle in selecting actions in the target domain, we propose to enforce knowledge transfer from the perspective of *planning*. Specifically, we use MPC in the target domain with the help of *cross-domain trajectory optimization* (CDTO). The details implementation is provided in Section 4.3.

4.2. Training Phase of CDPC: Learning a State Decoder

In the CD-PbRL setting, a well-trained state decoder ϕ_θ^{-1} should satisfy the following characteristics: (i) ϕ_θ^{-1} shall be able to ensure preference consistency between trajectories and (ii) meet the original cycle consistency conditions in both state construction and dynamics alignment. To learn the state decoder, we use the preference consistency loss as described in Section 4.1 as well as the dynamics cycle consistency loss and reconstruction loss.

- **Dynamics cycle consistency loss:** One common principle of learning state-action correspondence is through dynamics alignment, i.e., the next state obtained by the state decoder shall be consistent with that generated under the source-domain transition dynamics. Specifically, in this work, we use the following loss function to capture dynamics cycle consistency:

$$\mathcal{L}_{\text{dcc}}(\theta) = \mathbb{E}\left[\left\|\mathcal{T}_{\text{src}}\left(\phi_\theta^{-1}(s), a\right) - \phi_\theta^{-1}(s')\right\|^2\right], \quad (3)$$

  where the expectation is over the randomness of $s, s' \sim \mathcal{D}_{\text{tar}}$ and $a \sim \pi_{\text{src}}(\cdot|\phi^{-1}(s))$.

- **Reconstruction loss:** Additionally, the reconstruction loss (Zhang et al., 2021a; Gui et al., 2023; Zhu et al., 2017) is widely used in cross-domain tasks for its several advantages: (i) It acts as a regularization term, encouraging the decoder to produce outputs closely resembling the input data. This enhances reconstruction quality and generalization across domains. (ii) The loss fosters model stability by promoting consistency between input and reconstructed outputs, even in the presence of noise or domain variations. Minimizing the reconstruction loss leads to a more compact and meaningful data representation, facilitating better transfer learning and generalization capabilities. The reconstruction loss is defined as

$$\mathcal{L}_{\text{rec}}(\theta) := \mathbb{E}\left[\left\|\phi_\omega\left(\phi_\theta^{-1}(s)\right) - s\right\|^2\right], \quad (4)$$

  where the expectation is over the randomness of the state s drawn from the target-domain dataset $\mathcal{D}_{\text{tar}}$. Note that we presume the use of an autoencoder, where ϕ and ω represent the parameters of the state decoder and encoder, respectively. As we only need the decoder for inference, we ignore the dependency of $\mathcal{L}_{\text{rec}}(\theta)$ on ω in (4) for brevity.

In summary, the total loss of the state decoder can be expressed as follows:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{pref}}(\theta) + \beta_1 \mathcal{L}_{\text{dcc}}(\theta) + \beta_2 \mathcal{L}_{\text{rec}}(\theta), \quad (5)$$

where $\beta_1 > 0$ and $\beta_2 > 0$ are the weights for balancing the three loss terms. The overall pseudocode are provided in Algorithm 1.

4.3. Inference Phase of CDPC: Cross-Domain MPC

During the inference phase, given a well-trained state decoder, we propose to determine target-domain actions through planning via cross-domain MPC, which consists of two major components:

- **Cross-domain trajectory optimization (CDTO)**: As in typical MPC, at each time step t, based on the current observation s_t, we determine the action a_t by (i) generating multiple synthetic trajectories of length h with s_t as the starting state (denoted by $\mathcal{D}^{(t)}$) in the target domain, and then (ii) selecting one trajectory τ from $\mathcal{D}^{(t)}$ based on some performance metric, and (iii) choosing the first action of τ as the action a_t. Notably, to implement (ii), we propose to use the source-domain reward of the source-domain trajectory induced by the state decoder as the selection criterion for MPC.

- **Generation of synthetic trajectories for cross-domain MPC**: To implement the subroutine (i) in CDTO, we also learn two helper models based on the target-domain dataset $\mathcal{D}_{\text{tar}}$, namely a target-domain dynamics model (learned in a standard way by minimizing squared errors of next-state prediction) and a target-domain policy by

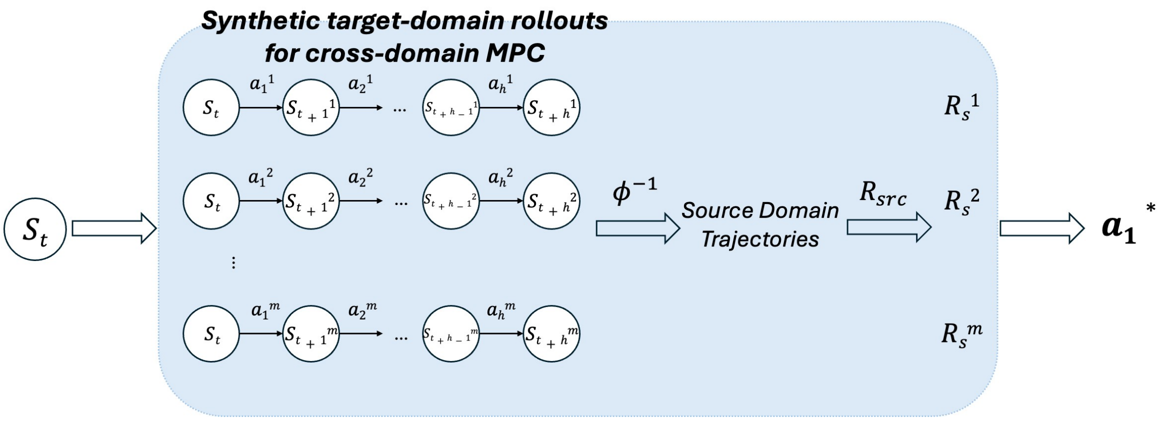

Synthetic target-domain rollouts for cross-domain MPC
S_t
a_1^1 a_2^1 a_h^1
S_{t+1}^1 … S_{t+h-1}^1 S_{t+h}^1
a_1^2 a_2^2 a_h^2
S_{t+1}^2 … S_{t+h-1}^2 S_{t+h}^2
a_1^m a_2^m a_h^m
S_{t+1}^m … S_{t+h-1}^m S_{t+h}^m
ϕ^{-1}
Source Domain Trajectories
R_{src}
R_s^1
R_s^2
R_s^m
a_1^*

