An advantage based policy transfer algorithm for reinforcement learning with measures of transferability

Anonymous authors
Paper under double-blind review

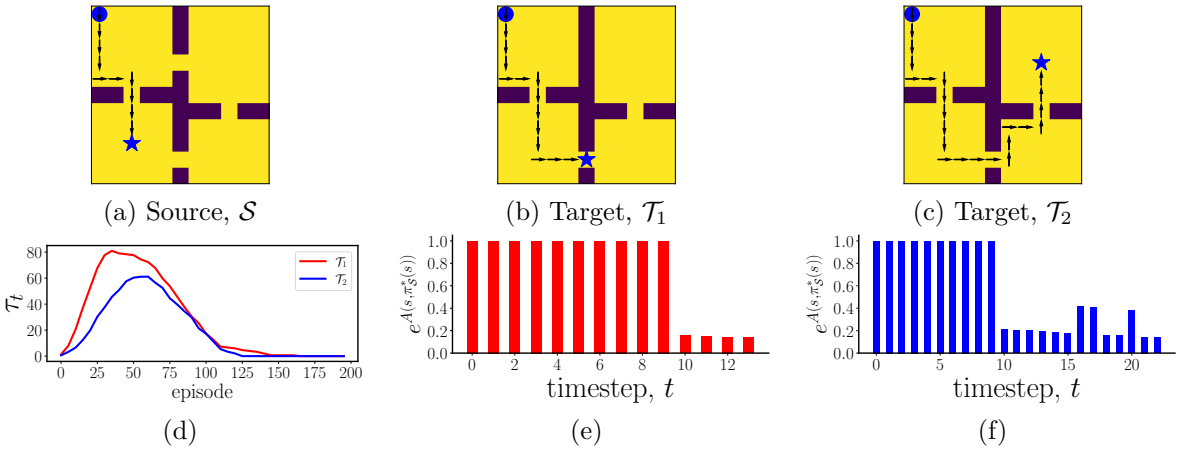
Abstract

Reinforcement learning (RL) enables sequential decision-making in complex and high-dimensional environments through interaction with the environment. In most real-world applications, however, a high number of interactions are infeasible. In these environments, transfer RL algorithms, which can be used for the transfer of knowledge from one or multiple source environments to a target environment, have been shown to increase learning speed and improve initial and asymptotic performance. However, most existing transfer RL algorithms are on-policy and sample inefficient, fail in adversarial target tasks, and often require heuristic choices in algorithm design. This paper proposes an off-policy Advantage-based Policy Transfer algorithm, APT-RL, for fixed domain environments. Its novelty is in using the popular notion of "advantage" as a regularizer, to weigh the knowledge that should be transferred from the source, relative to new knowledge learned in the target, removing the need for heuristic choices. Further, we propose a new transfer performance measure to evaluate the performance of our algorithm and unify existing transfer RL frameworks. Finally, we present a scalable, theoretically-backed task similarity measurement algorithm to illustrate the alignments between our proposed transferability measure and similarities between source and target environments. We compare APT-RL with several baselines, including existing transfer-RL algorithms, in three high-dimensional continuous control tasks. Our experiments demonstrate that APT-RL outperforms existing transfer RL algorithms and is at least as good as learning from scratch in adversarial tasks.

1 Introduction

A practical approach to implementing reinforcement learning (RL) in sequential decision-making problems is to utilize transfer learning. The transfer problem in reinforcement learning is the transfer of knowledge from one or multiple source environments to a target environment (Kaspar et al., 2020; Zhao et al., 2020; Bousmalis et al., 2018; Peng et al., 2018; Yu et al., 2017). The target environment is the intended environment, defined by the Markov decision process (MDP) $\mathcal{M}_{\mathcal{T}} = \langle \mathcal{X}, \mathcal{A}, \mathcal{R}_{\mathcal{T}}, \mathcal{P}_{\mathcal{T}} \rangle$, where $\mathcal{X}$ is the state-space, $\mathcal{A}$ is the action-space, $\mathcal{R}_{\mathcal{T}}$ is the target reward function and $\mathcal{P}_{\mathcal{T}}$ is the target transition dynamics. The source environment is a simulated or physical environment defined by the tuple $\mathcal{M}_{\mathcal{S}} = \langle \mathcal{X}, \mathcal{A}, \mathcal{R}_{\mathcal{S}}, \mathcal{P}_{\mathcal{S}} \rangle$ that, if learned, provides some sort of useful knowledge to be transferred to $\mathcal{M}_{\mathcal{T}}$. While in general $\mathcal{X}$ and $\mathcal{A}$ could be different between the source and target environments, we consider *fixed domain* environments here, where a domain is defined as $\langle \mathcal{X}, \mathcal{A} \rangle$ and is identical in the source and target.

For effective transfer of knowledge between fixed domain environments, it behooves the user to have $\mathcal{R}_{\mathcal{S}}$ and $\mathcal{P}_{\mathcal{S}}$ that are similar to $\mathcal{R}_{\mathcal{T}}$ and $\mathcal{P}_{\mathcal{T}}$, by either intuition, heuristics, or by some codified metric. Consider the single source transfer application to the four-room toy example for two different targets (Fig. 1). The objective is to learn a target policy, $\pi^*_{\mathcal{T}}$, in the corresponding target environments, $\mathcal{T}_1$ and $\mathcal{T}_2$ in Fig. 1(b) and (c) respectively, by utilizing knowledge from the source task, $\mathcal{S}$ in Fig. 1(a), such that the agent applies the optimal or near-optimal sequential actions to reach the goal state. Intuitively, target task $\mathcal{T}_1$ is more similar to the source task $\mathcal{S}$ when compared to target task $\mathcal{T}_2$, and thus we expect that knowledge transferred from $\mathcal{S}$ to $\mathcal{T}_1$ will be comparatively more useful. We propose a formal way to quantify the effectiveness of transfer through a *transferability evaluation measure*. Our proposed measure, shown as τ_1 for $\mathcal{T}_1$ in Fig.



(a) Source, $\mathcal{S}$ (b) Target, $\mathcal{T}_1$ (c) Target, $\mathcal{T}_2$

(d) (e) (f)

Figure 1: Knowledge transfer in the four-room toy problem: three tasks are presented in (a), (b) and (c), where ● represents the starting state and ★ represents the goal state; the goal state is moved further in (b) and (c) when compared to (a), and the doorways are also changed slightly. This makes the target task $\mathcal{T}_1$ in (b) more similar to the source task $\mathcal{S}$ in (a) when compared to the other target task $\mathcal{T}_2$ in (c). (d) An evaluation measure, τ, for transfer learning performance in tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ is shown, which calculates performance at each evaluation episode; (e)-(f) influence of source policy on the target tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ are shown in terms of $e^{A(\mathbf{x},\mathbf{a})}$ where $A(\mathbf{x},\mathbf{a}) = Q^*_{\mathcal{T}_i}(\mathbf{x},\mathbf{a}) - V^*(\mathbf{x})$ is the advantage function in task $\mathcal{T}_i$, and $i = 1, 2$. Note that the action is selected according to the source policy to calculate the advantage, which demonstrates the effect of the source policy on the target.

