
Beyond the Policy Gradient Theorem for Efficient Policy Updates in Actor-Critic Algorithms

Romain Laroche
Microsoft Research Montréal

Rémi Tachet des Combes
Microsoft Research Montréal

Abstract

In Reinforcement Learning, the optimal action at a given state is dependent on policy decisions at subsequent states. As a consequence, the learning targets evolve with time and the policy optimization process must be efficient at unlearning what it previously learnt. In this paper, we discover that the policy gradient theorem prescribes policy updates that are slow to unlearn because of their structural symmetry with respect to the value target. To increase the unlearning speed, we study a novel policy update: the gradient of the cross-entropy loss with respect to the action maximizing q , but find that such updates may lead to a decrease in value. Consequently, we introduce a modified policy update devoid of that flaw, and prove its guarantees of convergence to global optimality in $\mathcal{O}(t^{-1})$ under classic assumptions. Further, we assess standard policy updates and our cross-entropy policy updates along six analytical dimensions. Finally, we empirically validate our theoretical findings.

q , then at least as many policy gradient updates with respect to opposite values will be needed to return back to a uniform policy. We argue that unlearning in $\mathcal{O}(n)$ is too slow for efficient Reinforcement Learning (RL, Sutton and Barto, 1998). Indeed, the optimal action at a given state is dependent on policy decisions at subsequent states. As a consequence, the learning targets evolve with time and an efficient policy optimization process must be fast at unlearning what it previously learnt. We show further that the unlearning slowness of policy gradient updates critically compounds with the number of such chained decisions as well as with decaying learning rates and/or state visitations. The structural flaw in the policy gradient lies in its symmetry with respect to the q -function.

We therefore look for alternative solutions. As described in Agarwal et al. (2020), the direct parametrization update also applies a classic policy gradient update, but follows it with a projection on the probability simplex. This projection breaks the symmetry, like a wall preventing the parameters from going further forward but still allowing them to go backwards. Unfortunately, an adaptation of the direct parametrization to non-tabular settings, *i.e.* to function approximation, remains an open problem because the projection cannot be readily differentiated.

To overcome this limitation, we study a novel policy update that improves the unlearning speed: the cross-entropy policy update. It consists in updating the parameters with the gradient of the cross-entropy loss between the output of a softmax parametrization and the current local optimal action $a_q(s) = \operatorname{argmax}_{a \in \mathcal{A}} q(s, a)$. This policy update displays a consistent empiric convergence to global optimality in our experiments. But unfortunately, our analysis reveals that such updates may at times lead to a decrease in value, which is a serious dent in its theoretical grounding. We conjecture its convergence and global optimality to be true, but they remain an open problem.

In the meantime, we propose to alter the cross-entropy loss gradient in order to guarantee monotonicity of the value function. We prove that the resulting modified

1 INTRODUCTION

The policy gradient theorem derived for the first time in Williams (1992) is seminal to all the policy gradient theory (Sutton et al., 1999; Konda and Tsitsiklis, 1999; Ahmed et al., 2019; Kumar et al., 2019; Zhang et al., 2020a; Qiu et al., 2021), and the actor-critic algorithmic innovations (Mnih et al., 2016; Silver et al., 2016; Vinyals et al., 2019). In this paper, we discover that if, starting from a uniform policy, n policy gradient updates have been performed with respect to some values

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

cross-entropy policy update converges in $\mathcal{O}(t^{-1})$ to a global optimal under the set of assumption/conditions made in Laroche and Tachet des Combes (2021). We pursue our theoretical analysis with an overview of the main policy updates used in the literature, and analyze them along six axes: convergence to global optimality, asymptotic convergence rates, sensitivity to the gravity well exposed in Mei et al. (2020a), unlearning speed, compatibility with stochastic updates, and adaptability to function approximation. Due to space limitations, proofs for all propositions and theorems were moved to Appendix A.

Finally, we empirically validate our analysis on diverse finite MDPs. The results show that the cross entropy softmax updates are as efficient as the direct parametrization updates on hard planning tasks on which policy gradient methods fail to converge to optimality in a reasonable amount of time. Due to space limitations, some details regarding implementation choices, applicative domains, and experimental results were moved to Appendix B. Moreover, the code is attached to the proceedings.

Our contributions are summarized below:

- We identify the slow unlearning behaviour of policy updates following the policy gradient theorem.
- We develop two novel policy updates based on the cross-entropy loss tackling the aforementioned issue.
- We assess standard policy updates and our policy updates along six analytical dimensions.
- We empirically validate our theoretical findings.

2 FAMILIES OF POLICY UPDATES

The objective for an agent consists in maximizing the sum of discounted rewards:

$$\mathcal{J}(\pi) \doteq \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_t \middle| S_0 \sim p_0, S_{t+1} \sim p(\cdot | S_t, A_t), A_t \sim \pi(\cdot | S_t), R_t \sim r(\cdot | S_t, A_t) \right], \quad (1)$$

where state S_0 is sampled from the initial state distribution p_0 , and at each time step $t \geq 0$, action A_t is sampled from the current policy π , reward R_t is sampled according to the reward kernel r , and next state S_{t+1} is sampled according to the transition kernel p . $0 \leq \gamma < 1$ is the discount factor ensuring that the infinite sum of rewards converges.

In this paper, we will consider policies π_θ , parametrized by θ , that get recursively updated as follows:

$$\theta' \leftarrow U(\theta, d, q, \eta), \quad (2)$$

where $d(s)$ is classically the state-visit distribution induced by the current policy π_θ , but may be any state distribution following the generalized policy update in Laroche and Tachet des Combes (2021), q is the current state-action value function estimate for π_θ , and η is a learning rate scalar.

2.1 PG updates: $U_{\text{PG}}-U_{\text{PG-SM}}-U_{\text{PG-ES}}$

A natural approach is to take the gradient of $\mathcal{J}(\pi_\theta)$ with respect to its parameters. This corresponds to the update prescribed by the policy gradient theorem:

$$U_{\text{PG}}(\theta, d, q, \eta) \doteq \theta + \eta \sum_{s \in \mathcal{S}} d(s) \sum_{a \in \mathcal{A}} q(s, a) \nabla_\theta \pi_\theta(a|s), \quad (3)$$

where $\nabla_\theta \pi_\theta$ denotes the gradient of π_θ with respect to its parameters θ . In practice, the vast majority of practitioners use update U_{PG} in Eq. (3) with a softmax parametrization:

$$\pi_\theta(a|s) \doteq \frac{\exp(f_\theta(s, a))}{\sum_{a' \in \mathcal{A}} \exp(f_\theta(s, a'))} \quad (4)$$

$$\nabla_\theta \pi_\theta(a|s) = \pi_\theta(a|s) [\nabla_\theta f_\theta(s, a) - \sum_{a'} \nabla_\theta f_\theta(s, a') \pi_\theta(a'|s)] \quad (5)$$

where $f_\theta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is typically a neural network parametrized by its weights θ . We let $U_{\text{PG-SM}}$ denote the update resulting from this parametrization.

Recently, Mei et al. (2020a) discovered that softmax’s policy gradient has two issues: slow convergence (aka *damping*) and sensitivity to parameter initialization (aka *gravity well*). They propose the escort transform to address them:

$$\pi_\theta(a|s) \doteq \frac{|f_\theta(s, a)|^p}{\|f_\theta(s, \cdot)\|_p^p} \quad (6)$$

$$\nabla_\theta \pi_\theta(a|s) \propto \pi_\theta(a|s)^{1-\frac{1}{p}} [\nabla_\theta f_\theta(s, a) - \sum_{a'} \nabla_\theta f_\theta(s, a') \pi_\theta(a'|s)^{1/p} \pi_\theta(a|s)^{1-\frac{1}{p}}] \quad (7)$$

where p is the hyperparameter for the transformation, usually set to 2, $\|\cdot\|_p$ denotes the p -norm, and \propto means proportional to, hiding factors in p and $\|f_\theta(s, \cdot)\|_p$. We let $U_{\text{PG-ES}}$ denote the update resulting from this parametrization.

We will argue in Section 3.4 that updates of the form of U_{PG} generally take too much time to unlearn their past steps. Consequently, we investigate other policy updates.

2.2 Direct parametrization update: U_{DI}

The direct parametrization is arguably the simplest one:

$$\pi_\theta(a|s) \doteq f_\theta(s, a) \quad \nabla_\theta \pi_\theta = \nabla_\theta f_\theta, \quad (8)$$

but, since $f_\theta(s, \cdot)$ must live on the probability simplex $\Delta_{\mathcal{A}}$, an orthogonal projection on $\Delta_{\mathcal{A}}$ is required after each update U_{PG} :

$$U_{\text{DI}}(\theta, d, q, \eta) \doteq \text{Proj}_{\Delta_{\mathcal{A}}}[U_{\text{PG}}(\theta, d, q, \eta)] \quad (9)$$

The direct parametrization has been studied quite extensively in the context of finite MDPs. We will argue in Section 3.6 that U_{DI} cannot be readily applied with function approximation. We thus investigate other policy updates.

2.3 Cross-entropy update: U_{CE}

Instead of the gradient of the objective function, we propose to follow the gradient derived from the classification problem of selecting the best action according to the current value function q : the cross-entropy loss on a softmax parametrization (see Eq. (4)):

$$U_{\text{CE}}(\theta, d, q, \eta) \doteq \theta + \eta \sum_{s \in \mathcal{S}} d(s) \nabla_{\theta} \log \pi_{\theta}(a_q(s)|s), \quad (10)$$

$$(U_{\text{CE}})_{s,a} = \theta + \eta d(s) (\mathbb{1}_{a=a_q(s)} - \pi_{\theta}(a|s)), \quad (11)$$

where $a_q(s) = \arg\max_{a \in \mathcal{A}} q(s, a)$ is the action that maximizes q in state s . The cross-entropy update can be seen as a soft version of the SARSA algorithm (reached as η tends to infinity). Unfortunately, the cross-entropy update does not guarantee a monotonous increase of the value. Indeed, an imbalance of the policy for suboptimal actions may lead to an increase in the policy for some of them, and potentially to a decrease in the policy value.

Proposition 1 (Non-monotonicity of U_{CE}). *Updating a policy with U_{CE} may decrease its value.*

We will consider U_{CE} because it works well in practice but the non-monotonicity of its value compromises our proofs of convergence and optimality.

2.4 Modified cross-entropy update: U_{MCE}

In order to solve the monotonicity issue with U_{CE} , we propose to modify its update such that all suboptimal actions get penalised equally, thereby correcting the imbalance responsible for the non-monotonicity of the update:

$$(U_{\text{MCE}})_{s,a_q(s)} \doteq \theta + \eta d(s) (1 - \pi_{\theta}(a_q(s)|s)) \quad (12)$$

$$(U_{\text{MCE}})_{s,a \neq a_q(s)} \doteq \theta - \frac{\eta d(s)}{|\mathcal{A}| - 1} (1 - \pi_{\theta}(a_q(s)|s)). \quad (13)$$

Note that updating $a \neq a_q(s)$ is not necessary but allows $\sum_a \theta_{s,a}$ to remain constant over time, and thus prevents the weights from diverging artificially. We prove its monotonicity under the true updates.

Proposition 2 (Monotonicity of U_{MCE}). *Updating a policy with U_{MCE} increases its value.*

2.5 Cross-entropy related updates

In the context of transfer and multitask learning, Parisotto et al. (2016) train a set of experts on various tasks and then distill the learnt policies into a single agent via the cross-entropy loss. The idea of increasing the probability of greedy actions, while decreasing the probability of bad actions, has also been used in the field of Conservative Policy Iteration (Kakade and Langford, 2002; Pirodda et al., 2013; Scherrer, 2014). Finally, the Pursuit family of algorithms introduced in the automata literature also shares some common ground with the idea (Thathachar and Sastry, 1986; Agache and Oommen, 2002).

3 THEORETICAL ANALYSIS

In this section, we analyze the five policy updates presented in Section 2 across six dimensions summarized in Table 1. We first check whether there exists proof of their convergence to global optimality in Section 3.1. In Section 3.2, we look at their asymptotic convergence rates. In Section 3.3, we investigate their sensitivity to the gravity well (Mei et al., 2020a). In Section 3.4, we define the unlearning setting and assess the updates' unlearning speed. In Section 3.5, we discuss the compatibility of the update rules to stochastic updates. Finally, we discuss their deep implementations in Section 3.6.

3.1 Convergence and global optimality

The convergence and global optimality of $U_{\text{PG-SM}}$ has been proved under different sets of assumptions/conditions in Agarwal et al. (2020); Mei et al. (2020b); Laroche and Tachet des Combes (2021). The convergence and global optimality of $U_{\text{PG-ES}}$ has been proved in Mei et al. (2020a). The convergence and global optimality of U_{DI} has been proved under different sets of assumptions/conditions in Agarwal et al. (2020); Laroche and Tachet des Combes (2021).

While U_{CE} displays a consistent empiric convergence to optimality in our experiments, the non-monotonicity of its value function is a dent in its theoretical grounding. We conjecture its convergence and global optimality to be true, but they remain an open problem.

Using tools from Laroche and Tachet des Combes (2021), we prove the convergence of U_{MCE} to the global optimum on finite MDPs:

Theorem 3 (Convergence and optimality of U_{MCE}). *Starting from an arbitrary set of param-*

Id	Name	Conv. & Opt.	Rates	Gravity Well	Unlearning*	Stochastic	Deep Impl.
$U_{\text{PG-SM}}$	Policy gradient softmax	yes	$\mathcal{O}(t^{-1})$	strong	slow*	yes	yes
$U_{\text{PG-ES}}$	Policy gradient escort	yes	$\mathcal{O}(t^{-1})$	weak	slow*	yes	yes
U_{DI}	Direct parametrization	yes	exact	none*	fast*	no	no
U_{CE}	Cross-entropy softmax*	no*	$\mathcal{O}(t^{-1})$	none*	fast*	no	yes
U_{MCE}	Modified C-E softmax*	yes*	$\mathcal{O}(t^{-1})^*$	none*	fast*	no	yes

Table 1: Summary of the theoretical analysis. The five updates are evaluated across six dimensions. Conv. & Opt. relates to the existence of a proof of the convergence and global optimality of the update (see Section 3.1). Rates measures the asymptotic speed of convergence (see Section 3.2). Gravity Well reports how strong they endure a gravitational pull (see Section 3.3). Unlearning evaluates the speed at which an update can recover from its own past updates (see Section 3.4). Stochastic reports whether the update is compatible with stochastic updates (see Section 3.5). Deep Impl. assesses whether the method can be implemented with function approximation (see Section 3.6). The entries have a red font when the answer is considered as disadvantageous. The entries are associated with a star when the answer is novel.

ters θ_0 , we consider the process induced by $\theta_{t+1} = U_{\text{MCE}}(\theta_t, d_t, q_t, \eta_t)$, where $q_t = q_{\pi_{\theta_t}}$ is the state-action value of current policy π_{θ_t} . Then, under the assumption that the optimal policy is unique, the condition that each state s is updated with weights that sum to infinity over time: $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$, is necessary and sufficient to guarantee that the sequence of value functions (q_t) converges to global optimality.

