Lipschitz Lifelong Reinforcement Learning

Erwan Lecarpentier[1, 2], **David Abel**[3], **Kavosh Asadi**[3, 4*], **Yuu Jinnai**[3],
Emmanuel Rachelson[1], **Michael L. Littman**[3]
[1]ISAE-SUPAERO, Université de Toulouse, France
[2]ONERA, The French Aerospace Lab, Toulouse, France
[3]Brown University, Providence, Rhode Island, USA
[4]Amazon Web Service, Palo Alto, California, USA

Abstract

We consider the problem of knowledge transfer when an agent is facing a series of Reinforcement Learning (RL) tasks. We introduce a novel metric between Markov Decision Processes and establish that close MDPs have close optimal value functions. Formally, the optimal value functions are Lipschitz continuous with respect to the tasks space. These theoretical results lead us to a value-transfer method for Lifelong RL, which we use to build a PAC-MDP algorithm with improved convergence rate. Further, we show the method to experience no negative transfer with high probability. We illustrate the benefits of the method in Lifelong RL experiments.

1 Introduction

Lifelong Reinforcement Learning (RL) is an online problem where an agent faces a series of RL tasks, drawn sequentially. Transferring knowledge from prior experience to speed up the resolution of new tasks is a key question in that setting (Lazaric 2012; Taylor and Stone 2009). We elaborate on the intuitive idea that *similar* tasks should allow a large amount of transfer. An agent able to compute online a similarity measure between source tasks and the current target task could be able to perform transfer accordingly. By measuring the amount of inter-task similarity, we design a novel method for value transfer, practically deployable in the online Lifelong RL setting. Specifically, we introduce a metric between MDPs and prove that the optimal Q-value function is Lipschitz continuous with respect to the MDP space. This property makes it possible to compute a provable upper bound on the optimal Q-value function of an unknown target task, given the learned optimal Q-value function of a source task. Knowing this upper bound accelerates the convergence of an RMax-like algorithm (Brafman and Tennenholtz 2002), relying on an optimistic estimate of the optimal Q-value function. Overall, the proposed transfer method consists of computing online the distance between source and target tasks, deducing the upper bound on the optimal Q value function of the source task and using this bound to accelerate learning. Importantly, the method exhibits no negative transfer, *i.e.*, it cannot cause performance degradation, as the computed upper bound provably does not underestimate the optimal Q-value function.

Our contributions are as follows. First, we study theoretically the Lipschitz continuity of the optimal Q-value function in the task space by introducing a metric between MDPs (Section 3). Then, we use this continuity property to propose a value-transfer method based on a local distance between MDPs (Section 4). Full knowledge of both MDPs is not required and the transfer is non-negative, which makes the method applicable online and safe. In Section 4.3, we build a PAC-MDP algorithm called *Lipschitz RMax*, applying this transfer method in the online Lifelong RL setting. We provide sample and computational complexity bounds and showcase the algorithm in Lifelong RL experiments (Section 5).

2 Background and related work

Reinforcement Learning (RL) (Sutton and Barto 2018) is a framework for sequential decision making. The problem is typically modeled as a Markov Decision Process (MDP) (Puterman 2014) consisting of a 4-tuple $\langle \mathcal{S}, \mathcal{A}, R, T \rangle$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ an action space, R_s^a is the expected reward of taking action a in state s and $T_{ss'}^a$ is the transition probability of reaching state s' when taking action a in state s. Without loss of generality, we assume $R_s^a \in [0, 1]$. Given a discount factor $\gamma \in [0, 1)$, the expected cumulative return $\sum_t \gamma^t R_{s_t}^{a_t}$ obtained along a trajectory starting with state s and action a using policy π in MDP M is denoted by $Q_M^\pi(s, a)$ and called the Q-function. The optimal Q-function Q_M^* is the highest attainable expected return from s, a and $V_M^*(s) = \max_{a \in \mathcal{A}} Q_M^*(s, a)$ is the optimal value function in s. Notice that $R_s^a \leq 1$ implies $Q_M^*(s, a) \leq \frac{1}{1-\gamma}$ for all $s, a \in \mathcal{S} \times \mathcal{A}$. This maximum upper bound is used by the RMax algorithm as an optimistic initialization of the learned Q function. A key point to reduce the sample complexity of this algorithm is to benefit from a tighter upper bound, which is the purpose of our transfer method.

Lifelong RL (Silver, Yang, and Li 2013; Brunskill and Li 2014) is the problem of experiencing online a series of MDPs drawn from an unknown distribution. Each time an MDP is sampled, a classical RL problem takes place where the agent is able to interact with the environment to maximize its expected return. In this setting, it is reasonable to think that knowledge gained on previous MDPs could be re-used

[*] Kavosh Asadi finished working on this project before joining Amazon.

to improve the performance in new MDPs. In this paper, we provide a novel method for such transfer by characterizing the way the optimal Q-function can evolve across tasks. As commonly done (Wilson et al. 2007; Brunskill and Li 2014; Abel et al. 2018), we restrict the scope of the study to the case where sampled MDPs share the same state-action space $\mathcal{S} \times \mathcal{A}$. For brevity, we will refer indifferently to MDPs, models or tasks, and write them $M = \langle R, T \rangle$.

Using a metric between MDPs has the appealing characteristic of quantifying the amount of similarity between tasks, which intuitively should be linked to the amount of transfer achievable. Song et al. (2016) define a metric based on the bi-simulation metric introduced by Ferns, Panangaden, and Precup (2004) and the Wasserstein metric (Villani 2008). value transfer is performed between states with low bi-simulation distances. However, this metric requires knowing both MDPs completely and is thus unusable in the Lifelong RL setting where we expect to perform transfer before having learned the current MDP. Further, the transfer technique they propose does allow negative transfer (see Appendix, Section 1). Carroll and Seppi (2005) also define a value-transfer method based on a measure of similarity between tasks. However, this measure is not computable online and thus not applicable to the Lifelong RL setting. Mahmud et al. (2013) and Brunskill and Li (2013) propose MDP clustering methods; respectively using a metric quantifying the regret of running the optimal policy of one MDP in the other MDP and the $\mathcal{L}_1$ norm between the MDP models. An advantage of clustering is to prune the set of possible source tasks. They use their approach for policy transfer, which differs from the value-transfer method proposed in this paper. Ammar et al. (2014) learn the model of a source MDP and view the prediction error on a target MDP as a dissimilarity measure in the task space. Their method makes use of samples from both tasks and is not readily applicable to the online setting considered in this paper. Lazaric, Restelli, and Bonarini (2008) provide a practical method for sample transfer, computing a similarity metric reflecting the probability of the models to be identical. Their approach is applicable in a batch RL setting as opposed to the online setting considered in this paper. The approach developed by Sorg and Singh (2009) is very similar to ours in the sense that they prove bounds on the optimal Q-function for new tasks, assuming that both MDPs are known and that a soft homomorphism exists between the state spaces. Brunskill and Li (2013) also provide a method that can be used for Q-function bounding in multi-task RL.

