Lifelong Hyper-Policy Optimization with Multiple Importance Sampling Regularization

Pierre Liotet[1], Francesco Vidaich[2], Alberto Maria Metelli[1], Marcello Restelli[1]
[1]Politecnico di Milano
[2]University of Padova
pierre.liotet@polimi.it

Abstract

Learning in a lifelong setting, where the dynamics continually evolve, is a hard challenge for current reinforcement learning algorithms. Yet this would be a much needed feature for practical applications. In this paper, we propose an approach which learns a hyper-policy, whose input is time, that outputs the parameters of the policy to be queried at that time. This hyper-policy is trained to maximize the estimated future performance, efficiently reusing past data by means of importance sampling, at the cost of introducing a controlled bias. We combine the future performance estimate with the past performance to mitigate catastrophic forgetting. To avoid overfitting the collected data, we derive a differentiable variance bound that we embed as a penalization term. Finally, we empirically validate our approach, in comparison with state-of-the-art algorithms, on realistic environments, including water resource management and trading.

1 Introduction

In the most common setting, Reinforcement Learning (RL, Sutton and Barto 2018) considers the interaction between an agent and an environment in a sequence of episodes. The agent progressively adapts its policy, but the dynamics of the environment, typically, remain unchanged. Most importantly, the agent can experience multiple times the same portion of the environment. However, this usual setting is sometimes not met in real applications. Hence several modifications have been proposed to model different, more realistic, scenarios. One of them is *non-stationary* RL (Bowerman 1974), which considers that the episodes can follow different distributions, or even that the distribution changes within each episode. The change can either be *abrupt*, when a clear separation between tasks evolving through time is present, or *smooth*, when the environment's evolution displays some regularity w.r.t. time. Non-stationarity can arise from diverse causes and can be interpreted as a form of partial knowledge (Khetarpal et al. 2020). Learning in non-stationary environments has been diffusely addressed in the literature (Garcia and Smith 2000; Ghate and Smith 2013; Lesner and Scherrer 2015). Nevertheless, in these works, the agent-environment interaction based on episodes is preserved, so that the same region of non-stationary behavior can be experienced multiple times by the agent.

Still moving towards a more realistic setting, another modification is the *lifelong* interaction with the environment (Silver, Yang, and Li 2013; Brunskill and Li 2014). Here, the separation in episodes vanishes and, therefore, there is no clear distinction between learning and testing. Moreover, given the never-ending nature of this interaction, the agent is not allowed to reset the environment and might not be able to visit twice some portions of the environment. Thus, the agent aims at exploiting the experience collected in the past to optimize its future performance. In this sense, Lifelong Learning (LL) can be considered closer to the intuitive idea of learning for human agents. More technically, LL requires the agent to readily adapt its behavior to the environment's evolution, as well as keeping memory of past behaviors in order to leverage this knowledge on future similar phases (Khetarpal et al. 2020). This represents a critical trade-off, peculiar of the lifelong setting. Indeed, if the agent displays a highly non-stationary behavior, the samples collected in the past would be poorly informative and, consequently, hardly usable to estimate the future performance. Instead, being more stationary favors sample reuse, at the cost of sacrificing the optimality of the learned behavior.

In this paper, we consider the RL problem with a lifelong interaction between an agent and its environment, where the latter's dynamics *smoothly* evolve over time. We address this problem by designing a hyper-policy, responsible for selecting the best policy to be played at time t. This way, we decouple the problem of learning in a non-stationary setting, by assigning to the hyper-policy level the management of the dependence on time and to the policy level the action to be played given a state (Section 3). This hyper-policy is trained with an objective composed of the future performance, the ultimate quantity to be maximized, and the past performance. Although the past performance is not the direct interest of our agent, it is included to constrain the hyper-policy to perform well on past samples, thus mitigating *catastrophic forgetting*. Future performance is estimated through multiple importance sampling. To avoid overfitting, we additionally penalize the hyper-policy for the variance of the estimations. Rather than estimating this quantity, that would inject further uncertainty, we derive a differentiable upper-bound allowing a gradient based optimization. This penalization, involving a divergence between past and future hyper-policies, has the indirect effect of quantifying and controlling the "amount" of non-stationarity selected by the agent (Section 4). We propose a practical policy-gradient optimization of the objective, which we name POLIS, for Policy Optimization in Lifelong learning through Importance Sampling. After having revised the literature (Section 5), we provide an experimental evaluation on realistic domains, including a trading and water resource management, in comparison with state-of-the-art baselines (Section 6). The proofs of the results presented in the main paper are reported in Appendix A[1].

2 Preliminaries

In this section, we report the necessary background that will be employed in the following sections.

Lifelong RL A Non-Stationary Markov Decision Process (MDP, Puterman 2014) is defined as $\mathcal{M} = (\mathcal{X}, \mathcal{A}, P, R, \gamma, D_0)$, where $\mathcal{X}$ and $\mathcal{A}$ are the state and action spaces respectively, $P = (P_t)_{t\in\mathbb{N}}$ is the transition model that for every decision epoch $t \in \mathbb{N}$ and $(x, a) \in \mathcal{X} \times \mathcal{A}$ assigns a probability distribution over the next state $x' \sim P_t(\cdot|x, a)$, $R = (R_t)_{t\in\mathbb{N}}$ is the reward distribution assigning for every $t \in \mathbb{N}$ and $(x, a) \in \mathcal{X} \times \mathcal{A}$ the reward $r \sim R_t(\cdot|x, a)$ such that $\|r\|_\infty \leqslant R_{\max} < \infty$, $\gamma \in [0, 1]$ is the discount factor, and D_0 is the initial state distribution. A non-stationary policy $\pi = (\pi_t)_{t\in\mathbb{N}}$ assigns for every decision epoch $t \in \mathbb{N}$ and state $x \in \mathcal{X}$ a probability distribution over the actions $a_t \sim \pi_t(\cdot|x)$. Let $T \in \mathbb{N}$ be the current decision epoch, let $\beta \in \mathbb{N}_{\geqslant 1}$, we define the *$\beta$-step ahead expected return* as:

$$J_{T,\beta}(\pi) = \sum_{t=T+1}^{T+\beta} \widehat{\gamma}^t \mathbb{E}_t^\pi[r_t], \tag{1}$$

where $\widehat{\gamma}^t = \gamma^{t-T-1}$ and we denote with $\mathbb{E}_t^\pi$ the expectation under the visitation distribution induced by policy π in MDP $\mathcal{M}$ after t decision epochs. A policy $\pi^\star_{T,\beta}$ is β-step ahead optimal if $\pi^\star_{T,\beta} \in \arg\max_{\pi\in\Pi^\beta} J_{T,\beta}(\pi)$, where Π^β is the set of non-stationary policies operating over β decision epochs. In *classical RL*, the agent's goal consists in maximizing $J_{0,H}$, where H is the (possibly infinite) horizon of the task, having the possibility to collect *multiple* episodes (not necessarily of length H). Instead, from the *lifelong RL* perspective, the agent is interested in maximizing the ∞-step ahead expected return $J_{T,\infty}(\pi)$, having observed *in the past only one* episode of length T, i.e., optimizing for the future.

Multiple Importance Sampling Importance Sampling (IS, Owen 2013) allows estimating the expectation $\mu = \mathbb{E}_{x\sim P}[f(x)]$ of a function f under a *target* distribution P having samples collected with a sequence of *behavioral* distributions $(Q_j)_{j\in[\![1,J]\!]}$ such that $P \ll Q_j$, i.e., P is absolutely continuous w.r.t. Q_j, for all $j \in [\![1, J]\!]$. Let p and $(q_j)_{j\in[\![1,J]\!]}$ be the density functions corresponding to P and $(Q_j)_{j\in[\![1,J]\!]}$, then, the resulting unbiased estimator is:

$$\widehat{\mu} = \sum_{j=1}^{J} \frac{1}{N_j} \sum_{i=1}^{N_j} \beta_j(x_{ij}) \frac{p(x_{ij})}{q_j(x_{ij})} f(x_{ij}),$$

where $\{x_{ij}\}_{i=1}^{N_j} \sim Q_j$ and $(\beta_j(x))_{j\in[\![1,J]\!]}$ are a partition of the unit for every $x \in \mathcal{X}$. A common choice for the latter is to use the *balance heuristic* (BH, Veach and Guibas 1995), yielding $\beta_j(x) = \frac{N_j q_j(x)}{\sum_{k=1}^J N_k q_k(x)}$. Using BH, samples can be regarded as obtained from the mixture of the $(Q_j)_{j\in[\![1,J]\!]}$ distributions as $\Phi = \sum_{k=1}^J \frac{N_k}{N} Q_k$, with $N = \sum_{j=1}^J N_j$.

Rényi divergence Let $\alpha \in [0, \infty]$, the α-Rényi divergence between two probability distributions P and Q such that $P \ll Q$ is defined as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \int_{\mathcal{X}} p(x)^\alpha q(x)^{1-\alpha} \mathrm{d}x.$$

We denote with $d_\alpha(P\|Q) = \exp\{D_\alpha(P\|Q)\}$ the exponential α-Rényi divergence, linked to the α-moment of the importance weight, i.e., $\mathbb{E}_{x\sim Q}\left[\left(\frac{p(x)}{q(x)}\right)^\alpha\right] = d_\alpha(P\|Q)^{\alpha-1}$.

3 Lifelong Parameter-Based Policy Optimization

In this paper, we consider the Policy Optimization (PO, Deisenroth, Neumann, and Peters 2013) setting in which the policy belongs to a parametric set $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^{d_1}\}$. In particular, we focus on the *parameter-based* PO[2] in which the policy parameter $\boldsymbol{\theta}$ is sampled from a *hyper-policy* $\nu_{\boldsymbol{\rho}}$ belonging, in turn, to a parametric set $\mathcal{N}_{\mathcal{P}} = \{\nu_{\boldsymbol{\rho}} : \boldsymbol{\rho} \in \mathcal{P} \subseteq \mathbb{R}^{d_2}\}$ (Sehnke et al. 2008). As opposed to *action-based* PO in which policies $\pi_{\boldsymbol{\theta}}$ need to be stochastic for exploration, in parameter-based PO we move the stochasticity to the hyper-policy level and $\pi_{\boldsymbol{\theta}}$ can be deterministic.

Optimizing the β-step ahead expected return in eq. (1) requires, in general, considering non-stationary policies. From the PO perspective, this requirement can be fulfilled in two ways. The traditional way consists in augmenting the state x with the time t and, consequently, considering a policy of the form $\pi_{\boldsymbol{\theta}}(\cdot|(x, t))$. This approach highlights the direct dependence of the action $a_t \sim \pi_{\boldsymbol{\theta}}(\cdot|(x_t, t))$ on the time t. However, in several cases, it is convenient to track the evolution of the policy parameters $\boldsymbol{\theta}$ as a function of the time t, whose dependence might be simpler compared to that of the action. In this latter approach, the one we adopt in this work, the policy parameter is sampled from a *time-dependent* hyper-policy $\boldsymbol{\theta}_t \sim \nu_{\boldsymbol{\rho}}(\cdot|t)$ and the policy depends on the state only $\pi_{\boldsymbol{\theta}_t}(\cdot|x_t)$. We will refer to this setting as *lifelong parameter-based PO*. Refer to Figure 1 for a comparison of the graphical models of the two approaches.

In this setting, we aim at learning a hyper-policy parameter $\boldsymbol{\rho}^\star_{T,\beta}$ maximizing the *β-step ahead expected return*:

$$\boldsymbol{\rho}^\star_{T,\beta} \in \arg\max_{\boldsymbol{\rho}\in\mathcal{P}} J_{T,\beta}(\boldsymbol{\rho}) = \sum_{t=T+1}^{T+\beta} \widehat{\gamma}^t \mathbb{E}_t^{\boldsymbol{\rho}}[r_t], \tag{2}$$

where $\mathbb{E}_t^{\boldsymbol{\rho}}[\cdot]$ is a shorthand for $\mathbb{E}_{\boldsymbol{\theta}\sim\nu_{\boldsymbol{\rho}}(\cdot|t)}[\mathbb{E}_t^{\pi_{\boldsymbol{\theta}}}[\cdot]]$.