The necessary and sufficient condition on the infinite sum of weights is identical to that for the convergence of $U_{\text{PG-SM}}$ and U_{DI} in the same framework (Laroche and Tachet des Combes, 2021). The optimal policy uniqueness assumption is not required for $U_{\text{PG-SM}}$ and U_{DI} , but we conjecture that this is only required for technical reasons and that the theorem holds true even without uniqueness. We run in Section 4.2 an experiment to empirically support this conjecture.

3.2 Asymptotic convergence rates

Asymptotic convergence rates for $U_{\text{PG-SM}}$ is well documented to be in $\mathcal{O}(t^{-1})$ (Mei et al., 2020b; Laroche and Tachet des Combes, 2021). Mei et al. (2020a) prove a convergence rate in $\mathcal{O}(t^{-1})$ for $U_{\text{PG-ES}}$. Laroche and Tachet des Combes (2021) prove that U_{DI} will converge to an optimal policy in a finite number of steps.

The convergence of the softmax parametrization under the cross-entropic loss has been studied in the past with a rate of $\mathcal{O}(t^{-1})$ (Soudry et al., 2018). Under the assumption that U_{CE} converges, its rates should be the same. Theorem 4 shows the same holds true for U_{MCE} .

Theorem 4 (Convergence rates of U_{MCE}). *The process, assumption, and condition defined in Theorem 3 guarantee that the sequence of value functions (q_t) asymptotically converges in $\mathcal{O}\left(\left(\sum_{t'=0}^t \eta_{t'} \min_{s \in \mathcal{S}} d_{t'}(s)\right)^{-1}\right)$.*

This is the same rate as $U_{\text{PG-SM}}$ and it reaches rates in $\mathcal{O}(t^{-1})$, when the learning rates is kept constant and when off-policy updates prevent $d_t(s) \geq d_{\perp} > 0$ from decaying.

The various update rules all converge at least in $\mathcal{O}(t^{-1})$. Given that our setting of interest is RL, where the minimal theoretical cumulative regret is known to be $\mathcal{O}(\log t)$, there is no point in converging faster than $\mathcal{O}(t^{-1})$. For this reason, we consider all the updates converge *fast enough*.

3.3 Gravity well

Mei et al. (2020a) identify the softmax gravity well issue, *whereby gradient ascent trajectories are drawn towards suboptimal corners of the probability simplex and subsequently slowed in their progress toward the optimal vertex*. A condition for this to happen is for the action a_{q_t} maximizing q_t not to incur the maximal update step in policy space. Formally, the following condition guarantees the absence of gravity well:

$$a_{q_t}(s) \in \operatorname{argmax}_{a \in \mathcal{A}} (\pi_{t+1}(a|s) - \pi_t(a|s)). \quad (14)$$

Theorem 5 analyses condition 14 for all five updates $U_{\text{PG-SM}}$, $U_{\text{PG-ES}}$, U_{DI} , U_{CE} , and U_{MCE} .

Theorem 5 (Gravity well). *U_{DI} and U_{MCE} are guaranteed to satisfy Eq. (14). $U_{\text{PG-SM}}$, $U_{\text{PG-ES}}$, and U_{CE} may transgress Eq. (14). Furthermore, $U_{\text{PG-SM}}$ may not even satisfy $\pi_{t+1}(a_{q_t}(s)|s) - \pi_t(a_{q_t}(s)|s) \geq 0$.*

With policy gradient softmax $U_{\text{PG-SM}}$, we see in the proofs of Th. 5 that if $\pi_{\theta}(a_{q_t}(s)|s)$ is close to 0, it is possible, and even rather easy, for $U_{\text{PG-SM}}$ to induce a larger update step in policy space for a suboptimal action than for $a_{q_t}(s)$ itself. This can last for a significant amount of time because $\pi_{\theta}(a_{q_t}(s)|s)$ will only observe a small update, hence many steps will be needed to escape the gravity well, as Mei et al. (2020a) empirically

observed. Even worse, it may happen that the policy for $a_{q_t}(s)$ decreases.

Similarly to $U_{\text{PG-SM}}$, $U_{\text{PG-ES}}$ may induce a larger update step for a suboptimal action than for the optimal one (only when $p > 1$). However, that effect is dampened by the $1 - \frac{1}{p}$ power on $\pi_\theta(a|p)$ seen in Eq. (7), and the setting for condition (14) not to be satisfied is much more restricted. Consequently, the gravity well effect exists but is not strong enough to compromise the update efficiency, as predicted by Mei et al. (2020a).

Regarding U_{CE} , it is rather easy to construct examples where condition (14) is not satisfied. However, we argue that this differs from the gravity well issue, as the action benefiting from a higher update step is not the action with the highest policy. Our experiment confirms that U_{CE} does not suffer from the gravity well.

Finally, condition (14) is guaranteed to be satisfied with policy updates U_{DI} and U_{MCE} .

3.4 Unlearning what has been learnt

In RL, the optimal action at a given state is dependent on policy decisions at subsequent states. As a consequence, the learning targets evolve with time and the policy optimization process must be efficient at unlearning what it previously learnt. We use two settings to investigate this property: the unlearning setting measures how fast each algorithm is to recover from bad preliminary q targets, and the domino setting evaluates how this compounds when the bad targets are chained. Our analysis reveals that the symmetry of U_{PG} with respect to the q target strongly slows down unlearning and that this pitfall compounds geometrically in hard planning tasks.

3.4.1 Unlearning setting (constant weights)

To evaluate the ability to unlearn convergence stemming from bad preliminary q targets, we consider a single state MDP with two actions and a tabular parametrization $\theta \in \mathbb{R}^2$. Starting from an initial set of parameters θ_0 , we apply n updates with $(q(a_1) = 1, q(a_2) = 0)$ and then measure the number n' of ‘‘opposite’’ updates, i.e. with $(q(a_1) = 0, q(a_2) = 1)$ necessary to unlearn these steps, that is to recover a policy such that $\pi_{\theta_{n+n'}}(a_1) \leq \pi_{\theta_0}(a_1)$.

Theorem 6 (Unlearning setting – constant weights). *In the setting where η is constant:*

- (i) Under assumptions enunciated below—and verified by $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$, U_{PG} needs $n' \geq n$ updates.
- (ii) U_{DI} needs $n' = \min\{n; \lceil \frac{1}{\eta} \rceil\}$ updates.
- (iii) U_{CE} and U_{MCE} need $n' \leq 2 + \frac{1}{\eta} \log(1 + 2\eta n)$ updates.

Letting $(\theta)_1$ and $(\theta)_2$ denote θ 's components, Theorems 6(i), 7(i), and 8(i) require the following mild assumptions:

$$\text{Invariance w.r.t. } \theta: \pi_{\theta_1} = \pi_{\theta_2} \implies \nabla_{\theta} \pi_{\theta_1} = \nabla_{\theta} \pi_{\theta_2}$$

$$\text{Monotonicity: } \pi_{\theta}(a_1) \geq \pi_0(a_1) \implies \begin{cases} (\nabla_{\theta} \pi_{\theta}(a_1))_1 \geq 0 \\ (\nabla_{\theta} \pi_{\theta}(a_1))_2 \leq 0 \end{cases}$$

$$\text{Concavity: } \pi_{\theta}(a_1) \geq \pi_0(a_1) \implies \begin{cases} (\nabla_{\pi}(\nabla_{\theta} \pi_{\theta}(a_1)))_1 \leq 0 \\ (\nabla_{\pi}(\nabla_{\theta} \pi_{\theta}(a_1)))_2 \geq 0 \end{cases}$$

Invariance w.r.t. θ states that any two parameters implementing the same policy have equal gradients—our theorems do not deal with overparametrization. Monotonicity states that the gradient is monotonic with π : higher components imply higher policy. It is fairly standard to expect each parameter to be assigned to an action. Concavity states that, as the policy for a_1 grows, the absolute value of its gradient with respect to θ diminishes. Since $\pi_{\theta}(a_1)$ is a function of θ from \mathbb{R}^2 to $[0, 1]$, it is expected that the gradient diminishes as $\pi_{\theta}(a_1)$ grows to 1.

Theorem 6 states that to unlearn n steps towards a certain action, U_{PG} requires a number of updates n' at least as large as the number of steps taken initially. As shown in our experiments, this may significantly slow down convergence to the optimal policy. In contrast, U_{DI} updates display no asymptotic dependency in n , while U_{CE} and U_{MCE} updates require a logarithmic number of steps to recover.

3.4.2 Unlearning setting (decaying weights)

In practice, the learning weight $\eta_t d_t(s)$ applied to the gradient is likely to decay over time, either because the learning rate η_t is required to decay to guarantee convergence of stochastic policy updates, and/or because the state density $d_t(s)$ will decay (depending on s) as the policy converges.

In order to model this scenario, we reproduce the unlearning setting with a decaying learning rate: $\forall t \geq 1, \eta_t = \frac{\eta_1}{\sqrt{t}}$ for $\eta_1 > 0$. From time $t = 1$ to $t = n$, updates with η_t and $(q(a_1) = 1, q(a_2) = 0)$ are performed, from time $t = n + 1$ onwards, updates with η_t and $(q(a_1) = 0, q(a_2) = 1)$ are applied. We then study n' such that $\pi_{\theta_{n+n'}}(a_1) \leq \pi_{\theta_0}(a_1)$.

Theorem 7 (Unlearning setting – decaying weights). *In the setting where $\eta_t = \frac{\eta_1}{\sqrt{t}}$:*

- (i) Under the assumptions enunciated above—verified by $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$, U_{PG} needs $n' \geq 3n - 4\sqrt{n} + 1$.
- (ii) U_{DI} updates require at most n' equal to $\min \left\{ 3n + 4\sqrt{n} + 1 ; \left(\frac{1}{\eta_1} + 1 \right)^2 + \sqrt{n} \left(2 + \frac{2}{\eta_1} \right) \right\}$.

(iii) U_{CE} and U_{MCE} updates require at most n' equal to

$$\left(4 + \frac{\log(1+4\eta_1\sqrt{n})}{2\eta_1}\right)^2 + \sqrt{n} \left(8 + \frac{\log(1+4\eta_1\sqrt{n})}{\eta_1}\right).$$

All updates are severely affected by the decaying weights. Since the decay can stem from the learning rate actually decaying and/or from the state visit $d_t(s)$ decreasing as the behavioural policy converges, these worsened recovery rates are an argument in favor of (i) using expected updates in action space (Ciosek and Whiteson, 2018) to allow the use of a constant learning rate, and (ii) performing off-policy updates (Laroche and Tachet des Combes, 2021) to properly control the policy update state visitation density.

3.4.3 Domino setting

Next, we argue that the number of updates compounds exponentially with the horizon of the task in the following sense: for a q estimate to flip in one state, it is required that all future states have flipped beforehand. To illustrate this effect, we consider the domino setting: several binary decisions are taken sequentially in states $s \in \mathcal{S} = \{s_k\}_{k \in [|\mathcal{S}|]}$. The q -function is artificially designed as follows:

$$q(s_{|\mathcal{S}|}, a_1) = 0 \quad \text{and} \quad q(s_{|\mathcal{S}|}, a_2) = 1$$

$$\text{if } \pi_{\theta_t}(a_2|s_{k+1}) \leq \pi_{\theta_0}(a_2|s_{k+1}), \begin{cases} q(s_k, a_1) = 1 \\ q(s_k, a_2) = 0, \end{cases}$$

$$\text{otherwise, } \begin{cases} q(s_k, a_1) = 0 \\ q(s_k, a_2) = 1. \end{cases}$$

Intuitively, the decision in state s_{k+1} needs to be correct for the gradient in state s_k to point in the right direction. When this happens, we say that state s_{k+1} has flipped. We say that the domino setting has been solved in t steps when $\pi_{\theta_t}(a_2|s_1) > \pi_{\theta_0}(a_2|s_1)$.

Theorem 8 (Domino setting). *Under the assumption that η and $d(s_k) > 0$ are constant, in order to solve the setting,*

- (i) U_{PG} updates require at least $2^{|\mathcal{S}|-1}$ steps.
- (ii) U_{DI} updates require at most $1 + \frac{|\mathcal{S}|-1}{\eta}$ steps.
- (iii) U_{CE} and U_{MCE} updates require at most:

$$\frac{32e^{8\eta+3}}{\eta^3} (|\mathcal{S}|-1) \log(|\mathcal{S}|-1) \text{ steps.}$$

Disclaimer: The domino setting is a thought experiment for the backpropagation of an optimal policy through a decision chain. However, in practice, the updates will not flip this way. Moreover, its implementation requires a reward function of amplitude $2^{|\mathcal{S}|-1}$. We acknowledge that the domino setting makes things look worse than they really are, but we claim, with

the validation of our empirical experiments, that the unlearning slowness of the policy gradient updates are a critical burden for hard planning tasks.

3.5 Stochastic versus expected updates

In Section 3.4, we showed drawbacks of U_{PG} 's symmetry property w.r.t. the q function. It can, however, also be an asset, as it allows for stochastic updates. U_{DI} 's projection on the simplex breaks the symmetry but, fortunately, only at the frontier of its domain. So, U_{DI} also allows for stochastic updates as long as they obey the classic Robbins-Monro conditions. In contrast, the cross-entropy updates U_{CE} and U_{MCE} are asymmetric everywhere. Moreover, they ignore the q -value gaps, and are thus biased with stochastic updates. Therefore, they must use expected updates (Ciosek and Whiteson, 2018). Given Theorem 7's analysis, we argue that applying expected updates is actually necessary no matter the update type, so that constant learning rates can be used, and efficient unlearning performance attained.