Abel et al. (2018) present the MaxQInit algorithm, providing transferable bounds on the Q-function with high probability while preserving PAC-MDP guarantees (Strehl, Li, and Littman 2009). Given a set of solved tasks, they derive the probability that the maximum over the Q-values of previous MDPs is an upper bound on the current task's optimal Q-function. This approach results in a method for non-negative transfer with high probability once enough tasks have been sampled. The method developed by Abel et al. (2018) is similar to ours in two fundamental points: first, a theoretical upper bounds on optimal Q-values across the MDP space is built; secondly, this provable upper bound is used to transfer knowledge between MDPs by replacing the maximum $\frac{1}{1-\gamma}$ bound in an RMax-like algorithm, providing PAC guarantees. The difference between the two approaches is illustrated in Figure 1, where the MaxQInit bound is the one developed by Abel et al. (2018), and the LRMax bound is the one we present in this paper. On this figure, the essence of the LRMax bound is noticeable. It stems from the fact that the optimal Q value function is locally Lipschitz continuous in the MDP space w.r.t. a specific pseudometric. Confirming the intuition, close MDPs w.r.t. this metric have close optimal Q values. It should be noticed that no bound is uniformly better than the other as intuited by Figure 1. Hence, combining all the bounds results in a tighter upper bound as we will illustrate in experiments (Section 5). We first carry out the theoretical characterization of the Lipschitz continuity properties in the following section. Then, we build on this result to propose a practical transfer method for the online Lifelong RL setting.

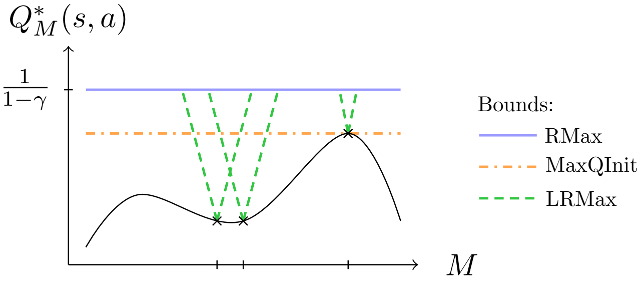


Figure 1: The optimal Q-value function represented for a particular s, a pair across the MDP space. The RMax, MaxQInit and LRMax bounds are represented for three sampled MDPs.

3 Lipschitz continuity of Q-functions

The intuition we build on is that similar MDPs should have similar optimal Q-functions. Formally, this insight can be translated into a continuity property of the optimal Q-function over the MDP space $\mathcal{M}$. The remainder of this section mathematically formalizes this intuition that will be used in the next section to derive a practical method for value transfer. To that end, we introduce a local pseudometric characterizing the distance between the models of two MDPs at a particular state-action pair. A reminder and a detailed discussion on the metrics used herein can be found in the Appendix, Section 2.

Definition 1. *Given two tasks $M = \langle R, T \rangle$, $\bar{M} = \langle \bar{R}, \bar{T} \rangle$, and a function $f : \mathcal{S} \rightarrow \mathbb{R}^+$, we define the* pseudometric between models *at $(s,a) \in \mathcal{S} \times \mathcal{A}$ w.r.t. f as:*

$$D^f_{sa}(M, \bar{M}) \triangleq |R^a_s - \bar{R}^a_s| + \sum_{s' \in \mathcal{S}} f(s') |T^a_{ss'} - \bar{T}^a_{ss'}|. \quad (1)$$

This pseudometric is relative to a positive function f. We implicitly cast this definition in the context of discrete state spaces. The extension to continuous spaces is straightforward but beyond the scope of this paper. For the sake of clarity in the remainder of this study, we introduce

$$D_{sa}(M \| \bar{M}) \triangleq D_{sa}^{\gamma V^*_{\bar{M}}}(M, \bar{M}),$$

corresponding to the pseudometric between models with the particular choice of $f = \gamma V^*_{\bar{M}}$. From this definition stems the following pseudo-Lipschitz continuity result.

Proposition 1 (Local pseudo-Lipschitz continuity). *For two MDPs $M, \bar{M}$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\left| Q^*_M(s,a) - Q^*_{\bar{M}}(s,a) \right| \leq \Delta_{sa}(M, \bar{M}), \qquad (2)$$

with the local MDP pseudometric $\Delta_{sa}(M, \bar{M}) \triangleq \min\left\{ d_{sa}(M \| \bar{M}), d_{sa}(\bar{M} \| M) \right\}$*, and the* local MDP dissimilarity $d_{sa}(M \| \bar{M})$ *is the unique solution to the following fixed-point equation for d_{sa}:*

$$d_{sa} = D_{sa}(M \| \bar{M}) + \gamma \sum_{s' \in \mathcal{S}} T^a_{ss'} \max_{a' \in \mathcal{A}} d_{s'a'}, \forall s, a. \qquad (3)$$

All the proofs of the paper can be found in the Appendix. This result establishes that the distance between the optimal Q-functions of two MDPs at $(s,a) \in \mathcal{S} \times \mathcal{A}$ is controlled by a local dissimilarity between the MDPs. The latter follows a fixed-point equation (Equation 3), which can be solved by Dynamic Programming (DP) (Bellman 1957). Note that, although the local MDP dissimilarity $d_{sa}(M \| \bar{M})$ is asymmetric, $\Delta_{sa}(M, \bar{M})$ *is* a pseudometric, hence the name *pseudo-Lipschitz continuity*. Notice that the policies in Equation 2 are the optimal ones for the two MDPs and thus are different. Proposition 1 is a mathematical result stemming from Definition 1 and should be distinguished from other frameworks of the literature that *assume* the continuity of the reward and transition models w.r.t. $\mathcal{S} \times \mathcal{A}$ (Rachelson and Lagoudakis 2010; Pirotta, Restelli, and Bascetta 2015; Asadi, Misra, and Littman 2018).This result establishes that the optimal Q-functions of two close MDPs, in the sense of Equation 1, are themselves close to each other. Hence, given $Q^*_{\bar{M}}$, the function

$$s, a \mapsto Q^*_{\bar{M}}(s,a) + \Delta_{sa}(M, \bar{M}) \qquad (4)$$

can be used as an upper bound on Q^*_M with M an unknown MDP. This is the idea on which we construct a computable and transferable upper bound in Section 4. In Figure 1, the upper bound of Equation 4 is represented by the LRMax bound. Noticeably, we provide a global pseudo-Lipschitz continuity property, along with similar results for the optimal value function V^*_M and the value function of a fixed policy. As these results do not directly serve the purpose of this article, we report them in the Appendix, Section 4.