[1]The extended version of the paper is available at https://arxiv.org/abs/2112.06625.

[2]We follow the taxonomy of (Metelli et al. 2018).

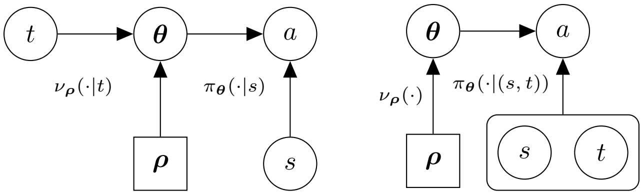


Figure 1: Graphical models of the two approaches to model non-stationarity of the environment: the non-stationarity is handled at the parameter selection level (left), the non-stationarity is handled at the action selection level (right).

4 Lifelong Parameter-Based PO via Multiple Importance Sampling

In this section, we propose an estimator for $J_{T,\beta}(\boldsymbol{\rho})$ (Section 4.1), we analyze its bias (Section 4.2) and variance (Section 4.3), and we propose a novel surrogate objective accounting for the estimation uncertainty (Section 4.4).

4.1 β-Step Ahead Expected Return Estimation

The main challenge we face in estimating $J_{T,\beta}(\boldsymbol{\rho})$ is that it requires evaluating hyper-policy $\nu_{\boldsymbol{\rho}}$ in the future, while having samples from the past only. Since the environment evolves smoothly, it is reasonable to use the past data to approximate the future dynamics and IS to correct the hyper-policy behavior mismatch from past to future. More specifically, in this section, we study how to leverage the history of the past α samples $\mathcal{H}_{T,\alpha} = (\boldsymbol{\theta}_t, r_t)_{t \in [\![T-\alpha+1, T]\!]}$ in order to estimate the β-step ahead expected return $J_{T,\beta}(\boldsymbol{\rho})$.

As a preliminary step, we illustrate the estimation of the s-step ahead expected reward $\mathbb{E}_s^{\boldsymbol{\rho}}[r_s]$. For every $s \in [\![T+1, T+\beta]\!]$ we employ the following MIS estimator that makes use of the history $\mathcal{H}_{T,\alpha}$:

$$\widehat{r}_s = \sum_{t=T-\alpha+1}^{T} \omega^{T-t} \frac{\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|s)}{\sum_{k=T-\alpha+1}^{T} \omega^{T-k} \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|k)} r_t, \quad (3)$$

where $\omega \in [0,1]$ is an exponential weighting parameter. The importance sampling correction $\frac{\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|s)}{\sum_{k=T-\alpha+1}^{T} \omega^{T-k} \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_k|k)}$ addresses the mismatch between the hyper-policies in the future $\nu_{\boldsymbol{\rho}}(\cdot|s)$ and those in the past $\nu_{\boldsymbol{\rho}}(\cdot|k)$. The reader may have noticed that these importance weights are not using the exact BH weights. Indeed, we have adapted the heuristic to include our knowledge that the environment is *smoothly changing*. With BH, each past sample would have been weighted equally whereas our sampling mixture probability, proportional to $\sum_{k=T-\alpha+1}^{T} \omega^{T-k} \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|k)$, gives more weight to recent samples thanks to the parameter ω which exponentially discounts samples as they are collected far from current time T (Jagerman, Markov, and de Rijke 2019).

Using $\widehat{r}_s$ as building block, we propose the estimator for the β-step ahead expected return $\widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$ that is obtained as the discounted sum of the s-step ahead expected reward estimators of Equation (3):

$$\widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho}) = \sum_{s=T+1}^{T+\beta} \widehat{\gamma}^s \widehat{r}_s$$

$$= \sum_{t=T-\alpha+1}^{T} \omega^{T-t} \frac{\sum_{s=T+1}^{T+\beta} \widehat{\gamma}^s \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|s)}{\sum_{k=T-\alpha+1}^{T} \omega^{T-k} \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|k)} r_t.$$

This estimator could be, in principle, directly optimized but, as common in IS-based estimators. However, we would incur in the following undesired effect. In order to increase $\widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$, the agent can either increase the probability of good actions for future policies $\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|s)$ (the numerator of the importance weight) or decrease the probability of the same good actions for past policies $\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_t|k)$ (the denominator). The latter phenomenon, akin to catastrophic forgetting, is clearly undesired, but can be easily spotted by looking at the past reward. Specifically, we propose to adjust the objective function with the return of the last α steps, called *α-step behind expected return*:

$$\breve{J}_{T,\alpha}(\boldsymbol{\rho}) = \frac{1}{C_\omega} \sum_{t=T-\alpha+1}^{T} \omega^{T-t} \breve{\gamma}^t r_t,$$

where $\breve{\gamma}^t = \breve{\gamma}^{t-T+\alpha-1}$, $C_\omega = \frac{1-\omega^\alpha}{1-\omega}$ if $\omega < 1$ otherwise $C_\omega = \alpha$. Putting all together, we obtain the objective:

$$\overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho}) = \widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho}) + \breve{J}_{T,\alpha}(\boldsymbol{\rho}).$$

4.2 Bias Analysis

In this section, we analyze the bias of estimator $\widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$, under suitable regularity conditions on the environment and on the hyper-policy model. In particular, we will require that the environment and the hyper-policy are *smoothly changing*. We formalize the intuition in the following assumptions.