3.6 Implementation with function approximation

To the best of our knowledge, all actor-critic algorithms with function approximations have implemented policy gradient updates U_{PG} with or without the true state visitation density: often omitting the discount factor in the state visitation (Thomas, 2014; Nota and Thomas, 2020; Zhang et al., 2020b), and sometimes not correcting off-policy updates (Wang et al., 2017; Espeholt et al., 2018; Vinyals et al., 2019; Schmitt et al., 2020; Zahavy et al., 2020). A vast majority of the implementations also chose by default the softmax parametrization $U_{\text{PG-SM}}$. However, given their shape, implementing cross-entropy updates U_{CE} and U_{MCE} should be straightforward. The assessment of their actual efficiency is left for future work. Implementing $U_{\text{PG-ES}}$ ought not be of any challenge. Nevertheless, since it was already an issue with tabular representations, the instability of $U_{\text{PG-ES}}$ when the parameters cross 0 may be a source of concern with function approximation.

The direct parametrization U_{DI} remains to be discussed for function approximation. Sparsemax (Martins and Astudillo, 2016; Laha et al., 2018) could be seen as an implementation, but, since it omits the projection step, it misses one of its fundamental feature: asymmetry. Indeed, if an action is already assigned a 0 probability, a proper direct update would either decrease it and then project it back to 0, or increase it. In contrast, Sparsemax exhibits a null gradient and therefore immobility in both directions. Adapting U_{DI}

to function approximation remains an open problem to this date.

4 EMPIRICAL ANALYSIS

This section intends to compare the five updates studied until now in RL experiments (with $p = 2$ for $U_{\text{PG-ES}}$). In RL, many confounding factors such as exploration or the nature of the environment may compromise the empirical analysis. We will endeavour to control these factors by:

Investigating three exploratory and off-policy updates schemes designed within the J&H algorithm¹ (Laroche and Tachet des Combes, 2021), where ϵ_t denotes the probability to give control to Hyde, a pure exploratory agent, at the beginning of each trajectory, and where o_t denotes the probability of updating the actor with a sample collected with Hyde (therefore likely to be off-policy):

- **NOEXPLO**: no exploration: $\epsilon_t = 0$ and $o_t = 0$,
- **LOWOFFPOL**: exploration and low off-policy updates: $\epsilon_t = \min\{1; \frac{10}{\sqrt{t}}\}$ and $o_t = \min\{1; \frac{10}{\sqrt{t}}\}$,
- **HI OFFPOL**: exploration and high off-policy updates: $\epsilon_t = \min\{1; \frac{10}{\sqrt{t}}\}$ and $o_t = \frac{1}{2}$.

Evaluating on three RL environments

- **Random MDPs**: procedurally generated environments where efficient planning is unlikely to matter and where stochasticity plays an important role.
- **Chain domain**: a deterministic domain with rewards misleading towards suboptimal policies, and thus where planning is the main challenge. The chain domain evaluates the unlearning ability of the updates.
- **Cliff domain**: a slight modification of the chain domain in order to create gravity wells. In addition to their unlearning abilities, the cliff domain should evaluate the updates' resilience to the gravity well pitfall.

4.1 Random MDPs experiments

A Random MDP environment with $|\mathcal{S}| = 100$ and $|\mathcal{A}| = 4$ is procedurally generated at the start of each run. The transition from each state action pair

¹The precise J&H implementation is detailed in Appendix B.2.

stochastically connects to two states uniformly sampled from \mathcal{S} and with probability generated from a uniform partition of the segment $[0,1]$. In most cases, the obtained MDP is strongly connected and exploration is barely an issue. However, stochasticity is strong and an accurate q estimate is necessary to find a policy with a good performance.

We evaluate the policies with the number of steps (each step is a transition and an update) they need to reach a normalized performance $\bar{\mathcal{J}}_\pi$ equal to 95% of the gap between the performance of the optimal policy π_* and that of the uniform one $\pi_u(\cdot|\cdot) = \frac{1}{|\mathcal{A}|}$, formally $\bar{\mathcal{J}}_\pi = \frac{\mathcal{J}(\pi) - \mathcal{J}(\pi_u)}{\mathcal{J}(\pi_*) - \mathcal{J}(\pi_u)}$.

The result of the Random MDPs experiments is presented on Figure 2, where we display the number of steps to reach $\bar{\mathcal{J}}(\pi) = 0.95$ versus the learning rate for the actor. The three exploration/off-policiness settings are shown on three separate subfigures: NOEXPLO in 2a, LOWOFFPOL in 2b, and HI OFFPOL in 2c.

The similarity of NOEXPLO and LOWOFFPOL confirms that the exploration plays a minimal role in this setting. More generally, the policy gradient updates $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$ are quite unchanged by the presence of off-policy updates, maybe indicating that their slowness to change policies makes them less likely to follow fine q optimality. It is also worth noting that $U_{\text{PG-ES}}$ has the narrowest learning rate bandwidth inside which it is efficient: lower, the updates are too slow, higher, the updates get too large and induce divergence because of the *bouncing* behaviour in 0. U_{DI} gets the best performance in all settings.

U_{DI} , U_{CE} , and U_{MCE} all converge to SARSA when η tends to ∞ and by extrapolation, we may imagine where their curves would meet and therefore SARSA's performance. Thus, we observe that all the policy update algorithms do much better than SARSA on NOEXPLO and LOWOFFPOL. U_{CE} and U_{MCE} perform equally, but their strong similarity to SARSA make them the worst on NOEXPLO and LOWOFFPOL. Indeed, the stochasticity in the environment makes the q predictions of the critic unstable, which prevents the cross-entropy updates to converge. Off-policy updates help because they allow to quickly fix bad q prediction. With high off-policy updates, U_{CE} and U_{MCE} perform much better with a wide range for the learning rate.

4.2 Chain experiment

Framing the domino setting in an RL task would have been possible, but we opted for a less processed environment, already existing in the literature. The chain domain, depicted on Figure 1, implements a deterministic walk along a line, starting from s_0 to $s_{|\mathcal{S}|-1}$. In

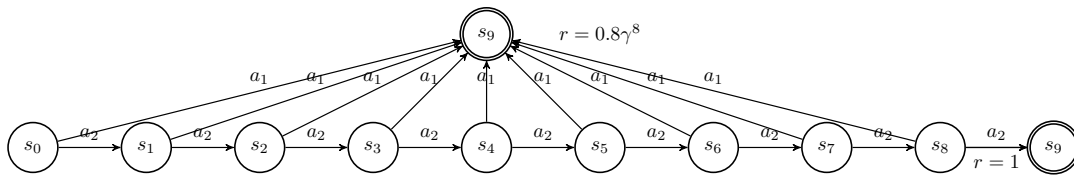


Figure 1: Deterministic chain with $|\mathcal{S}| = 10$. Initial state is s_0 . Reward is 0 except when entering final state s_9 , which is represented with two icons for clarity but is a single state. $r(\cdot, a_2)$ is set such that $q(s_0, a_1) = 0.8q_*(s_0, a_2)$.

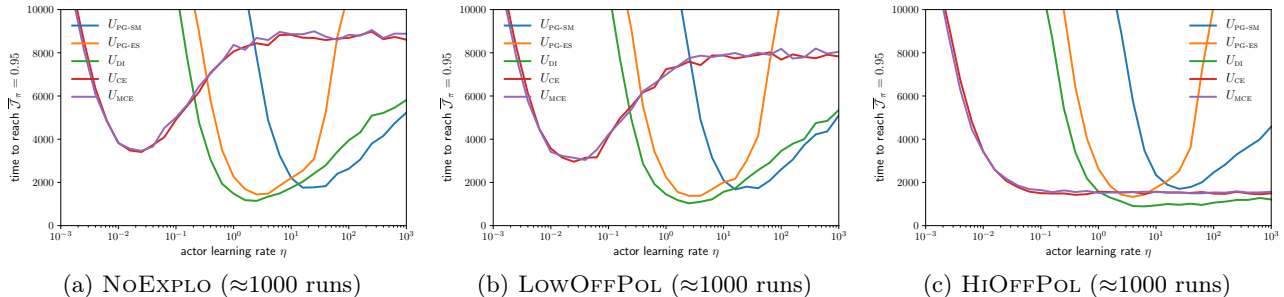


Figure 2: Random MDPs: number of steps to obtain performance equal to 95% of the gap between π_* and π_u .

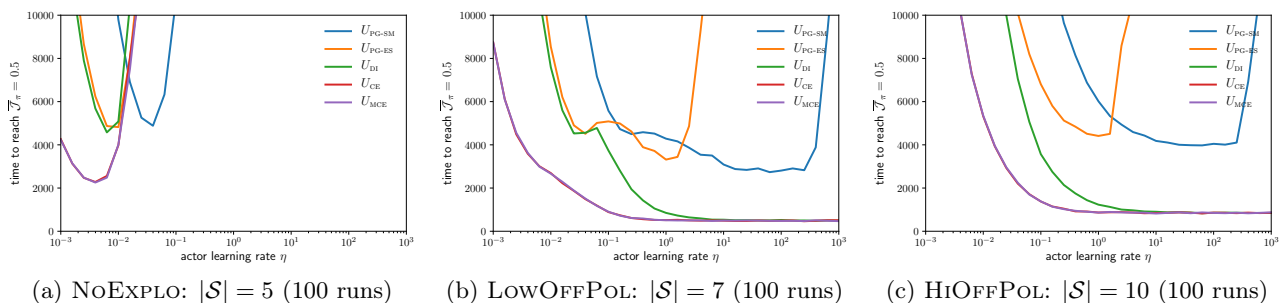


Figure 3: Chain: number of steps to obtain performance equal to 50% of the gap between π_* and π_s .

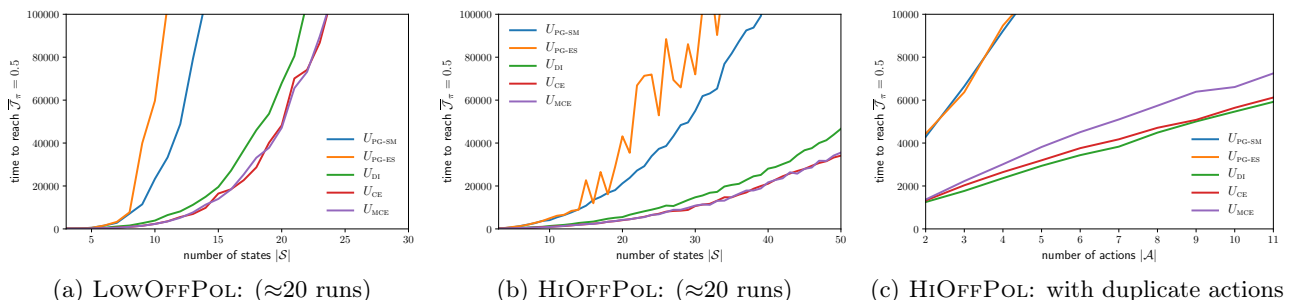


Figure 4: Chain: number of steps to obtain performance equal to 50% of the gap between π_* and π_s .

any state, it is possible to jump off the chain (action a_1) and get an immediate reward, but the optimal policy consists in walking (action a_2) throughout the chain. We expect the PUs to first guide the agent towards a_1 , and require exploration and updates to propagate the optimal policy $\pi_*(a_2|s) \approx 1$ and value through the chain. Thus, starting from the end of the chain, the policy in each state will need to be flipped in order to choose the best action at the first state.

We evaluate the policies with the number of steps (each step is a transition and an update) they need to reach a normalized performance $\bar{\mathcal{J}}_\pi$ that is equal to half the gap between that of the optimal policy π_* and that of the suboptimal one $\pi_s(a_1|\cdot) = 1$, formally $\bar{\mathcal{J}}_\pi = \frac{\mathcal{J}(\pi) - \mathcal{J}(\pi_s)}{\mathcal{J}(\pi_*) - \mathcal{J}(\pi_s)}$.

Similarly to the Random MDPs, we first look at the incidence of the actor learning rate for each update rule. The results are presented in Figure 3. First, we notice

that the size of the chain had to be adapted to each setting. Indeed, with no exploration and no off-policy updates, policy updates tends to completely converge to the suboptimal policy and stop updating the subsequent states much more easily. In all the settings, the cross-entropy updates U_{CE} and U_{MCE} yield the best results even better than U_{DI} . In LOWOFFPOL (Figure 3b) and HIOFFPOL (Figure 3c), this might just be a hyperparameter shift, but in NOEXPLO (Figure 3a), the advantage is real, which was unexpected to us since it unlearns a bit more slowly in theory. This might be due to the fact that it can both learn and unlearn fast but still explore. In contrast, either U_{DI} has a small learning rate, and it is slow to unlearn, or it has a large learning rate and it completely stops exploring too fast. This analysis seems confirmed by the fact that U_{DI} yields a performance similar to that of U_{PG-SM} and U_{PG-ES} , suggesting it never gets the benefit of its projection’s step ability to break the symmetry of the policy gradient.

We conduct an additional experiment where we observe how each update behaves when the length of the chain grows (Figures 4a and 4b). We set their best learning rate: $\eta_{PG-SM} = 10$, $\eta_{PG-ES} = 1$, $\eta_{DI} = 1$, $\eta_{CE} = 1$, and $\eta_{MCE} = 1$ for LOWOFFPOL and HIOFFPOL. We observe that U_{PG-SM} and U_{PG-ES} have approximately the same behaviour and are much slower to converge to the optimal policy. U_{PG-ES} has more visible variance because it diverges sometimes (and less runs have been performed for this experiment). With sufficient exploration, U_{DI} , U_{CE} , and U_{MCE} have the same behaviour granted that their learning rate is set sufficiently high (setting $\eta_{DI} = 1$ was slightly low and this is the reason why U_{DI} is a bit slower).

To further investigate it, we run our chain environment ($|\mathcal{S}| = 10$) where optimal actions have been duplicated $|\mathcal{A}| - 1$ times. The results are displayed on Figure 5. The results show no convergence delay for U_{CE} , as compared with U_{DI} , and a slight delay for U_{MCE} . Overall, this experiment supports our conjecture that the uniqueness assumption is an artefact of our proof technique and not a structural flaw of the policy update.

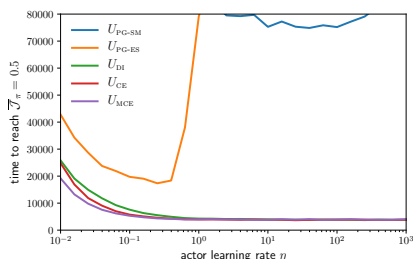


Figure 5: Cliff experiment HIOFFPOL: $|\mathcal{S}| = 7$.