4 Transfer using the Lipschitz continuity

A purpose of value transfer, when interacting online with a new MDP, is to initialize the value function and drive the exploration to accelerate learning. We aim to exploit value transfer in a method guaranteeing three conditions:

C1. the resulting algorithm is PAC-MDP;

C2. the transfer accelerates learning;

C3. the transfer is non-negative.

To achieve these conditions, we first present a transferable upper bound on Q^*_M in Section 4.1. This upper bound stems from the Lipschitz continuity result of Proposition 1. Then, we propose a practical way to *compute* this upper bound in Section 4.2. Precisely, we propose a surrogate bound that can be calculated online in the Lifelong RL setting, without having explored the source and target tasks completely. Finally, we implement the method in an algorithm described in Section 4.3, and demonstrate formally that it meets conditions C1, C2 and C3. Improvements are discussed in Section 4.4.

4.1 A transferable upper bound on Q^*_M

From Proposition 1, one can naturally define a local upper bound on the optimal Q-function of an MDP given the optimal Q-function of another MDP.

Definition 2. *Given two tasks M and $\bar{M}$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, the* Lipschitz upper bound on Q^*_M induced by $Q^*_{\bar{M}}$ *is defined as $U_{\bar{M}}(s,a) \geq Q^*_M(s,a)$ with:*

$$U_{\bar{M}}(s,a) \triangleq Q^*_{\bar{M}}(s,a) + \Delta_{sa}(M, \bar{M}). \qquad (5)$$

The *optimism in the face of uncertainty* principle leads to considering that the long-term expected return from any state is the $\frac{1}{1-\gamma}$ maximum return, unless proven otherwise. Particularly, the RMax algorithm (Brafman and Tennenholtz 2002), explores an MDP so as to shrink this upper bound. RMax is a model-based, online RL algorithm with PAC-MDP guarantees (Strehl, Li, and Littman 2009), meaning that convergence to a near-optimal policy is guaranteed in a polynomial number of missteps with high probability. It relies on an optimistic model initialization that yields an optimistic upper bound U on the optimal Q-function, then acts greedily w.r.t. U. By default, it takes the maximum value $U(s,a) = \frac{1}{1-\gamma}$, but any tighter upper bound is admissible. Thus, shrinking U with Equation 5 is expected to improve the learning speed or sample complexity for new tasks in Lifelong RL.

In RMax, during the resolution of a task M, $\mathcal{S} \times \mathcal{A}$ is split into a subset of known state-action pairs K and its complement K^c of unknown pairs. A state-action pair is known if the number of collected reward and transition samples allows estimating an ϵ-accurate model in $\mathcal{L}_1$-norm with probability higher than $1 - \delta$. We refer to ϵ and δ as the *RMax precision parameters*. This results in a threshold n_{known} on the number of visits $n(s,a)$ to a pair s, a that are necessary to reach this precision. Given the experience of a set of m MDPs $\bar{\mathcal{M}} = \{\bar{M}_1, \dots, \bar{M}_m\}$, we define the total bound as the minimum over all the induced Lipschitz bounds.

Proposition 2. *Given a partially known task $M = \langle R, T \rangle$, the set of known state-action pairs K, and the set of Lipschitz bounds on Q^*_M induced by previous tasks $\left\{ U_{\bar{M}_1}, \dots, U_{\bar{M}_m} \right\}$, the function Q defined below is an upper bound on Q^*_M for all $s, a \in \mathcal{S} \times \mathcal{A}$.*

$$Q(s,a) \triangleq \begin{cases} R^a_s + \gamma \sum\limits_{s' \in \mathcal{S}} T^a_{ss'} \max\limits_{a' \in \mathcal{A}} Q(s', a') \\ \qquad\qquad\qquad\qquad \textit{if } (s,a) \in K, \\ U(s,a) \textit{ otherwise,} \end{cases} \qquad (6)$$

with $U(s,a) = \min\left\{ \frac{1}{1-\gamma}, U_{\bar{M}_1}(s,a), \dots, U_{\bar{M}_m}(s,a) \right\}$.

Commonly in RMax, Equation 6 is solved to a precision ϵ_Q via Value Iteration. This yields a function Q that is a valid heuristic (bound on Q^*_M) for the exploration of MDP M.

4.2 A computable upper bound on Q^*_M

The key issue addressed in this section is how to actually compute $U(s,a)$, particularly when both source and target tasks are partially explored. Consider two tasks M and $\bar{M}$, on which vanilla RMax has been applied, yielding the respective sets of known state-action pairs K and $\bar{K}$, along with the learned models $\hat{M} = \langle \hat{T}, \hat{R} \rangle$ and $\hat{\bar{M}} = \langle \hat{\bar{T}}, \hat{\bar{R}} \rangle$, and the upper bounds Q and $\bar{Q}$ respectively on Q^*_M and $Q^*_{\bar{M}}$. Notice that, if $K = \emptyset$, then $Q(s,a) = \frac{1}{1-\gamma}$ for all s, a pairs. Conversely, if $K^c = \emptyset$, Q is an ϵ-accurate estimate of Q^*_M in $\mathcal{L}_1$-norm with high probability. Equation 6 allows the transfer of knowledge from $\bar{M}$ to M if $U_{\bar{M}}(s,a)$ can be computed. Unfortunately, the true model and optimal value functions, necessary to compute $U_{\bar{M}}$, are *partially* known (see Equation 5). Thus, we propose to compute a looser upper bound based on the learned models and value functions. First, we provide an upper bound $\hat{D}_{sa}(M \| \bar{M})$ on $D_{sa}(M \| \bar{M})$ (Definition 1).

