KNOWLEDGE TRANSFER THROUGH VALUE FUNCTION FOR COMPOSITIONAL TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Reinforcement Learning methods are sample inefficient when exploring the environment from scratch. In this work, we introduce an approach of knowledge transfer using the value function combined with curriculum learning, which aims to leverage the learning process by transferring knowledge among progressively increasing task complexity. Our main contribution is demonstrating the effectiveness of this approach by modifying the *degrees of freedom* of the target task, breaking it down into simpler sub-tasks, and leveraging learning by transferring the knowledge along the curriculum steps. We empirically demonstrate the broad possibilities of modifying the degrees of freedom of the target task to leverage learning in classical Reinforcement Learning problems and a real-world control task.

1 INTRODUCTION

Deep Reinforcement Learning (DRL) (Mnih et al., 2013; 2015), which leverages the scalability of Reinforcement Learning (RL) by combining it with Deep Learning, has demonstrated remarkable success in diverse applications ranging from games (Vinyals et al., 2019; Wurman et al., 2022) to control scenarios (Wei et al., 2017; Yang et al., 2018; Degrave et al., 2022). Nonetheless, DRL encounters challenges in sample efficiency when exploring environments from scratch. These challenges arise from various factors, including the lack of reward signals in environments with sparse and delayed feedback (Bellemare et al., 2016), the expansive exploration space driven by the high-dimensionality of state-action pairs, and the complexity in trading off competing objectives, each may demand a diverse skill set to be accomplished (Mataric, 1994; Hayes et al., 2022).

Organizing and structuring the learning process can significantly enhance sample efficiency, reducing the need for exhaustive exploration across the state space. Curriculum Learning (CL) (Elman, 1993; Bengio et al., 2009; Narvekar et al., 2020) draws inspiration from educational curricula by gradually introducing increasingly complex tasks. Though formulating a curriculum of tasks presents challenges and might demand domain-specific knowledge and control over the environment, it can result in resource efficiency as easier tasks facilitate faster convergence. Additionally, this method can strengthen robustness and enhance generalization by systematically encompassing various data distribution aspects in a meaningful sequence rather than arbitrary sampling (Schaul et al., 2016; Soviany et al., 2021).

In the context of RL, the challenges posed by tasks can be eased by adjusting their *degrees of freedom* (Narvekar et al., 2016). This entails modifying factors like the state-action space, reward structure, episode length, and the initial state distribution. For example, converting a continuous action space into a discrete one can facilitate exploration (Farquhar et al., 2020), and aligning actions with specific objectives can expedite task completion (Wang et al., 2023). In scenarios where positive outcomes are only observed upon reaching the goal, linking intermediate actions to their final result becomes challenging. This issue can be mitigated by changing the initial states of episodes, bringing the agent closer to the target states (Dai et al., 2021), or shaping rewards to observe positive outcomes (Ng et al., 1999; Andrychowicz et al., 2017). As such, a policy proficient at a simpler source task has the potential to harness its acquired knowledge to facilitate the learning of a complex target task.

This work introduces an approach for knowledge transfer across tasks with progressive complexity structured within a curriculum framework. While CL can speed up the learning process by breaking down a target task into simpler subtasks through adjustments in its degree of freedom, transfer learning leverages the learning by transferring knowledge acquired in an easier task to harness the learning to a target task. Our method employs the value function, which encodes granular information about task execution, and utilizes the softmax function to facilitate the interleaving of actions between source and target policies. By interleaving actions from source and target policies, the latter can play actions from its estimations, an essential mechanism for self-correction Ostrovski et al. (2021). Furthermore, the method allows the expansion of the artificial neural network architecture (ANN) to accommodate an eventual growth in the search space without requiring mappings from different state-action spaces.

Transfer Learning has been an active research topic in RL (Zhu et al., 2020), taking many forms, such as using logged data from an expert policy (Hussein et al., 2017; Levine et al., 2020) to more high-level knowledge, such as partial policies Sutton et al. (1999). Closely related to our work, in (Taylor et al., 2007), the authors proposed *Transfer via inter-task mapping (TVITM)*, which employed the value function to transfer between tasks with different state and action spaces. Our method differs from previous ones by transferring knowledge without reusing artificial neural network (ANN) weights or demanding state/action mappings for different tasks, allowing the search space to grow organically. We empirically evaluate this approach in classical environments sourced from OpenAI Gym (Brockman et al., 2016) and a control task of pump scheduling for water distribution networks (Donâncio et al., 2022). The outcomes highlight the benefits of this strategy in certain domains, yielding policies with better asymptotic performance compared to policies learned from scratch through conventional exploration techniques.

2 BACKGROUND

2.1 REINFORCEMENT LEARNING

We adopt the framework of episodic Markov Decision Processes (MDPs) (Sutton and Barto, 2018) to model learning tasks, defined as $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \mathcal{T}, \mu)$ where $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow dist(S')$ the transition probability, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ the reward function, $\gamma \in [0, 1]$ the discount factor, $\mathcal{T}$ the horizon length, and μ the initial state distribution. The objective of an agent is to learn a policy $\pi(s|a)$ which maps the state space to the action distribution aiming to maximize returns $r(\pi) = \mathbb{E}_{s\sim\mathcal{P},a\sim\pi} \sum_{t=0}^{\mathcal{T}} \gamma^t r(s_t, a_t)$. This work is independent of the particular value function-based learning algorithm, and for simplicity, we assume Deep Q-Networks (DQN) Mnih et al. (2013), which update the policy π represented by the set of weights θ by minimizing the loss δ:

$$\delta_i(\theta) = \mathbb{E}_{s,a,s',r\sim\mathcal{D}}[r(s, a) + \gamma max_{a'} Q_{\theta^-}(s', a') - Q_\theta(s, a)]^2. \tag{1}$$

In the RL framework, an agent interacts with the environment starting from a state $s_0 \sim \mu$ and at each timestep $t = 0, 1, 2, ..., \mathcal{T}$, takes an action $a \in \mathcal{A}$ given a state $s \in \mathcal{S}$, receiving a reward $r(s, a) \in \mathcal{R}$, and transitioning to a new state $s' \in \mathcal{S}$. DQN achieved remarkable performance by storing these experiences $< s, a, r, s' >$ in the replay memory $\mathcal{D}$ (Lin, 1992) and shuffling them to break their correlation. Furthermore, it also employs a target network with weights θ^- to estimate future returns. The target network usually updates at frequency λ by copying the learning weights θ, a critical component to enhance the stability of the learning process.