Figure 3. **An illustration of cross-domain MPC:** During inference, based on the current state s_t, we generate m synthetic trajectories of length h by using a learned target-domain dynamics model and utilizing a behavior-cloned policy π_ι from $\mathcal{D}_{\text{tar}}$. These m trajectories are then mapped into the corresponding source trajectories using the trained state decoder ϕ_θ^{-1}. We compute the total return for each trajectory separately using the source-domain reward function (available in the cross-domain RL setting). Finally, the first action a_1^* from the sequence with the highest total return is adopted.

behavior cloning. This can be viewed as a variant of the random shooting technique in the model-based RL literature (e.g., (Nagabandi et al., 2018; 2020)) but with a behavior-cloned policy.

The cross-domain MPC approach is illustrated in Figure 3.

Remark 4.1. Note that here we choose the most basic variant of MPC during inference mainly to show the effectiveness of CDPC framework. The proposed framework can be readily enhanced and integrated with more sophisticated MPC methods, such as the popular cross-entropy method (Botev et al., 2013) and the filtering and reward-weighted refinement (Nagabandi et al., 2020).

The overall pseudocode are provided in Algorithm 2.

4.4. Algorithm

Algorithm 1 Cross-Domain Preference Consistency (CDPC)

```
Require:
  target domain trajectory training data buffer $\mathcal{D}_{tar}$
for each $episode\ k$ do
    // Training
    $\tau_i, \tau_j \sim \mathcal{D}_{tar}$
    Query human for preference
    Update state decoder $\phi_\theta^{-1}$ using $\mathcal{L}_{\text{total}}(\theta)$ (Equation 5)
    // Validation
    for each $timestep\ t = 1..T$ do
        $s_t \leftarrow$ current state in target domain environment
        Select optimal action $a_t$ using Algo. 2
        Take a step with $a_t$ in the environment
    end for
end for
```

5. Experimental Results

5.1. Experimental Configuration

Environment Domains. We utilize MuJoCo and Robosuite to simulate robot locomotion and manipulation, respectively. While MuJoCo and Robosuite already have pre-configured reward functions, given the CD-PbRL problem setting, we will not utilize them during training; they will only serve as performance metrics for evaluation.

- **MuJoCo.** We consider three MuJoCo tasks, namely Reacher, HalfCheetah, and Walker. Regarding the cross-domain setting, we use the original MuJoCo environments as the source domains and consider robots of more complex morphologies (and hence with higher state and action dimensionalities) as the target domains, The detailed description about the source domain and target domain can be found in Table 4 and Figure 9 in Appendix C.
- **Robosuite.** We set the source domain and target domain as two structurally different robot arms, namely Panda and IIWA, which have distinct state-action representations. We let the two types of robot arms perform the same set of tasks, including Lift, Door, and Assembly. The detailed description of the source domain and target domain can be found in Table 5 and Figure 10. All of the detailed information about the environments is provided in Appendix C.

Benchmark Methods. We compare CDPC with multi-

Algorithm 2 Cross-Domain Trajectory Optimization (CDTO)

```
Require:
  state $s_t$, state decoder $\phi^{-1}$
Ensure:
  action $a_t$
Initialize $\mathcal{D}^{(t)} \leftarrow \emptyset$
Generate synthetic trajectories $\tau_{1:m}$ using policy network $\pi_\iota(s)$ and dynamic model $F_\gamma(s, a)$
$\mathcal{D}^{(t)} \leftarrow \mathcal{D}^{(t)} \cup \{\tau_1, \tau_2, ..., \tau_m\}$
Decode $\tau_{1:m}$ using state decoder $\phi_\theta^{-1}$
Compute $R_s^{1:m}$ using source domain reward function $R_{src}$
Sort $\tau_{1:m}$ by $R_s^{1:m}$ in descending order
$\tau^* \leftarrow \mathcal{D}^{(t)}[0]$
$a^* \leftarrow$ first action of $\tau^*$
return $a^*$
```

ple benchmark algorithms, including: (i) **SAC$_T$:** Training an SAC agent directly in the target domain using the true environmental rewards; (ii) **SAC$_R$:** Learning a reward model using the PbRL method as in (1) and then training an SAC agent in the target domain using the learned reward model; (iii) **Dynamics Cycle-Consistency (DCC):** DCC is an unsupervised CDRL method that learns state-action correspondence by cycle consistency in dynamics and reconstruction. (iv) **Cross-Morphology Domain policy adaptation (CMD):** CMD is a more recent CDRL method proposed by (Gui et al., 2023) specifically for transfer in cross-morphology problems.

Dataset. As described in the problem formulation of CD-PbRL, a target-domain dataset $\mathcal{D}_{\text{tar}}$ is provided to the learner. To implement this, we follow the data collection method of D4RL (Fu et al., 2020). Specifically, we mix the expert demonstrations (by an expert policy learned under SAC (Haarnoja et al., 2018)) and sub-optimal data generated by unrolling a uniform-at-random policy. For a fair comparison, this dataset is shared by all the algorithms in the experiments. The comparison of experiments with varying mixing proportions is included in our ablation study.