Proposition 3. *Given two tasks M, $\bar{M}$ and respectively K, $\bar{K}$ the subsets of $\mathcal{S} \times \mathcal{A}$ where their models are known with accuracy ϵ in $\mathcal{L}_1$-norm with probability at least $1 - \delta$,*

$$\boldsymbol{Pr}\left(\hat{D}_{sa}(M \| \bar{M}) \geq D_{sa}(M \| \bar{M})\right) \geq 1 - \delta$$

with $\hat{D}_{sa}(M \| \bar{M})$ the upper bound on the pseudometric between models *defined below for $B = \epsilon\left(1 + \gamma \max_{s'} \bar{V}(s')\right)$.*

$$\hat{D}_{sa}(M \| \bar{M}) \triangleq \begin{cases} D^{\gamma \bar{V}}_{sa}(\hat{M}, \hat{\bar{M}}) + 2B & \textit{if } (s,a) \in K \cap \bar{K} \\ \max\limits_{\bar{\mu} \in \mathcal{M}} D^{\gamma \bar{V}}_{sa}(\hat{M}, \bar{\mu}) + B & \textit{if } (s,a) \in K \cap \bar{K}^c \\ \max\limits_{\mu \in \mathcal{M}} D^{\gamma \bar{V}}_{sa}(\mu, \hat{\bar{M}}) + B & \textit{if } (s,a) \in K^c \cap \bar{K} \\ \max\limits_{\mu, \bar{\mu} \in \mathcal{M}^2} D^{\gamma \bar{V}}_{sa}(\mu, \bar{\mu}) & \textit{if } (s,a) \in K^c \cap \bar{K}^c \end{cases} \tag{7}$$

Importantly, this upper bound $\hat{D}_{sa}(M \| \bar{M})$ can be calculated analytically (see Appendix, Section 7). This makes $\hat{D}_{sa}(M \| \bar{M})$ usable in the online Lifelong RL setting, where already explored tasks may be partially learned, and little knowledge has been gathered on the current task. The magnitude of the B term is controlled by ϵ. In the case where no information is available on the maximum value of $\bar{V}$, we have that $B = \frac{\epsilon}{1-\gamma}$. ϵ measures the accuracy with which the tasks are known: the smaller ϵ, the tighter the B bound. Note that $\bar{V}$ is used as an upper bound on the true $V^*_{\bar{M}}$. In many cases, $\max_{s'} V^*_{\bar{M}}(s') \leq \frac{1}{1-\gamma}$; *e.g.* for stochastic shortest path problems, which feature rewards only upon reaching terminal states, we have that $\max_{s'} V^*_{\bar{M}}(s') = 1$ and thus $B = (1 + \gamma)\epsilon$ is a tighter bound for transfer. Combining $\hat{D}_{sa}(M \| \bar{M})$ and Equation 3, one can derive an upper bound $\hat{d}_{sa}(M \| \bar{M})$ on $d_{sa}(M \| \bar{M})$, detailed in Proposition 4.

Proposition 4. *Given two tasks $M, \bar{M} \in \mathcal{M}$, K the set of state-action pairs where (R, T) is known with accuracy ϵ in $\mathcal{L}_1$-norm with probability at least $1 - \delta$. If $\gamma(1 + \epsilon) < 1$, the solution $\hat{d}_{sa}(M \| \bar{M})$ of the following fixed-point equation on $\hat{d}_{sa}$ (for all $s, a \in \mathcal{S} \times \mathcal{A}$) is an upper bound on $d_{sa}(M \| \bar{M})$ with probability at least $1 - \delta$:*

$$\hat{d}_{sa} = \hat{D}_{sa}(M \| \bar{M}) + \tag{8}$$

$$\begin{cases} \gamma \left(\sum\limits_{s' \in \mathcal{S}} \hat{T}^a_{ss'} \max\limits_{a' \in \mathcal{A}} \hat{d}_{s'a'} + \epsilon \max\limits_{s', a' \in \mathcal{S} \times \mathcal{A}} \hat{d}_{s'a'} \right) \textit{ if } s, a \in K, \\ \gamma \max\limits_{s', a' \in \mathcal{S} \times \mathcal{A}} \hat{d}_{s'a'} \textit{ otherwise.} \end{cases}$$

Similarly as in Proposition 3, the condition $\gamma(1 + \epsilon) < 1$ illustrates the fact that for a large return horizon (large γ), a high accuracy (small ϵ) is needed for the bound to be computable. Eventually, a computable upper bound on Q^*_M given $\bar{M}$ with high probability is given by

$$\hat{U}_{\bar{M}}(s,a) = \bar{Q}(s,a) + \min\left\{ \hat{d}_{sa}(M \| \bar{M}), \hat{d}_{sa}(\bar{M} \| M) \right\}. \tag{9}$$

The associated upper bound on $U(s,a)$ (Equation 6) given the set of previous tasks $\bar{\mathcal{M}} = \{\bar{M}_i\}_{i=1}^m$ is defined by

$$\hat{U}(s,a) = \min\left\{ \tfrac{1}{1-\gamma}, \hat{U}_{\bar{M}_1}(s,a), \ldots, \hat{U}_{\bar{M}_m}(s,a) \right\}. \tag{10}$$

This upper bound can be used to transfer knowledge from a partially solved source task to a target task. If $\hat{U}(s,a) \leq \frac{1}{1-\gamma}$ on a subset of $\mathcal{S} \times \mathcal{A}$, then the convergence rate can be improved. As complete knowledge of both tasks is not needed to compute the upper bound, it can be applied online in the Lifelong RL setting. In the next section, we explicit an algorithm that leverages this value-transfer method.

4.3 Lipschitz RMax algorithm

In Lifelong RL, MDPs are encountered sequentially. Applying RMax to task M yields the set of known state-action pairs K, the learned models $\hat{T}$ and $\hat{R}$, and the upper bound Q on Q^*_M. Saving this information when the task changes allows computing the upper bound of Equation 10 for the new target task, and using it to shrink the optimistic heuristic of RMax. This computation effectively transfers value functions between tasks based on task similarity. As the new task is explored online, the task similarity is progressively assessed with better confidence, refining the values of $\hat{D}_{sa}(M \| \bar{M})$, $\hat{d}_{sa}(M \| \bar{M})$ and eventually $\hat{U}$, allowing for more efficient transfer where the task similarity is appraised. The resulting algorithm, Lipschitz RMax (LRMax), is presented in Algorithm 1. To avoid ambiguities with $\bar{\mathcal{M}}$, we use $\hat{\mathcal{M}}$ to store learned features ($\hat{T}$, $\hat{R}$, K, Q) about previous MDPs. In a nutshell, the behavior of LRMax is precisely that of RMax, but with a tighter admissible heuristic $\hat{U}$ that becomes better as the new task is explored (while this heuristic remains constant in vanilla RMax). LRMax is PAC-MDP (Condition C1) as stated in Propositions 5 and 6 below. With $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$, the sample complexity of vanilla RMax is $\tilde{\mathcal{O}}(S^2 A / (\epsilon^3 (1-\gamma)^3))$, which is improved by LRMax in Proposition 5 and meets Condition C2. Finally, $\hat{U}$ is a provable upper bound with high probability on Q^*_M, which avoids negative transfer and meets Condition C3.