Assumption 4.1 (Smoothly Changing Environment) *For every $t, t' \in \mathbb{N}$, and for every policy π it holds for some Lipschitz constant $0 \leqslant L_{\mathcal{M}} < \infty$:*

$$\left|\left(\mathbb{E}_t^\pi - \mathbb{E}_{t'}^\pi\right)[r]\right| \leqslant L_{\mathcal{M}} \left|t - t'\right|.$$

Assumption 4.2 (Smoothly Changing Hyper-policy) *For every $t, t' \in \mathbb{N}$, and for every time-dependent hyper-policy $\boldsymbol{\rho} \in \mathcal{P}$ it holds for some Lipschitz constant $0 \leqslant L_\nu < \infty$:*

$$\left\|\nu_{\boldsymbol{\rho}}(\cdot|t) - \nu_{\boldsymbol{\rho}}(\cdot|t')\right\|_1 \leqslant L_\nu \left|t - t'\right|.$$

Thus, Assumption 4.1 prescribes that executing the same policy π at different times t and t' yields expected rewards whose difference can be bounded proportionally to the time distance. A similar requirement is given by Assumption 4.2, involving the total variation distance between time-dependent hyper-policies. Under these assumptions, we provide the following bias bound. We denote with $\mathbb{E}_{T,\alpha}^{\boldsymbol{\rho}}$ the expectation under the probability distribution induced by the joint hyper-policy $\prod_{t=T-\alpha+1}^{T} \nu_{\boldsymbol{\rho}}(\cdot|t)$ in the MDP.

Lemma 4.1 *Under Assumptions 4.1 and 4.2, the bias of the estimator $\widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$, for $\omega < 1$, can be bounded as:*

$$\left|J_{T,\beta}(\boldsymbol{\rho}) - \mathbb{E}_{T,\alpha}^{\boldsymbol{\rho}}[\widehat{J}_{T,\alpha,\beta}]\right| \leqslant (L_{\mathcal{M}} + 2R_{\max}L_\nu) C_\gamma(\beta) \left(\frac{\omega}{1-\omega} + \frac{1}{1-\gamma}\right),$$

where, for $\xi \geqslant 1$, $C_\gamma(\xi) = \frac{1-\gamma^\xi}{1-\gamma}$ if $\gamma < 1$ otherwise $C_\gamma(\xi) = \xi$.

A tighter, but more intricate bias bound, and a derivation for the case $\omega = 1$ can be found in Appendix A. Some observations are in order. First, we note the role of ω in controlling the bias: the smaller ω, the smaller the bias. Second, the bound is proportional to the Lipschitz constants governing the non-starionarity of the environment and of the hyper-policy. It is worth noting that in a fully stationary setting (i.e., $L_{\mathcal{M}} = L_\nu = 0$), the estimator is unbiased.

4.3 Variance Analysis

Before showing the construction of the surrogate objective, we derive in this section a bound on the variance of $\overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$ that involves the Rényi divergence. To this purpose, we denote with $\mathbb{V}\mathrm{ar}^{\boldsymbol{\rho}}_{T,\alpha}$ the variance under the probability distribution induced by the joint hyper-policy $\prod_{t=T-\alpha+1}^{T} \nu_{\boldsymbol{\rho}}(\cdot|t)$ in the MDP.

Lemma 4.2 *The variance of the objective $\overline{J}_{T,\alpha,\beta}$ can be bounded as:*

$$\mathbb{V}\mathrm{ar}^{\boldsymbol{\rho}}_{T,\alpha}\left[\overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho})\right] \leqslant 2R^2_{\max}\Bigg(C_\gamma(\alpha)^2 + C_\gamma(\beta)^2 \times d_2\left(\sum_{s=T+1}^{T+\beta} \frac{\widehat{\gamma}^s}{C_\gamma(\beta)}\nu_{\boldsymbol{\rho}}(\cdot|s)\middle\| \sum_{t=T-\alpha+1}^{T} \frac{\omega^{T-t}}{C_\omega}\nu_{\boldsymbol{\rho}}(\cdot|t)\right)\Bigg).$$

The variance bound resembles the ones usually provided in the context of off-policy estimation and learning (e.g., Metelli et al. 2018; Papini et al. 2019; Metelli et al. 2020). The first addendum accounts for the variance of the estimator component $\check{J}_{T,\alpha}(\boldsymbol{\rho})$ that does not involve importance sampling, whereas the second refers to $\widehat{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$, based on importance sampling. Indeed, this latter term comprises the exponentiated 2-Rényi divergence between two mixture hyper-policies. Unfortunately, even in presence of convenient distributions, like Gaussians, the Rényi divergence between mixtures does not admit a closed form (Papini et al. 2019). In Appendix B, we discuss several approaches, based on variational upper-bounds, to provide a usable version of such a divergence. In the following, we report the upper-bound that we will use in practice.

Lemma 4.3 *The divergence between mixtures of Lemma 4.2 can be bounded as:*

$$d_2\left(\sum_{s=T+1}^{T+\beta} \frac{\widehat{\gamma}^s}{C_\gamma(\beta)}\nu_{\boldsymbol{\rho}}(\cdot|s)\middle\| \sum_{t=T-\alpha+1}^{T} \frac{\omega^{T-t}}{C_\omega}\nu_{\boldsymbol{\rho}}(\cdot|t)\right) \leqslant \frac{C_\omega}{C_\gamma(\beta)^2}\underbrace{\left(\sum_{s=T+1}^{T+\beta} \frac{\widetilde{\gamma}^s}{\left(\sum_{t=T-\alpha+1}^{T} \frac{\omega^{T-t}}{d_2(\nu_{\boldsymbol{\rho}}(\cdot|s)\|\nu_{\boldsymbol{\rho}}(\cdot|t))}\right)^{\frac{1}{2}}}\right)^2}_{B_{T,\alpha,\beta}(\boldsymbol{\rho})}.$$

4.4 Surrogate Objective

The direct optimization of the objective $\overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho})$ makes the hyper-policy overfit the non-stationary process on the last α steps. To allow for a better generalization on future unseen variations, following the idea of Metelli et al. (2018), we regularize the objective with the bound on the variance of Lemma 4.3. The following concentration bound, based on Cantelli's inequality, is the theoretical grounding of our surrogate objective.