5.2. Results and Discussions

Q: Does CDPC achieve effective cross-domain transfer? The results of final total rewards are shown in Figure 4, indicating that CDPC converges faster and performs better than the baselines. Table 1 shows the results for the target environments. Notably, CDPC can achieve higher total rewards than all the benchmark methods, even than SAC with true reward signals.

Q: Does CDPC achieve data-efficient cross-domain transfer in RL? As shown in Figure 4, compared to the other methods, CDPC already performs better during the initial training phase because it well utilizes the source-domain knowledge. CDPC achieves a good total return with only a small number of training iterations, addressing the important data inefficiency issue in RL. The reason why DCC and CMD perform relatively poorly is that they suffer from the identifiability issue as they only focus on learning the state-action correspondence between two domains. SAC$_R$, on the other hand, needs to first learn a reward model, and if the reward model is inaccurate, it greatly impacts the results. SAC$_T$ converges more slowly as it does not involve any knowledge transfer from the source domain.

Q: Does CDPC learn a state decoder that can effectively achieve cross-domain preference consistency? We provide an ablation study and investigate the significance of the preference consistency loss. The results showed that the preference consistency loss has a highly significant effect. Without using $\mathcal{L}_{\text{pref}}(\theta)$, the decoder encounters identifiability issues, making it unable to decode good trajectories into corresponding source trajectories. Consequently, it also becomes unable to utilize the MPC module to select suitable actions. The comparison results are shown in Figure 5 and Table 3. We also provide a Reacher example for visualization (with the link provided in Appendix D). In the video, we can see that the decoder with preference consistency loss can maintain preference consistency across domains. In contrast, the decoder without preference consistency loss cannot achieve such consistency.

Moreover, we also compare the state decoders learned by CDPC, DCC, and CMD in terms of their capabilities to maintain preference consistency across domains. The results, as shown in Figure 6, indicate that the CDPC decoder is significantly better in achieving preference consistency.

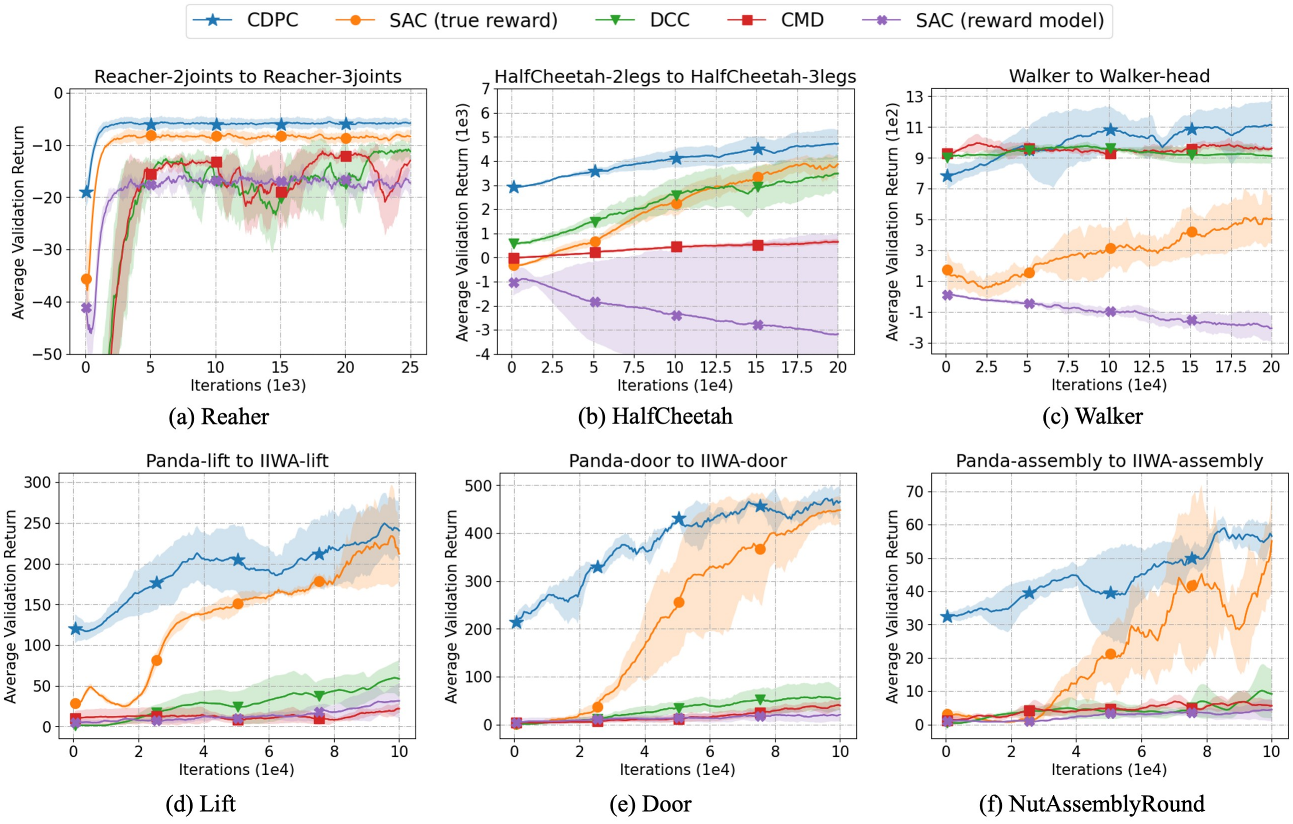
Q: Does the quality of the target-domain data have a significant impact on CDPC? Recall that CDPC learns from a target-domain $\mathcal{D}_{\text{tar}}$ with mixed samples collected by an expert policy and a uniform-at-random policy. Let $\alpha \in [0, 1]$ denote the mixing rate of expert data. We evaluate CDPC with four choices of mixture proportions and observe that CDPC is not very sensitive to the data quality. The results are shown in Figure 7 and Table 2. Even without any expert data, the performance of CDPC remains very competitive compared to the baselines.