Proposition 5 (Sample complexity (Strehl, Li, and Littman 2009))**.** *With probability $1 - \delta$, the greedy policy w.r.t. Q computed by LRMax achieves an ϵ-optimal return in MDP M after*

$$\tilde{\mathcal{O}}\left(\frac{S \left| \{ s, a \in \mathcal{S} \times \mathcal{A} \mid \hat{U}(s,a) \geq V^*_M(s) - \epsilon \} \right|}{\epsilon^3 (1-\gamma)^3} \right)$$

samples (when logarithmic factors are ignored), with $\hat{U}$ defined in Equation 10 a non-static, decreasing quantity, upper bounded by $\frac{1}{1-\gamma}$.

Algorithm 1: Lipschitz RMax algorithm

```
Initialize $\hat{\mathcal{M}} = \emptyset$.
for each newly sampled MDP $M$ do
    Initialize $Q(s,a) = \frac{1}{1-\gamma}, \forall s, a$, and $K = \emptyset$
    Initialize $\hat{T}$ and $\hat{R}$ (RMax initialization)
    $Q \leftarrow$ UpdateQ($\hat{\mathcal{M}}, \hat{T}, \hat{R}$)
    for $t \in [1$, max number of steps$]$ do
        $s$ = current state, $a = \arg\max_{a'} Q(s, a')$
        Observe reward $r$ and next state $s'$
        $n(s,a) \leftarrow n(s,a) + 1$
        if $n(s,a) < n_{known}$ then
            Store $(s, a, r, s')$
        if $n(s,a) = n_{known}$ then
            Update $K$ and $(\hat{T}^a_{ss'}, \hat{R}^a_s)$ (learned model)
            $Q \leftarrow$ UpdateQ($\hat{\mathcal{M}}, \hat{T}, \hat{R}$)
    Save $\hat{M} = \left(\hat{T}, \hat{R}, K, Q\right)$ in $\hat{\mathcal{M}}$
Function UpdateQ($\hat{\mathcal{M}}, \hat{T}, \hat{R}$):
for $\bar{M} \in \hat{\mathcal{M}}$ do
    Compute $\hat{D}_{sa}(M \| \bar{M})$, $\hat{D}_{sa}(\bar{M} \| M)$ (Eq. 7)
    Compute $\hat{d}_{sa}(M \| \bar{M})$, $\hat{d}_{sa}(\bar{M} \| M)$ (DP on Eq. 8)
    Compute $\hat{U}_{\bar{M}}$ (Eq. 9)
Compute $\hat{U}$ (Eq. 10)
Compute and return $Q$ (DP on Eq. 6 using $\hat{U}$)
```

Proposition 5 shows that the sample complexity of LRMax is no worse than that of RMax. Consequently, in the worst case, LRMax performs as badly as learning from scratch, which is to say that the transfer method is not negative as it cannot degrade the performance.

Proposition 6 (Computational complexity)**.** *The total computational complexity of LRMax (Algorithm 1) is*

$$\tilde{\mathcal{O}}\left(\tau + \frac{S^3 A^2 N}{(1-\gamma)} \ln\left(\frac{1}{\epsilon_Q(1-\gamma)}\right)\right)$$

with τ the number of interaction steps, ϵ_Q the precision of value iteration and N the number of source tasks.

4.4 Refining the LRMax bounds

LRMax relies on bounds on the local MDP dissimilarity (Equation 8). The quality of the Lipschitz bound on Q^*_M can be improved according to the quality of those estimates. We discuss two methods to provide finer estimates.

Refining with prior knowledge. First, from the definition of $D_{sa}(M \| \bar{M})$, it is easy to show that this pseudometric between models is always upper bounded by $\frac{1+\gamma}{1-\gamma}$. However, in practice, the tasks experienced in a Lifelong RL experiment might not cover the full span of possible MDPs $\mathcal{M}$ and may systematically be closer to each other than $\frac{1+\gamma}{1-\gamma}$. For instance, the distance between two games in the Arcade Learning Environment (ALE) (Bellemare et al. 2013), is smaller than the maximum distance between any two MDPs defined on the common state-action space of the ALE (extended discussion in Appendix, Section 11). Let us note $\tilde{\mathcal{M}} \subset \mathcal{M}$ the set of possible MDPs for a particular Lifelong RL experiment. Let $D_{\max}(s,a) \triangleq \max_{M, \bar{M} \in \tilde{\mathcal{M}}^2} \left(D_{sa}(M \| \bar{M})\right)$ be the *maximum model pseudo-distance* at a particular s, a pair on the subset $\tilde{\mathcal{M}}$. *Prior knowledge* might indicate a smaller upper bound for $D_{\max}(s,a)$ than $\frac{1+\gamma}{1-\gamma}$. We will note such an upper bound $D_{\max}$, considered valid for all s, a pairs, *i.e.*, such that $D_{\max} \geq \max_{s,a,M,\bar{M} \in \mathcal{S} \times \mathcal{A} \times \tilde{\mathcal{M}}^2} \left(D_{sa}(M \| \bar{M})\right)$. In a Lifelong RL experiment, $D_{\max}$ can be seen as a rough estimate of the maximum model discrepancy an agent may encounter. Figure 2 illustrates the relative importance of $D_{\max}$ *vs.* $\frac{1+\gamma}{1-\gamma}$. Solving Equation 8 boils down to accumulating $\hat{D}_{sa}(M \| \bar{M})$ values in $\hat{d}_{sa}(M \| \bar{M})$. Hence, reducing a $\hat{D}_{sa}(M \| \bar{M})$ estimate in a single s, a pair actually reduces $\hat{d}_{sa}(M \| \bar{M})$ in *all* s, a pairs. Thus, replacing $\hat{D}_{sa}(M \| \bar{M})$ in Equation 8 by $\min\{D_{\max}, \hat{D}_{sa}(M \| \bar{M})\}$, provides a smaller upper bound $\hat{d}_{sa}(M \| \bar{M})$ on $d_{sa}(M \| \bar{M})$, and thus a smaller $\hat{U}$ which allows transfer if it is less than $\frac{1}{1-\gamma}$. Consequently, the knowledge of such a bound $D_{\max}$ can make a difference between successful and unsuccessful transfer, even if its value is of little importance. Conversely, setting a value for $D_{\max}$ quantifies the distance between MDPs where transfer is efficient.