4.3 Cliff experiment

The cliff experiment is similar to the chain except it includes a third action: a_3 jumps and leads to the abyss: $r(s, a_3) = 0$ and terminates. It has been designed to create gravity wells: the policy will first converge to a_1 , and will eventually be able to converge to optimal a_2 with enough exploration and updates. Like with the chain domain, we report the time to reach $\bar{\mathcal{J}}_\pi = 0.5$.

The results are presented in Figure 4c. As expected, despite the cliff being rather short: $|\mathcal{S}| = 7$, U_{PG-SM} is very slow to converge to the optimal policy, which we interpret as a manifestation of its gravity well. U_{PG-ES} does significantly better but still hits a performance wall with high learning rates because of its divergence behaviour. With its best learning rate, U_{PG-ES} shows more or less the same relative performance gap with the best updates as in the chain domain. We thereby attribute the gap to its unlearning slowness rather than to a gravity well effect.

U_{DI} , U_{CE} , and U_{MCE} display the best performance in a similar fashion than in the chain domain at the exception that U_{CE} and U_{MCE} are not behaving exactly identically (but still similarly). Indeed, with small learning rates, U_{MCE} does a bit better than U_{CE} . We naturally attribute this to its ability to perform monotonic updates that should help to converge faster when starting from a bad parameter initialization, because of past convergence to the suboptimal solution.

5 CONCLUSION

In this paper, we identified the unlearning slowness implied by policy gradient updates in actor-critic algorithms. We proposed several alternatives to these updates: the direct parametrization, and two novel cross-entropy-based updates, including one for which we prove convergence to optimality at a rate of $\mathcal{O}(t^{-1})$, under classic assumptions. We further extend the analysis to the study of their ability to enter a gravity well, their unlearning speed, and various implementation constraints. Finally, we empirically validate our theoretical findings on finite MDPs.

In the future, we would like to extend convergence/optimality proofs of policy updates to the stochastic and approximate case, as this kind of results is starting to emerge for softmax policy gradient updates (Zhang et al., 2021, 2022). We would also like to study other non policy gradient policy updates, and find some that solve the both the policy gradient unlearning slowness and the stochasticity issue related to the cross-entropy updates. Finally, we would like to investigate the relationship of these policy updates with the function approximators used in complex tasks.

Bibliography

- Agache, M. and Oommen, B. J. (2002). Generalized pursuit learning schemes: new families of continuous and discretized learning automata. IEEE Trans. Syst. Man Cybern. Part B, 32(6):738–749.
- Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2020). Optimality and approximation with policy gradient methods in markov decision processes. In Abernethy, J. and Agarwal, S., editors, Proceedings of 3rd Conference on Learning Theory (COLT), volume 125 of Proceedings of Machine Learning Research, pages 64–66. PMLR.
- Ahmed, Z., Le Roux, N., Norouzi, M., and Schuurmans, D. (2019). Understanding the impact of entropy on policy optimization. In Proceedings of the 37th International Conference on Machine Learning (ICML), pages 151–160. PMLR.
- Ciosek, K. and Whiteson, S. (2018). Expected policy gradients. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, volume 32.
- Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. (2018). IMPALA: scalable distributed deep-rl with importance weighted actor-learner architectures. In Proceedings of the 35th International Conference on Machine Learning (ICML).
- Kakade, S. M. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In Sammut, C. and Hoffmann, A. G., editors, ICML, pages 267–274. Morgan Kaufmann.
- Konda, V. R. and Tsitsiklis, J. N. (1999). Actor-critic algorithms. In Proceedings of the 12th Advances in Neural Information Processing Systems (NeurIPS).
- Kumar, H., Koppel, A., and Ribeiro, A. (2019). On the sample complexity of actor-critic method for reinforcement learning with function approximation. arXiv preprint arXiv:1910.08412.
- Laha, A., Chemmengath, S. A., Agrawal, P., Khapra, M., Sankaranarayanan, K., and Ramaswamy, H. G. (2018). On controllable sparse alternatives to softmax. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, Proceedings of the 31st Advances in Neural Information Processing Systems (NeurIPS), volume 31. Curran Associates, Inc.
- Laroche, R. and Tachet des Combes, R. (2021). Dr jekyll and mr hyde: The strange case of off-policy policy updates. In Proceedings of the 34th Advances in Neural Information Processing Systems (NeurIPS, to appear).
- Laroche, R., Trichelair, P., and Tachet des Combes, R. (2019). Safe policy improvement with baseline bootstrapping. In Proceedings of the 36th International Conference on Machine Learning (ICML).
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR, poster).
- Martins, A. and Astudillo, R. (2016). From softmax to sparsemax: A sparse model of attention and multi-label classification. In Proceedings of the 33rd International Conference on Machine Learning (ICML), pages 1614–1623. PMLR.
- Mei, J., Xiao, C., Dai, B., Li, L., Szepesvari, C., and Schuurmans, D. (2020a). Escaping the gravitational pull of softmax. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, Proceedings of the 33rd Advances in Neural Information Processing Systems (NeurIPS), volume 33, pages 21130–21140. Curran Associates, Inc.
- Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. (2020b). On the global convergence rates of softmax policy gradient methods. In Proceedings of the 37th International Conference on Machine Learning (ICML), pages 6820–6829. PMLR.
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (ICML).
- Nadjahi*, K., Laroche*, R., and Tachet des Combes, R. (2019). Safe policy improvement with soft baseline bootstrapping. In Proceedings of the 17th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD).
- Nota, C. and Thomas, P. S. (2020). Is the policy gradient a gradient? In Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS).
- Parisotto, E., Ba, J. L., and Salakhutdinov, R. (2016). Actor-mimic: Deep multitask and transfer reinforcement learning.
- Pirotta, M., Restelli, M., Pecorino, A., and Calandriello, D. (2013). Safe policy iteration. In ICML (3), volume 28 of JMLR Workshop and Conference Proceedings, pages 307–315. JMLR.org.
- Qiu, S., Yang, Z., Ye, J., and Wang, Z. (2021). On finite-time convergence of actor-critic algorithm.

- IEEE Journal on Selected Areas in Information Theory, 2(2):652–664.
- Scherrer, B. (2014). Approximate policy iteration schemes: A comparison.
- Schmitt, S., Hessel, M., and Simonyan, K. (2020). Off-policy actor-critic with shared experience replay. In Proceedings of the 37th International Conference on Machine Learning (ICML), pages 8545–8554. PMLR.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T. P., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. Nature.
- Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning (ICML), pages 387–395. PMLR.
- Simão, T. D., Laroche, R., and Tachet des Combes, R. (2020). Safe policy improvement with estimated baseline bootstrapping. In Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS, in review).
- Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. Journal of Machine Learning Research, 19(70):1–57.
- Sutton, R. S. and Barto, A. G. (1998). Reinforcement Learning: An Introduction. The MIT Press.
- Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 12th Advances in Neural Information Processing Systems (NeurIPS).
- Thathachar, M. A. L. and Sastry, P. S. (1986). Estimator algorithms for learning automata.
- Thomas, P. (2014). Bias in natural actor-critic algorithms. In Proceedings of the 31st International Conference on Machine Learning (ICML).
- Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, Ç., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in starcraft II using multi-agent reinforcement learning. Nature.
- Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. (2017). Sample efficient actor-critic with experience replay. In Proceedings of the 5th International Conference on Learning Representations (ICLR).
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3–4):229–256.
- Zahavy, T., Xu, Z., Veeriah, V., Hessel, M., Oh, J., van Hasselt, H. P., Silver, D., and Singh, S. (2020). A self-tuning actor-critic algorithm. In Laroche, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, Proceedings of the 33rd Advances in Neural Information Processing Systems (NeurIPS), volume 33, pages 20913–20924. Curran Associates, Inc.
- Zhang, K., Koppel, A., Zhu, H., and Basar, T. (2020a). Global convergence of policy gradient methods to (almost) locally optimal policies. SIAM Journal on Control and Optimization, 58(6):3586–3612.
- Zhang, S., Laroche, R., van Seijen, H., Whiteson, S., and Tachet des Combes, R. (2020b). A deeper look at discounting mismatch in actor-critic algorithms. arXiv preprint arXiv:2010.01069.
- Zhang, S., Tachet, R., and Laroche, R. (2022). On the chattering of sarsa with linear function approximation.
- Zhang, S., Tachet des Combes, R., and Laroche, R. (2021). Global optimality and finite sample analysis of softmax off-policy actor critic under state distribution mismatch. arXiv preprint arXiv:2111.02997.

A THEORY

A.1 Notations and preliminaries

A.1.1 Softmax parametrization: $U_{\text{PG-SM}}$

For completeness, we reproduce the computations for the softmax parametrization present in e.g. [Agarwal et al. \(2020\)](#):

$$\nabla_{\theta}\pi_{\theta}(a|s) = \nabla_{\theta} \left(\frac{e^{f_{\theta}(s,a)}}{\sum_{a'} e^{f_{\theta}(s,a)}} \right) \quad (15)$$

$$= \nabla_{\theta} f_{\theta}(s,a) \frac{e^{f_{\theta}(s,a)}}{\sum_{a'} e^{f_{\theta}(s,a)}} - \sum_{a''} e^{f_{\theta}(s,a)} \nabla_{\theta} f_{\theta}(s,a'') \frac{e^{f_{\theta}(s,a'')}}{(\sum_{a'} e^{f_{\theta}(s,a')})^2} \quad (16)$$

$$= \pi_{\theta}(s,a) \left[\nabla_{\theta} f_{\theta}(s,a) - \sum_{a''} \nabla_{\theta} f_{\theta}(s,a'') \pi_{\theta}(s,a'') \right]. \quad (17)$$

This leads to the following update step for $U_{\text{PG-SM}}$ with a tabular parametrization: $\theta \in \mathbb{R}^{\mathcal{A}}$ and $f_{\theta}(s,a) = \theta_a$. In that case, $(\nabla_{\theta} f_{\theta}(s,a))_a = 1$ and its other coordinates are 0. This gives:

$$(U_{\text{PG-SM}}(\theta))_{s,a} - (\theta)_{s,a} \propto \sum_{a' \in \mathcal{A}} q(s,a') \nabla_{\theta} \pi_{\theta}(a'|s) \quad (18)$$

$$= \pi_{\theta}(s,a) q(s,a) - \sum_{a'} \pi_{\theta}(s,a') q(s,a') \quad (19)$$

$$= \pi_{\theta}(s,a) \text{adv}(s,a). \quad (20)$$

A.1.2 Escort parametrization: $U_{\text{PG-ES}}$

Let us compute the policy gradients for the escort parametrization, assuming for the sake of simplicity that for all a , $f_{\theta}(s,a) \geq 0$ (accounting for the sign simply amounts to multiplying any gradient $\nabla_{\theta} f_{\theta}(s,a)$ by $\text{sign} f_{\theta}(s,a)$):

$$\nabla_{\theta}\pi_{\theta}(a|s) = \nabla_{\theta} \left(\frac{|f_{\theta}(s,a)|^p}{\sum_{a'} |f_{\theta}(s,a')|^p} \right) \quad (21)$$

$$= p \nabla_{\theta} f_{\theta}(s,a) \frac{|f_{\theta}(s,a)|^{p-1}}{\sum_{a'} |f_{\theta}(s,a')|^p} - \sum_{a''} |f_{\theta}(s,a)|^p p \nabla_{\theta} f_{\theta}(s,a'') \frac{|f_{\theta}(s,a'')|^{p-1}}{(\sum_{a'} |f_{\theta}(s,a')|^p)^2} \quad (22)$$

$$= p \nabla_{\theta} f_{\theta}(s,a) \frac{\pi_{\theta}(s,a)^{1-\frac{1}{p}}}{(\sum_{a'} |f_{\theta}(s,a')|^p)^{1/p}} - \pi_{\theta}(s,a) \sum_{a''} p \nabla_{\theta} f_{\theta}(s,a'') \frac{|f_{\theta}(s,a'')|^{p-1}}{\sum_{a'} |f_{\theta}(s,a')|^p} \quad (23)$$

$$= \frac{p}{\|f_{\theta}(s,\cdot)\|_p} \pi_{\theta}(s,a)^{1-\frac{1}{p}} \left[\nabla_{\theta} f_{\theta}(s,a) - \sum_{a''} \nabla_{\theta} f_{\theta}(s,a'') \pi_{\theta}(s,a)^{1/p} \pi_{\theta}(s,a'')^{1-\frac{1}{p}} \right]. \quad (24)$$

We now compute the update step for the escort update $U_{\text{PG-ES}}$ with a tabular parametrization: $\theta \in \mathbb{R}^{\mathcal{A}}$ and $f_{\theta}(s,a) = \theta_a$. In that case, $(\nabla_{\theta} f_{\theta}(s,a))_a = 1$ and its other coordinates are 0. This gives:

$$(U_{\text{PG-ES}}(\theta))_{s,a} - (\theta)_{s,a} \propto \sum_{a' \in \mathcal{A}} q(s,a') \nabla_{\theta} \pi_{\theta}(a'|s) \quad (25)$$

$$= \frac{p}{\|f_{\theta}(s,\cdot)\|_p} \left[\pi_{\theta}(s,a)^{1-\frac{1}{p}} (1 - \pi_{\theta}(s,a)) q(s,a) - \sum_{a' \neq a} \pi_{\theta}(s,a') \pi_{\theta}(s,a)^{1-\frac{1}{p}} q(s,a') \right] \quad (26)$$

$$= \frac{p}{\|f_{\theta}(s,\cdot)\|_p} \pi_{\theta}(s,a)^{1-\frac{1}{p}} \left[q(s,a) - \sum_{a'} \pi_{\theta}(s,a') q(s,a') \right] \quad (27)$$

$$= \frac{p}{\|f_{\theta}(s,\cdot)\|_p} \pi_{\theta}(s,a)^{1-\frac{1}{p}} \text{adv}(s,a). \quad (28)$$

A.2 Convergence, optimality, and rates proofs (Sections 2, 3.1, and 3.2)

Proposition 1 (Non-monotonicity of U_{CE}). *Updating a policy with U_{CE} may decrease its value.*