2.2 DEGREES OF FREEDOM OF A TARGET TASK

CL hierarchically arranges tasks based on complexity levels. In the context of RL, this can be achieved by breaking down a target task into more straightforward source tasks by modifying the *degrees of freedom* of the former (Narvekar et al., 2016). In (Zhu et al., 2020), the authors shed light on various domain differences that can exist between the source task $\mathcal{M}_s$ and the target task $\mathcal{M}_t$.

The state-space $\mathcal{S}$: the state-space can be modified by constraining it to filter out irrelevant information, focusing on critical aspects. On the contrary, it can also be augmented, enriching the agent's observation by providing additional information about the environment;

The action-space $\mathcal{A}$: the action-space can be constrained to its subspace to reduce the task complexity, for instance, focusing on a meaningful subset of actions for a given subtask;

The reward function $\mathcal{R}$: the reward signals can be manipulated to enhance the learning process for some tasks. That is particularly useful in scenarios with sparse and binary rewards. For example, shaping the reward considering the *distance* to some goal can speed up the observation of positive outcomes;

Transition dynamics $\mathcal{P}$: the dynamics of the interactions can be affected by, for example, the distinct physical properties between the source task and the target task, thus altering the probability distribution of reaching a state s' by applying a given action a on the state s;

The initial state distribution μ: source and target tasks can have different initial state distributions. In this way, the agent could have more chances to experience meaningful outcomes early in the learning process;

The trajectories length $\mathcal{T}$: the task length can differ between source and target tasks, allowing the agent to interact longer with the environment, quickly reaching states that are rare to observe with the standard task length.

3 Related Work

Curriculum Learning (CL) (Elman, 1993; Bengio et al., 2009) systematically organizes the learning process by initially presenting "easier" tasks and gradually increasing the difficulty as the agent's proficiency in these tasks improves. This approach is grounded in the belief that arranging tasks in a meaningful sequence, instead of random sampling, enhances learning effectiveness (Soviany et al., 2021). This spans from manual curriculum design to semi- or fully automated methods. Recent advancements in CL have showcased its efficacy in DRL, improving sample efficiency and enhancing generalization (Portelas et al., 2020). Various strategies can be employed to induce a curriculum, including prioritizing meaningful experiences (Schaul et al., 2016), shaping the reward to provide more reward signals (Ng et al., 1999; Andrychowicz et al., 2017), or exploring unknown states in the environment (Subramanian et al., 2016; Bellemare et al., 2016; Tang et al., 2017; Pathak et al., 2019). Generative Adversarial Networks (GANs) based approaches have recently attained attention due to their capacity to generate curricula automatically. These techniques can entail goal sampling based on the agent's skill level (Sharma et al., 2021), augment reward signals for the learning agent (Campero et al., 2021), or follow a teacher-student goal sampling paradigm (Florensa et al., 2018). Closer to our approach, in (Farquhar et al., 2020; Wang et al., 2023), the authors propose a curriculum to break down a complex task by reducing its action space. That can be achieved, for example, by discretizing a continuous action space, leading to a smaller search space and facilitating exploration.

While CL focuses on structuring and organizing the learning process, knowledge transfer, also known as transfer learning (Taylor and Stone, 2009), aims to leverage knowledge acquired from prior tasks, resulting in resource savings compared to starting from scratch. Transfer learning in RL is an active research area (Zhu et al., 2020), with various forms such as utilizing logged data (Hussein et al., 2017; Hester et al., 2018; Levine et al., 2020), employing a teacher to interleave rollouts with the student policy Dai et al. (2021), using the value function (Taylor et al., 2007), or transferring partial policies (e.g. options) (Sutton et al., 1999; Jinnai et al., 2020). Additionally, agents can have their capabilities enhanced by incrementally expanding the ANN architecture as they tackle new tasks (Rusu et al., 2016b; Yoon et al., 2018), enabling adaptation to novel challenges while retaining previously acquired knowledge (Khetarpal et al., 2022; Abel et al., 2023). Policy distillation-based approaches (Hinton et al., 2014; Rusu et al., 2016a; Teh et al., 2017; Berseth et al., 2018; Czarnecki et al., 2018; Ghosh et al., 2018) can be considered implicit curriculum combined with a transfer learning method that results in a compressed model capable of handling multiple tasks (Guillet et al., 2022). Our work stands apart from previous approaches by modifying the degrees of freedom of the target task and employing the value function for knowledge transfer. In a related context, Self-Imitation via Reduction (SIR) (Li et al., 2021) proposes a compositional approach to solving RL tasks. Their idea involves learning a reduced task and using self-imitation learning to transfer knowledge to the target task.

4 METHOD

This work employs the value function to transfer learning between increasingly complex tasks. The process involves modifying the degrees of freedom within a target task to create one or more easier source tasks, which are subsequently sampled based on their complexity. To structure and organize the learning process, we assume the existence of a curriculum generation method that defines these tasks and establishes a sequential hierarchy of complexity for sampling. Once the tasks are defined and their complexity order established, we employ a transfer learning method that relies on the softmax function. This method utilizes the Q-estimations derived from both the source task ($\mathcal{M}_s$) and the target task ($\mathcal{M}_t$) while introducing a temperature parameter τ to control the entropy between source and target policies when selecting actions. As the curriculum steps unfold, a target task $\mathcal{M}_{t_i}$ becomes the source task $\mathcal{M}_{s_{i+1}}$ for the subsequential step until convergence to a final target task $\mathcal{M}_{t_n}$.