Theorem 4.1 *For every $\delta \in (0,1)$, with probability at least $1-\delta$, it holds that:*

$$\mathbb{E}^{\boldsymbol{\rho}}_{T,\alpha}\left[\overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho})\right] \geqslant \overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho}) - \sqrt{\frac{1-\delta}{\delta}2R^2_{\max}\left(C_\gamma(\alpha)^2 + C_\omega B_{T,\alpha,\beta}(\boldsymbol{\rho})\right)}.$$

We are now ready to construct the surrogate objective function and show how to optimize it. Following the path of Metelli et al. (2018), we take an *uncertainty-averse* approach, by maximizing a probabilistic lower bound of the quantity of interest, i.e., the one presented in Theorem 4.1. Renaming $\lambda = \sqrt{\frac{1-\delta}{\delta}2R^2_{\max}}$, and treating it as a hyperparameter, we get the following surrogate objective:

$$\mathcal{L}_\lambda(\boldsymbol{\rho}) = \overline{J}_{T,\alpha,\beta}(\boldsymbol{\rho}) - \lambda\sqrt{C_\gamma(\alpha)^2 + C_\omega B_{T,\alpha,\beta}(\boldsymbol{\rho})}. \quad (4)$$

In order to optimize this objective, we use a policy-gradient approach that is discussed in Appendix A.1. We call this algorithm POLIS (Policy Optimization in Lifelong learning through Importance Sampling) and provide its pseudocode in algorithm 1.

Algorithm 1: Lifelong learning with POLIS

Input: steps behind α, steps ahead β, regularization λ, discount factor ω, training period h, training epochs N,

```
Initialize ν_ρ, t ← 0
while True do                                  ▷ Possibly never-ending loop
    Sample θ_t ~ ν_ρ(t)
    Collect new state s_t and reward r_t using π_θt
    if t mod h = 0 and t > 1 then
        for i ∈ {1, ..., N} do                 ▷ POLIS' training loop
            Compute J̆_t,α(ρ), J̄_t,α,β(ρ) and B_t,α,β(ρ)
            ρ ← arg max_ρ L_λ(ρ)               ▷ See eq. (4)
    t ← t + 1
```

5 Related Works

The problem of lifelong RL is not new to the community. Nevertheless, the term encapsulates problems which can have slightly differing definitions, thus hindering the comparison between existing approaches. In this section, we will briefly discuss the most relevant ones. A more complete overview can be found in (Padakandla 2021). Lifelong RL approaches handle finite-horizon settings (Hallak, Di Castro, and Mannor 2015; Ortner, Gajane, and Auer 2020) as well as infinite-horizon settings. In this second case, many works focus on the detection of abrupt changes in the dynamics (da Silva et al. 2006; Hadoux, Beynier, and Weng 2014) or on scenarios where the non-stationarity arises from switching between stationary dynamics, where the number of such dynamics is known (Choi, Yeung, and Zhang 2000). When this number is unknown, Mendez, Wang, and Eaton (2020) propose learning a factored representation of the policy composed of a shared dictionary of coefficients trained to perform well on average on the set of tasks encountered so far and of task-specific coefficients trained on the current task. Their algorithm, LPG-FTW, is therefore able to adapt quickly to new task while avoiding catastrophic forgetting.

Non-stationarity in the MDP dynamics is not only bound to LL, since it is also a core element of *continual learning*. To adapt to the evolving environment, continually learning agents need to find structure in the world to tackle new tasks by decomposing them in smaller sub-problems through function composition (Griffiths et al. 2019) or by extracting meaningful information in the form of abstract concepts (Zhang, Satija, and Pineau 2018; Francois-Lavet et al. 2019). Other approaches focus on capturing task-agnostic underlying dynamics of the world, by building auxiliary tasks like reward prediction (Jaderberg et al. 2017) or using inverse dynamics prediction (Shelhamer et al. 2017) to provide denser training signal. A general overview of continual RL approaches can be found in (Khetarpal et al. 2020).

Another relevant approach to Lifelong RL is Meta RL, which leverages past experience to learn new skills more efficiently, i.e. using a small amount of new data. Usual Meta RL algorithms can be adapted to non-stationarity by modelling the consecutive tasks as a Markov chain model (Al-Shedivat et al. 2018), using experience replay (Riemer et al. 2019) or learning a latent model of the environment which can then be predicted (Poiani, Tirinzoni, and Restelli 2021).

Lastly, more similar to our approach, is the one of Chandak et al. (2020) where the policy is trained to optimize the future predicted performance. To this end, the past performance is first of all estimated through importance sampling and then used to forecast future performance. All these steps are differentiable, which allows optimizing the policy through gradient ascent. The authors propose two algorithms, Pro-OLS, forecasting the performance using an ordinary least-squares regression and Pro-WLS, where the regression takes into account the importance weights inside a weighted least-squares. The latter reduces the variance of the estimates at the expense of adding some bias. One major difference with our approach is that their method is designed for episodic RL, where the non-stationarity arises from one episode to the next. We instead consider a truly lifelong framework, where that are no episodes and non-stationarity arises at the single step level. Our approach is different for three other reasons. First, while our estimate of the β-step ahead performance has in common with the aforementioned paper the use of importance sampling, our objective is however greatly different as we add an extra discounting parameter to control the bias due to non-stationarity and two terms, the α-step behind performance and a variance regularization. Second, our surrogate objective can be optimized at any point in time, meaning that if a significant shift in dynamics is detected, one has the opportunity to retrain the algorithm suddenly. Third, we consider a parameter-based approach in which the hyper-policy depends only upon time, the policy may thus change at every step. Chandak et al. (2020) consider an action-based approach where policy's parameters are fixed during an episode.

6 Experiments

In this section, we report the experimental evaluation of our algorithm in comparison with state-of-the-art baselines.

6.1 Lifelong Learning Framework

The schedule for a lifelong interaction with the environment is divided in two periods. In the first, which we refer to as *behavioural period*, a behavioural hyper-policy is queried to sample data in order to gather enough samples to compute the first α-step behind expected return. In the second period, referred to as *target period*, the agent continues interacting with same environment, but is now training its hyper-policy every few steps (50 in all experiments) for a given number of gradient steps (100 in all experiments). For all tasks, we set $\gamma = \omega = 1$.[3]

We consider a particular subclass of non-stationary environments, frequently encountered in practice. The state $x = (x^c, x^u)$ is decomposed into a controllable x^c and a non-controllable x^u part. The controllable part evolves according to stationary dynamics and depends on the action $P^c((x')^c|x^c, a)$. The non-controllable part instead is not affected by the action and follows non-stationary dynamics $P^u_t((x')^u|x^u)$.