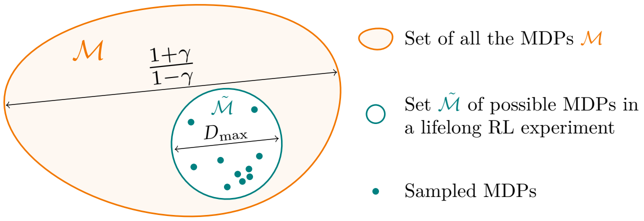

$\mathcal{M}$
$\frac{1+\gamma}{1-\gamma}$
$\tilde{\mathcal{M}}$
$D_{\max}$
Set of all the MDPs $\mathcal{M}$
Set $\tilde{\mathcal{M}}$ of possible MDPs in a lifelong RL experiment
Sampled MDPs

Figure 2: Illustration of the prior knowledge on the maximum pseudo-distance between models for a particular s, a pair.

Refining by learning the maximum distance. The value of $D_{\max}(s,a)$ can be estimated online for each s, a pair, discarding the hypothesis of available prior knowledge. We propose to use an empirical estimate of the maximum model distance at s, a: $\hat{D}_{\max}(s,a) \triangleq \max_{M, \bar{M} \in \hat{\mathcal{M}}^2} \{\hat{D}_{sa}(M \| \bar{M})\}$, with $\hat{\mathcal{M}}$ the set of explored tasks. The pitfall of this approach is that, with few explored tasks, $\hat{D}_{\max}(s,a)$ could underestimate $D_{\max}(s,a)$. Proposition 7 provides a lower bound on the probability that $\hat{D}_{\max}(s,a) + \epsilon$ does not underestimate $D_{\max}(s,a)$, depending on the number of sampled tasks.

Proposition 7. *Consider an algorithm producing ϵ-accurate model estimates $\hat{D}_{sa}(M \| \bar{M})$ for a subset K of $\mathcal{S} \times \mathcal{A}$ after interacting with any two MDPs $M, \bar{M} \in \mathcal{M}$. Assume $\hat{D}_{sa}(M \| \bar{M}) \geq D_{sa}(M \| \bar{M})$ for any $s, a \notin K$. For all $s, a \in \mathcal{S} \times \mathcal{A}$, $\delta \in (0, 1]$, after sampling m tasks, if m is large enough to verify $2(1-p_{\min})^m - (1-2p_{\min})^m \leq \delta$,*

$$\textbf{\textit{Pr}}\left(\hat{D}_{\max}(s,a) + \epsilon \geq D_{\max}(s,a)\right) \geq 1 - \delta\,.$$

This result indicates when $\hat{D}_{\max}(s,a) + \epsilon$ upper bounds $D_{\max}(s,a)$ with high probability. In such a case, $\hat{D}_{sa}(M\|\bar{M})$ of Equation 8 can be replaced by $\min\{\hat{D}_{\max}(s,a)+\epsilon, \hat{D}_{sa}(M\|\bar{M})\}$ to tighten the bound on $d_{sa}(M\|\bar{M})$. Assuming a lower bound $p_{\min}$ on the sampling probability of a task implies that $\mathcal{M}$ is finite and is seen as a non-adversarial task sampling rule (Abel et al. 2018).

5 Experiments

The experiments reported here[1] illustrate how the Lipschitz bound (Equation 9) provides a tighter upper bound on Q^*, improving the sample complexity of LRMax compared to RMax, and making the transfer of inter-task knowledge effective. Graphs are displayed with 95% confidence intervals. For information in line with the Machine Learning Reproducibility Check-list (Pineau 2019) see the Appendix, Section 16.

We evaluate different variants of LRMax in a Lifelong RL experiment. The RMax algorithm will be used as a no-transfer baseline. LRMax(x) denotes Algorithm 1 with prior $D_{\max} = x$. MaxQInit denotes the MAXQINIT algorithm from Abel et al. (2018), consisting in a state-of-the art PAC-MDP algorithm. Both LRMax and MaxQInit algorithms achieve value transfer by providing a tighter upper bound on Q^* than $\frac{1}{1-\gamma}$. Computing both upper bounds and taking the minimum results in combining the two approaches. We include such a combination in our study with the LRMaxQInit algorithm. Similarly, the latter algorithm benefiting from prior knowledge $D_{\max} = x$ is denoted by LRMaxQInit(x). For the sake of comparison, we only compare algorithms with the same features, namely, tabular, online, PAC-MDP methods, presenting non-negative transfer.

The environment used in all experiments is a variant of the "tight" task used by Abel et al. (2018). It is an 11×11 grid-world, the initial state is in the centre, actions are the cardinal moves (Appendix, Section 12). The reward is always zero except for the three goal cells in the upper-right corner. Each sampled task has its own reward values, drawn from $[0.8, 1]$ for each of the three goal cells and its own probability of slipping (performing a different action than the one selected), picked in $[0, 0.1]$. Hence, tasks have different reward and transition functions. Notice the distinction in applicability between MaxQInit, that requires the set of MDPs to be finite, and LRMax, that can be used with any set of MDPs. For the comparison between both to be possible, we drew tasks from a finite set of 5 MDPs. We sample 15 tasks sequentially among this set, each run for 2000 episodes of length 10. The operation is repeated 10 times to narrow the confidence intervals. We set $n_{known} = 10$, $\delta = 0.05$, and $\epsilon = 0.01$ (discussion in Appendix, Section 15). Other Lifelong RL experiments are reported in Appendix, Section 13.

The results are reported in Figure 3. Figure 3a displays the discounted return for each task, averaged across episodes. Similarly, Figure 3b displays the discounted return for each episode, averaged across tasks (same color code as Figure 3a). Figure 3c displays the discounted return for five specific instances, detailed below. To avoid inter-task disparities, all the aforementioned discounted returns are displayed relative to an estimator of the optimal expected return for each task. For readability, Figures 3b and 3c display a moving average over 100 episodes. Figure 3d reports the benefits of various values of $D_{\max}$ on the algorithmic properties.