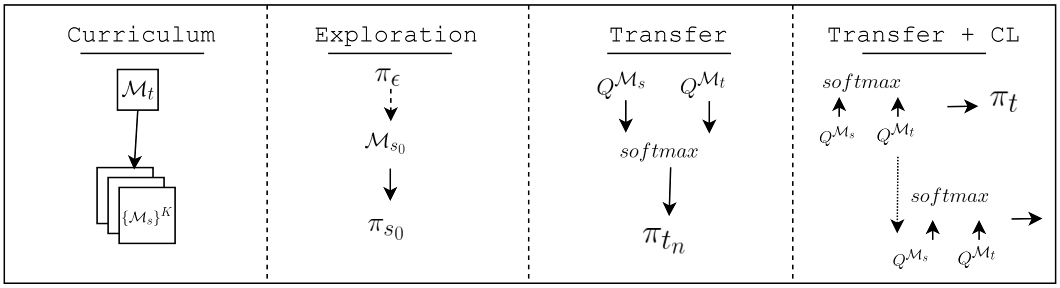


Figure 1: Knowledge transfer steps through a curriculum

4.1 KNOWLEDGE TRANSFER METHOD

We introduce two variations of the knowledge transfer methods tailored to the degrees of freedom between tasks $\mathcal{M}_s$ and $\mathcal{M}_t$, specifically regarding their action space $\mathcal{A}$. In the first scenario, we consider unnormalized Q-estimations, denoted as $Q^{\mathcal{M}_s}$ from task $\mathcal{M}_s$ and $Q^{\mathcal{M}_t}$ from task $\mathcal{M}_t$, where both tasks share an identical action space. When selecting an action during the learning process of the target task, we employ the softmax function having as input the maximum output values from both $Q^{\mathcal{M}_s}$ and $Q^{\mathcal{M}_t}$. The application of softmax results in a probability distribution, which guides the selection of the action with the highest value from the respective Q-estimation:

$$\text{softmax}(\{\max(Q^{\mathcal{M}_s}), \max(Q^{\mathcal{M}_t})\} = \frac{e^{\{max(Q^{\mathcal{M}_s}),max(Q^{\mathcal{M}_t})\}\backslash\tau}}{e^{max(Q^{\mathcal{M}_s})\backslash\tau} + e^{max(Q^{\mathcal{M}_t})\backslash\tau}}$$

$$\text{action } a = \begin{cases} \text{argmax}(Q^{\mathcal{M}_s}), & \text{with probability softmax}_{Q^{\mathcal{M}_s}} \\ \text{argmax}(Q^{\mathcal{M}_t}), & \text{with probability softmax}_{Q^{\mathcal{M}_t}} \end{cases}$$

Figure 1 provides an overview of our knowledge transfer approach. Initially, a target task $\mathcal{M}_t$ into K source subtasks $\mathcal{M}_s$, which are sampled based on complexity. For the sake of simplicity, we focus on one pair of $< \mathcal{M}_s, \mathcal{M}_t >$ at a time, although we can use a set of source policies $\{\pi_{s_i}\}_{i=1}^{K}$ to leverage the learning to a target policy π_t. In the first curriculum step, where no source policy is available, we employ an exploration strategy, such as ϵ-greedy, to learn a policy π_{s_0}. We use this policy π_{s_0} to interleave actions with a target policy learning how to perform the subsequent task $\mathcal{M}_{t_0}$[1]. To transfer knowledge between two policies, a *softmax* function is employed, combining Q-estimations from source $max(Q^{\mathcal{M}_s})$ and target $max(Q^{\mathcal{M}_t})$ tasks. Once the state space between the tasks can be different, we need a function $s_{\mathcal{M}_s}(\phi) = h(s_{\mathcal{M}_t}(\phi))$ which contracts or expands the current observations $s_{\mathcal{M}_t}(\phi)$ to match the previously learned state space by π_s.

[1] It's noteworthy that we can use $\mathcal{M}s_1$ and $\mathcal{M}t_0$ interchangeably

In the following, a policy that has mastered a target task $\mathcal{M}_{t_i}$ becomes the source task $\mathcal{M}_{s_{i+1}}$ for the subsequent target task $\mathcal{M}_{t_{i+1}}$, and this process continues until we have unfold all the curriculum steps, ultimately converging to a final target task $\mathcal{M}_{t_n}$. As the complexity of the learning tasks increases, and the search space between two curriculum steps may expand, a necessity arises to augment the artificial neural network (ANN) architecture to accommodate the knowledge requirements. This expansion ensures that the network can effectively absorb and apply the additional knowledge required to master the more intricate tasks.

However, we may have a set of tasks learned simultaneously for a single curriculum step. For that, our second case considers a set of policies $\{\pi_{s_j}\}_{j=1}^{K}$ transferring knowledge to a target policy π_t being the action space $\mathcal{A}(\mathcal{M}_{s_j}) \neq \mathcal{A}(\mathcal{M}_{s_{l \neq j}})$ and $\mathcal{A}(\mathcal{M}_{s_1}) \cup \mathcal{A}(\mathcal{M}_{s_2}) \cup ... \cup \mathcal{A}(\mathcal{M}_{s_K}) = \mathcal{A}(\mathcal{M}_t)$. In other words, the action spaces of two source tasks can be different, but the set of source tasks has the same action spaces as the target task, such that $\mathcal{A}(\mathcal{M}_{s_j}\}_{j=1}^{K}) = \mathcal{A}(\mathcal{M}_t)$. Thus, Q-estimations for overlapping actions are averaged, while those for distinct actions are combined to obtain $Q^{\mathcal{M}_s}$. Finally, we select actions using the probability distribution outputted by the softmax$(\{\max(Q^{\mathcal{M}_s}), \max(Q^{\mathcal{M}_t})\})$.

By employing the value function, which estimates the *quality* of state-action pairs, we can transfer low-level information, which carries out more granular information regarding how to perform a task Taylor et al. (2007). However, the value function can be biased, leading to overestimation when learning passively. By interleaving actions sample from $Q^{\mathcal{M}_s}$ and $Q^{\mathcal{M}_t}$ allows the target task policy to explore its underlying estimations, an essential mechanism for self-correcting (van Hasselt et al., 2016; Ostrovski et al., 2021). Finally, the temperature parameter τ can lead to distinct entropy between the policies $\pi_{\mathcal{M}_s}$ and $\pi_{\mathcal{M}_t}$ during the transfer learning process. In the following section, we delve into the impact of reward function normalization on this entropy.