Proof. Consider an MDP with 3 actions $\mathcal{A} = \{a_1, a_2, a_3\}$. Assume the parameters in state s are:

$$\theta_{s,a_1} = 10 \quad \theta_{s,a_2} = 0 \quad \theta_{s,a_3} = 0,$$

and the value function is:

$$q(s, a_1) = 0.99 \quad q(s, a_2) = 1 \quad q(s, a_3) = 0.$$

Let us choose $\eta = d(s) = 1$ for simplicity. Then, update U_{CE} , defined as $\theta'_{s,a} \doteq \theta_{s,a} + \mathbb{1}_{a=a_q(s)} - \pi_{\theta_t}(a|s)$, leads to:

$$\theta'_{s,a_1} \approx 10 - 1 = 9 \quad \theta'_{s,a_2} \approx 0 + 1 = 1 \quad \theta'_{s,a_3} \approx 0 - 0 = 0,$$

meaning that $\pi_{\theta'}(a_2|s) \approx e^2 \pi_{\theta}(a_3|s)$, while $\pi_{\theta'}(a_3|s) \approx e \pi_{\theta}(a_3|s)$, inducing a disadvantageous update, given how $q(s, a_2)$ is closer to $q(s, a_1)$ than $q(s, a_3)$ is. Numerically we find that:

$$v_{\pi_{\theta}}(s) \approx 0.98996 > 0.98988 \approx v_{\pi_{\theta'}}(s),$$

which confirms the result. \square

Proposition 2 (Monotonicity of U_{MCE}). *Updating a policy with U_{MCE} increases its value.*

Proof. Let us fix any $s \in \mathcal{S}$. We recall that

$$(\theta_{t+1})_{s,a_q(s)} \doteq (\theta_t)_{s,a_q(s)} + \eta_t d_t(s) \nabla_{\theta} \log \pi_{\theta_t}(a_q(s)|s) \quad (29)$$

$$= (\theta_t)_{s,a_q(s)} + \eta_t d_t(s) (1 - \pi_{\theta_t}(a_q(s)|s)) \quad (30)$$

$$\forall a \neq a_q(s), \quad (\theta_{t+1})_{s,a} \doteq (\theta_t)_{s,a} - \frac{1}{|\mathcal{A}| - 1} \eta_t d_t(s) (1 - \pi_{\theta_t}(a_q(s)|s)) \quad (31)$$

$$\text{with } \begin{cases} \pi_{\theta_t}(a|s) = \frac{\exp((\theta_t)_{s,a})}{\sum_{a' \in \mathcal{A}} \exp((\theta_t)_{s,a'})} \\ q_t(s, a) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_t \mid S_0 = s, A_0 = a, S_{t+1} \sim p(\cdot | S_t, A_t), \right. \\ \left. A_t \sim \pi_{\theta_t}(\cdot | S_t), R_t \sim r(\cdot | S_t, A_t) \right] \\ a_q(s) = \operatorname{argmax}_{a \in \mathcal{A}} q_t(s, a). \end{cases} \quad (32)$$

Let $u_t \doteq \eta_t d_t(s) (1 - \pi_{\theta_t}(a_q(s)|s))$. For all $a \neq a_q(s)$:

$$\pi_{t+1}(a|s) = \frac{\exp((\theta_{t+1})_{s,a})}{\sum_{a' \in \mathcal{A}} \exp((\theta_{t+1})_{s,a'})} \quad (33)$$

$$= \frac{\exp((\theta_t)_{s,a} - \frac{u_t}{|\mathcal{A}| - 1})}{\exp((\theta_t)_{s,a_q(s)} + u_t) + \sum_{a' \neq a_q(s)} \exp((\theta_t)_{s,a'} - \frac{u_t}{|\mathcal{A}| - 1})} \quad (34)$$

$$= \frac{\exp((\theta_t)_{s,a})}{\exp\left((\theta_t)_{s,a_q(s)} + u_t + \frac{u_t}{|\mathcal{A}| - 1}\right) + \sum_{a' \neq a_q(s)} \exp((\theta_t)_{s,a'})} \quad (35)$$

$$\leq \frac{\exp((\theta_t)_{s,a})}{\exp((\theta_t)_{s,a_q(s)}) + \sum_{a' \neq a_q(s)} \exp((\theta_t)_{s,a'})} \quad (36)$$

$$= \pi_t(a|s). \quad (37)$$

We prove now that the new policy is advantageous by rearranging the policy masses:

$$\text{adv}_t(s, \pi_{t+1}) \doteq \sum_{a \in \mathcal{A}} (\pi_{t+1}(a|s) - \pi_t(a|s)) q_t(s, a) \quad (38)$$

$$= (\pi_{t+1}(a_q(s)|s) - \pi_t(a_q(s)|s)) q_t(s, a_q(s)) + \sum_{a \neq a_q(s)} (\pi_{t+1}(a|s) - \pi_t(a|s)) q_t(s, a) \quad (39)$$

$$= \sum_{a \neq a_q(s)} \underbrace{(\pi_{t+1}(a|s) - \pi_t(a|s))}_{\leq 0 \text{ from Eq. (37)}} \underbrace{(q_t(s, a) - q_t(s, a_q(s)))}_{\leq 0 \text{ from } a_q(s)\text{'s optimality}} \quad (40)$$

$$\geq 0. \quad (41)$$

which is true for all s and therefore allows to apply the policy improvement theorem to conclude the proof. \square

Theorem 3 (Convergence and optimality of U_{MCE}). *Starting from an arbitrary set of parameters θ_0 , we consider the process induced by $\theta_{t+1} = U_{\text{MCE}}(\theta_t, d_t, q_t, \eta_t)$, where $q_t = q_{\pi_{\theta_t}}$ is the state-action value of current policy π_{θ_t} . Then, under the assumption that the optimal policy is unique, the condition that each state s is updated with weights that sum to infinity over time: $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$, is necessary and sufficient to guarantee that the sequence of value functions (q_t) converges to global optimality.*

Proof. We use the same structure of proof as Laroche and Tachet des Combes (2021). Monotonicity of value functions has been proven in Proposition 2. Next, we prove the convergence of the value functions q_t .

Corollary 9 (Convergence under U_{MCE}). *The sequence of values (q_t) converges to some value function: $q_{\infty} \doteq \lim_{t \rightarrow \infty} q_t$.*

Proof. By the monotonicity and boundedness of the state value functions, the monotone convergence theorem guarantees the sequence (v_t) converges to $v_{\infty} \doteq \lim_{t \rightarrow \infty} v_t$. Applying Bellman's equation then proves the existence of $q_{\infty} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ that is the limit of the sequence of q_t . \square

We now show that condition $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$ is sufficient for optimality.

Lemma 10 (Optimality under U_{MCE} (sufficiency)). *It is sufficient to assume that $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$ to guarantee optimality of update U_{MCE} : $q_{\infty} = q_{\star} \doteq \max_{\pi \in \Pi} q_{\pi}$.*

Proof. Let us assume that $q_{\infty} < q_{\star}$, then, by the policy improvement theorem, there must be some state s for which an advantage $q_{\infty}(s, a_{\top}) - v_{\infty}(s) = \epsilon(s) > 0$ over π_t exists, with $a_{\top} \in \mathcal{A}_{\top}(s) = \text{argmax}_{a \in \mathcal{A}} q_{\infty}(s, a)$. Let us define the state value-gap $\delta(s) := q_{\infty}(s, a_{\top}) - \max_{a_{\perp} \in \mathcal{A}_{\perp}(s)} q_{\infty}(s, a_{\perp}) > 0$, with $\mathcal{A}_{\perp}(s) := \mathcal{A} / \mathcal{A}_{\top}(s)$.

Since we proved that $q_t \rightarrow_{t \rightarrow \infty} q_{\infty}$, there exists t_0 such that for all $t \geq t_0$ and $a \in \mathcal{A}$, $q_{\infty}(s, a) - q_t(s, a) \leq \frac{\delta(s)}{2}$. This guarantees two things for any $t \geq t_0$:

$$\forall a \in \mathcal{A}_{\top}(s), q_t(s, a) \geq q_{\infty}(s, a_{\top}) - \frac{\delta(s)}{2}, \quad (42)$$

$$\forall a \in \mathcal{A}_{\perp}(s), q_t(s, a) \leq q_{\infty}(s, a_{\top}) - \delta(s). \quad (43)$$

Let us bound the advantage function from Eq. (40):

$$\text{adv}_t(s, \pi_{t+1}) = \sum_{a \in \mathcal{A}} (\pi_{t+1}(a|s) - \pi_t(a|s)) (q_t(s, a) - q_t(s, a_q(s))), \quad (44)$$

with $a_q(s) \doteq \operatorname{argmax}_{a \in \mathcal{A}} q_t(s, a) \in \mathcal{A}_\top(s)$. We have:

$$\operatorname{adv}_t(s, \pi_{t+1}) = \sum_{a \in \mathcal{A}_\top(s)} \underbrace{(\pi_{t+1}(a|s) - \pi_t(a|s))}_{\leq 0 \text{ from Eq. (37)}} \underbrace{(q_t(s, a) - q_t(s, a_q(s)))}_{\leq 0 \text{ from } a_q(s)\text{'s optimality}} \quad (45)$$

$$+ \sum_{a \in \mathcal{A}_\perp(s)} (\pi_{t+1}(a|s) - \pi_t(a|s))(q_t(s, a) - q_t(s, a_q(s))) \quad (46)$$

$$\geq \sum_{a \in \mathcal{A}_\perp(s)} \underbrace{(\pi_{t+1}(a|s) - \pi_t(a|s))}_{\leq 0 \text{ from Eq. (37)}} (q_t(s, a) - q_t(s, a_q(s))) \quad (47)$$

$$\geq \sum_{a \in \mathcal{A}_\perp(s)} (\pi_{t+1}(a|s) - \pi_t(a|s))(q_\infty(s, a_\top) - \delta(s) - q_\infty(s, a_\top) + \frac{\delta(s)}{2}) \quad (48)$$

$$\geq \frac{\delta(s)}{2} \sum_{a \in \mathcal{A}_\perp(s)} (\pi_t(a|s) - \pi_{t+1}(a|s)) \quad (49)$$

$$\geq \frac{\delta(s)}{2} \sum_{a \in \mathcal{A}_\perp(s)} \left(\frac{\exp((\theta_t)_{s,a})}{\sum_{a' \in \mathcal{A}} \exp((\theta_t)_{s,a'})} - \frac{\exp((\theta_t)_{s,a})}{\exp((\theta_t)_{s,a_q(s)} + \frac{u_t|\mathcal{A}|}{|\mathcal{A}|-1}) + \sum_{a' \neq a_q(s)} \exp((\theta_t)_{s,a'})} \right) \quad (50)$$

where $u_t \doteq \eta_t d_t(s) (1 - \pi_{\theta_t}(a_q(s)|s))$ (from Eq. (35)). We proceed further:

$$\operatorname{adv}_t(s, \pi_{t+1}) \geq \frac{\delta(s)}{2} \sum_{a \in \mathcal{A}_\perp(s)} \frac{\exp((\theta_t)_{s,a} + (\theta_t)_{s,a_q(s)} + \frac{u_t|\mathcal{A}|}{|\mathcal{A}|-1}) - \exp((\theta_t)_{s,a} + (\theta_t)_{s,a_q(s)})}{(\sum_{a' \in \mathcal{A}} \exp((\theta_t)_{s,a'})) \left(\exp((\theta_t)_{s,a_q(s)} + \frac{u_t|\mathcal{A}|}{|\mathcal{A}|-1}) + \sum_{a' \neq a_q(s)} \exp((\theta_t)_{s,a'}) \right)} \quad (51)$$

$$= \frac{\delta(s)}{2} \sum_{a \in \mathcal{A}_\perp(s)} \pi_t(a|s) \pi_{t+1}(a_q(s)|s) \left(1 - \exp\left(-\frac{u_t|\mathcal{A}|}{|\mathcal{A}|-1}\right) \right) \quad (52)$$

$$\geq \frac{\delta(s)}{2} \sum_{a \in \mathcal{A}_\perp(s)} \pi_t(a|s) \pi_t(a_q(s)|s) \left(1 - \exp\left(-\frac{u_t|\mathcal{A}|}{|\mathcal{A}|-1}\right) \right) \quad (53)$$

$$\geq \frac{\delta(s)}{2} \sum_{a \in \mathcal{A}_\perp(s)} \pi_t(a|s) \pi_t(a_q(s)|s) \min\left(\frac{u_t|\mathcal{A}|}{2(|\mathcal{A}|-1)}, \frac{1}{2}\right) \quad (54)$$

$$= \frac{\delta(s)}{4} \pi_t(a_q(s)|s) \min\left(\frac{u_t|\mathcal{A}|}{|\mathcal{A}|-1}, 1\right) \sum_{a \in \mathcal{A}_\perp(s)} \pi_t(a|s), \quad (55)$$

where on Eq. (54) we used the fact that $1 - e^{-x} \geq \min(\frac{x}{2}, \frac{1}{2})$ for $x \geq 0$.

By assumption, we know that $\pi_t(a_q(s)|s) \leq \sum_{a \in \mathcal{A}_\top(s)} \pi_t(a|s) \leq \lim_{t' \rightarrow \infty} \sum_{a \in \mathcal{A}_\top(s)} \pi_{t'}(a|s) < 1$. Let m_\perp denote the policy mass that must remain outside of $\mathcal{A}_\top(s)$ at all t . We get that $\forall t \geq t_0$:

$$\operatorname{adv}_t(s, \pi_{t+1}) \geq \frac{\delta(s)m_\perp}{4} \min\left(\eta_t d_t(s) m_\perp \frac{|\mathcal{A}|}{|\mathcal{A}|-1}, 1\right) \pi_t(a_q(s)|s). \quad (56)$$

If the optimal action is unique, it is direct to observe that the sum of the advantage gained over all t is lower bounded by a term that diverges to ∞ under the condition $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$. However, we face a technical issue with non unique optimal policy: how can we guarantee that $\pi_t(a_q(s)|s)$ is large enough? It is intuitive that it is going to be the case "on average", but its formal proof remains an open problem. In consequence, we limit our current proof to the assumption of uniqueness of the optimal policy. (we will try to solve it for camera-ready version) \square

Finally, we prove that condition $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$ is necessary for optimality.