6. Conclusion

We study a new cross-domain RL problem with preference feedback and propose a generic CDPC framework that enforces preference alignment between the source and target domains. Based on this concept, we propose a CDPC algorithm that combines a state decoder learned by preference consistency loss for training and a cross-domain

Table 1. **Final total rewards of CDPC and the benchmark methods.**

Tasks	SAC_T	SAC_R	DCC	CMD	CDPC
Reacher-3joints	-8.3±1.1	-17.2±2.4	-11.2±1.5	-12.9±1.6	**-5.7±0.6**
HalfCheetah-3legs	3866.5±362.3	-3171.3±44214.7	3479.5±599.5	646.9±129.9	**4716.4±588.9**
Walker-head	505.2±158.5	-207.1±82.3	911.4±49.6	961.9±8.8	**1111.8±162.7**
IIWA-Lift	212.2±33.4	31.5±21.2	58.4±22.8	21.9±5.4	**240.6±34.9**
IIWA-Door	448.1±29.0	20.6±12.8	54.7±24.6	39.7±9.1	**465.8±33.2**
IIWA-Assembly	55.0±14.8	4.4±3.3	9.1±7.2	5.6±1.6	**56.5±3.1**



(a) Reaher

(b) HalfCheetah

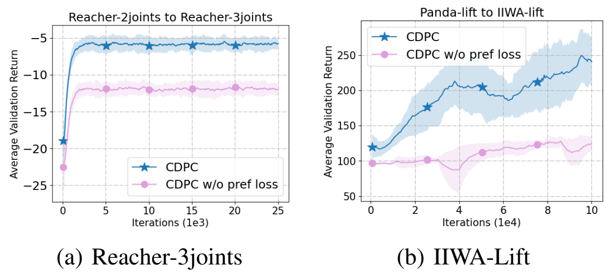
(c) Walker

(d) Lift

(e) Door

(f) NutAssemblyRound

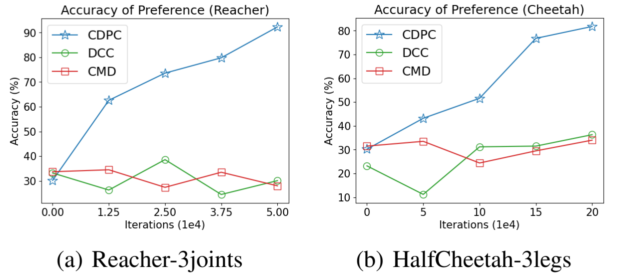
Figure 4. Learning curves of CDPC and the benchmark methods.



(a) Reacher-3joints

(b) IIWA-Lift

Figure 5. Ablation study: Learning curves of CDPC with and without the preference consistency loss.



(a) Reacher-3joints

(b) HalfCheetah-3legs

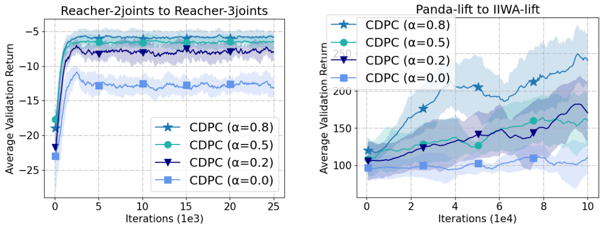
Figure 6. A comparison of the preference accuracy of the state decoders learned by CDPC, DCC, and CMD.

Table 2. **Total final rewards of CDPC under different mixing rates of expert data α.**

Tasks	$\alpha = 0.8$	$\alpha = 0.5$	$\alpha = 0.2$	$\alpha = 0.0$
Reacher-3joints	**-5.77±0.69**	-6.43±1.04	-7.93±1.30	-13.17±1.45
IIWA-Lift	**240.67±34.92**	160.09±36.17	170.45±41.49	110.22±21.52

Table 3. **Ablation study: Final total rewards of CDPC with and without the preference consistency loss.**

Tasks	w/ pref loss	w/o pref loss
Reacher-3joints	**-5.77±0.59**	-12.02±0.83
IIWA-Lift	**240.67±34.92**	124.27±10.31



(a) Reacher-3joints

(b) IIWA-Lift

Figure 7. Learning curves of CDPC under different mixing rates of expert data α.

MPC method for inference. Through extensive experiments on various robotic tasks, we confirm that CDPC indeed serves as a promising solution to achieving effective and data-efficient cross-domain transfer across domains.

Acknowledgements

This material is based upon work partially supported by the National Science and Technology Council (NSTC), Taiwan under Contract No. 112-2628-E-A49-023 and Contract No. 112-2634-F-A49-001-MBK, also partially supported by the NVIDIA Taiwan R&D Center, and also supported by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University and Ministry of Education (MOE), Taiwan. We also thank the National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Akrour, R., Schoenauer, M., and Sebag, M. Preference-based policy learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*, pp. 12–27. Springer, 2011.

Akrour, R., Schoenauer, M., and Sebag, M. April: Active preference learning-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pp. 116–131. Springer, 2012.

Ammar, H. B. and Taylor, M. E. Reinforcement learning transfer via common subspaces. In *Adaptive and Learning Agents: International Workshop, ALA 2011, Held at AAMAS 2011, Taipei, Taiwan, May 2, 2011, Revised Selected Papers*, pp. 21–36. Springer, 2012.

Ammar, H. B., Eaton, E., Luna, J. M., and Ruvolo, P. Autonomous Cross-Domain Knowledge Transfer in Lifelong Policy Gradient Reinforcement Learning.