1(d) as a function of evaluation episode t, indicates that the source knowledge from $\mathcal{S}$ is highly transferable to the target $\mathcal{T}_1$. In contrast, target task $\mathcal{T}_2$ is less similar to $\mathcal{S}$ and thus knowledge transferred from $\mathcal{S}$ to $\mathcal{T}_2$ may be useful, but should be less effective than that transferred to $\mathcal{T}_1$. Accordingly, the transferability evaluation measure τ_2 for $\mathcal{T}_2$ in Fig. 1(d) indicates that the knowledge is useful ($\tau_2 > 0$), but not as useful as the transfer to $\mathcal{T}_1$ ($\tau_1 > \tau_2$). This task similarity measurement approach can provide critical insights about the usefulness of the source knowledge, and as we show, can also be leveraged in comparing the performance of different transfer RL algorithms. We also propose a theoretically-backed, model-based *task similarity measurement* algorithm, and use it to show that our proposed transferability measure closely aligns with the similarities between source and target tasks. Furthermore, we propose a new transfer RL algorithm that uses the popular notion of "advantage" as a regularizer, to weigh the knowledge that should be transferred from the source, relative to new knowledge learned in the target.

Our key idea is the following: calculate the advantage function based on actions selected using the source policy, and then use this advantage function as a regularization coefficient to weigh the influence of the source policy on the target. We can observe the effectiveness of this idea in our earlier toy example: Fig. 1(e) and (f) show the exponential of the advantage function as the regularization weight. By using the optimal source policy $\pi^*_{\mathcal{S}}$ to obtain an action, a, and calculating the corresponding advantage function $A(\mathbf{x},\mathbf{a})$, it is easy to see that $e^{A(\mathbf{x},\mathbf{a})}$ is lower in $\mathcal{T}_2$ when compared to $\mathcal{T}_1$. This result intuitively means that $\mathcal{S}$ can provide useful guidance for most of the actions selected by the target policy except the last four actions in $\mathcal{T}_1$, whereas, in contrast, the guidance is poor for the last thirteen actions in $\mathcal{T}_2$. We show that this simple yet scalable framework can improve transfer learning capabilities in RL. Our proposed advantage-based policy transfer algorithm is straightforward to implement and, as we show empirically on several continuous control benchmark gym environments, can be at least as good as learning from scratch.

Our main contributions are the following:

- We propose a novel advantage-based policy transfer algorithm, APT-RL, that enables a data-efficient transfer learning strategy between fixed-domain tasks in RL (Algorithm 1). This algorithm incorporates two new ideas: advantage-based regularization of gradient updates, and synchronous updates of the source policy with target data.

- We propose a new *relative transfer performance* measure to evaluate and compare the performance of transfer RL algorithms. Our idea extends the previously proposed formal definition of transfer in RL by Lazaric (2012), and unifies previous approaches proposed by (Taylor et al., 2007; Zhu et al., 2020). We provide theoretical support for the effectiveness of this measure (Theorem 1) and demonstrate its use in the evaluation of APT-RL on different benchmarks and against other algorithms (Section 6).

- We propose a model-based *task similarity measurement* algorithm, and use it to illustrate the relationship between source and target task similarities and our proposed transferability measure (Section 6). We motivate this algorithm by providing new theoretical bounds on the difference in the optimal policies' action-value functions between the source and target environments in terms of gaps in their environment dynamics and rewards (Theorem 2).

- We demonstrate the performance of APT-RL in twelve high-dimensional continuous control tasks which are created by varying three OpenAI gym environments, "halfcheetah", "ant", and "humanoid". Across these tasks, we compare APT-RL against four benchmarks: zero-shot source policy transfer, SAC (Haarnoja et al., 2018) without any source knowledge, fine-tuning the source policy with target data, and REPAINT, a state-of-the-art transfer RL algorithm (Tao et al., 2020). Our experiments show that APT-RL outperforms existing transfer RL algorithms on most tasks, and performs as good as learning from scratch in adversarial tasks.

The paper is organized as follows: we begin by reviewing related work in section 2, We propose and explain our off-policy transfer RL algorithm, APT-RL, in section 3. Next, we propose evaluation measures for APT-RL and a scalable task similarity measurement algorithm between fixed domain environments in section 4. We outline our experiment setup in section 5, and use the proposed transferability metric and task similarity measurement algorithm to evaluate the performance of APT-RL and compare it against other algorithms in section 6.

2 Related work

Traditionally, transfer in RL is described as the transfer of knowledge from one or multiple source tasks to a target task to help the agent learn in the target task. Transfer achieves one of the following: a) increases the learning speed in the target task, b) jumpstarts initial performance, and/or c) improves asymptotic performance (Taylor & Stone, 2009). Transfer in RL has been studied extensively in the literature, as evidenced by the three surveys (Taylor & Stone, 2009; Lazaric, 2012; Zhu et al., 2020) on the topic.

The prior literature proposes several approaches for transferring knowledge in RL. One such approach is the transfer of instances or samples (Lazaric et al., 2008; Taylor et al., 2008). Another approach is learning some representation from the source and then transferring it to the target task (Taylor & Stone, 2007b; 2005). Sometimes the transfer is also used in RL for better generalization between several environments instead of focusing on sample efficiency. For example, Barreto et al. (2017); Zhang et al. (2017) used successor feature representation to decouple the reward and dynamics model for better generalization across several tasks with similar dynamics but different reward functions. Another approach considered policy transfer where the KL divergence between point-wise local target trajectory deviation is minimized and an additional intrinsic reward is calculated at each timestep to adapt to a new similar task (Joshi & Chowdhary, 2021). In contrast to these works, our proposed approach simply uses the notion of advantage function to transfer *policy parameters* to take knowledge from the source policy to the target task and thus the transfer of knowledge is automated without depending on any heuristic choice.

There have also been different approaches to comparing source and target tasks and evaluating task similarity. A few studies have focused on identifying similarities between MDPs in terms of state similarities or policy similarities (Ferns et al., 2004; Castro, 2020; Agarwal et al., 2021). A couple of studies also focused on transfer RL where each of the MDP elements is varying (Taylor & Stone, 2007a; Gupta et al., 2017). Most of these approaches either require a heuristic mapping or consider a high level of similarity between the source and target tasks. In contrast to these works, we develop a scalable task similarity measurement algorithm for fixed domain environments that does not require the learning of the optimal policy.

A few recent studies have focused on fixed domain transfer RL problems for high-dimensional control tasks (Zhang et al., 2018; Tao et al., 2020). Most of these studies are built upon on-policy algorithms which require online data collection and tend to be less data efficient than an off-policy algorithm (which we consider here). Although Zhang et al. (2018) discussed off-policy algorithms briefly along with decoupled reward and transition dynamics, a formal framework is absent. Additionally, learning decoupled dynamics and reward models accurately is highly non-trivial and requires a multitude of efforts. More recently, Tao et al. (2020) proposed an on-policy actor-critic framework that utilizes the source policy and off-policy instance transfer for learning a target policy. This idea is similar to our approach, but different in two main ways. First, we consider an entirely off-policy algorithm, unlike Tao et al. (2020), and second, our approach does not require a manually tuned hyperparameter for regularization. Additionally, Tao et al. (2020) discards samples collected from the target environment that do not follow a certain threshold value which hampers data efficiency. Finally, Tao et al. (2020) only considers environments where source and target only vary by rewards and not dynamics. In contrast, we account for varying dynamics, which we believe to be of practical importance for transfer RL applications.