Assumption 6.1 *The transition model $P = (P_t)_{t\in\mathbb{N}}$ factorizes as follows, for every $x = (x^c, x^u) \in \mathcal{X}$, $a \in \mathcal{A}$, and $t \in \mathbb{N}$:*

$$P_t(x'|x, a) = P^c((x')^c|x^c, x^u, a)P^u_t((x')^u|x^u). \tag{5}$$

Under the assumption, we can sample new trajectories from the last α steps, where the non-stationary part of the state x^u is kept fixed but the sampled policy at each time and therefore the stationary controllable part of the state x^c changes. Therefore, we have access to the value of the gradient of the α-step behind expected return by direct estimation, without requiring importance sampling.

6.2 Trading Environment

The first task is the daily trading of the EUR-USD (€/$) currency pair from the Foreign Exchange (Forex). Following (Bisi et al. 2020), we allow the agent to trade up to a fixed quantity of 100k$ USD with a per-transaction fee f of 1$. The agent has a continuous actions space in $[-1, 1]$, where 1 and -1 correspond to buy or sell with the maximum order size, while 0 corresponds to staying flat. We do not model the effect of our agent's trades on the market,[4] thus satisfying assumption 6.1. The state of the agent is composed of its actual portfolio (x^c_t, which also corresponds to its previous action) and the current rate of the currency (x^u_t). The reward is defined as $r_t = a_t(x^u_{t+1} - x^u_t) - f|a_t - x^c_t|$.

We consider three datasets of historical data, 2009-2012, 2013-2016, and 2017-2020; each period having a little more

[3]The code is available at https://github.com/pierresdr/polis.

[4]We assume that the order size of the agent is negligible w.r.t. the market liquidity.

than 1000 data points. α is set to 500 and we consider a target period of 500 steps.

Finding a satisfactory set of hyperparameters (in the sense of parameters of the algorithm itself, not parameters of the hyper-policy) can be problematic in our lifelong scenario. Indeed, here, there is no distinction between training and testing since the parameters are continually updated. Selecting hyperparameters for future interactions with the environment by evaluating past performances is thus prone to overfitting on the past performance. To account for this problem, we compare two hyperparameter selection approaches. In the first, we select the best performing hyperparameters from the dataset 2009-2012 and evaluate the selection on the other two datasets. In the second approach, we both select the hyperparameters and evaluate on the last two datasets.

The trading of the EUR-USD currency pair is a highly complex task. To give more chance to the algorithm to exploit potential patterns of the series, we provide another trading task on a simulated series. The framework is the same, only changes the underlying rate process which will now be a Vasicek process. In this scenario, the rate $(p_t)_{t\geqslant 1}$ satisfies $p_{t+1} = 0.9p_t + u_t$, where $u_t \sim \mathcal{N}(0, 1)$. On this task we will test the set of hyperparameters selected on the EUR-USD.

6.3 Dam Environment

The second environment is a water resource management problem. A dam is used to save water from rains and possibly release it to meet a certain demand for water (e.g., the needs of a town). We model the environment following (Castelletti et al. 2010; Tirinzoni et al. 2018). The inflow (e.g., rain) is the non-stationary process and the agent has obviously no impact on it, thus satisfying assumption 6.1. The mean inflow follows one of either 3 profiles given in Appendix C.2. The state observed by the agent is the day's lake level. The agent does not observe the day of the year, contrarily to (Tirinzoni et al. 2018), in order to ensure non-stationarity. The agent's action is continuous and corresponds to selecting the daily amount of water to release in order to avoid flooding and meet the demand. Considering the flooding level $F = 300$ and the daily demand for water $D = 10$, the penalty that the agent gets for each is respectively $c_F = (\max(x - F, 0))^2$ and $c_D = (\max(a - D, 0))^2$, where a is the action of the agent and x the current lake level. The final cost is a convex combination of those costs, whose weights depend on the inflow profile (see Appendix C.2). In this environment, α is set to 1000 in order to include enough years of past data in the estimator. We provide results for a target period of 500 steps. Because the results for this environment are less sensitive to the choice of hyperparameters, we only select them according to the performance given the first inflow profile.

6.4 The Hyper-Policy and Policy

We now describe the hyper-policy used in all experiments. It is composed of two modules. The first is *positional encoding* introduced in (Vaswani et al. 2017). It embeds its input, time, as a vector of Fourier basis. Therefore, it does not add learnable parameters to the hyper-policy. This module has two main advantages. First, its output dimension is free to be

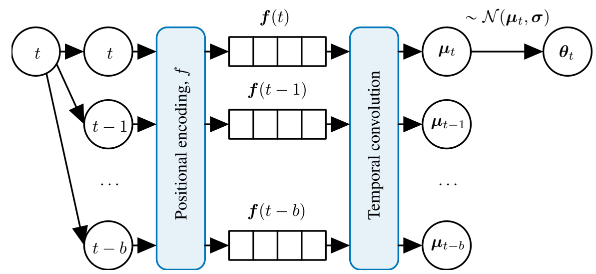


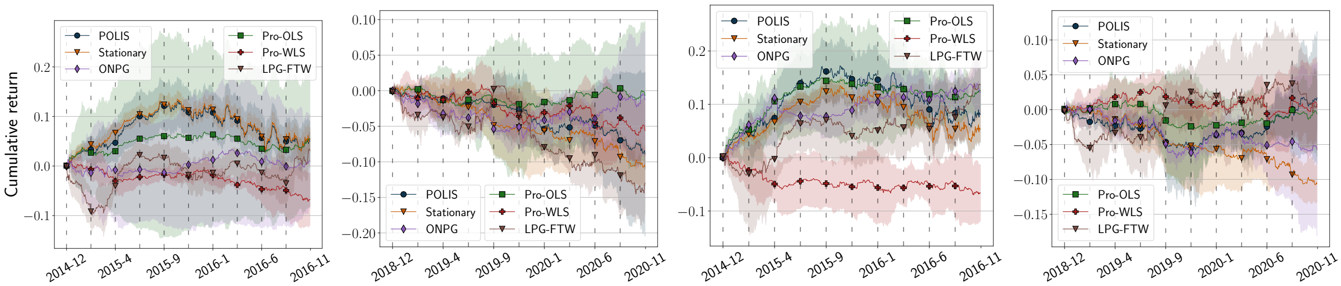
Figure 2: The hyper-policy queried at time t. First the last b times are appended to t, b the receptive field of the temporal convolution. They are fed to a positional encoding which outputs a vector $\boldsymbol{f}(t)$ for each. The latter are then fed to a temporal convolution which returns the mean $\boldsymbol{\mu}_t$ of $\boldsymbol{\theta}_t$.