4.2 The Reward Normalization Matters

The normalization of the reward function depends on the task. Typically, the scale varies using a normalization $[0, 1]$, where the policy tries to maximize the cumulative rewards, and $[-1, 1]$, where the goal is to minimize negative outcomes and maximize positive ones. In some tasks, normalizing the reward using negative values can be valuable to facilitate interpretation by the learning algorithm, highlighting penalized actions. However, if most of the rewards observed during an episode's rollout are negatives, the Q-estimations for this task would also inherently output negative values.

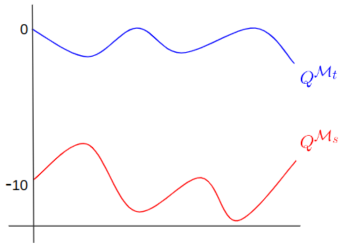


Figure 2: Q-values distribution from source $\mathcal{M}_s$ and target $\mathcal{M}_t$ tasks.

Since our approach relies on the magnitude of the value function to sample actions, the normalization scale directly impacts the entropy between source and target policy. As illustrated in Figure 2, a source policy that mostly observes negative rewards, even though it performs the task well, has lower Q-estimations than a target policy started with random weights without any training. Thus, most actions in the early training come from this suboptimal target policy, hampering the overall transfer learning performance. To mitigate this problem, we propose to combine estimates $Q^{\mathcal{M}_s}$ and $Q^{\mathcal{M}_t}$ when updating estimates of the former while learning a target task by modifying the Equation 1:

$$\delta_i = \mathbb{E}_{s,a,s',r \sim \mathcal{D}}[r(s,a) + \gamma max_{a'}(\alpha Q^{\mathcal{M}_t}(s',a') + \beta Q^{\mathcal{M}_s}(s',a')) - Q^{\mathcal{M}_t}(s,a)]^2.$$

Thus, when predicting Q-values for the future state, the estimates from the source and target tasks are combined such that $Q(s', a') = \alpha Q^{\mathcal{M}_t}(s', a') + \beta Q^{\mathcal{M}_s}(s', a')$, being $\alpha + \beta = 1$ and $\alpha > \beta = 1$. This linear combination leverages the knowledge transfer process since the $Q^{\mathcal{M}_t}$ estimations have a dependency with values $Q^{\mathcal{M}_s}$, which can be seen as a form of reward shaping.

5 EXPERIMENTS

5.1 ENVIRONMENTS

To demonstrate the effectiveness of our approach, we conducted experiments on classical reinforcement learning (RL) problems sourced from the OpenAI Gym library (Brockman et al., 2016) as well as a real-world control task centered on optimizing pump scheduling within a water distribution system (Donâncio et al., 2022). In the following, we outline the specific adjustments made to the degrees of freedom for each task, with further comprehensive details about the environments available in section A:

Mountain Car: to facilitate the learning of the source task, we adjust the degree of freedom regarding the transition probability $\mathcal{P}$ by manipulating the task's physical properties concerning friction. This modification enables the agent to achieve higher speeds rapidly. Additionally, we extended the episode length $\mathcal{T}$ from 200 to 5000 timesteps to facilitate exploration. During the exploration, the probability ϵ of taking random actions is gradually reduced from 1 to 0.01, decaying every time the goal is reached.

Taxi Cab: in our approach to the Taxi Cab environment, we introduce a reward shaping mechanism that rewards the agent with +10 for passengers' successful pickup and drop-off during the source task. Furthermore, we introduce a bonus to the reward at each timestep, considering the grid distance required to achieve these subgoals. The exploration parameter ϵ gradually decreases using a decay factor considering the number of episodes, initially set at 1 and eventually reaching 0.2.

Frozen Lake: In the Frozen Lake scenario, our approach begins by initiating the source task with a modified initial state that situates the agent closer to the goal state. Also, mirroring the Mountain Car strategy, we implement an ϵ-greedy decay schedule that reduces the exploration rate ϵ each time the agent successfully reaches the goal. As ϵ decreases and eventually falls below a predefined threshold, we transition back to utilizing the standard initial state while continuing to learn the source task.

Pump Scheduling: The reward function proposed for pump scheduling encapsulates three subgoals: (i) diminishing electricity consumption of pump operation; (ii) the maintenance of safe water tank levels; and (iii) discouraging frequent pump switches (`ON`/`OFF`) while distributing their usage. The task involves managing four distinct pumps with varying supply capacities and electricity consumption profiles. Moreover, a tank with a 10m length is used for water storage. To tackle this problem, we devise a curriculum-based approach divided into three sequential steps. Initially, we train individual policies, focusing on controlling a single pump with a reduced state space, limited to observations related to tank levels. Subsequently, we transfer the acquired knowledge to the subsequent task, which introduces observations related to pump operation, including running time and switches. The final step entails combining these distinct individual policies into a comprehensive one capable of managing all four pumps while incorporating observations about their operation.

Our objective in proposing these curricula is to establish a proof-of-concept that focuses not on identifying the one with the highest learning enhancement but rather on demonstrating the efficacy of transfer learning by modifying various aspects of a target task's degrees of freedom.

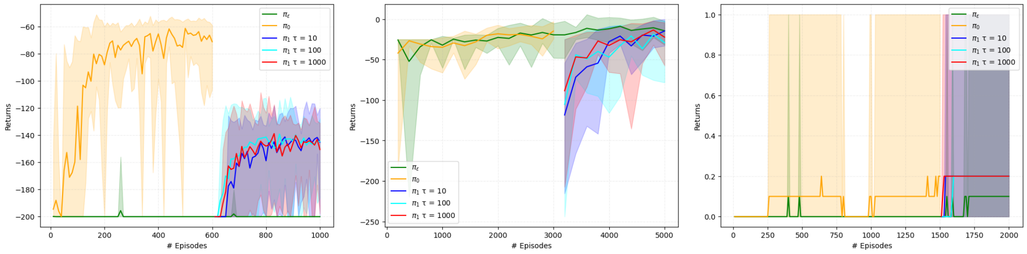
5.2 RESULTS

In Figure 3, we present the results obtained from experiments conducted using OpenAI's scenarios. The depicted outcomes illustrate the average, maximum, and minimum returns across 10 runs for policies trained on the source task (π_0) with modified degrees of freedom, policies tackling the target task from scratch (π_ϵ) employing ϵ-greedy exploration, and policies addressing the target task via transfer learning (π_1) while varying the temperature parameter τ. Notably, the MDP employed throughout each curriculum step for learning and evaluation remains largely consistent, except for the reward function, which aligns with those of the target task to establish a grounded comparison.