3 APT-RL: An off-policy advantage based policy transfer algorithm

In this section, we present our proposed transfer RL algorithm. We explain two main ideas that are novel to this algorithm: advantage-based regularization, and synchronous updates of the source policy. As our proposed algorithm notably utilizes advantage estimates to control the weight of the policy from a source task, we call it **A**dvantage based **P**olicy **T**ransfer in **RL** (APT-RL). We build upon soft-actor-critic (SAC) (Haarnoja et al., 2018), a state-of-the-art off-policy RL algorithm for model-free learning. We use off-policy learning to re-use past experiences and make learning sample-efficient. In contrast, an on-policy algorithm would require collecting new samples for each gradient update, which often makes the number of samples required for learning considerably high. Our choice of off-policy learning therefore helps with the scalability of APT-RL to higher dimensional problems, which we will show in the case studies.

3.1 Advantage-based policy regularization

The first new idea in our algorithm is to consider utilizing source knowledge during each gradient step of the policy update. Our intuition is that the current policy, π_ϕ parameterized by ϕ, should be close to the source optimal policy, $\pi^*_{\mathcal{S}}$, when the source can provide useful knowledge. In contrast, when the source knowledge does not aid learning in the target, then less weight should be put on the source knowledge. Based on this intuition, we modify the policy update formula of SAC as follows: we use an additional regularization loss with a temperature parameter, along with the original SAC policy update loss, to control the effect of the added regularization loss. Formally, the policy parameter has the update formula:

$$\phi \leftarrow \phi + \alpha_\pi \left[\hat{\nabla}_\phi J_1(\phi) + \beta_t \hat{\nabla}_\phi J_2(\phi) \right] \tag{1}$$

where $J_1(\cdot)$ is the usual SAC policy update loss, which uses soft-Q values instead of Q-values, and Q-function is parameterized by θ for the dataset $\mathcal{D}_\mathcal{T}$ and learning rate α_π,

$$J_1(\phi) = \mathbb{E}_{\mathbf{x}_t \sim \mathcal{D}_\mathcal{T}} \left[\mathbb{E}_{\mathbf{a}_t \sim \pi_\phi} [\alpha \log(\pi_\phi(\mathbf{a}_t|\mathbf{x}_t)) - Q_\theta(\mathbf{x}_t, \mathbf{a}_t)] \right], \tag{2}$$

and $J_2(\cdot)$ is the cross-entropy loss, $\mathcal{H}(\cdot,\cdot)$, between the source policy and the current policy,

$$J_2(\phi) = \mathcal{H}(\mu_\psi(\mathbf{a}|\mathbf{x}), \pi_\phi(\mathbf{a}|\mathbf{x})) = \mathbb{E}_{\mu_\psi(\mathbf{a}|\mathbf{x})}[-\log \pi_\phi(\mathbf{a}|\mathbf{x})], \tag{3}$$

where μ_ψ represents the optimal source policy $\pi^*_{\mathcal{S}}$ parameterized by ψ.

Thus, we are biasing the current policy π_ϕ to stay close to the source optimal policy $\pi^*_{\mathcal{S}}$ by minimizing the cross-entropy between these two policies while using the temperature parameter β to control the effect of the source policy. Typically, this type of temperature parameter is considered a hyper-parameter and requires (manual) fine-tuning. Finding an appropriate value for this parameter is highly non-trivial and maybe even task-specific. Additionally, if the value of β is not appropriately chosen, then the effect of the source policy may be detrimental to learning in the target task.

To overcome these limitations, we propose an *advantage-based* control of the temperature parameter, β. The main motivation here is to make the source influence adaptive, which means that we do not treat β as a hyper-parameter. The core intuition of our idea is that the second term of Equation 1 should have more weight when the average action taken according to μ_ψ is better than the random action. If the source policy provides an action that is worse than a random action in the target, then μ_ψ should be regularized to have less weight. This is equivalent to taking the difference of the advantages based on the current policy and the source policy, respectively. In addition, we consider the exponential, rather than the absolute value, of this difference, so that the temperature approaches zero when the source provides adversarial knowledge. Formally, our proposed advantage-based temperature parameter is determined as follows:

$$\beta_t = e^{A^t_{\mathcal{S}} - A^t_{\mathcal{T}}}, A^t_{\mathcal{T}} = Q_\theta(\mathbf{x}_t, \pi_\phi(\mathbf{x}_t)) - V(\mathbf{x}_t), A^t_{\mathcal{S}} = Q_\theta(\mathbf{x}_t, \mu_\psi(\mathbf{x}_t)) - V(\mathbf{x}_t) \tag{4}$$

Note that we can leverage the relationship between the soft-Q values and soft-value functions to represent the advantages from Equation 4 in a more convenient way that follows from (Haarnoja et al., 2018):

$$\begin{aligned} A^t_{\mathcal{T}} &= Q_\theta(\mathbf{x}_t, \pi_\phi(\mathbf{x}_t)) - \mathbb{E}[Q_\theta(\mathbf{x}_t, \mu_\psi(\mathbf{x}_t)) - \alpha \log \mu_\psi(\mathbf{a}_t|\mathbf{x}_t)] \\ A^t_{\mathcal{S}} &= Q_\theta(\mathbf{x}_t, \pi_\phi(\mathbf{x}_t)) - \mathbb{E}[Q_\theta(\mathbf{x}_t, \pi_\phi(\mathbf{x}_t)) - \alpha \log \pi_\phi(\mathbf{a}_t|\mathbf{x}_t)] \end{aligned} \tag{5}$$

3.2 Synchronous update of the source policy

We propose an additional improvement over the advantage-based policy transfer idea to further improve sample efficiency. As we consider a parameterized source optimal policy, μ_ψ, it is possible to update the parameters of the source policy with the target data, $\mathcal{D}_\mathcal{T}$, by minimizing the SAC loss. The benefits of this approach is two-fold: 1) If the source optimal policy provides useful information to the target, then this will accelerate the policy optimization procedure by working as a regularization term in Equation 1, and 2) This approach enables sample transfer to the target policy. This is because the initial source policy is learned using the source data; when this source policy is further updated with the target data, it can be viewed as augmenting the source dataset with the latest target data. Formally, the source policy will be updated as follows: $\psi \leftarrow \psi + \alpha_\psi \hat{\nabla}_\psi J_1(\psi)$, where $J_1(\psi)$ is the typical SAC loss for the source policy with parameters ψ and learning rate α_ψ. The pseudo-code for APT-RL is shown in Algorithm 1.

Algorithm 1 APT-RL: **A**dvantage based **P**olicy **T**ransfer in **R**einforcement **L**earning

Given: parameterized source optimal policy, μ_ψ, source learning step $\alpha_\mathcal{S}$
Initialize: current target policy, π_ϕ, target buffer $\mathcal{D}_\mathcal{T} = \emptyset$
for each iteration **do**
  for each target environment step **do**
    $\mathbf{a} \sim \pi_\phi(\mathbf{a}_t|\mathbf{x}_t)$
    $\mathbf{x}' \sim p(\mathbf{x}'|\mathbf{x}, \mathbf{a})$
    $\mathcal{D}_\mathcal{T} \leftarrow \mathcal{D}_\mathcal{T} \cup \{(\mathbf{x}, \mathbf{a}, \mathcal{R}(\mathbf{x}, \mathbf{a}), \mathbf{x}')\}$
  end for
  for G gradient updates **do**
    $\psi \leftarrow \psi + \alpha_\mathcal{S} \hat{\nabla}_\psi J_1(\psi)$ where J_1 is defined in Equation 2
    calculate $A^t_\mathcal{T}$ and $A^t_\mathcal{S}$ using Equation 5
    $\beta_t \leftarrow e^{A^t_\mathcal{S} - A^t_\mathcal{T}}$
    $\phi \leftarrow \phi + \alpha_\pi \left[\hat{\nabla}_\phi J_1(\phi) + \beta_t \hat{\nabla}_\phi J_2(\phi)\right]$
  end for
end for