	Inflow 1	Inflow 2	Inflow 3
POLIS	-2.2 ± 0.2	-1.5 ± 0.0	-3.2 ± 0.2
Stationary	-2.2 ± 0.1	-1.5 ± 0.0	-3.2 ± 0.2
ONPG	-5.1 ± 3.2	-1.4 ± 0.2	-4.1 ± 0.5
Pro-OLS	-2.6 ± 0.4	-5.2 ± 5.1	-3.8 ± 0.7
Pro-WLS	-5.5 ± 4.3	-8.5 ± 9.7	-8.4 ± 4.2
LPG-FTW	-2.3 ± 0.3	-11.9 ± 8.6	-4.7 ± 1.9

Table 1: Lifelong learning on the Dam environment for each of 3 inflow profile. Mean return on the target period and standard deviation over 3 seeds. Reported results are divided by an order of $1e3$ for aesthetic.

chosen, allowing to control the input size of the next module. Second, while time eventually becomes large, the output of positional encoding is bounded, which is a valuable property when then fed to a neural network. The second module are convolutions scanning through time. We chose convolutions as they generally excel in finding patterns in time series. Moreover, they allow processing inputs of variable length and are easily parallelizable. We use a particular type of convolutions, *temporal convolutions* (Oord et al. 2016) which preserve time causality. Obviously, the convolutions require several time-steps in order to scan through with their kernel. However our hyper-policy takes only the current time as input, $\nu_{\boldsymbol{\rho}}(\cdot|t)$. Nevertheless, we can freely decide to consider the positional encoding of t and a few previous times to reach the length of the *receptive field* of the convolutions. Its length is $b = 2^{l-1}(k-1)$, where l is the number of layers and k the kernel size of the temporal convolution. Another advantage of using temporal convolutions is that the computation of the policy parameters $\boldsymbol{\theta}$ can be made in parallel. This is an interesting property in practice as between two updates of the hyper-policy, we can already sample in parallel all the policy parameters to be used. The output of the temporal convolutions is the mean $\boldsymbol{\mu}_t$ of the normally distributed policy parameter $\boldsymbol{\theta}_t$. The standard deviation of each entry of $\boldsymbol{\theta}_t$ is not time dependent and can be either learned or fixed during training. A schematic representation of the hyper-policy is given in fig. 2.

At the policy level, in all the experiments, we use an affine



(a) Hyperparameters selected on 2009-2012.

(b) Hyperparameters selected on 2013-2016 and 2017-2020.

Figure 3: Lifelong learning on the EUR-USD currency pair. Mean cumulative returns on the target period with one standard deviation shaded area, over 3 seeds. Vertical dashed line indicate retrain.

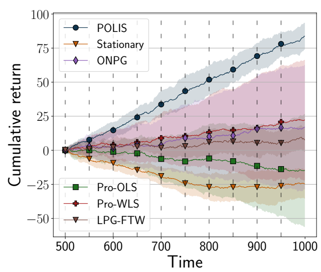


Figure 4: Lifelong learning on the Vasicek process. Mean cumulative returns on the target period with one standard deviation shaded area, over 10 seeds. Vertical dashed line indicate retrain.

policy with bounded outputs.

6.5 Baselines

The first baseline is the stationary hyper-policy which can be seen as a special case of POLIS when $\nu_{\boldsymbol{\rho}}(\cdot|t) = \nu_{\boldsymbol{\rho}}(\cdot)$. We consider this hyper-policy along with the same affine policy. Note that, although stationary between trainings, the hyper-policy's parameters are retrained every 50 steps.

We then consider baselines from the literature, including Pro-OLS and Pro-WLS (Chandak et al. 2020) and LPG-FTW (Mendez, Wang, and Eaton 2020). In their experiments, Chandak et al. (2020) use a baseline which they refer to as ONPG, replicating the idea of (Al-Shedivat et al. 2018). We also include this baseline and thank Chandak et al. (2020) for providing their code.

6.6 Results

Trading environment The cumulative returns obtained for the hyperparameters selected on 2009-2012 are given in Figure 3a. Interestingly, on the period 2013-2016, POLIS has a performance similar to the stationary policy, which is comparable or superior to baselines. On the period 2017-2020, POLIS under-performs the baselines, but the stationary one. When selecting the set of hyperparameters from the testing dataset, we obtain the results shown in Figure 3b. This time, POLIS obtains more similar performance to the baselines, closing the gap on the period 2017-2020.

The cumulative returns obtained for the trading on the Vasicek process are reported in Figure 4. On this tasks, specifically designed to highlight the smooth non-stationarity, POLIS is clearly superior to baselines, particularly the stationary one. Note also its smaller variance.

Dam environment We report the results of the experiment in Table 1. Surprisingly, out of all the baselines, the stationary hyper-policy obtains the best performance over the 3 inflows. Other baselines sometimes obtain a comparable performance but exhibit a tendency to have a higher standard deviation. Remarkably, POLIS is able to match the stationary policy's performance and variance on each inflow profile. This indicates that our approach is able to avoid extra non-stationarity in tasks where it is not needed.

7 Conclusion

In this paper, we proposed to address the *lifelong* RL problem by using a hyper-policy mapping time to policy parameters. To grasp the objective of LL, i.e., the future performance, we designed an estimator of such quantity, making use of the past collected experience via importance sampling. The estimator has a controllable bias which vanishes as the environment and the hyper-policy become stationary. Besides, we add two terms to the objective: an estimation of the past performance preventing catastrophic forgetting and a penalization based on an upper-bound on the variance, which prevents overfitting the past and favors generalization to future non-stationarity. We proposed an implementation of such hyper-policy which we tested in several scenarios, demonstrating that our approach can exploit predictable non-stationarity, control for its variance and avoid excessive non-stationarity when non necessary. Our approach tackled exploration via the stochasticity of the hyper-policy. Future work include a more principled and explicit treatment of the exploration problem in the lifelong RL setting.