Ammar, H. B., Eaton, E., Ruvolo, P., and Taylor, M. Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Botev, Z. I., Kroese, D. P., Rubinstein, R. Y., and L'Ecuyer, P. The cross-entropy method for optimization. In *Handbook of statistics*, volume 31, pp. 35–59. Elsevier, 2013.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Busa-Fekete, R. and Hüllermeier, E. A survey of preference-based online learning with bandit algorithms. In *Algorithmic Learning Theory: 25th International Conference, ALT 2014, Bled, Slovenia, October 8-10, 2014. Proceedings 25*, pp. 18–39. Springer, 2014.

Chakraborty, S., Bedi, A. S., Koppel, A., Manocha, D., Wang, H., Wang, M., and Huang, F. Parl: A unified framework for policy alignment in reinforcement learning. *arXiv preprint arXiv:2308.02585*, 2023.

Chen, L. Y., Hari, K., Dharmarajan, K., Xu, C., Vuong, Q., and Goldberg, K. Mirage: Cross-Embodiment Zero-Shot Policy Transfer with Cross-Painting. *arXiv preprint arXiv:2402.19249*, 2024.

Devin, C., Gupta, A., Darrell, T., Abbeel, P., and Levine, S. Learning modular neural network policies for multi-task and multi-robot transfer. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2169–2176. IEEE, 2017.

Fickinger, A., Cohen, S., Russell, S., and Amos, B. Cross-Domain Imitation Learning via Optimal Transport. In *International Conference on Learning Representations*, 2021.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. 2020.

Gui, H., Pang, S., Yu, S., Qiao, S., Qi, Y., He, X., Wang, M., and Zhai, X. Cross-domain policy adaptation with dynamics alignment. *Neural Networks*, 167:104–117, 2023.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Kamishima, T., Kazawa, H., and Akaho, S. A survey and empirical comparison of object ranking methods. In *Preference learning*, pp. 181–201. Springer, 2010.

Kim, K., Gu, Y., Song, J., Zhao, S., and Ermon, S. Domain adaptive imitation learning. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2020.

Kirk, R., Mediratta, I., Nalmpantis, C., Luketina, J., Hambro, E., Grefenstette, E., and Raileanu, R. Understanding the Effects of RLHF on LLM Generalisation and Diversity. In *The Twelfth International Conference on Learning Representations*, 2023.

Knox, W. B. and Stone, P. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009.

Lee, H., Phatale, S., Mansoor, H., Lu, K. R., Mesnard, T., Ferret, J., Bishop, C., Hall, E., Carbune, V., and Rastogi, A. RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. 2023.

Lee, K., Smith, L., Dragan, A., and Abbeel, P. B-Pref: Benchmarking Preference-Based Reinforcement Learning.

Lee, K., Smith, L. M., and Abbeel, P. PEBBLE: Feedback-Efficient Interactive Reinforcement Learning via Relabeling Experience and Unsupervised Pre-training. In *International Conference on Machine Learning*, pp. 6152–6163. PMLR, 2021.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Li, D., Meng, L., Li, J., Lu, K., and Yang, Y. Domain adaptive state representation alignment for reinforcement learning. *Information Sciences*, 609:1353–1368, 2022.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. Statistical Rejection Sampling Improves Preference Optimization. In *The Twelfth International Conference on Learning Representations*, 2023.

Memarian, F., Goo, W., Lioutikov, R., Niekum, S., and Topcu, U. Self-supervised online reward shaping in sparse-reward environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2369–2375. IEEE, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Mounsif, M., Lengagne, S., Thuilot, B., and Adouane, L. CoachGAN: fast adversarial transfer learning between differently shaped entities. In *17th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2020)*, volume 1, pp. 89–96, 2020.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566, 2018.

Nagabandi, A., Konolige, K., Levine, S., and Kumar, V. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pp. 1101–1112, 2020.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pilarski, P. M., Dawson, M. R., Degris, T., Fahimi, F., Carey, J. P., and Sutton, R. S. Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning. In *2011 IEEE international conference on rehabilitation robotics*, pp. 1–7. IEEE, 2011.

Qian, Y., Xiong, F., and Liu, Z. Intra-domain knowledge generalization in cross-domain lifelong reinforcement learning. In *International Conference on Neural Information Processing*, pp. 386–394. Springer, 2020.

Raychaudhuri, D. S., Paul, S., Vanbaar, J., and Roy-Chowdhury, A. K. Cross-domain imitation from observations. In *International Conference on Machine Learning*, pp. 8902–8912. PMLR, 2021.

Serrano, S. A., Martinez-Carranza, J., and Sucar, L. E. Intertask similarity measure for heterogeneous tasks. In *Robot World Cup*, pp. 40–52. Springer, 2021.

Serrano, S. A., Martinez-Carranza, J., and Sucar, L. E. Knowledge Transfer for Cross-Domain Reinforcement Learning: A Systematic Review. *arXiv preprint arXiv:2404.17687*, 2024.

Shankar, T., Lin, Y., Rajeswaran, A., Kumar, V., Anderson, S., and Oh, J. Translating robot skills: Learning unsupervised skill correspondences across robots. In *International Conference on Machine Learning*, pp. 19626–19644. PMLR, 2022.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Sun, S., Gupta, D., and Iyyer, M. Exploring the impact of low-rank adaptation on the performance, efficiency, and regularization of rlhf. *arXiv preprint arXiv:2309.09055*, 2023a.

Sun, Z., Shen, Y., Zhang, H., Zhou, Q., Chen, Z., Cox, D. D., Yang, Y., and Gan, C. SALMON: Self-Alignment with Principle-Following Reward Models. In *The Twelfth International Conference on Learning Representations*, 2023b.

Talvitie, E. and Singh, S. An experts algorithm for transfer learning. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pp. 1065–1070, 2007.