4 An evaluation framework for transfer RL

To formally quantify the performance of APT-RL, including against other algorithms, in this section, we propose notions of transferability and task similarity, as discussed in Section 1. First, we propose a formal notion of transferability and use this notion to calculate a "relative transfer performance" measure. We

source knowledge, $\mathcal{K}_{\mathcal{S}}$	target knowledge, $\mathcal{K}_{\mathcal{T},t}$	evaluation measure, ρ_t
samples, $\mathcal{D}_{\mathcal{S}}$	samples, $\mathcal{D}_{\mathcal{T},t}$	average returns, $G_t = \mathbb{E}^{\pi_{\mathcal{T},t}}\left[\sum_{k=0}^{H} r_k\right]$
policy, μ_ψ	current policy, $\pi_{\mathcal{T},t}$	samples required to obtain reward threshold, n_{th}
models $\mathcal{R}_{\mathcal{S}}, \mathcal{P}_{\mathcal{S}}$	models, $\mathcal{R}_{\mathcal{T},t}, \mathcal{P}_{\mathcal{T},t}$	area under the reward curve, Δ_t
value functions $Q^*_{\mathcal{S}}, V^*_{\mathcal{S}}$	value functions $Q_{\mathcal{T},t}, V_{\mathcal{T},t}$	samples required to obtain asymptotic returns, n_∞

Table 1: A list of potential source knowledge, target knowledge and evaluation performance. This list unifies previous approaches proposed by (Taylor et al., 2007; Zhu et al., 2020). Note that target knowledge and evaluation performance is represented for the t^{th} episode, and that $\pi_{\mathcal{T},t}$ denotes the optimal target policy at the evaluation episode t.

demonstrate how this measure can be utilized to assess and compare the performance of APT-RL and similar algorithms. Then, we propose a scalable task similarity measurement algorithm for high-dimensional environments. We motivate this algorithm by providing a new theoretical bound on the difference in the optimal policies action-value functions between the source and target environments in terms of gaps in their environment dynamics and rewards. This task similarity measurement algorithm can be used to identify the best source task for transfer. Further, we use this algorithm in the experiment section, to illustrate how our proposed relative transfer performance closely aligns with the similarities between the source and target environment.

4.1 A measure of transferability

We formally define *transferability* as a mapping from stationary source knowledge and non-stationary target knowledge accumulated until timestep t, to learning performance, ρ_t.

Definition 4.1 (Single-task transferability)**.** Let $\mathcal{K}_{\mathcal{S}}$ be the transferred knowledge from a source task $\mathcal{M}_{\mathcal{S}}$ to a target task $\mathcal{M}_{\mathcal{T}}$, and let $\mathcal{K}_{\mathcal{T},t}$ be the available knowledge in $\mathcal{M}_{\mathcal{T}}$ at timestep t. Let ρ_t denote a measure that evaluates the learning performance in $\mathcal{M}_{\mathcal{T}}$ at timestep t. Then, transferability is defined as the mapping

$$\Lambda : \mathcal{K}_{\mathcal{S}} \times \mathcal{K}_{\mathcal{T},t} \rightarrow \rho_t \ .$$

Intuitively, this means that $\Lambda(\cdot)$ takes prior source knowledge and accumulated target knowledge to evaluate the learning performance in the target task.

As an example, if the collection of source data samples $\mathcal{D}_{\mathcal{S}}$ are utilized as the transferred knowledge and the average returns using the latest target policy, $G_t = \mathbb{E}^{\pi_{\mathcal{T},t}}\left[\sum_{k=0}^{H} r_k\right]$, is used as the evaluation performance of a certain transfer algorithm i, then the transferability of algorithm i at episode t, can be written as the following, $\Lambda_i : \mathcal{D}_{\mathcal{S}} \times \mathcal{D}_{\mathcal{T},t} \rightarrow G_t$.

Notice that we leave the choice of input and output of this mapping as user-defined task-specific parameters. Any traditional transfer methods can be represented using the idea of transferability. Potential choices for source and target knowledge and evaluation measures are listed in Table 1. Our idea extends the previously proposed formal definition of transfer in RL by Lazaric (2012), and unifies previous approaches proposed by Taylor et al. (2007); Zhu et al. (2020). Expressing transfer learning algorithms in terms of this notion of transferability has a number of advantages. First, this problem formulation can be easily extended to unify other important RL settings. For example, this definition can be extended to offline RL (Levine et al., 2020) by considering $\mathcal{K}_{\mathcal{S}} = \mathcal{D}_{\text{source}}$, and $\mathcal{K}_{\mathcal{T},t} = \emptyset, \ \forall t$. Second, the comparison of two transfer methods becomes convenient if they have the same evaluation criteria. For instance, one way to construct evaluation criteria may be to use sample complexity in the target task to achieve a desired return. Subsequently, the transferability measure can be used to assess "relative transfer performance", which can act as a tool for comparing two different transfer methods conveniently.

Definition 4.2 (Relative transfer performance, τ)**.** Given the transferability mapping of algorithm i, Λ_i, the relative transfer performance is defined as the difference between the corresponding learning performance ρ^i_t and learning performance from a base RL algorithm ρ^b_t at evaluation episode t. Formally, $\tau_t = \rho^i_t - \rho^b_t$,

where the base RL algorithm represents learning from scratch in the target task (meaning $\mathcal{K}_{\mathcal{S}} = \emptyset$), and ρ_t^i, ρ_t^b are the same evaluation criteria of learning performances.

4.1.1 Theoretical support

We first formally show that, with an appropriate definition of the evaluation measure, non-negative relative transfer performance leads to a policy in the target task which is at least as good as learning from scratch.

Theorem 1. ***(Relative transfer performance and policy improvement)*** *Consider* $\rho_t^i = \mathbb{E}^{\pi_{i,t}}\left[\sum_{k=0}^{H} r_k | \mathbf{x}_0\right]$ *for policy* π_i *and* $\rho_t^b = \mathbb{E}^{\pi_{b,t}}\left[\sum_{k=0}^{H} r_k | \mathbf{x}_0\right]$ *for policy* π_b*, where* $\mathbf{x}_0$ *is the starting state and each policy is executed for* H *timesteps. Then, the learned policy,* $\pi_{i,t}$ *using algorithm* i*, in the target at episode* t *is at least as good as the source optimal policy* $\pi_{b,t}$ *if* $\tau_t \geq 0$*.*

The proof can be found in the Appendix A.