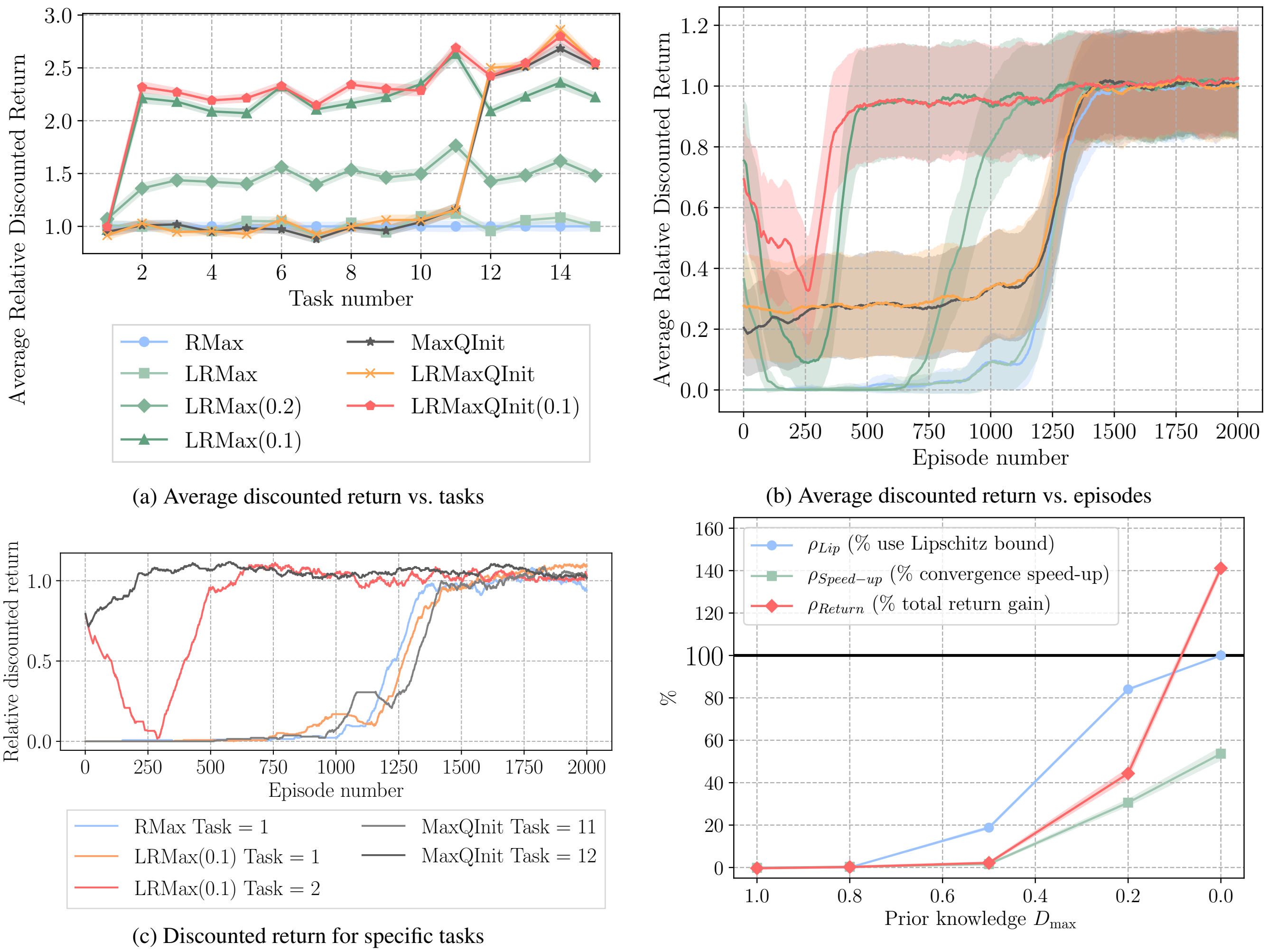
In Figure 3a, we first observe that LRMax benefits from the transfer method, as the average discounted return increases as more tasks are experienced. Moreover, this advantage appears as early as the second task. In contrast, MaxQInit requires to wait for task 12 before benefiting from transfer. As suggested in Section 4.4, increasing amounts of prior knowledge allow the LRMax transfer method to be more efficient: a smaller known upper bound $D_{\max}$ on $\hat{D}_{sa}(M\|\bar{M})$ accelerates convergence. Combining both approaches in the LRMaxQInit algorithm outperforms all other methods. Episode-wise, we observe in Figure 3b that the LRMax transfer method allows for faster convergence, *i.e.*, lower sample complexity. Interestingly, LRMax exhibits three stages in the learning process. 1) The first episodes are characterized by a direct exploitation of the transferred knowledge, causing these episodes to yield high payoff. This behavior is a consequence of the combined facts that the Lipschitz bound (Equation 9) is larger on promising regions of $\mathcal{S} \times \mathcal{A}$ seen on previous tasks and the fact that LRMax acts greedily w.r.t. that bound. 2) This high performance regime is followed by the exploration of unknown regions of $\mathcal{S} \times \mathcal{A}$, in our case yielding low returns. Indeed, as promising regions are explored first, the bound becomes tighter for the corresponding state-action pairs, enough for the Lipschitz bound of unknown pairs to become larger, thus driving the exploration towards low payoff regions. Such regions are then identified and never revisited. 3) Eventually, LRMax stops exploring and converges to the optimal policy. Importantly, in all experiments, LRMax never experiences negative transfer, as supported by the provability of the Lipschitz upper bound with high probability. LRMax is at least as efficient as the no-transfer RMax baseline.

Figure 3c displays the collected returns of RMax, LRMax(0.1), and MaxQInit for specific tasks. We observe that LRMax benefits from transfer as early as Task 2, where the previous 3-stage behavior is visible. MaxQInit takes until task 12 to leverage the transfer method. However, the bound it provides is tight enough that it does not have to explore.

In Figure 3d, we display the following quantities for various values of $D_{\max}$: ρ_{Lip}, the fraction of the time the Lipschitz bound was tighter than the RMax bound $\frac{1}{1-\gamma}$; $\rho_{Speed\text{-}up}$, is the relative gain of time steps before convergence when comparing LRMax to RMax. This quantity is estimated based on the last updates of the empirical model $\bar{M}$; ρ_{Return}, is the relative total return gain on 2000 episodes of LRMax w.r.t. RMax. First, we observe an increase of ρ_{Lip} as $D_{\max}$ becomes tighter. This means that the Lipschitz bound of Equation 9 becomes effectively smaller than $\frac{1}{1-\gamma}$. This phenomenon leads to faster convergence, indicated by $\rho_{Speed\text{-}up}$. Eventually, this increased convergence rate allows for a net total return gain, as can be seen with the increase of ρ_{Return}.

Overall, in this analysis, we have showed that LRMax benefits from an enhanced sample complexity thanks to the value-transfer method. The knowledge of a prior $D_{\max}$ increases

[1] Code available at https://github.com/SuReLI/llrl



(a) Average discounted return vs. tasks

(b) Average discounted return vs. episodes

(c) Discounted return for specific tasks

(d) Algorithmic properties vs. $D_{\max}$

Figure 3: Experimental results. LRMax benefits from an enhanced sample complexity thanks to the value-transfer method.

this benefit. The method is comparable to the MaxQInit method and has some advantages such as the early fitness for use and the applicability to infinite sets of tasks. Moreover, the transfer is non-negative while preserving the PAC-MDP guarantees of the algorithm. Additionally, we show in Appendix, Section 14 that, when provided with any prior $D_{\max}$, LRMax increasingly stops using it during exploration, confirming the claim of Section 4.4 that providing $D_{\max}$ enables transfer even if its value is of little importance.