Our outcomes when learning from scratch in the Mountain Car scenario align with findings reported in (Farquhar et al., 2020). Specifically, the use of ϵ-greedy exploration in the Mountain Car task often struggles to achieve the goal due to the limited exposure to positive outcomes. However, our curriculum-based approach significantly enhances the learning process for the source task, as evident in Figure 3(a). Subsequently, the transfer learning process retrieves that performance being the discrepancy in cumulative returns due to different physical settings between the source and target task.

As depicted in Figure 3(b), our curriculum approach applied to the Taxi Cab problem demonstrates limited learning enhancement for the source task. We attributed this to the intrinsic exploration challenges posed by the task itself. Consequently, when transitioning from this suboptimal policy obtained in the source task, the transfer learning method fails to surpass the performance achieved through exploration from scratch. It's worth noting that we don't view this outcome as a limitation of our approach but rather as an indicator of the importance of well-designed source tasks to expedite the learning process.

The results for the Frozen Lake scenario are depicted in Figure 3(c), showing an "dip" in performance in the range between 750∼1250 episodes resulting from the modification in the initial state during the source task learning. However, after a few subsequent episodes, the agent could once again attain the goal, starting from a state identical to that of the target task. Remarkably, our knowledge transfer approach once more effectively leverages the target task's learning process, resulting in superior performance compared to the target task learned from scratch utilizing ϵ-greedy exploration.



(a) Mountain Car (b) Taxi Cab (c) Frozen Lake

Figure 3: OpenAI Gym results

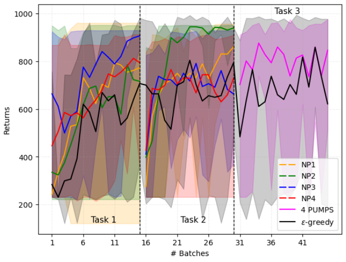


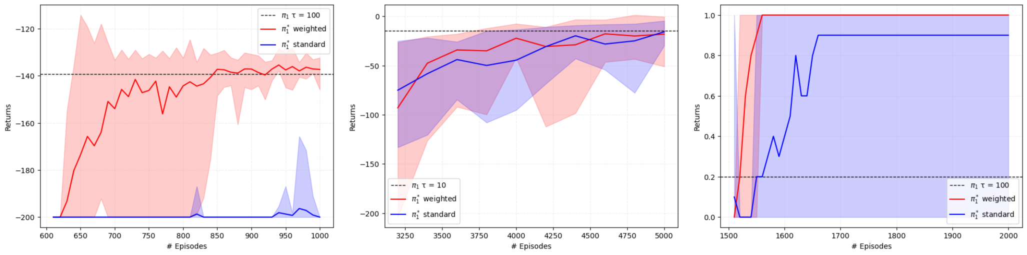
Figure 4: Pump scheduling results

In Figure 4, we compare the results for the pump scheduling problem with policies learned using ϵ-greedy exploration with an ANN architecture identical to the curriculum's final step. We split the three-year dataset regarding water consumption into two years for the learning process and one for policy evaluation. This water consumption data serves as input at each timestep for the water distribution simulator once other states' features are fully transition-dependent. To address fairness regarding samples, we divide the training in "batches" once our curriculum approach learns four individual policies during Task 1 ($\mathcal{M}_1$) and Task 2 ($\mathcal{M}_2$). As such, each batch of policies with action space $\mathcal{A} = \{\text{NOP}, \text{NP\#}\}$ has a single episode, while for ϵ-greedy and Task 3 ($\mathcal{M}_3$) contains four episodes. We employ a reset mechanism for each episode rollout, having the initial tank levels defined by the logged data, matching the water consumption at t_0. Finally, ϵ decays from $1 \to 0.1$ with a discount factor of 0.95 for $\mathcal{M}_1$ and 0.85 for ϵ-greedy exploration.

The results display the average, maximum, and minimum returns for policy performance after updates using each batch. We have observed that policies that control single pumps (Task 1 and 2) demonstrate better jumpstart performance and lower variance than the full action space learned through ϵ-greedy. Additionally, although a full-action space (Task 3) policy achieved a lower average return than some single-pump policies, which could be explained by the reduced search space, it holds the highest peak for maximum return. Finally, for Task 3, our approach attained a maximum average return peak with lower variance than a policy learned from scratch through exploration.

5.3 Ablation Analyses of Q-weighted Combination

We performed an ablation study to assess the effectiveness of the Q-weighted sum in composing $Q(s', a')$. We employ a single source policy π_0^* that exhibited the highest cumulative reward over the learning process of $\mathcal{M}_s$ to transfer learning to target policies, one employing the Q-weighted sum (red line) and the other without (blue line). Furthermore, we set fixed values for $\alpha = 0.7$ and $\beta = 0.3$ to assess the method's robustness across all scenarios. Lastly, the black dashed line (π_1) works as a baseline representing the maximum average peak observed in the previous experiment employing the same temperature value τ, transferring using any source policy π_0.



(a) Mountain Car

(b) Taxi Cab

(c) Frozen Lake

Figure 5: Ablation analyses of Q-weighted sum.

The results in Figure 5 show that the Q-weighted sum can be the cornerstone to effective knowledge transfer in scenarios such as Mountain Car, where most observed rewards are negative. Indeed, our transfer learning strategy can work in two manifolds: interleaving actions from source and target policies while shaping the reward by transferring Q-estimations. In addition, the source policy's performance can substantially impact leveraging the learning process. As seen in the results for the Frozen Lake (see Figure 5(c), our approach could retrieve the optimality of the source policy π_0^* which its final step has the same settings as the target task, in very few episodes.