4.1.2 Revisiting the toy problem

We also leverage the toy example presented in Fig. 1 to explain the proposed concepts. The performance evaluation is chosen as the average returns, collected from an evaluation episode t, that is $\rho_t = \sum_{k=0}^{H} r_k$. For transfer, we choose the low-level direct knowledge Q-values. At first, we initialize the target Q-values with pre-trained source Q-values, $Q_{\mathcal{S}}^*$. Thus, at each episode, t, the updated Q-values in the target are a combination of both source and target knowledge. Thus, the transferability mapping can be expressed as $\Lambda_{\text{Q-learning}} : Q_{\mathcal{S}}^* \times Q_{\mathcal{T},t} \rightarrow \mathbb{E}^{\pi_{\mathcal{T},t}}\left[\sum_{k=0}^{H} r_k\right]$. We calculate ρ_t after every 10 timestep by executing the greedy policy from the updated Q-values for a fixed time horizon. Relative transfer performance, τ_t, remains non-negative for both the target tasks $\mathcal{T}_1$ and $\mathcal{T}_2$, for up to around 125 evaluation episodes. Intuitively this means that the learning performance of both policies is better than a base algorithm for all of the evaluation episodes. Also, τ_t is higher for $\mathcal{T}_1$ than $\mathcal{T}_2$ which means that transferring knowledge from $\mathcal{S}$ leads to better learning performance in $\mathcal{T}_1$ compared to $\mathcal{T}_2$. This can be explained by the fact that the dynamics in $\mathcal{T}_1$ are more similar to $\mathcal{S}$ than $\mathcal{T}_2$.

4.2 Measuring task similarity

As seen in the toy problem above, a measure of task similarity would help us illustrate the close alignment of our proposed transferability metric with the similarities of the source and target tasks. Beyond this, measuring task similarity can provide additional insights into why a particular source task is more appropriate to transfer knowledge to the target task. Motivated by these, we propose an algorithm for measuring task similarity in this section.

4.2.1 Theoretical motivation

To motivate the idea behind our proposed algorithm, we first investigate theoretical bounds on the expected discrepancies between the policies learned in the source and target environments. One effective way for this analysis is to calculate the upper bound on differences between the optimal policies' action-values. Previously, action-value bounds have been proposed for similar problems by Csáji & Monostori (2008); Abdolshah et al. (2021). We extend these ideas to the transfer learning setting where we derive the bound between the target action-values under target optimal policy, $\pi_{\mathcal{T}}^*$ and target action-values under source optimal policy, $\pi_{\mathcal{S}}^*$.

Theorem 2. ***(Action-value bound between fixed-domain environments)*** *If* $\pi_{\mathcal{S}}^*$ *and* $\pi_{\mathcal{T}}^*$ *are the optimal policies in the MDPs* $\mathcal{M}_{\mathcal{S}} = \langle \mathcal{X}, \mathcal{A}, \mathcal{R}_{\mathcal{S}}, \mathcal{P}_{\mathcal{S}} \rangle$ *and* $\mathcal{M}_{\mathcal{T}} = \langle \mathcal{X}, \mathcal{A}, \mathcal{R}_{\mathcal{T}}, \mathcal{P}_{\mathcal{T}} \rangle$ *respectively, then the corresponding action-value functions is upper bounded by*

$$||\mathbf{Q}_{\mathcal{T}}^{\pi_{\mathcal{T}}^*} - \mathbf{Q}_{\mathcal{T}}^{\pi_{\mathcal{S}}^*}||_\infty \leq \frac{2\delta_{\mathcal{ST}}^r}{1-\gamma} + \frac{2\gamma\delta_{\mathcal{ST}}^{TV}(R_{max,\mathcal{S}} + R_{max,\mathcal{T}})}{(1-\gamma)^2} \tag{6}$$

where $\delta_{\mathcal{ST}}^r = ||\mathcal{R}_{\mathcal{S}}(\mathbf{x}, \mathbf{a}) - \mathcal{R}_{\mathcal{T}}(\mathbf{x}, \mathbf{a}))||_\infty$*,* $\delta_{\mathcal{ST}}^{TV}$ *is the total variation distance between* $\mathcal{P}_{\mathcal{S}}$ *and* $\mathcal{P}_{\mathcal{T}}$*,* γ *is the discount factor and* $R_{max,\mathcal{S}} = ||\mathcal{R}_{\mathcal{S}}(\mathbf{x}, \mathbf{a})||_\infty$*,* $R_{max,\mathcal{T}} = ||\mathcal{R}_{\mathcal{T}}(\mathbf{x}, \mathbf{a})||_\infty$*.*

The proof of this theorem can be found in the Appendix B. Note that as we propose APT-RL for fixed-domain environments, the bound can be expressed in terms of differences in the remaining environment parameters: the total variation distance between the source and target transition probabilities, and the maximum reward difference between the source and the target. Intuitively this bound means that a lower total variation distance between the transition dynamics can provide a tighter bound on the deviation between the action-values from the target and source optimal policies. Similarly, having a smaller reward difference also helps in getting lower action-value deviations in the target. Also note that, if the reward function or transition dynamics remain identical between source and target, then the corresponding term on the right side of Equation 6 vanishes.

Next, motivated by this bound, we propose a task similarity measurement algorithm that assesses the differences between source and target dynamics and rewards in order to evaluate their similarity.

4.2.2 A model-based task similarity measurement algorithm

Previous attempts for measuring task similarity include behavioral similarities in MDP in terms of state-similarity or bisimulation metric, and policy similarity (Ferns et al., 2004; Castro, 2020; Agarwal et al., 2021). Calculating such metrics in practice is often challenging due to scalability issues and computation limits. Additionally, our key motivation is to find a similarity measurement that does not require solving for the optimal policy *apriori*, as the latter is often the key challenge in RL. To this end, we propose a new model-based method for calculating similarities between tasks.

We propose an encoder-decoder based deep neural network model at the core of this idea. For any source or target task, a dataset, $\mathcal{D} = \{(\mathbf{x}, \mathbf{a}, r, \mathbf{x}')\}$, is collected by executing a random policy. Next, a dynamics model, $f_{\mathcal{P}}(\mathbf{x}, \mathbf{a})$, is trained by minimizing the mean-squared-error (MSE) loss using stochastic gradient descent, $\mathcal{L}_{\text{dyn}} = ||\mathbf{x}' - f_{\mathcal{P}}(\mathbf{x}, \mathbf{a})||_2$. Similarly, a reward model, $f_{\mathcal{R}}(\mathbf{x}, \mathbf{a})$ is trained using the collected data to minimize the following MSE loss, $\mathcal{L}_{\text{rew}} = ||r - f_{\mathcal{R}}(\mathbf{x}, \mathbf{a})||_2$. The encoder portion of the neural network model encodes state and action inputs into a latent vector. Then, the decoder portion uses this latent vector for the prediction of the next state or reward. We consider decoupled models for this purpose, meaning that we learn separate models for reward and transition dynamics from the same dataset. This allows us to identify whether only reward or transition dynamics or both vary between tasks. Once these models are learned, the source model is used to predict the target data and calculate the L_2 distance between the predicted and actual data as the similarity error. If $\xi_k^{\mathcal{P}}$ and $\xi_k^{\mathcal{R}}$ are the similarity errors in target dynamics and rewards, respectively, we can calculate the dynamics and reward similarity separately as follows,

$$\text{dynamics similarity: } \Xi_{\mathcal{S},\mathcal{T}}^{\mathcal{P}} = \frac{1}{|\mathcal{D}_{\mathcal{T}}|} \sum_{k=1}^{|\mathcal{D}_{\mathcal{T}}|} \xi_k^{\mathcal{P}}, \text{reward similarity: } \Xi_{\mathcal{S},\mathcal{T}}^{\mathcal{R}} = \frac{1}{|\mathcal{D}_{\mathcal{T}}|} \sum_{k=1}^{|\mathcal{D}_{\mathcal{T}}|} \xi_k^{\mathcal{R}} \tag{7}$$

Our approach is summarized in Algorithm 2. This approach can be viewed as a modern version of (Ammar et al., 2014), but instead of using Restricted Boltzmann Machines, we use deep neural network-based encoder-decoder architecture to learn the models, and we do it in a decoupled way.