Taylor, M. E. and Stone, P. Transfer Learning for Reinforcement Learning Domains: A Survey. *The Journal of Machine Learning Research*, 10:1633–1685, 2009.

Taylor, M. E., Kuhlmann, G., and Stone, P. Autonomous Transfer for Reinforcement Learning. 2008.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017.

Torrey, L., Shavlik, J., Walker, T., and Maclin, R. Relational macros for transfer in reinforcement learning. In *Inductive Logic Programming: 17th International Conference, ILP 2007, Corvallis, OR, USA, June 19-21, 2007, Revised Selected Papers 17*, pp. 254–268. Springer, 2008.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Wang, Z., Cao, Z., Hao, Y., and Sadigh, D. Weakly supervised correspondence learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 469–476. IEEE, 2022.

Warnell, G., Waytowich, N., Lawhern, V., and Stone, P. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Watahiki, H., Iwase, R., Unno, R., and Tsuruoka, Y. Learning a Domain-Agnostic Policy through Adversarial Representation Matching for Cross-Domain Policy Transfer. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

Watahiki, H., Iwase, R., Unno, R., and Tsuruoka, Y. Leveraging Behavioral Cloning for Representation Alignment in Cross-Domain Policy Transfer. 2023.

Wilson, A., Fern, A., and Tadepalli, P. A bayesian approach for policy learning from trajectory preference queries. *Advances in neural information processing systems*, 25, 2012.

Wirth, C. and Fürnkranz, J. Preference-based reinforcement learning: A preliminary survey. Citeseer.

Wirth, C., Akrour, R., Neumann, G., and Fürnkranz, J. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136): 1–46, 2017.

Wu, J., Xie, Z., Yu, T., Zhao, H., Zhang, R., and Li, S. Dynamics-aware adaptation for reinforcement learning based cross-domain interactive recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 290–300, 2022.

Wulfmeier, M., Posner, I., and Abbeel, P. Mutual alignment transfer learning. In *Conference on Robot Learning*, pp. 281–290. PMLR, 2017.

Xu, K., Bai, C., Ma, X., Wang, D., Zhao, B., Wang, Z., Li, X., and Li, W. Cross-domain policy adaptation via value-guided data filtering. *Advances in Neural Information Processing Systems*, 36, 2023.

Yang, Q., Stork, J. A., and Stoyanov, T. Learn from Robot: Transferring Skills for Diverse Manipulation via Cycle Generative Networks. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pp. 1–6. IEEE, 2023.

You, H., Yang, T., Zheng, Y., Hao, J., and E Taylor, M. Cross-domain adaptive transfer reinforcement learning based on state-action correspondence. In *Uncertainty in Artificial Intelligence*, pp. 2299–2309. PMLR, 2022.

Zhang, Q., Xiao, T., Efros, A. A., Pinto, L., and Wang, X. Learning Cross-Domain Correspondence for Control with Dynamics Cycle-Consistency. In *International Conference on Learning Representations*, 2021a.

Zhang, Y., Zhang, X., Shen, T., Zhou, Y., and Wang, Z. Feature-option-action: a domain adaption transfer reinforcement learning framework. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–12. IEEE, 2021b.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Zhu, Z., Lin, K., Jain, A. K., and Zhou, J. Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Appendix

A. Identifiability Issue

Here, we explain the detailed steps of the gridworld example in Figure 1.

Given : one target domain trajectory τ, two state decoder ϕ_α^{-1} and ϕ_β^{-1}, one well-trained source domain policy π_{src}, source domain reward function R_{src} (defined as follows: reaching the endpoint $(2, 2)$ yields a reward of +1, while all other states yield a reward of +0.) Assume discount factor γ equals to 1.

For ϕ_α^{-1}, the process of decoding can be described as follows:

1. $\phi_\alpha^{-1}(00, 00) \Rightarrow (0, 0)$, $\pi_{src}(0, 0) = \rightarrow$, go to $(0, 1)$, reward = +0
2. $\phi_\alpha^{-1}(00, 01) \Rightarrow (0, 1)$, $\pi_{src}(0, 1) = \rightarrow$, go to $(0, 2)$, reward = +0
3. $\phi_\alpha^{-1}(00, 10) \Rightarrow (0, 2)$, $\pi_{src}(0, 2) = \downarrow$, go to $(1, 2)$, reward = +0
4. $\phi_\alpha^{-1}(01, 10) \Rightarrow (1, 2)$, $\pi_{src}(1, 2) = \downarrow$, go to $(2, 2)$, reward = +1
5. $\phi_\alpha^{-1}(10, 10) \Rightarrow (2, 2)$, total return = 1

For ϕ_β^{-1}, the process of decoding can be described as follows:

1. $\phi_\beta^{-1}(00, 00) \Rightarrow (0, 0)$, $\pi_{src}(0, 0) = \downarrow$, go to $(1, 0)$, reward = +0
2. $\phi_\beta^{-1}(00, 01) \Rightarrow (1, 0)$, $\pi_{src}(1, 0) = \rightarrow$, go to $(1, 1)$, reward = +0
3. $\phi_\beta^{-1}(00, 10) \Rightarrow (1, 1)$, $\pi_{src}(1, 1) = \downarrow$, go to $(2, 1)$, reward = +0
4. $\phi_\beta^{-1}(01, 10) \Rightarrow (2, 1)$, $\pi_{src}(2, 1) = \rightarrow$, go to $(2, 2)$, reward = +1
5. $\phi_\beta^{-1}(10, 10) \Rightarrow (2, 2)$, total return = 1

However, we cannot determine whether τ'_α or τ'_β is better, even after looking at the reward information; it is still not easy to distinguish between them. Without a suitable mechanism for choosing between ϕ_α^{-1} or ϕ_β^{-1}, an identifiability issue may arise.