Lemma 11 (Optimality under U_{MCE} (necessity)). *It is necessary to assume that $\sum_{t=0}^{\infty} \eta_t d_t(s) = \infty$ to guarantee optimality of update U_{MCE} : $q_\infty = q_\star \doteq \max_{\pi \in \Pi} q_\pi$.*

Proof. We assume $\sum_{t=0}^{\infty} \eta_t d_t(s) = M < \infty$. We upper bound the parameter update u_t used in Proposition 2 and Lemma 10:

$$u_t \doteq \eta_t d_t(s) (1 - \pi_{\theta_t}(a_q(s)|s)) \leq \eta_t d_t(s). \quad (57)$$

For all t , we have:

$$\theta_t - u_t \leq \theta_{t+1} \leq \theta_t + u_t \quad (58)$$

$$\theta_0 - \sum_{t'=0}^t u_{t'} \leq \theta_{t+1} \leq \theta_0 + \sum_{t'=0}^t u_{t'} \quad (59)$$

$$\theta_0 - \sum_{t'=0}^t \eta_{t'} d_{t'}(s) \leq \theta_{t+1} \leq \theta_0 + \sum_{t'=0}^t \eta_{t'} d_{t'}(s) \quad (60)$$

$$\theta_0 - M \leq \theta_{t+1} \leq \theta_0 + M \quad (61)$$

With a softmax parametrization, some parameters need to diverge for the policy to converge to 0 on suboptimal actions. The boundedness exhibited in Eq. (61) prevents that from happening, which concludes the proof of the lemma. \square

This concludes the proof of the theorem. \square

Theorem 4 (Convergence rates of U_{MCE}). *The process, assumption, and condition defined in Theorem 3 guarantee that the sequence of value functions (q_t) asymptotically converges in $\mathcal{O}\left(\left(\sum_{t'=0}^t \eta_{t'} \min_{s \in \mathcal{S}} d_{t'}(s)\right)^{-1}\right)$.*

Proof. We consider the following definition in state s :

$$a_s^\top = \operatorname{argmax}_{a \in \mathcal{A}} q_*(s, a) \quad (62)$$

$$\mathcal{A}_s^\perp = \{a \in \mathcal{A} | a \neq a_s^\top\}. \quad (63)$$

Since we assumed the unicity of the optimal policy, a_s^\top must be unique and $|\mathcal{A}_s^\perp| = |\mathcal{A}| - 1$. We define $\delta(s)$ as the gap with the best suboptimal action:

$$\delta(s) := v_*(s) - \max_{a \in \mathcal{A}_s^\perp} q_*(s, a) > 0. \quad (64)$$

From the convergence of $v_t(s)$ and $q_t(s, a)$, we also know that for any $\xi > 0$, there exists t_0 such that for any s, a and $t \geq t_0$:

$$v_*(s) - \xi \leq v_t(s) \leq v_*(s) \quad (65)$$

$$q_*(s, a) - \xi \leq q_t(s, a) \leq q_*(s, a) \quad (66)$$

$$\pi_{\theta_t}(a_s^\top | s) \geq \frac{1}{2} \geq \pi_{\theta_t}(a | s). \quad (67)$$

We fix $\xi \doteq \frac{\delta(s)}{2}$. We then know that for $t \geq t_0$:

$$a_q(s) \doteq \operatorname{argmax}_{a \in \mathcal{A}} q_t(s, a) = a_s^\top \quad (68)$$

As a consequence, from t_0 , update U_{MCE} will be:

$$(\theta_{t+1})_{s, a_s^\top} = (\theta_t)_{s, a_s^\top} + \eta_t d_t(s) (1 - \pi_{\theta_t}(a_s^\top | s)) \quad (69)$$

$$\forall a \in \mathcal{A}_s^\perp, \quad (\theta_{t+1})_{s, a} = (\theta_t)_{s, a} - \frac{\eta_t d_t(s)}{|\mathcal{A}| - 1} (1 - \pi_{\theta_t}(a_s^\top | s)) \quad (70)$$

From these last lines, we observe that $(\theta_{t+1})_{s,a_s^\top}$ and $(\theta_{t+1})_{s,a}$ evolve symmetrically with some fixed ratio. We can therefore state that:

$$(\theta_t)_{s,a_s^\top} = (\theta_{t_0})_{s,a_s^\top} + X_t \quad \text{and} \quad \forall a \in \mathcal{A}_s^\perp, \quad (\theta_t)_{s,a} = (\theta_{t_0})_{s,a_s^\top} - \frac{X_t}{|\mathcal{A}| - 1}, \quad (71)$$

with:

$$X_{t_0} = 0 \quad \text{and} \quad X_{t+1} = X_t + \eta_t d_t(s)(1 - \pi_{\theta_t}(a_s^\top | s)) \quad (72)$$

Therefore:

$$1 - \pi_{\theta_t}(a_s^\top | s) = \frac{\sum_{a \in \mathcal{A}_s^\perp} \exp((\theta_t)_{s,a})}{\exp((\theta_t)_{s,a_s^\top}) + \sum_{a \in \mathcal{A}_s^\perp} \exp((\theta_t)_{s,a})} \quad (73)$$

$$= \frac{\sum_{a \in \mathcal{A}_s^\perp} \exp\left((\theta_{t_0})_{s,a_s^\top} - \frac{X_t}{|\mathcal{A}| - 1}\right)}{\exp((\theta_{t_0})_{s,a_s^\top} + X_t) + \sum_{a \in \mathcal{A}_s^\perp} \exp\left((\theta_{t_0})_{s,a_s^\top} - \frac{X_t}{|\mathcal{A}| - 1}\right)} \quad (74)$$

$$= \frac{\sum_{a \in \mathcal{A}_s^\perp} \exp((\theta_{t_0})_{s,a_s^\top})}{\exp\left((\theta_{t_0})_{s,a_s^\top} + \frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_t\right) + \sum_{a \in \mathcal{A}_s^\perp} \exp((\theta_{t_0})_{s,a_s^\top})} \quad (75)$$

$$= \frac{1}{\frac{\exp((\theta_{t_0})_{s,a_s^\top})}{\sum_{a \in \mathcal{A}_s^\perp} \exp((\theta_{t_0})_{s,a_s^\top})} \exp\left(\frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_t\right) + 1} \quad (76)$$

$$= \frac{1}{\frac{\pi_{\theta_{t_0}}(a_s^\top | s)}{1 - \pi_{\theta_{t_0}}(a_s^\top | s)} \exp\left(\frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_t\right) + 1} \quad (77)$$

Setting $\kappa \doteq \frac{\pi_{\theta_{t_0}}(a_s^\top | s)}{1 - \pi_{\theta_{t_0}}(a_s^\top | s)}$, we obtain the following sequence:

$$X_{t_0} = 0 \quad \text{and} \quad X_{t+1} = X_t + \frac{\eta_t d_t(s)}{\kappa \exp\left(\frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_t\right) + 1} \geq X_t + \frac{\eta_t d_t(s)}{2\kappa} \exp\left(-\frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_t\right) \quad (78)$$

From there, we reproduce for completeness the end of Theorem 4 in Laroche and Tachet des Combes (2021). Let us now study the sequence $(X_t)_{t \geq t_0}$. To that end, we define the function $f(t)$ solution on $[t_0, +\infty)$ of the ordinary differential equation (note that t is now a continuous variable):

$$\begin{cases} f(t_0) = \frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_{t_0} = 0 \\ \frac{df(t)}{dt} = \frac{\eta_t d_t(s) |\mathcal{A}|}{2\kappa (|\mathcal{A}| - 1)} e^{-f(t)}, \end{cases} \quad (79)$$

where $\eta_t d_t(s)$ is the piece-wise constant function defined as $\eta_t d_t(s) = \eta_{\lfloor t \rfloor} d_{\lfloor t \rfloor}(s)$.

From the evolution equations of X_t and $f(t)$, we see that $\forall t \in \mathbb{N}$, $\frac{|\mathcal{A}|}{|\mathcal{A}| - 1} X_t \geq f(t)$. Additionally, we have:

$$f(t) = \log\left(\frac{|\mathcal{A}|}{2\kappa (|\mathcal{A}| - 1)} \int_{t_0}^t \eta_{t'} d_{t'}(s) dt' + e^{f(t_0)}\right). \quad (80)$$

In particular, going back to $t \in \mathbb{N}$, we obtain:

$$X_t \geq \frac{|\mathcal{A}| - 1}{|\mathcal{A}|} \log\left(\frac{|\mathcal{A}|}{2\kappa (|\mathcal{A}| - 1)} \int_{t_0}^t \eta_{t'} d_{t'}(s) dt' + e^{f(t_0)}\right) = \frac{|\mathcal{A}| - 1}{|\mathcal{A}|} \log\left(\frac{|\mathcal{A}|}{2\kappa (|\mathcal{A}| - 1)} \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s) + 1\right). \quad (81)$$

We can now write the following rate in policy convergence:

$$1 - \pi_t(a_s^\top | s) = \frac{1}{\kappa \exp\left(\frac{|\mathcal{A}|}{|\mathcal{A}|-1} X_t\right) + 1} \quad (82)$$

$$\geq \frac{1}{\kappa \frac{|\mathcal{A}|}{2\kappa(|\mathcal{A}|-1)} \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s) + 1 + 1} \quad (83)$$

$$\geq \frac{1}{\frac{|\mathcal{A}|}{2(|\mathcal{A}|-1)} \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s) + 2} \quad (84)$$

On the value side, we further get:

$$v_\star(s) - v_t(s) = \mathbb{P}[A = a_s^\top | A \sim \pi_t(\cdot | s)] \times \gamma \mathbb{E}[v_\star(S') - v_t(S') | S' \sim p(\cdot | s, a_s^\top)] \quad (85)$$

$$+ \mathbb{P}[A \in \mathcal{A} / \{a_s^\top\} | A \sim \pi_t(\cdot | s)] (v_\star(s) - \mathbb{E}[q_t(s, A) | A \sim \pi_t(\cdot | s) \cap \mathcal{A} / \{a_s^\top\}])$$

$$\leq \gamma \mathbb{E}[v_\star(S') - v_t(S') | S' \sim p(\cdot | s, a_s^\top)] + \frac{v_\star(s) - \min_{a \in \mathcal{A}} q_t(s, a)}{2 + \frac{|\mathcal{A}|}{2(|\mathcal{A}|-1)} \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s)} \quad (86)$$

$$\leq \frac{v_\top - v_\perp}{(1 - \gamma) \left(2 + \frac{|\mathcal{A}|}{2(|\mathcal{A}|-1)} \min_{s \in \text{supp}(d_{\pi_\star, \gamma})} \sum_{t'=t_0}^{t-1} \eta_{t'} d_{t'}(s)\right)}, \quad (87)$$

where v_\top (resp. v_\perp) stand for the maximal (resp. minimal) value, which is upper bounded by $\frac{r_\top}{1-\gamma}$ (resp. $\frac{r_\perp}{1-\gamma}$), often times much smaller (resp. larger), and where $\text{supp}(d_{\pi_\star, \gamma})$ denotes the support of the distribution of the optimal policy. This concludes the proof. \square

A.3 Gravity well proofs (Section 3.3)

Theorem 5 (Gravity well). U_{DI} and U_{MCE} are guaranteed to satisfy Eq. (14). $U_{\text{PG-SM}}$, $U_{\text{PG-ES}}$, and U_{CE} may transgress Eq. (14). Furthermore, $U_{\text{PG-SM}}$ may not even satisfy $\pi_{t+1}(a_{q_t}(s) | s) - \pi_t(a_{q_t}(s) | s) \geq 0$.

Proof. We deal with each $U_{\text{PG-SM}}$, $U_{\text{PG-ES}}$, U_{DI} , U_{CE} , and U_{MCE} separately.

Proof for $U_{\text{PG-SM}}$: Update of $U_{\text{PG-SM}}$:

$$(U_{\text{PG-SM}}(\theta))_{s,a} - (\theta)_{s,a} = \eta d(s) \text{adv}(s, a) \pi_\theta(a | s) \quad (88)$$

Assuming the existence of a suboptimal advantageous action b such that $\sum_a \pi_\theta(a | s) q(s, a) < q(s, b) < q(s, a_q(s))$, we get:

$$\pi_{U_{\text{PG-SM}}(\theta)}(a_q(s) | s) = \frac{\exp((\theta)_{s,a_q(s)} + \eta d(s) \text{adv}(s, a_q(s)) \pi_\theta(a_q(s) | s))}{\sum_{a \in \mathcal{A}} \exp((\theta)_{s,a} + \eta d(s) \text{adv}(s, a) \pi_\theta(a | s))} \quad (89)$$

$$< \frac{\exp((\theta)_{s,a_q(s)} + \eta d(s) \text{adv}(s, a_q(s)) \pi_\theta(a_q(s) | s))}{\exp((\theta)_{s,b} + \eta d(s) \text{adv}(s, b) \pi_\theta(b | s))} \quad (90)$$

$$= \exp((\theta)_{s,a_q(s)} - (\theta)_{s,b}) \exp(\eta d(s) (\text{adv}(s, a_q(s)) \pi_\theta(a_q(s) | s) - \text{adv}(s, b) \pi_\theta(b | s))) \quad (91)$$

$$= \frac{\pi_\theta(a_q(s) | s)}{\pi_\theta(b | s)} \exp(\eta d(s) (\text{adv}(s, a_q(s)) \pi_\theta(a_q(s) | s) - \text{adv}(s, b) \pi_\theta(b | s))). \quad (92)$$

Therefore, we get $\pi_{U_{\text{PG-SM}}(\theta)}(a_q(s) | s) < \pi_\theta(a_q(s) | s)$ granted that:

$$\pi_\theta(b | s) \geq \exp(\eta d(s) (\text{adv}(s, a_q(s)) \pi_\theta(a_q(s) | s) - \text{adv}(s, b) \pi_\theta(b | s))), \quad (93)$$

which can be obtained quite easily when $\pi_\theta(a_q(s) | s)$ is close to 0 and $\pi_\theta(b | s)$ is close to 1. If this is the case, then, the policy mass lost by $a_q(s)$ from θ to $U_{\text{PG-SM}}(\theta)$ must have been gained by at least another action, and condition (14) cannot be satisfied.