Algorithm 2 Model-based task similarity measurement

1: Collect m data samples from source $\mathcal{M}_{\mathcal{S}}$ using a random policy, $\mathcal{D}_{\mathcal{S}} = \{(\mathbf{x}_{\mathcal{S}}, \mathbf{a}_{\mathcal{S}}, r_{\mathcal{S}}, \mathbf{s}'_{\mathcal{S}})\}$
2: Collect m data samples from target $\mathcal{M}_{\mathcal{T}}$ using a random policy, $\mathcal{D}_{\mathcal{T}} = \{(\mathbf{x}_{\mathcal{T}}, \mathbf{a}_{\mathcal{T}}, r_{\mathcal{T}}, \mathbf{s}'_{\mathcal{T}})\}$
3: Learn dynamics models, $f_{\mathcal{P}}^{\mathcal{S}}(\mathbf{x}, \mathbf{a}), f_{\mathcal{P}}^{\mathcal{T}}(\mathbf{x}, \mathbf{a})$ using $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$ and minimizing $\mathcal{L}_{\text{dyn}} = ||\mathbf{x}' - f_{\mathcal{P}}(\mathbf{x}, \mathbf{a})||_2$
4: Learn reward models, $f_{\mathcal{R}_{\mathcal{S}}}(\mathbf{x}, \mathbf{a}), f_{\mathcal{R}_{\mathcal{T}}}(\mathbf{x}, \mathbf{a})$ using $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$ and minimizing $\mathcal{L}_{\text{rew}} = ||r - f_{\mathcal{R}}(\mathbf{x}, \mathbf{a})||_2$
5: **for** each $(\mathbf{x}_{\mathcal{T}}, \mathbf{a}_{\mathcal{T}}, r_{\mathcal{T}}, \mathbf{x}'_{\mathcal{T}}) \in \mathcal{D}_{\mathcal{T}}$ **do**
6: $\quad \hat{\mathbf{x}}'_{\mathcal{T}} = f_{\mathcal{P}_{\mathcal{T}}}(\mathbf{x}_{\mathcal{T}}, \mathbf{a}_{\mathcal{T}}), \hat{r}_{\mathcal{T}} = f_{\mathcal{R}_{\mathcal{T}}}(\mathbf{x}_{\mathcal{T}}, \mathbf{a}_{\mathcal{T}})$
7: $\quad \xi_k^{\mathcal{P}} = ||\hat{\mathbf{x}}'_{\mathcal{T}} - \mathbf{x}'_{\mathcal{T}}||_2, \xi_k^{\mathcal{R}} = ||\hat{r}_{\mathcal{T}} - r_{\mathcal{T}}||_2$
8: **end for**
9: dynamics similarity, $\Xi_{\mathcal{S},\mathcal{T}}^{\mathcal{P}} = \frac{1}{|\mathcal{D}^{\mathcal{T}}|} \sum_{k=1}^{|\mathcal{D}^{\mathcal{T}}|} \xi_k^{\mathcal{P}}$
10: reward similarity, $\Xi_{\mathcal{S},\mathcal{T}}^{\mathcal{R}} = \frac{1}{|\mathcal{D}^{\mathcal{T}}|} \sum_{k=1}^{|\mathcal{D}^{\mathcal{T}}|} \xi_k^{\mathcal{R}}$

5 Experiment Setup

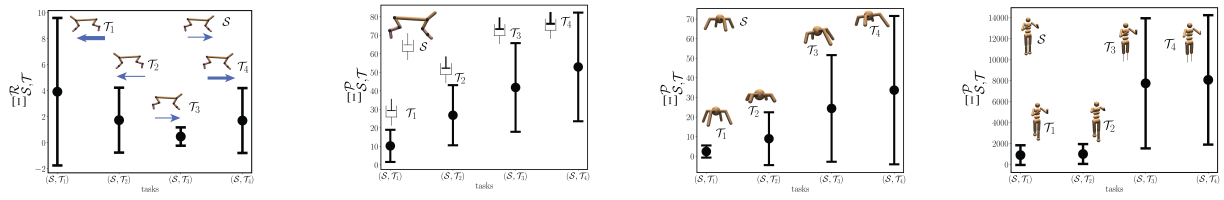
We apply the described methods to three popular high-dimensional continuous control benchmark gym environments (Brockman et al., 2016): 1) 'HalfCheetah-v3', 2) 'Ant-v3', and 3) 'Humanoid-v3'. The vanilla Gym environments are not suitable for transfer learning settings. Thus we create four different perturbations of the dynamics of each environment. The source HalfCheetah-v3 environment has the standard gym values for damping and the four target environments have different values of damping increased gradually in each task. The least similar task has the highest damping values in the joints. For the Ant-v3 environment, the source task has the standard gym robot and the four target environments have varied dynamics by changing the leg lengths of the robot. Similarly, for the Humanoid-v3 environment, the source task has the standard gym robot and the four target environments have varied dynamics by changing the leg length as well as the size of the hand and legs. For the HalfCheetah-v3 environment, we create additional four tasks by changing the reward function to show reward variation. Details of the environments and the algorithm hyperparameters can be found in Appendices C and D.

For each environment, the knowledge transferred is the source optimal policy, data collected from the target task is used as the target knowledge, and average returns during each evaluation episode are used as the performance evaluation metric, $\Lambda_i : \pi^*_{\mathcal{S}} \times \mathcal{D}_{\mathcal{T},t} \rightarrow \mathbb{E}^{\pi_{\mathcal{T},t}} [\sum_k r_k]$. To obtain the source optimal policy, the SAC algorithm is utilized to train a policy from scratch in each source environment. In each of these experiments, we perform and compare our algorithm APT-RL against one of the recent benchmarks on transfer RL, the REPAINT algorithm proposed by Tao et al. (2020). We also compare APT-RL against zero-shot source policy transfer, policy fine-tuning on the target task, and SAC without any source knowledge.

6 Experiment Results

6.1 Task similarity

We leverage Algorithm 2 to calculate the task similarity between the source and each of the target tasks. The empirical task similarity is shown in Fig. 2. Fig. 2(a) shows the similarity in tasks for the half-cheetah environment with varying rewards. For reward similarity, we can see the highest dissimilarity when the robot is provided a negative reward instead of a positive reward. Similarly, for the half-cheetah environment with varying dynamics in Fig. 2(b), we can see an approximately linear trend in the similarity of the dynamics. This makes sense as the change in joint damping is gradual and constant. In Fig. 2(c) we show the task similarity in the Ant environment with varying dynamics. As we change the dynamics of each of the target environments by changing the length of the legs of the robot, the similarity between each target and the source task reduces monotonically. Finally, task similarity of the humanoid environments are shown in Fig. 2(d) where first two tasks are relatively more similar to the source and the final two tasks are less similar.



(a) Halfcheetah (reward) (b) Halfcheetah (dynamics) (c) Ant (dynamics) (d) Humanoid (dynamics)

Figure 2: **Task dissimilarity:** Empirical task similarity between several variations of Half-cheetah, Ant, and Humanoid environments

.