6 Conclusions

This work introduces a knowledge transfer approach to effectively leverage the learning process for solving challenging reinforcement learning tasks. The fundamental idea revolves around knowledge transfer through a curriculum of tasks that systematically increase in complexity. We assume a curricula generator method responsible for reducing target task complexity by manipulating its degrees of freedom and hierarchically organizing the derivated simpler sub-task(s). As the curriculum steps unfold, we incorporate the softmax function with a parameter τ, which employs the value function for knowledge transfer across these tasks. Given that our approach relies on the magnitude of value function estimations for action selection, we also present a solution to address the challenge of reward range normalization. We conduct extensive evaluations across various classical reinforcement learning environments and control tasks to empirically validate our approach. These experiments serve as a proof-of-concept by modifying the degrees of freedom in different target tasks, and our results provide compelling evidence of the effectiveness of our approach across four distinct environments.

References

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado van Hasselt, and Satinder Singh. A definition of continual reinforcement learning, 2023. URL `https://arxiv.org/abs/2307.11046`.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/file/afda332245e2af431fb7b672a68b659d-Paper.pdf.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA, 2009. Association for Computing Machinery. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Glen Berseth, Cheng Xie, Paul Cernek, and Michiel van de Panne. Progressive reinforcement learning with distillation for multi-skilled motion control. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

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

Andres Campero, Roberta Raileanu, Heinrich Küttler, Joshua B. Tenenbaum, Tim Rocktäschel, and Edward Grefenstette. Learning with amigo: Adversarially motivated intrinsic goals. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

Wojciech Czarnecki, Siddhant Jayakumar, Max Jaderberg, Leonard Hasenclever, Yee Whye Teh, Nicolas Heess, Simon Osindero, and Razvan Pascanu. Mix & match agent curricula for reinforcement learning. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1087–1095. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/czarnecki18a.html.

Siyu Dai, Andreas G. Hofmann, and Brian Williams. Automatic curricula via expert demonstrations. *CoRR*, abs/2106.09159, 2021. URL https://arxiv.org/abs/2106.09159.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Thomas G. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Int. Res.*, 13(1):227–303, nov 2000. ISSN 1076-9757.

Henrique Donâncio, Laurent Vercouter, and Harald Roclawski. The pump scheduling problem: A real-world scenario for reinforcement learning. *CoRR*, abs/2210.11111, 2022. doi: 10.48550/ARXIV.2210.11111. URL https://arxiv.org/abs/2210.11111.

Jeffrey L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99, 1993. ISSN 0010-0277. doi: https://doi.org/10.1016/0010-0277(93)90058-4. URL https://www.sciencedirect.com/science/article/pii/0010027793900584.

Gregory Farquhar, Laura Gustafson, Zeming Lin, Shimon Whiteson, Nicolas Usunier, and Gabriel Synnaeve. Growing action spaces. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3040–3051. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/farquhar20a.html.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1515–1528. PMLR, 10–15 Jul 2018.

Dibya Ghosh, Avi Singh, Aravind Rajeswaran, Vikash Kumar, and Sergey Levine. Divide-and-conquer reinforcement learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Valentin Guillet, Dennis Wilson, Carlos Aguilar-Melchor, and Emmanuel Rachelson. On neural consolidation for transfer in reinforcement learning. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 867–874. IEEE, 2022.

Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), apr 2022. ISSN 1387-2532. URL `https://doi.org/10.1007/s10458-022-09552-y`.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. URL `https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16976`.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2014. URL `https://arxiv.org/abs/1503.02531`.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), apr 2017. ISSN 0360-0300. doi: 10.1145/3054912. URL `https://doi.org/10.1145/3054912`.

Yuu Jinnai, Jee Won Park, Marlos C. Machado, and George Dimitri Konidaris. Exploration in reinforcement learning with deep covering options. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL `https://arxiv.org/abs/2005.01643`.

Yunfei Li, Yilin Wu, Huazhe Xu, Xiaolong Wang, and Yi Wu. Solving compositional reinforcement learning problems via task reduction. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

Long Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.*, 8:293–321, 1992.

Maja J Mataric. Reward functions for accelerated learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 181–189. Morgan Kaufmann, San Francisco (CA), 1994. ISBN 978-1-55860-335-6. doi: https://doi.org/10.1016/B978-1-55860-335-6.50030-1. URL `https://www.sciencedirect.com/science/article/pii/B9781558603356500301`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

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

Andrew Moore. Efficient memory-based learning for robot control. Technical report, Carnegie Mellon University, Pittsburgh, PA, November 1990.

Sanmit Narvekar, Jivko Sinapov, Matteo Leonetti, and Peter Stone. Source task creation for curriculum learning. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, page 566–574, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450342391.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E. Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020. URL `http://jmlr.org/papers/v21/20-212.html`.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1558606122.

Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. The difficulty of passive learning in deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL `https://papers.nips.cc/paper/2021/file/c3e0c62ee91db8dc7382bde7419bb573-Paper.pdf`.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5062–5071. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/pathak19a.html`.

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4819–4825. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/671. URL `https://doi.org/10.24963/ijcai.2020/671`. Survey track.

Andrei A. Rusu, Sergio Gomez Colmenarejo, Çaglar Gülçehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016a.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016b. URL `http://arxiv.org/abs/1606.04671`.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Archit Sharma, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Autonomous reinforcement learning via subgoal curricula. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18474–18486. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/99c83c904d0d64fbef50d919a5c66a80-Paper.pdf`.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. Curriculum learning: A survey. *CoRR*, 2021. URL `https://arxiv.org/abs/2101.10382`.

Kaushik Subramanian, Charles L. Isbell, and Andrea L. Thomaz. Exploration from demonstration for interactive reinforcement learning. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, page 447–456, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(99)00052-1. URL https://www.sciencedirect.com/science/article/pii/S0004370299000521.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3a20f62a0af1aa152670bab3c602feed-Paper.pdf.

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009. ISSN 1532-4435.

Matthew E. Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.

Yee Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/0abdc563a06105aee3c6136871c9f4d1-Paper.pdf.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2016. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

S Wang, R Papallas, M Leonetti, and M Dogar. Goal-conditioned action space reduction for deformable object manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023. URL https://eprints.whiterose.ac.uk/197058/.