Proof for $U_{\text{PG=ES}}$: Update of $U_{\text{PG=ES}}$ ²:

$$(U_{\text{PG=ES}}(\theta))_{s,a} - (\theta)_{s,a} = \eta d(s) \text{adv}(s, a) \pi_\theta(a|s)^{1-\frac{1}{p}} \quad (94)$$

Let us assume that the action set contains only three actions: $\mathcal{A} \doteq \{a, b, c\}$. Let us drop the dependencies on s , and set $\varepsilon \doteq \pi_\theta(a) = \pi_\theta(c) \ll \pi_\theta(b)$ for concision. Let $q(c) = 1 = q(b) + \delta > q(a) = 0$ and assume $\delta \ll \varepsilon \ll 1$. Therefore $\theta_a = \theta_c$. Assuming the existence of a suboptimal advantageous action b such that $\pi_\theta(a)q(a) + \pi_\theta(b)q(b) + \pi_\theta(c)q(c) < q(b) < q(c)$, we get:

$$\pi_{U_{\text{PG-SM}}(\theta)}(c) = \frac{|\theta_c + \eta \text{adv}(c) \varepsilon^{1-\frac{1}{p}}|^p}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (95)$$

$$= \frac{|\theta_c + \eta(\varepsilon + (1-2\varepsilon)\delta) \varepsilon^{1-\frac{1}{p}}|^p}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (96)$$

$$= \frac{|\theta_c + \eta \varepsilon^{2-\frac{1}{p}} + \mathcal{O}(\varepsilon^{3-\frac{1}{p}} + \delta)|^p}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (97)$$

$$= \frac{\theta_c^p + p\eta \varepsilon^{2-\frac{1}{p}} \theta_c^{p-1} + \mathcal{O}(\varepsilon^{3-\frac{1}{p}} + \delta)}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (98)$$

$$\pi_{U_{\text{PG-SM}}(\theta)}(b) = \frac{|\theta_b + \eta \text{adv}(b)(1-2\varepsilon)^{1-\frac{1}{p}}|^p}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (99)$$

$$= \frac{|\theta_b + \eta \varepsilon(1-2\delta)(1-2\varepsilon)^{1-\frac{1}{p}}|^p}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (100)$$

$$= \frac{|\theta_b + \eta \varepsilon + \mathcal{O}(\varepsilon^{2-\frac{1}{p}} + \delta)|^p}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (101)$$

$$= \frac{\theta_b^p + p\eta \varepsilon \theta_b^{p-1} + \mathcal{O}(\varepsilon^{2-\frac{1}{p}} + \delta)}{\|U_{\text{PG-SM}}(\theta)\|_p^p} \quad (102)$$

We notice that, as long as $p > 1$, $\pi_{U_{\text{PG-SM}}(\theta)}(b)$ grows with in larger order of magnitude than $\pi_{U_{\text{PG-SM}}(\theta)}(c)$, and therefore that we can construct an update such that condition 14 is not satisfied. Note that the situation for it to happen is much more stringent than that with $U_{\text{PG-SM}}$.

Proof for U_{DI} : Update of $U_{\text{DI}} \doteq \text{Proj}_{\Delta_{\mathcal{A}}}(U_{\text{PG-DI}}(\theta))$:

$$(U_{\text{PG-DI}}(\theta))_{s,a_q(s)} - (\theta)_{s,a_q(s)} = \eta d(s) q(s, a_q(s)) \quad (103)$$

$$(U_{\text{PG-DI}}(\theta))_{s,a} - (\theta)_{s,a} = \eta d(s) q(s, a) \quad (104)$$

$$\leq (U_{\text{PG-DI}}(\theta))_{s,a_q(s)} - (\theta)_{s,a_q(s)}. \quad (105)$$

Since $a_q(s) \doteq \text{argmax}_a q(s, a)$, we may apply Lemma 12 and obtain:

$$(\text{Proj}_{\Delta_{\mathcal{A}}}(U_{\text{PG-DI}}(\theta)))_{s,a} - (\theta)_{s,a} \leq (\text{Proj}_{\Delta_{\mathcal{A}}}(U_{\text{PG-DI}}(\theta)))_{s,a_q(s)} - (\theta)_{s,a_q(s)} \quad (106)$$

$$\iff (U_{\text{DI}}(\theta))_{s,a} - (\theta)_{s,a} \leq (U_{\text{DI}}(\theta))_{s,a_q(s)} - (\theta)_{s,a_q(s)}, \quad (107)$$

which proves that condition (14) is satisfied.

Proof for U_{CE} : The non-monotonicity example of Proposition 1 serves also as an example where U_{CE} does not satisfy condition (14).

Proof for U_{MCE} : In Proposition 2, it has been established for all actions $s \neq a_q(s)$ in Eq. (37) that:

$$\pi_{U_{\text{MCE}}(\theta)}(a|s) \leq \pi_\theta(a|s), \quad (108)$$

²We assume the positivity of parameters θ without loss of generality: $\nabla_\theta \pi_\theta = \nabla_\theta \pi_{-\theta}$.

which implies that:

$$\pi_{U_{\text{MCE}}(\theta)}(a_q(s)|s) \geq \pi_\theta(a_q(s)|s), \quad (109)$$

which in turn implies that condition (14) is satisfied. \square

A.4 Unlearning proofs (Section 3.4)

Theorem 6 (Unlearning setting – constant weights). *In the setting where η is constant:*

- (i) *Under assumptions enunciated below—and verified by $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$, U_{PG} needs $n' \geq n$ updates.*
- (ii) *U_{DI} needs $n' = \min\{n; \lceil \frac{1}{\eta} \rceil\}$ updates.*
- (iii) *U_{CE} and U_{MCE} need $n' \leq 2 + \frac{1}{\eta} \log(1 + 2\eta n)$ updates.*

Proof. We deal with each (i), (ii), and (iii) separately.

Proof of (i): We start with the proof of (i) with update U_{PG} . We recall that $\theta \in \mathbb{R}^2$, let θ_1 and θ_2 denote its two components and make the following assumptions (verified by both $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$):

- $\nabla_\theta \pi_\theta(a_1)$ only depends on θ via π_θ ,
- $(\nabla_\theta \pi_\theta(a_1))_1$ (resp. $(\nabla_\theta \pi_\theta(a_1))_2$) is a positive and decreasing (resp. negative and increasing) function of $\pi_\theta(a_1)$ when $\pi_\theta(a_1) \geq \pi_0(a_1)$.

We now define recursively the process $\theta_n^{\leftrightarrow}$ of, starting from θ_0 , performing n updates with $q(a_1) = 1, q(a_2) = 0$ and then n updates with $q(a_1) = 0, q(a_2) = 1$:

$$\theta_0^{\leftrightarrow} \doteq \theta_0 \quad \text{and} \quad \theta_{n+1}^{\leftrightarrow} \doteq U_{\text{PG}}((U_{\text{PG}}(\theta_0, d, (1, 0), \eta))_n^{\leftrightarrow}, d, (0, 1), \eta). \quad (110)$$

We prove by induction that for any θ_0 such that $\pi_{\theta_0}(a_1) \geq \frac{1}{2}$:

$$(\theta_n^{\leftrightarrow})_1 \geq \theta_1, \quad (111)$$

$$(\theta_n^{\leftrightarrow})_2 \leq \theta_2. \quad (112)$$

Given the signs of the components of $(\nabla_\theta \pi_\theta(a_1))_1$, the above guarantees that $\pi_{\theta_n^{\leftrightarrow}}(a_1) \geq \pi_{\theta_0}(a_1)$, i.e. that the policy has not yet recovered its initial value.

The property is direct for $n = 0$. Next, we make the hypothesis that the property holds for n , and prove it for $n + 1$. We also assume in the following that θ_0 is such that $\pi_{\theta_0}(a_1) \geq 0.5$. We compute the policy gradient updates:

- with $q(a_1) = 1, q(a_2) = 0$: $U_{\text{PG}}(\theta, d, (1, 0), \eta) \doteq \theta + \eta \nabla_\theta \pi_\theta(a_1)$,
- with $q(a_1) = 0, q(a_2) = 1$: $U_{\text{PG}}(\theta, d, (0, 1), \eta) \doteq \theta + \eta \nabla_\theta \pi_\theta(a_2) = \theta - \eta \nabla_\theta \pi_\theta(a_1)$,

since a_1 and a_2 are playing symmetrical roles ($\pi_\theta(a_1) + \pi_\theta(a_2) = 1$). Then:

$$(\theta)_{n+1}^{\leftrightarrow} = U_{\text{PG}}((U_{\text{PG}}(\theta_0, d, (1, 0), \eta))_n^{\leftrightarrow}, d, (0, 1), \eta) \quad (113)$$

$$= (\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow} - \eta \nabla_\theta \pi_{(\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow}}(a_1). \quad (114)$$

First, by the assumptions on the monotonicity of $\pi_\theta(a_1)$ with respect to its components, we know that $\pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1) \geq \pi_{\theta_0}(a_1) \geq \frac{1}{2}$. We may therefore apply the induction hypothesis to $\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)$ and obtain that

$$((\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow})_1 \geq (\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_1, \quad (115)$$

$$((\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow})_2 \leq (\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_2, \quad (116)$$

$$\pi_{(\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow}}(a_1) \geq \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1). \quad (117)$$

Given our assumptions on the partial derivatives of $\pi_\theta(a_1)$, we infer that:

$$(\nabla_\theta \pi_{(\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow}}(a_1))_1 \leq (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_1, \quad (118)$$

$$(\nabla_\theta \pi_{(\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow}}(a_1))_2 \geq (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_2, \quad (119)$$

where the 1 and 2 subscripts still refer to the first and second coordinate of the gradient respectively. We therefore obtain that:

$$(\theta_{n+1}^{\leftrightarrow})_1 = ((\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow})_1 - \eta (\nabla_\theta \pi_{(\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow}}(a_1))_1 \quad (120)$$

$$\geq ((\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow})_1 - \eta (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_1 \quad (121)$$

$$\geq (\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_1 - \eta (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_1 \quad (122)$$

$$= (\theta_0)_1 + \eta ((\nabla_\theta \pi_{\theta_0}(a_1))_1 - (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_1) \quad (123)$$

$$\leq (\theta_0)_1, \quad (124)$$

the last step using again the decrease of $(\nabla_\theta \pi_\theta(a_1))_1$ with respect to $\pi_\theta(a_1)$. Similarly, as $(\nabla_\theta \pi_\theta(a_1))_2$ is an increasing function of $\pi_\theta(a_1)$:

$$(\theta_{n+1}^{\leftrightarrow})_2 = ((\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow})_2 - \eta (\nabla_\theta \pi_{(\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow}}(a_1))_2 \quad (125)$$

$$\leq ((\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_n^{\leftrightarrow})_2 - \eta (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_2 \quad (126)$$

$$\leq (\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1))_2 - \eta (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_2 \quad (127)$$

$$= (\theta_0)_2 + \eta ((\nabla_\theta \pi_{\theta_0}(a_1))_2 - (\nabla_\theta \pi_{\theta_0 + \eta \nabla_\theta \pi_{\theta_0}(a_1)}(a_1))_2) \quad (128)$$

$$\leq (\theta_0)_2. \quad (129)$$

We conclude that the policy gradient update has still not reached the initial θ_0 after n updates back and therefore that n' must be larger than n .

Let us now show that $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$ verify the assumptions. For $U_{\text{PG-SM}}$:

$$\nabla_\theta \pi_\theta(a_1) = \begin{bmatrix} \pi_\theta(a_1)(1 - \pi_\theta(a_1)) \\ -\pi_\theta(a_1)(1 - \pi_\theta(a_1)) \end{bmatrix}, \quad (130)$$

which is indeed only a function of $\pi_\theta(a_1)$, has a first component that is positive and decreasing as a function of $\pi_\theta(a_1)$ and a second one that is negative and increasing (for $\pi_\theta(a_1) \geq 0.5$). For $U_{\text{PG-ES}}$:

$$\nabla_\theta \pi_\theta(a_1) = \begin{bmatrix} \pi_\theta(a_1)^{1-\frac{1}{p}}(1 - \pi_\theta(a_1)) \\ -\pi_\theta(a_1)(1 - \pi_\theta(a_1))^{1-\frac{1}{p}} \end{bmatrix} \quad (131)$$

which is indeed only a function of $\pi_\theta(a_1)$, has a first component that is positive and decreasing as a function of $\pi_\theta(a_1)$ and a second one that is negative and increasing (for $\pi_\theta(a_1) \geq \frac{p}{2p-1}$).

Proof of (ii): We continue with the proof of (ii) and therefore consider the direct parametrization. We compute the policy gradient updates:

- with $q(a_1) = 1, q(a_2) = 0$: $U_{\text{Di}}(\theta, d, (1, 0), \eta) \doteq \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \begin{bmatrix} \eta \\ 0 \end{bmatrix} \right) = \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \frac{1}{2} \begin{bmatrix} \eta \\ -\eta \end{bmatrix} \right),$
- with $q(a_1) = 0, q(a_2) = 1$: $U_{\text{Di}}(\theta, d, (0, 1), \eta) \doteq \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \begin{bmatrix} 0 \\ \eta \end{bmatrix} \right) = \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \frac{1}{2} \begin{bmatrix} -\eta \\ \eta \end{bmatrix} \right).$

As a consequence applying n times update $U_{\text{Di}}(\theta, d, (1, 0), \eta)$ is equivalent to applying one update $U_{\text{Di}}(\theta, d, (1, 0), n\eta)$. Then we face two cases:

- $n\eta \leq 1$: starting from $\theta_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$, the 2D-projection does not have any effect, and it will take $n' = n$ steps to recover $\pi(a_1) \leq 0.5$.

- $n\eta \geq 1$: the 2D-projection projects $U_{\text{DI}}(\theta, d, (1, 0), n\eta)$ on $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and it will take $n' = \lceil \frac{1}{\eta} \rceil$ steps to recover $\pi(a_1) \leq 0.5$.

As a conclusion, it will take exactly $n' = \min\{n; \lceil \frac{1}{\eta} \rceil\}$ to recover from the n gradient steps.