6.2 Transferability of APT-RL, $\Lambda_{\text{APT-RL}}$

6.2.1 Half-cheetah-v3

Fig. 3 (a)-(c) shows the transfer evaluation performance for three target tasks with varying dynamics and Fig. 3(d) shows the performance for one target task with negative reward. In all cases, APT-RL learns faster and also achieves higher average returns than learning from scratch. While fine-tuning the source policy performs better than APT-RL in task $\mathcal{T}_1$, it performs worse than both APT-RL and learning from scratch in rest of the tasks in $\mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4$. Most importantly, APT-RL performs as good as learning from scratch for target task $\mathcal{T}_4$ with a negative reward. Note that, this is an adversarial source task as the robot needs to learn to run in the opposite direction in the target task. In contrast, the REPAINT algorithm fails to achieve similar evaluation performance and in most cases, obtains evaluation performance lower than learning from scratch using SAC. As REPAINT is a PPO-based on-policy algorithm, this result aligns with the previously reported performance of SAC and PPO algorithms (Haarnoja et al., 2018). The performance of APT-RL may be explained by the fact that the source optimal policy jumpstarts the target policy. This increase in learning performance, in turn, provides a positive relative transfer measure over time in tasks $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ as shown in Fig. 4. Note that τ remains higher for the most similar task $\mathcal{T}_1$ and relatively lower for the least similar task $\mathcal{T}_2$. Temperature parameter β decreases quickly initially, then increases slightly and stays approximately constant over time. This can be explained by the fact that more weight is put into the regularization loss initially and once the target policy becomes better the effect reduces. As we keep utilizing the target data to update the source policy, the source policy improves over time and provides useful information during the later timesteps. Finally, we observe that the effect of the source policy remains almost constant with respect to task similarity. This makes sense due to the particular change in dynamics of the environment. We anticipate that changing only the damping values make the tasks less adversarial to the source in terms of dynamics.

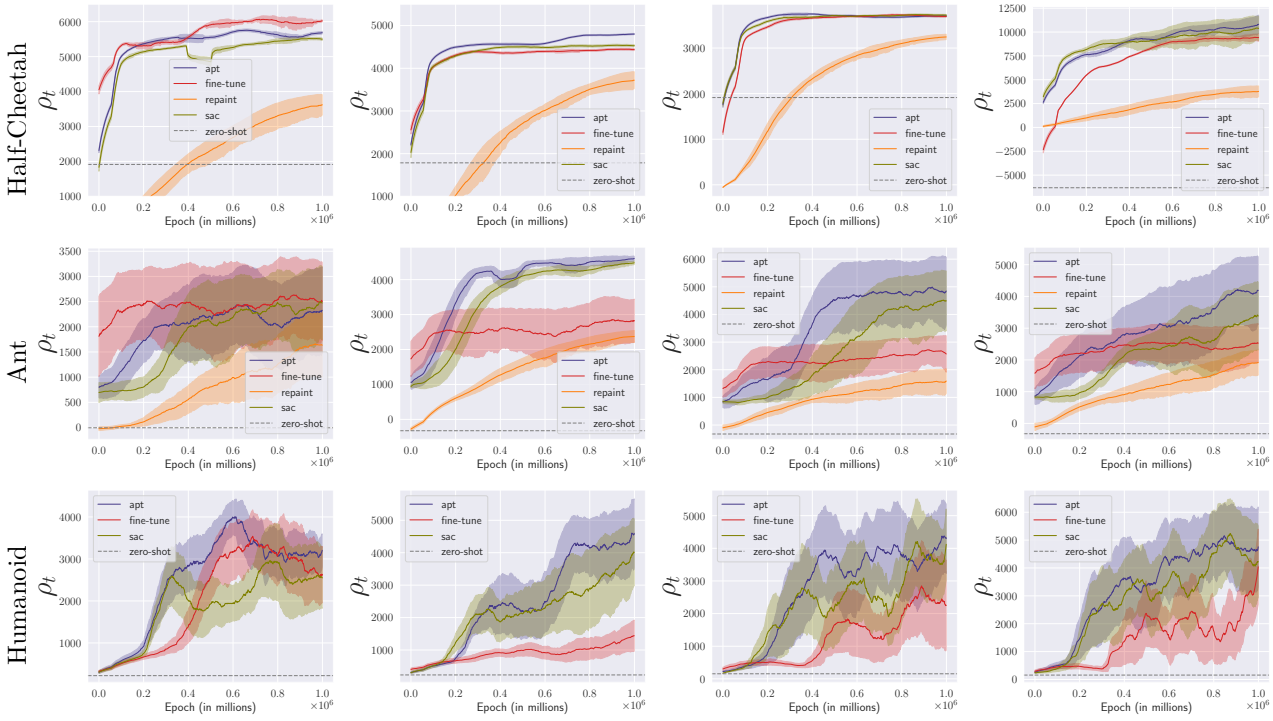
6.2.2 Ant-v3

For the ant environment, we observe significant performance gain of APT-RL in the target task against learning from scratch, zero-shot policy transfer, fine-tuning, and the REPAINT algorithm (Fig. 3). For target tasks that are very similar to the source, we observe a fast convergence of the policy in the target. For less similar source and target, APT-RL can even achieve higher learning performance than learning from scratch. This might happen due to the jumpstart of the target policy and also due to the synchronous improvement of the source policy. The latter characteristic of APT-RL accelerates policy updates. Similar to the half-cheetah environment, we observe that the temperature parameter decreases with timesteps and decreases more when task similarity is lower (Fig. 4). In all of these examples, APT-RL outperforms the REPAINT algorithm.

6.3 Humanoid-v3

For the humanoid environment, APT-RL outperforms all the baselines with significant initial performance gain (Fig. 3). For the final target task, $\mathcal{T}_4$, APT-RL performs similarly to learning from scratch. We anticipate that this behavior is mainly due to the difficulty of the task. The change in the dynamics of the environment make the target an adversarial task which is relatively more difficult to solve than the rest of the tasks. This can be also supported by the fact that $\mathcal{T}_4$ is the least similar task among all of the target tasks. Similar to the ant and half-cheetah environments, we observe that the temperature parameter decreases with less task similarity (Fig. 4). We do not show results from the REPAINT algorithm as it fails to solve even the source task.

Finally, we show an ablation study on the effect of the temperature parameter, β, in Fig. 5. Notice that APT-RL outperforms manual choice of β in all tasks except the halfcheetah environment where $\beta = 1.0$ performs slightly better than APT-RL. Interestingly, Fig. 4 shows that APT-RL also converges to $\beta = 1.0$ without any manual choice. We argue that it shows further evidence on the strength of APT-RL where we do not need to manually choose the temperature parameter.



(a) Target task, $\mathcal{T}_1$ (b) Target task, $\mathcal{T}_2$ (c) Target task, $\mathcal{T}_3$ (d) Target task, $\mathcal{T}_4$

Figure 3: **APT-RL transferability, $\Lambda_{\text{APT-RL}}$:** APT-RL is compared against vanilla SAC (learning from scratch), REPAINT, zero-shot policy, and fine-tuned policy. Average return during the evaluation episode is taken as ρ_t, meaning $\rho_t = \mathbb{E}^{\pi^*_{\mathcal{T}_i}}[\sum_t r_k]$. We do not show Repaint for the humanoid environment as it fails to solve the tasks. Results are shown with one standard deviation range.