Tianshu Wei, Yanzhi Wang, and Qi Zhu. Deep reinforcement learning for building hvac control. In *Proceedings of the 54th Annual Design Automation Conference 2017*, DAC '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349277. doi: 10.1145/3061639.3062224. URL https://doi.org/10.1145/3061639.3062224.

Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896): 223–228, 2022.

Yaodong Yang, Jianye Hao, Mingyang Sun, Zan Wang, Changjie Fan, and Goran Strbac. Recurrent deep multiagent q-learning for autonomous brokers in smart grid. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 569–575. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/79. URL https://doi.org/10.24963/ijcai.2018/79.

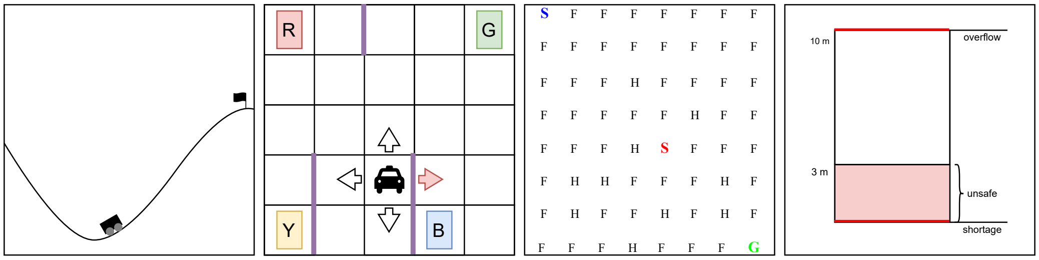
Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *CoRR*, abs/2009.07888, 2020. URL https://arxiv.org/abs/2009.07888.

Appendix

A Further environments descriptions

In this section, we provide a comprehensive overview of each stage within the curricula that were designed to assess the effectiveness of our knowledge transfer approach.



S	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F
F	F	F	H	F	F	F	F
F	F	F	F	F	H	F	F
F	F	F	H	S	F	F	F
F	H	H	F	F	F	H	F
F	H	F	F	H	F	H	F
F	F	F	H	F	F	F	G

(a) Mountain Car (b) Taxi Cab (c) Frozen Lake (d) Pump Scheduling

Figure 6: Environments.

A.1 Mountain Car

The Mountain Car scenario Moore (1990) presents a challenging task where the agent must navigate a valley to reach the "top of the right hill" (see Figure 6(a)). With discrete actions for acceleration (right, none, left), the agent faces limited positive reward observations and the need to balance accelerations to gain momentum toward reaching the goal. Each episode typically lasts for 200 timesteps.

In this scenario, the absence of positive rewards leads to challenges in achieving optimal performance, and their lack during early exploration phases may result in convergence to suboptimal actions. To address this, we propose breaking down the task through a curriculum. Initially, the agent learns the task with a modified friction force, enabling faster speed gain. Moreover, we extend the episode duration by removing the reset mechanism until policy updates, allowing continuous interaction with the environment for up to 5000 timesteps. Using this strategy, we aim to leverage knowledge acquired in the modified setting to improve performance when transitioning to the standard task with original physics and episode length.

A.2 Taxi Cab

The Taxi Cab scenario (Dietterich, 2000) involves picking up and dropping off passengers at specified locations R, Y, G, B in Figure 6(b). The agent navigates a grid environment with actions (north, south, west, east, pickup, dropoff) while avoiding internal walls (purple lines). For each illegal pickup or dropoff action, the agent receives a negative reward of -10; otherwise, a positive reward of +20 if the passenger is delivered to the correct location. The agent also gets a -1 penalty for any state-action pair that does not overlap previous rewards. During the exploration, we allow the agent to perform up to 100 actions while evaluating up to 25.

A significant learning challenge arises due to the delayed positive rewards, which only occur after successful dropoffs. Consequently, the agent may struggle to grasp the importance of passenger pickups. To address this issue, we propose a curriculum where the source task uses a modified reward function, receiving a positive reward of +10 for picking up a passenger and an additional +10 for successfully dropping off the passenger at the correct location. Furthermore, to guide the agent efficiently, we introduce a reward bonus based on the distance d to the current goal (pickup or dropoff). For cardinal direction actions (north, south, west, east), the reward is determined by $r = -1 + 1/(d + 1)$. This modification allows the agent to recognize the significance of passenger pickups as part of the task, while the reward bonus based on distance aids navigation toward each subtask. Subsequently, we transfer the learned policy with modified rewards to a policy learning task using the standard reward function.

A.3 Frozen Lake

The Frozen Lake scenario involves guiding the agent from the starting point (S) in the top left corner of a grid to the ending point (G) on the right bottom corner, avoiding falling in holes (H). Each movement action (north, south, west, east) yields a 0 reward while reaching the objective provides a +1 reward. The challenge lies in the absence of positive reward, necessitating the agent to explore and discover a sequence of actions that lead to successful goal-reaching while avoiding premature episode endings caused by falling into holes.

We introduce a curriculum where we initially learn a source task by penalizing premature episode endings through a negative reward of -1 for landing in a hole (H). Additionally, we implement a modification to the initial state, relocating the agent from the top left corner (S) to a more centrally situated state (S), as illustrated in Figure 6(c). As the agent observes positive rewards upon reaching the goal, the likelihood ϵ for random actions diminishes over time. We use the standard initial state once the ϵ value becomes ≤ 0.36 while still learning the source task. Subsequently, we transfer the acquired knowledge to a target task featuring the standard initial state configuration. By manipulating the initial state, we aim to bring the agent closer to the goal state, reducing the need for exploration by facilitating the observation of positive rewards.