References

Al-Shedivat, M.; Bansal, T.; Burda, Y.; Sutskever, I.; Mordatch, I.; and Abbeel, P. 2018. Continuous Adaptation via Meta-Learning in Nonstationary and Competitive Environments. In *ICLR 2018*.

Bisi, L.; Liotet, P.; Sabbioni, L.; Reho, G.; Montali, N.; Corno, C.; and Restelli, M. 2020. Foreign Exchange Trading: A Risk-Averse Batch Reinforcement Learning Approach. In *ICAIF '20*.

Bowerman, B. L. 1974. *Nonstationary Markov decision processes and related topics in nonstationary Markov chains*. Ph.D. thesis, Iowa State University.

Brunskill, E.; and Li, L. 2014. PAC-inspired Option Discovery in Lifelong Reinforcement Learning. In *ICML 2014*.

Castelletti, A.; Galelli, S.; Restelli, M.; and Soncini-Sessa, R. 2010. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46(9).

Chandak, Y.; Theocharous, G.; Shankar, S.; Mahadevan, S.; White, M.; and Thomas, P. S. 2020. Optimizing for the Future in Non-Stationary MDPs. *ICML 2020*.

Choi, S. P.; Yeung, D.-Y.; and Zhang, N. L. 2000. Hidden-mode markov decision processes for nonstationary sequential decision making. In *Sequence Learning*, 264–287. Springer.

da Silva, B. C.; Basso, E. W.; Bazzan, A. L. C.; and Engel, P. M. 2006. Dealing with Non-Stationary Environments Using Context Detection. In *ICML 2006*.

Deisenroth, M. P.; Neumann, G.; and Peters, J. 2013. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2(1-2).

Francois-Lavet, V.; Bengio, Y.; Precup, D.; and Pineau, J. 2019. Combined Reinforcement Learning via Abstract Representations. In *AAAI 2019*.

Garcia, A.; and Smith, R. L. 2000. Solving nonstationary infinite horizon stochastic production planning problems. *Oper. Res. Lett.*, 27(3).

Ghate, A.; and Smith, R. L. 2013. A Linear Programming Approach to Nonstationary Infinite-Horizon Markov Decision Processes. *Oper. Res.*, 61(2): 413–425.

Griffiths, T. L.; Callaway, F.; Chang, M. B.; Grant, E.; Krueger, P. M.; and Lieder, F. 2019. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29: 24–30. Artificial Intelligence.

Hadoux, E.; Beynier, A.; and Weng, P. 2014. Sequential decision-making under non-stationary environments via sequential change-point detection. In *Learning over multiple contexts (LMCE)*.

Hallak, A.; Di Castro, D.; and Mannor, S. 2015. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*.

Jaderberg, M.; Mnih, V.; Czarnecki, W. M.; Schaul, T.; Leibo, J. Z.; Silver, D.; and Kavukcuoglu, K. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *ICLR 2017*.

Jagerman, R.; Markov, I.; and de Rijke, M. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 447–455.

Khetarpal, K.; Riemer, M.; Rish, I.; and Precup, D. 2020. Towards continual reinforcement learning: A review and perspectives. *arXiv preprint arXiv:2012.13490*.

Lesner, B.; and Scherrer, B. 2015. Non-Stationary Approximate Modified Policy Iteration. In Bach, F. R.; and Blei, D. M., eds., *ICML 2015*.

Mendez, J.; Wang, B.; and Eaton, E. 2020. Lifelong policy gradient learning of factored policies for faster training without forgetting. *Advances in Neural Information Processing Systems*, 33: 14398–14409.

Metelli, A. M.; Papini, M.; Faccio, F.; and Restelli, M. 2018. Policy optimization via importance sampling. In *NeurIPS 2018*.

Metelli, A. M.; Papini, M.; Montali, N.; and Restelli, M. 2020. Importance Sampling Techniques for Policy Optimization. *Journal of Machine Learning Research*, 21(141): 1–75.

Oord, A. v. d.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Ortner, R.; Gajane, P.; and Auer, P. 2020. Variational Regret Bounds for Reinforcement Learning. In *UAI 2020*.

Owen, A. B. 2013. *Monte Carlo theory, methods and examples*. http://statweb.stanford.edu/ owen/mc/. Online book; accessed 11-April-2022.

Padakandla, S. 2021. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*, 54(6): 1–25.

Papini, M.; Metelli, A. M.; Lupo, L.; and Restelli, M. 2019. Optimistic policy optimization via multiple importance sampling. In *ICML 2019*.

Poiani, R.; Tirinzoni, A.; and Restelli, M. 2021. Meta-Reinforcement Learning by Tracking Task Non-stationarity. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2019. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *In International Conference on Learning Representations (ICLR)*.

Sehnke, F.; Osendorfer, C.; Rückstieß, T.; Graves, A.; Peters, J.; and Schmidhuber, J. 2008. Policy Gradients with Parameter-Based Exploration for Control. In Kurková, V.; Neruda, R.; and Koutník, J., eds., *ICANN 2008*.

Shelhamer, E.; Mahmoudieh, P.; Argus, M.; and Darrell, T. 2017. Loss is its own Reward: Self-Supervision for Reinforcement Learning. In *ICLR 2017*.

Silver, D. L.; Yang, Q.; and Li, L. 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *Lifelong Machine Learning, Papers from the 2013 AAAI Spring Symposium, Palo Alto, California, USA, March 25-27, 2013*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Tirinzoni, A.; Sessa, A.; Pirotta, M.; and Restelli, M. 2018. Importance Weighted Transfer of Samples in Reinforcement Learning. In *ICML 2018*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS 2017*, 5998–6008.

Veach, E.; and Guibas, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In Mair, S. G.; and Cook, R., eds., *SIGGRAPH 1995*.

Zhang, A.; Satija, H.; and Pineau, J. 2018. Decoupling dynamics and reward for transfer learning. In *ICLR (workshop) 2018*.