6 Conclusion

We have studied theoretically the Lipschitz continuity property of the optimal Q-function in the MDP space w.r.t. a new metric. We proved a local Lipschitz continuity result, establishing that the optimal Q-functions of two close MDPs are themselves close to each other. We then proposed a value-transfer method using this continuity property with the Lipschitz RMax algorithm, practically implementing this approach in the Lifelong RL setting. The algorithm preserves PAC-MDP guarantees, accelerates learning in subsequent tasks and exhibits no negative transfer. Improvements of the algorithm were discussed in the form of prior knowledge on the maximum distance between models and online estimation of this distance. As a non-negative, similarity-based, PAC-MDP transfer method, the LRMax algorithm is the first method of the literature combining those three appealing features. We showcased the algorithm in Lifelong RL experiments and demonstrated empirically its ability to accelerate learning while not experiencing any negative transfer. Notably, our approach can directly extend other PAC-MDP algorithms (Szita and Szepesvári 2010; Rao and Whiteson 2012; Pazis, Parr, and How 2016; Dann, Lattimore, and Brunskill 2017) to the Lifelong setting. In hindsight, we believe this contribution provides a sound basis to non-negative value transfer via MDP similarity, a study that was lacking in the literature. Key insights for the practitioner lie both in the theoretical analysis and in the practical derivation of a transfer scheme achieving non-negative transfer with PAC guarantees. Further, designing scalable methods conveying the same intuition could be a promising research direction.

Acknowledgments

We would like to thank Dennis Wilson for fruitful discussions and comments on the paper. This research was supported by the Occitanie region, France; ISAE-SUPAERO (Institut Supérieur de l'Aéronautique et de l'Espace); fondation ISAE-SUPAERO; École Doctorale Systèmes; and ONERA, the French Aerospace Lab.

References

Abel, D.; Jinnai, Y.; Guo, S. Y.; Konidaris, G.; and Littman, M. L. 2018. Policy and Value Transfer in Lifelong Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 20–29.

Ammar, H. B.; Eaton, E.; Taylor, M. E.; Mocanu, D. C.; Driessens, K.; Weiss, G.; and Tuyls, K. 2014. An Automated Measure of MDP Similarity for Transfer in Reinforcement Learning. In *Workshops at the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*.

Asadi, K.; Misra, D.; and Littman, M. L. 2018. Lipschitz Continuity in Model-Based Reinforcement Learning. *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)* .

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47: 253–279.

Bellman, R. 1957. *Dynamic Programming*. Princeton, USA: Princeton University Press.

Brafman, R. I.; and Tennenholtz, M. 2002. R-max - a General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning. *Journal of Machine Learning Research* 3(Oct): 213–231.

Brunskill, E.; and Li, L. 2013. Sample Complexity of Multitask Reinforcement Learning. In *Proceedings of the 29th conference on Uncertainty in Artificial Intelligence (UAI 2013)*.

Brunskill, E.; and Li, L. 2014. PAC-inspired Option Discovery in Lifelong Reinforcement Learning. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, 316–324.

Carroll, J. L.; and Seppi, K. 2005. Task Similarity Measures for Transfer in Reinforcement Learning Task Libraries. In *Proceedings of the 5th International Joint Conference on Neural Networks (IJCNN 2005)*, volume 2, 803–808. IEEE.

Dann, C.; Lattimore, T.; and Brunskill, E. 2017. Unifying PAC and Regret: Uniform PAC Bounds for Episodic Reinforcement Learning. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 5713–5723.

Ferns, N.; Panangaden, P.; and Precup, D. 2004. Metrics for Finite Markov Decision Processes. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence (UAI 2004)*, 162–169. AUAI Press.

Lazaric, A. 2012. Transfer in Reinforcement Learning: a Framework and a Survey. In *Reinforcement Learning*, 143–173. Springer.

Lazaric, A.; Restelli, M.; and Bonarini, A. 2008. Transfer of Samples in Batch Reinforcement Learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, 544–551.

Mahmud, M. M.; Hawasly, M.; Rosman, B.; and Ramamoorthy, S. 2013. Clustering Markov Decision Processes for Continual Transfer. Technical report.

Pazis, J.; Parr, R. E.; and How, J. P. 2016. Improving PAC Exploration using the Median of Means. In *Advances in Neural Information Processing Systems 29 (NeurIPS 2016)*, 3898–3906.

Pineau, J. 2019. Machine Learning Reproducibility Checklist. https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf. Version 1.2, Mar.27 2019.

Pirotta, M.; Restelli, M.; and Bascetta, L. 2015. Policy gradient in Lipschitz Markov Decision Processes. *Machine Learning* 100(2-3): 255–283.

Puterman, M. L. 2014. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Rachelson, E.; and Lagoudakis, M. G. 2010. On the Locality of Action Domination in Sequential Decision Making. In *Proceedings of the 11th International Symposium on Artificial Intelligence and Mathematics (ISAIM 2010)*.

Rao, K.; and Whiteson, S. 2012. V-MAX: Tempered Optimism for Better PAC Reinforcement Learning. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, 375–382.

Silver, D. L.; Yang, Q.; and Li, L. 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, volume 13, 05.

Song, J.; Gao, Y.; Wang, H.; and An, B. 2016. Measuring the Distance Between Finite Markov Decision Processes. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, 468–476.

Sorg, J.; and Singh, S. 2009. Transfer via Soft Homomorphisms. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, 741–748. International Foundation for Autonomous Agents and Multiagent Systems.

Strehl, A. L.; Li, L.; and Littman, M. L. 2009. Reinforcement Learning in Finite MDPs: PAC Analysis. *Journal of Machine Learning Research* 10(Nov): 2413–2444.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT press, Cambridge.

Szita, I.; and Szepesvári, C. 2010. Model-Based Reinforcement Learning with Nearly Tight Exploration Complexity Bounds. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, 1031–1038.

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

Villani, C. 2008. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media.

Wilson, A.; Fern, A.; Ray, S.; and Tadepalli, P. 2007. Multi-Task Reinforcement Learning: A Hierarchical Bayesian Approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML 2007)*, 1015–1022.