A.4 Pump Scheduling

In (Donâncio et al., 2022), the authors introduce the pump scheduling scenario and its representation as a Partially Observable MDP (POMDP). This control optimization problem relies on minimizing the energy consumption of a set of pumps while meeting the safety constraints of water reservoirs. In this scenario, four pumps (NP1 to NP4) pump water into the system. These pumps have different sizes and electricity consumption. Moreover, a tank with a 10m length is used for water storage with constraints depicted in Figure 6(d). The proposed reward function for the pump scheduling scenario is given by:

$$r_t = -e^{(-1/kW_t)} - B * \psi + \log(1/(P + \omega)),$$

where the term $-e^{(-1/kW_t)}$ penalizes the electricity consumption (kW) associated with the operation of pumps $\{(\text{NP\#})\}_{\#=1}^{4}$ at timestep t. Secondly, $B * \psi$ penalizes the agent when the tank levels fail to meet safety constraints. Specifically, safety operation necessitates maintaining the tank's water level at a minimum of 3m. The term $\log(1/(P + \omega))$ aims to lead to an action distribution and avoid pump switches once it decreases the asset's lifetime. Here, P is a non-markovian feature representing the cumulative time the current action has been applied to the system throughout the episode. Lastly, the variable ω is a penalty for action switches, encouraging the maintenance of more continuous pump operation.

To trade off these sub-goals, the agent needs to obtain information about the environment through states/observations $s \in \mathcal{S}$ defined by the following set of features ϕ, where:

- T is the tank level at some timestep t;
- O is the sensor's data regarding water consumption for some timestep t;
- Γ is time of day;
- X is the last action performed;
- P is the cumulative time that the current action has been performed along the episode;

Once an observation is triggered, the agent has to decide between choosing (or keeping) one of the pumps running or turning (or keeping) it off. At each timestep, only one of the pumps can be running, or none of them (NOP). Thus, the action space is such that $\mathcal{A}$ = {NP1, NP2, NP3, NP4, NOP}.

Our approach to addressing the pump scheduling problem follows an incremental strategy, introducing sub-goals one by one, progressively expanding the source task to encompass the target task. To achieve this, we first establish a task where a set of policies is individually trained to control a single pump while meeting the tank level constraints. In this initial task $\mathcal{M}_1$, the observation space is defined as $\mathcal{S} = \phi(T, O)$, while the action space consists of $\mathcal{A} = \{\text{NOP}, \text{NP\#}\}$.

Once the task $\mathcal{M}_1$ is *mastered*, we transfer the knowledge to the target task $\mathcal{M}_2$, where observations extend to the full state representation $s_t = \phi(T_t, O_t, \Gamma_t, X_t, P_t)$. The next step involves combining these source tasks from $\mathcal{M}_2$ into a final target task. To achieve this, we average the Q-values for each NOP action once each policy maps its estimation. We then combine the estimations for each action using different pumps, resulting in the estimation $Q^{\mathcal{M}_s}$:

$$Q^{\mathcal{M}_s} = \{avg(Q(s, \text{NOP})), Q(s, \text{NP1}), Q(s, \text{NP2}), Q(s, \text{NP3}), Q(s, \text{NP4})\},$$

Finally, the softmax using $Q^{\mathcal{M}_s}$ and $Q^{\mathcal{M}_t}$ is used to output actions while learning the last curriculum step $\mathcal{M}_3$. This enables the agent to effectively leverage the knowledge acquired from the policies controlling individual pumps to address the complete pump scheduling problem. Along with these curriculum steps, the network architecture is adapted to accommodate the increase in task complexity, as shown in the implementation and training details B.

B Implementation and training details

B.1 Learning algorithm

We employ Double DQN (DDQN) (van Hasselt et al., 2016) as a learning algorithm in this work. DDQN mitigates the overestimation problem for under-represented state-action pairs by selecting the action using the current Q-estimations but updates it using the target net θ^- values:

$$\delta_i(\theta) = \mathbb{E}_{s,a,s',r\sim\mathcal{D}} \left[r(s,a) + \gamma Q_{\theta^-}\left(s', \arg\max_{a'} Q_\theta(s', a')\right) - Q_\theta(s,a) \right]^2. \tag{2}$$

Thus, we employ the linear combination $Q_{\theta^-} = \alpha Q_{\theta^-}^{\mathcal{M}_t} + \beta Q_{\theta}^{\mathcal{M}_s}$ using learning weights θ^- from the task $\mathcal{M}_t$ and consolidate ones θ from the task $\mathcal{M}_s$, to update the estimations $Q^{\mathcal{M}_t}$.

B.2 ANN architecure

Mountain Car: The neural network architecture for this task consists of dense layers (64, 64) with ReLU activation functions. The training utilizes an Adam optimizer with a learning rate α of 0.001, a minibatch size of 64, target network updates every 100 samples and a discount factor γ of 0.99.

Taxi Cab: For the Taxi Cab problem, the neural network architecture includes an embedding layer followed by dense layers (50, 50, 50) using ReLU activation functions. Training employs an Adam optimizer with a learning rate α of 0.001, a minibatch size of 32, target network updates every 500 samples, and a discount factor γ of 0.99.

Frozen Lake: In the case of Frozen Lake, the neural network architecture features an embedding layer followed by dense layers (128, 64) with ReLU activation functions. The training process uses an Adam optimizer with a learning rate α of 0.001, a minibatch size of 64, target network updates every 100 samples and a discount factor γ of 0.99.

Pump Scheduling: The ANN architecture in the pump scheduling task varies in size across different curriculum steps. The architecture consists of layers (LSTM, Dense, Dense), and all dense layers have the ReLU activation function. The same architecture is employed for the policy learned using ϵ-greedy from scratch and the task $\mathcal{M}_3$. The training uses an Adam optimizer with a learning rate α of 0.00025, a minibatch size of 32, target network updates every 300 samples, L_2 regularization (dense layers) of 0.000001, ϵ-greedy policy interpolation ranging from 1 to 0.1, and a discount factor γ of 0.99. The expansion of the ANN architecture's hidden nodes for each task is illustrated in Table 1.

Actions $\mathcal{A}$	Task $\mathcal{M}_1$	Task $\mathcal{M}_2$	Task $\mathcal{M}_3$
(NOP, NP1)	(25, 25, 25)	(75, 75, 75)	–
(NOP, NP2)	(25, 25, 25)	(75, 75, 75)	–
(NOP, NP3)	(50, 50, 50)	(75, 75, 75)	–
(NOP, NP4)	(75, 75, 75)	(75, 75, 75)	–
Full action space	–	–	(125, 125, 125)

Table 1: ANN architecture progress for the pump scheduling problem