B. Discussion: A Naive CD-PbRL Approach With an Action Encoder

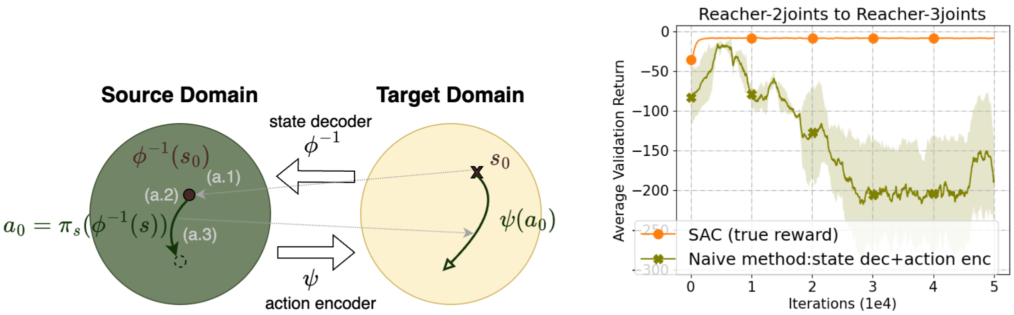
The most naive approach to addressing inter-task mapping problems is to train mapping functions for both state and action. A simple illustration and explanation are provided in Figure 8. Initially, we employed the concept of preference consistency to train an autoencoder for both state and action. However, the results were highly unstable, and since there was no information available regarding the target domain's reward, we needed to additionally train a reward model in the target domain to ensure both domains had preference information to maintain bidirectional mapping. A particularly tricky aspect is that if the reward model is not well-trained easily, the preference labels provided by the reward model will be incorrect, which will lead to poor performance of the action encoder. We also included the training results of this naive method in Figure 8.

Finally, we cleverly combined the preference consistency state decoder with MPC, which only required finding a decoder that could ensure consistent preferences, guaranteeing the effectiveness of the MPC approach.

C. Detailed Configurations of the Environments

The detailed descriptions of the environments of our experiments are as follows:

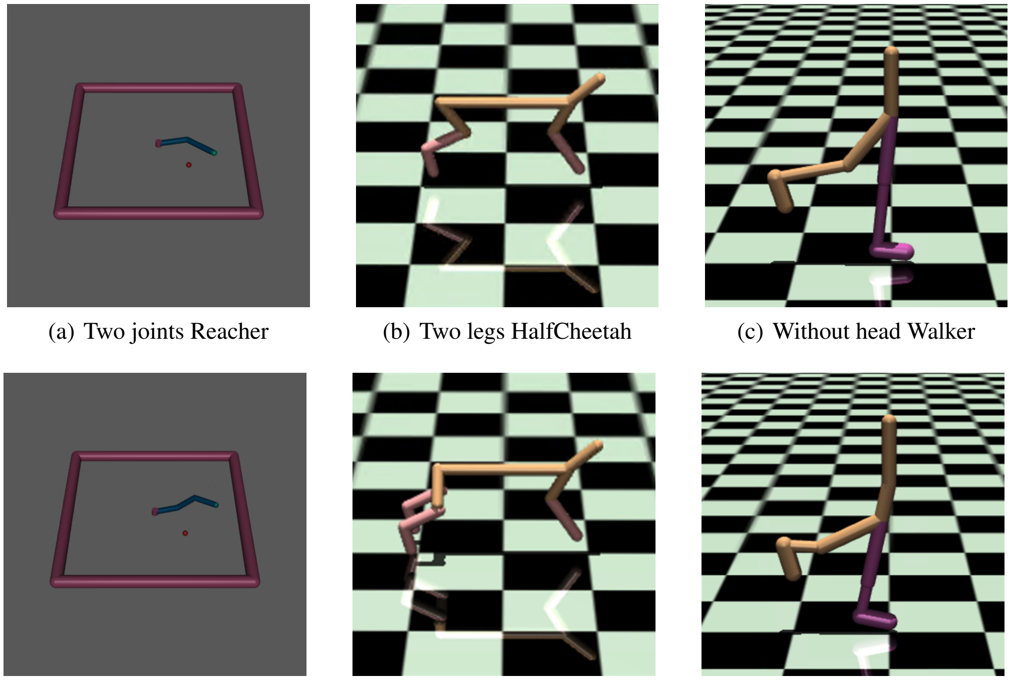
- **Reacher:** MuJoCo Reacher is an environment commonly used in reinforcement learning research. In this environment, an agent, typically a robotic arm, must learn to control its movements to reach a target location. The agent receives observations such as position and velocity of its joints, and its goal is to learn a policy that enables it to efficiently navigate its arm to the target.



(a) Naive method

(b) Total return of naive method

Figure 8. **Naive method:** (a) (a.1)First, the target state is transformed into the corresponding source state through the decoder. (a.2)Second, Using the known source domain policy, an action is selected in the source domain. (a.3)Finally, the action encoder transforms this action into the corresponding target action to complete one step. This process is repeated until termination. (b) Performance of naive method is poor and unstable.