7 Limitations

The task similarity algorithm presented in section 4.2.2 uses the random policy to learn models for the dynamics and the reward. While using a random policy for model learning is fairly common in the literature Zhang et al. (2018); Moerland et al. (2023), a key challenge is to learn a reasonable model for complex tasks. We anticipate that the task similarity algorithm might not be effective in environments where a random policy cannot efficiently capture the underlying complexities of the dynamics and the reward. We would also like to focus on the limitations of APT-RL. One of the key challenges of transfer RL is to identify useful source tasks and reduce the impact of adversarial sources. While APT-RL shows strong performance in most of the cases, as shown in the Fig. 3(a) for the 'Half-Cheetah-V3' environment, simple transfer approaches such as fine-tuning the source policy is more convenient. This behavior can be explained by the close similarity of the source to the target. Therefore, identifying measures of when to opt for more advanced transfer algorithms such as APT-RL remains an interesting challenge.

8 Conclusion

In this paper, we proposed the APT-RL algorithm to transfer knowledge from a source task in an off-policy fashion. Through advantage-based regularization, our algorithm does not require any heuristic or manual fine-tuning of the objective function. We also introduced a new relative transfer performance measure, which can help evaluate and compare transfer learning approaches in RL. We also provided a simple, theoretically-backed algorithm to calculate task similarity, and demonstrated the alignment of our proposed transfer performance measure with source and target task similarities. We demonstrated the effectiveness of APT-RL in continuous control tasks and showed its superior performance against benchmark transfer RL algorithms.

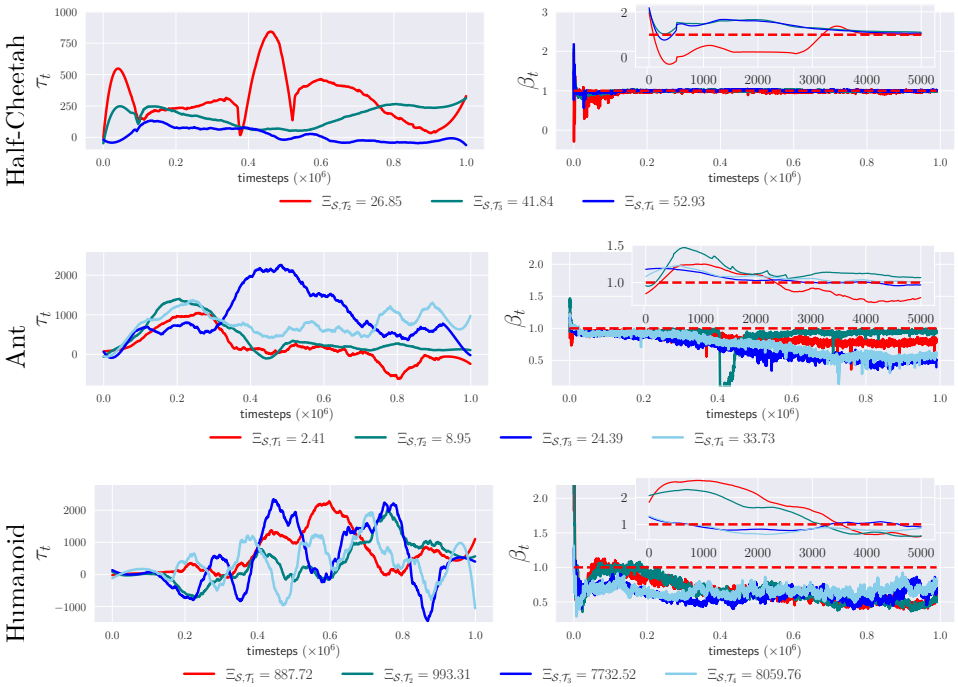
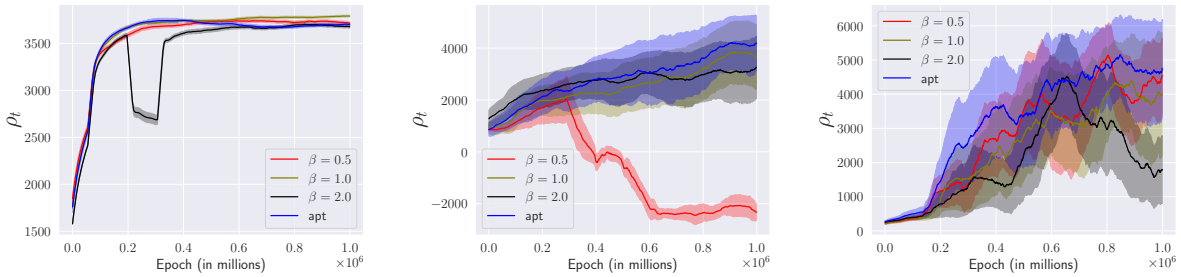


Figure 4: Left: Relative transfer performance, τ_t are shown with corresponding mean similarity scores. Right: Regularization co-efficient, β_t, is shown for all tasks with corresponding mean similarity scores.



(a) Half-Cheetah (b) Ant (c) Humanoid

Figure 5: Ablation study of β parameter in APT-RL: manual tuning of hyperparameter β is shown against APT-RL in the least similar tasks for all three environments.

Future directions may include considering similar concepts for multi-task transfer learning scenarios, as well as benchmarking the performance of various transfer learning algorithms with the help of the transferability measures introduced in this paper.

References

Majid Abdolshah, Hung Le, Thommen Karimpanal George, Sunil Gupta, Santu Rana, and Svetha Venkatesh. A new representation of successor features for transfer across dissimilar environments. In *International Conference on Machine Learning*, pp. 1–9. PMLR, 2021.

Rishabh Agarwal, Marlos C Machado, Pablo Samuel Castro, and Marc G Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. *arXiv preprint arXiv:2101.05265*, 2021.

Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of mdp similarity for transfer in reinforcement learning.

In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4243–4250. IEEE, 2018.

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

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Balázs Csanád Csáji and László Monostori. Value function based reinforcement learning in changing markovian environments. *Journal of Machine Learning Research*, 9(8), 2008.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. *arXiv preprint arXiv:1703.02949*, 2017.

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

Girish Joshi and Girish Chowdhary. Adaptive policy transfer in reinforcement learning. *arXiv preprint arXiv:2105.04699*, 2021.

Manuel Kaspar, Juan D Muñoz Osorio, and Jürgen Bock. Sim2real transfer for reinforcement learning without dynamics randomization. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4383–4388. IEEE, 2020.

Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pp. 143–173. Springer, 2012.

Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 544–551, 2008.

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.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

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

Yunzhe Tao, Sahika Genc, Jonathan Chung, Tao Sun, and Sunil Mallya. Repaint: Knowledge transfer in deep reinforcement learning. *arXiv preprint arXiv:2011.11827*, 2020.

Matthew E Taylor and Peter Stone. Behavior transfer for value-function-based reinforcement learning. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pp. 53–59, 2005.

Matthew E Taylor and Peter Stone. Cross-domain transfer for reinforcement learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 879–886, 2007a.

Matthew E Taylor and Peter Stone. Representation transfer for reinforcement learning. In *AAAI Fall Symposium: Computational Approaches to Representation Change during Learning and Development*, pp. 78–85, 2007b.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Matthew E Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(9), 2007.

Matthew E Taylor, Nicholas K Jong, and Peter Stone. Transferring instances for model-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II 19*, pp. 488–505. Springer, 2008.

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

Amy Zhang, Harsh Satija, and Joelle Pineau. Decoupling dynamics and reward for transfer learning. *arXiv preprint arXiv:1804.10689*, 2018.

Jingwei Zhang, Jost Tobias Springenberg, Joschka Boedecker, and Wolfram Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2371–2378. IEEE, 2017.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*, 2020.