(a) Two joints Reacher

(b) Two legs HalfCheetah

(c) Without head Walker

(d) Three joints Reacher

(e) Three legs HalfCheetah

(f) With head Walker

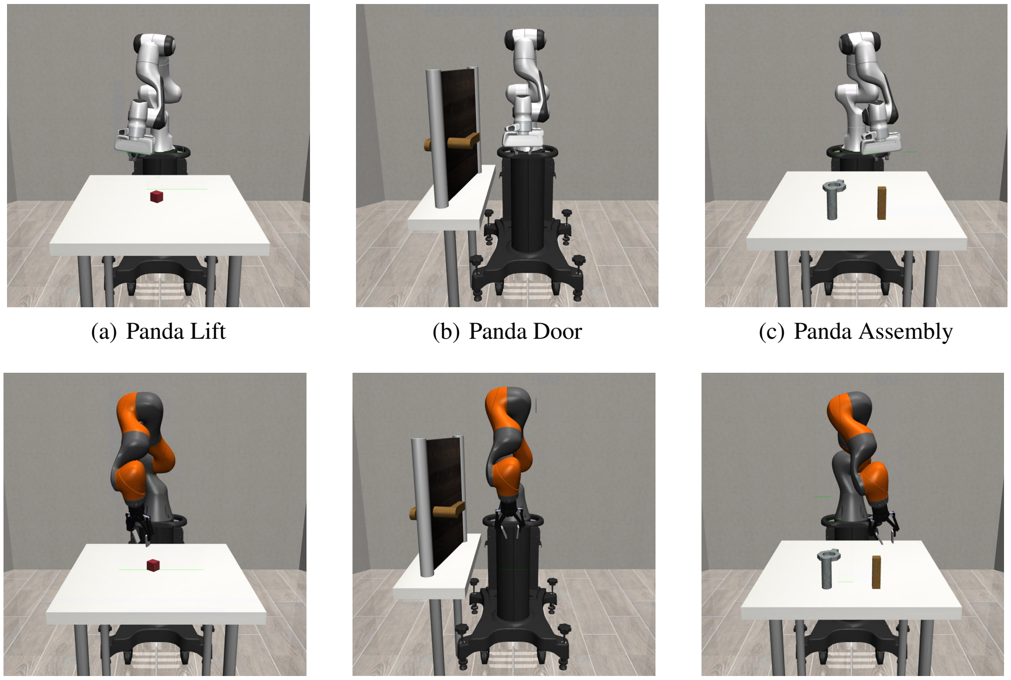
Figure 9. Agent morphologies of source domain and target domain in MuJoCo

- **HalfCheetah:** MuJoCo HalfCheetah is a simulated environment frequently utilized in reinforcement learning research. In this environment, an agent, typically a virtual half-cheetah, learns to navigate and control its movements in a physics-based simulation. The primary objective for the agent is to achieve efficient locomotion while adhering to physical constraints. The HalfCheetah environment offers a continuous control task, where the agent must learn to balance speed and stability to achieve optimal performance.

- **Walker:** MuJoCo Walker is a simulated environment frequently utilized in reinforcement learning research. In this environment, an agent, typically a virtual bipedal walker, learns to navigate and control its movements in a physics-based simulation. The primary objective for the agent is to achieve efficient and stable bipedal locomotion while adhering to physical constraints. The Walker environment offers a continuous control task, where the agent must learn to balance, walk, and sometimes recover from disturbances to achieve optimal performance.

Table 4. **Differences between source and target domain in MuJoCo**

		Reacher	**HalfCheetah**	**Walker**
Source	state dim	11	17	17
Domain	action dim	2	6	6
Target	state dim	12	23	19
Domain	action dim	3	9	7



(a) Panda Lift (b) Panda Door (c) Panda Assembly

(d) IIWA Lift (e) IIWA Door (f) IIWA Assembly

Figure 10. Agent morphologies of source domain and target domain in Robosuite

- **Panda:** RoboSuite Panda is a versatile robotic platform featuring a highly dexterous Panda robot arm. It's designed for research and development in robotics, offering flexibility for various tasks like manipulation and assembly. With its user-friendly interface and comprehensive software framework, it fosters innovation and collaboration in both academic and industrial settings. Our experimental tasks include Block Lifting, Door Opening, and Nut Assembly Round.

- **IIWA:** RoboSuite IIWA presents an advanced robotic platform centered around the highly sensitive and versatile IIWA robotic arm. Tailored for research and development, it excels in precision tasks like assembly and pick-and-place operations. Its intuitive interface and robust software framework support experimentation with cutting-edge robotics algorithms. Whether in academia or industry, RoboSuite IIWA empowers users to explore the forefront of robotic technology.

D. Video

The link to the video is https://imgur.com/a/cdpc-decoder-visualization-KvzLOqA. A small note we must clarify is that you might be curious about why the target point in the decoded trajectory keeps moving while the robotic arm doesn't move much. This is because our decoder takes the entire state as input, and the target point position is included in the state. Practically, it's challenging to ensure that the decoded target point position remains the same each time. However, in the Reacher environment, a trajectory can be considered good if the total distance between the fingertip position and the target point position is minimized throughout the episode. The decoder ensures that the decoded trajectory maintains preference consistency, and we can leverage this characteristic with MPC to select the optimal actions.

Table 5. **Differences between source and target domain in Robosuite**

		Lift	**Door**	**NutAssemblyRound**
Source	state dim	42	46	46
Domain	action dim	7	7	7
Target	state dim	50	54	54
Domain	action dim	7	7	7

E. Experimental Setting

- Hyperparameter
  We train source domain policy using SAC for 1e6 episodes, 128 for batch size, 3e-4 for Q network, policy and alpha learning rate. Target domain expert policy using SAC for 1e5 episodes, 128 for batch size, 3e-4 for Q network, policy and alpha learning rate. Decoder using LSTM for batch size 32, 1e-3 for learning rate run for 5 random seeds.

- Device
  GeForce RTX 2080 Ti, GeForce RTX 3090, GeForce RTX 4060, GeForce RTX 4090.

- Codebase
  For the implementation of SAC, we follow the github (https://github.com/quantumiracle/Popular-RL-Algorithms/tree/master)
  For the implementation of Robosuite policy, we follow the github (https://github.com/ARISE-Initiative/robosuite-benchmark/tree/master)
  For the implementation of DCC and CMD, we follow the github (https://github.com/sjtuzq/Cycle_Dynamics/tree/master)