Proof of (iii): We compute the policy gradient updates for classic cross-entropy and notice that they are equal to that of modified cross entropy when $|\mathcal{A}| = 2$:

- with $q(a_1) = 1, q(a_2) = 0$:

$$U_{\text{CE}}(\theta, d, (1, 0), \eta) \doteq \theta + \eta \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \pi_\theta \right) = \theta + \eta(1 - \pi_\theta(a_1)) \begin{bmatrix} 1 \\ -1 \end{bmatrix} = U_{\text{MCE}}(\theta, d, (1, 0), \eta),$$

- with $q(a_1) = 0, q(a_2) = 1$:

$$U_{\text{CE}}(\theta, d, (0, 1), \eta) \doteq \theta + \eta \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \pi_\theta \right) = \theta + \eta(1 - \pi_\theta(a_2)) \begin{bmatrix} -1 \\ 1 \end{bmatrix} = U_{\text{MCE}}(\theta, d, (0, 1), \eta).$$

We focus on $\delta_n^\rightarrow \doteq (\theta_n)_{a_1} - (\theta_n)_{a_2}$ after n updates with $q(a_1) = 1, q(a_2) = 0$:

$$\delta_n^\rightarrow = \delta_{n-1}^\rightarrow + \eta(1 - \pi_{\theta_{n-1}}(a_1)) + \eta\pi_{\theta_{n-1}}(a_2) \quad (132)$$

$$= \delta_{n-1}^\rightarrow + \eta - \frac{\eta \exp((\theta_{n-1})_{a_1})}{\exp((\theta_{n-1})_{a_1}) + \exp((\theta_{n-1})_{a_2})} + \frac{\eta \exp((\theta_{n-1})_{a_2})}{\exp((\theta_{n-1})_{a_1}) + \exp((\theta_{n-1})_{a_2})} \quad (133)$$

$$= \delta_{n-1}^\rightarrow + 2\eta \frac{\exp((\theta_{n-1})_{a_2})}{\exp((\theta_{n-1})_{a_1}) + \exp((\theta_{n-1})_{a_2})} \quad (134)$$

$$\leq \delta_{n-1}^\rightarrow + 2\eta \exp(-\delta_{n-1}^\rightarrow). \quad (135)$$

Applying Lemma 13 to δ_n^\rightarrow establishes that $\delta_n^\rightarrow \leq 2\eta + \log(1 + 2\eta n)$.

We now focus on a single update with $q(a_1) = 0, q(a_2) = 1$: $\delta^\leftarrow \doteq U_{\text{CE}}(\theta, d, (0, 1), \eta)_{a_2} - U_{\text{CE}}(\theta, d, (0, 1), \eta)_{a_1}$, with $\delta \doteq \theta_{a_2} - \theta_{a_1} \geq 0$:

$$\delta^\leftarrow = \delta + \eta(1 - \pi_{\theta_{n-1}}(a_2)) + \eta\pi_{\theta_{n-1}}(a_1) \quad (136)$$

$$= \delta + 2\eta \frac{\exp((\theta_{n-1})_{a_1})}{\exp((\theta_{n-1})_{a_1}) + \exp((\theta_{n-1})_{a_2})} \quad (137)$$

$$\geq \delta + \eta \quad (138)$$

As consequence, the number n' of updates δ^\leftarrow for recovering from a convergence in δ_n^\rightarrow is lower than:

$$n'\eta \leq 2\eta + \log(1 + 2\eta n) \quad (139)$$

$$n' \leq 2 + \frac{1}{\eta} \log(1 + 2\eta n), \quad (140)$$

which concludes the proof. \square

Theorem 7 (Unlearning setting – decaying weights). *In the setting where $\eta_t = \frac{\eta_1}{\sqrt{t}}$:*

(i) *Under the assumptions enunciated above-verified by $U_{\text{PG-SM}}$ and $U_{\text{PG-ES}}$, U_{PG} needs $n' \geq 3n - 4\sqrt{n} + 1$.*

(ii) *U_{DI} updates require at most n' equal to*

$$\min \left\{ 3n + 4\sqrt{n} + 1 ; \left(\frac{1}{\eta_1} + 1 \right)^2 + \sqrt{n} \left(2 + \frac{2}{\eta_1} \right) \right\}.$$

(iii) *U_{CE} and U_{MCE} updates require at most n' equal to*

$$\left(4 + \frac{\log(1+4\eta_1\sqrt{n})}{2\eta_1} \right)^2 + \sqrt{n} \left(8 + \frac{\log(1+4\eta_1\sqrt{n})}{\eta_1} \right).$$

Proof. We deal with each (i), (ii), and (iii) separately.

Proof of (i): We start with the proof of (i), for update U_{PG} . We make the similar assumptions as in Theorem 6. We prove the result by measuring the steps made in parameter space and noting (given the behavior of $\pi_\theta(a_1)$ with respect to the components of θ) that the sum of steps in one direction must equal the sum of steps in the other direction for at least one of the components of θ , in other words:

$$\sum_{t=1}^n \eta_t (\nabla_\theta \pi_{\theta_t}(a_1))_1 \leq \sum_{t=n+1}^{n+n'} \eta_t (\nabla_\theta \pi_{\theta_t}(a_1))_1, \quad (141)$$

or

$$\sum_{t=1}^n \eta_t (\nabla_\theta \pi_{\theta_t}(a_1))_2 \geq \sum_{t=n+1}^{n+n'} \eta_t (\nabla_\theta \pi_{\theta_t}(a_1))_2. \quad (142)$$

Both cases being equivalent, we focus on the first one:

$$\sum_{t=1}^n \eta_t (\nabla_\theta \pi_{\theta_t}(a_1))_1 \leq \sum_{t=n+1}^{n+n'} \eta_t (\nabla_\theta \pi_{\theta_t}(a_1))_1 \quad (143)$$

$$\iff \sum_{t=1}^n \frac{\eta_1}{\sqrt{t}} (\nabla_\theta \pi_{\theta_t}(a_1))_1 \leq \sum_{t=1}^{n'} \frac{\eta_1}{\sqrt{t+n}} (\nabla_\theta \pi_{\theta_{t+n}}(a_1))_1 \quad (144)$$

$$\iff \sum_{t=1}^n \frac{1}{\sqrt{t}} (\nabla_\theta \pi_{\theta_t}(a_1))_1 \leq \sum_{t=1}^{n'} \frac{1}{\sqrt{t+n}} (\nabla_\theta \pi_{\theta_{t+n}}(a_1))_1. \quad (145)$$

By the concavity assumption, all the gradient $(\nabla_\theta \pi_{\theta_{t+n}}(a_1))_1$ may be upper bounded by some $(\nabla_\theta \pi_{\sigma(t)}(a_1))_1$ with $\sigma(t) \in \{1, \dots, n\}$, a decreasing function such that:

$$(\nabla_\theta \pi_{\sigma(t)+1}(a_1))_1 \leq (\nabla_\theta \pi_{\theta_{t+n}}(a_1))_1 \leq (\nabla_\theta \pi_{\sigma(t)}(a_1))_1. \quad (146)$$

In particular, $\sigma(1) = n$, and in that case, the lower-bound is realized. We therefore get:

$$\sum_{t=1}^n \frac{1}{\sqrt{t}} (\nabla_\theta \pi_{\theta_t}(a_1))_1 \leq \sum_{t=1}^{n'} \frac{1}{\sqrt{t+n}} (\nabla_\theta \pi_{\sigma(t)}(a_1))_1. \quad (147)$$

Given the fact that $\forall 1 \leq t \leq n, (\nabla_\theta \pi_{\theta_t}(a_1))_1 \geq (\nabla_\theta \pi_{\theta_{t+1}}(a_1))_1$, the smallest n' that can verify this last inequality is realized when all the gradients are equal, implying the following condition on n' :

$$\sum_{t=1}^n \frac{1}{\sqrt{t}} \leq \sum_{t=1}^{n'} \frac{1}{\sqrt{t+n}} \quad (148)$$

$$\implies 2(\sqrt{n+1} - 1) \leq \sqrt{n'+n} - 2\sqrt{n} \quad (149)$$

$$\implies 4\sqrt{n} - 2 \leq 2\sqrt{n'+n} \quad (150)$$

$$\iff 4n - 4\sqrt{n} + 1 \leq n' + n \quad (151)$$

$$\iff n' \geq 3n - 4\sqrt{n} + 1. \quad (152)$$

Proof of (ii): Exactly like in Theorem 6, the policy gradient updates for the direct parametrization are:

- with $q(a_1) = 1, q(a_2) = 0$: $U_{\text{Di}}(\theta, d, (1, 0), \eta) \doteq \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \begin{bmatrix} \eta \\ 0 \end{bmatrix} \right) = \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \frac{1}{2} \begin{bmatrix} \eta \\ -\eta \end{bmatrix} \right),$
- with $q(a_1) = 0, q(a_2) = 1$: $U_{\text{Di}}(\theta, d, (0, 1), \eta) \doteq \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \begin{bmatrix} 0 \\ \eta \end{bmatrix} \right) = \text{Proj}_{\Delta_{\mathcal{A}}} \left(\theta + \frac{1}{2} \begin{bmatrix} -\eta \\ \eta \end{bmatrix} \right).$

Applying update $U_{\text{DI}}(\theta, d, (1, 0), \eta_t)$ n times is equivalent to apply one update $U_{\text{DI}}(\theta, d, (1, 0), \sum_{t=1}^n \eta_t)$. Then, we face two cases:

- $\sum_{t=1}^n \eta_t \leq 1 \implies n < \frac{1}{\eta_1^2}$: starting from $\theta_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$, the 2D-projection does not have any effect.
- $\sum_{t=1}^n \eta_t \geq 1 \implies n \geq \frac{1}{\eta_1^2}$: the 2D-projection projects $U_{\text{DI}}(\theta, d, (1, 0), n\eta)$ on $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

In the first case, since we are looking for an upper bound of n' , we look for the maximal value of n' such that:

$$\sum_{t=1}^n \eta_t \geq \sum_{t=n+1}^{n+n'-1} \eta_t \quad (153)$$

$$\iff \sum_{t=1}^n \frac{\eta_1}{\sqrt{t}} \geq \sum_{t=n+1}^{n+n'-1} \frac{\eta_1}{\sqrt{t}} \quad (154)$$

$$\iff 2 \sum_{t=1}^n \frac{\eta_1}{\sqrt{t}} \geq \sum_{t=1}^{n+n'-1} \frac{\eta_1}{\sqrt{t}} \quad (155)$$

$$\implies 2\sqrt{n} \geq \sqrt{n+n'} - 1 \quad (156)$$

$$\iff n' \leq 3n + 4\sqrt{n} + 1 \quad (157)$$

In the second case, since we are looking for an upper bound of n' , we look for the maximal value of n' such that:

$$1 \geq \sum_{t=n+1}^{n+n'-1} \eta_t \quad (158)$$

$$\iff 1 \geq \sum_{t=n+1}^{n+n'-1} \frac{\eta_1}{\sqrt{t}} \quad (159)$$

$$\iff 1 + \sum_{t=1}^n \frac{\eta_1}{\sqrt{t}} \geq \sum_{t=1}^{n+n'-1} \frac{\eta_1}{\sqrt{t}} \quad (160)$$

$$\implies 1 + \eta_1 \sqrt{n} \geq \eta_1 \sqrt{n+n'} - \eta_1 \quad (161)$$

$$\iff \frac{1}{\eta_1} + \sqrt{n} \geq \sqrt{n+n'} - 1 \quad (162)$$

$$\iff \left(\frac{1}{\eta_1} + \sqrt{n} + 1 \right)^2 \geq n + n' \quad (163)$$

$$\iff \left(\frac{1}{\eta_1} + 1 \right)^2 + 2\sqrt{n} \left(1 + \frac{1}{\eta_1} \right) \geq n' \quad (164)$$

As a conclusion, it will take at most $n' = \min\{3n + 4\sqrt{n} + 1; (\frac{1}{\eta_1} + 1)^2 + 2\sqrt{n}(1 + \frac{1}{\eta_1})\}$ to recover from the n gradient steps.

Proof of (iii): We compute the policy gradient updates for classic cross-entropy and notice that they are equal to that of modified cross entropy when $|\mathcal{A}| = 2$:

- with $q(a_1) = 1, q(a_2) = 0$:

$$U_{\text{CE}}(\theta, d, (1, 0), \eta) \doteq \theta + \eta \left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} - \pi_\theta \right) = \theta + \eta(1 - \pi_\theta(a_1)) \begin{bmatrix} 1 \\ -1 \end{bmatrix} = U_{\text{MCE}}(\theta, d, (1, 0), \eta),$$

- with $q(a_1) = 0, q(a_2) = 1$:

$$U_{\text{CE}}(\theta, d, (0, 1), \eta) \doteq \theta + \eta \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \pi_\theta \right) = \theta + \eta(1 - \pi_\theta(a_1)) \begin{bmatrix} -1 \\ 1 \end{bmatrix} = U_{\text{MCE}}(\theta, d, (1, 0), \eta).$$

We focus on $\delta_t^{\rightarrow} \doteq (\theta_t)_{a_1} - (\theta_t)_{a_2}$ after t updates with $q(a_1) = 1, q(a_2) = 0$:

$$\delta_t^{\rightarrow} = \delta_{t-1}^{\rightarrow} + \eta_t(1 - \pi_{\theta_{t-1}}(a_1)) + \eta_t \pi_{\theta_{t-1}}(a_2) \quad (165)$$

$$= \delta_{t-1}^{\rightarrow} + 2\eta_t \frac{\exp((\theta_{t-1})_{a_2})}{\exp((\theta_{t-1})_{a_1}) + \exp((\theta_{t-1})_{a_2})} \quad (166)$$

$$\leq \delta_{t-1}^{\rightarrow} + 2 \frac{\eta_1}{\sqrt{t}} \exp(-\delta_{t-1}^{\rightarrow}) \quad (167)$$

$$\leq 2\eta_1 + \log(1 + 4\eta_1 \sqrt{t}) \quad \text{from Lemma 14.} \quad (168)$$

We now focus on $\delta_t^{\leftarrow} \doteq (\theta_t)_{a_1} - (\theta_t)_{a_2}$ after t updates with $q(a_1) = 0, q(a_2) = 1$, as long as $\delta_{t-1}^{\leftarrow} \geq 0$:

$$\delta_t^{\leftarrow} = \delta_{t-1}^{\leftarrow} - \eta_{t+n}(1 - \pi_{\theta_{t-1}}(a_2)) - \eta_t \pi_{\theta_{t-1}}(a_1) \quad (169)$$

$$= \delta_{t-1}^{\leftarrow} + 2\eta_t \frac{\exp((\theta_{t-1})_{a_1})}{\exp((\theta_{t-1})_{a_1}) + \exp((\theta_{t-1})_{a_2})} \quad (170)$$

$$\geq \delta_{t-1}^{\leftarrow} + \eta_t \quad (171)$$

As consequence, the number n' of updates δ^{\leftarrow} required to recover from a convergence to δ_n^{\rightarrow} is lower than the smallest n' for which $\sum_{t=1}^{n'} \eta_{t+n} \geq 2\eta_1 + \log(1 + 4\eta_1 \sqrt{n})$. Further analysis is similar to that of (ii) and gives:

$$n' \leq \left(4 + \frac{\log(1 + 4\eta_1 \sqrt{n})}{2\eta_1}\right)^2 + \sqrt{n} \left(8 + \frac{\log(1 + 4\eta_1 \sqrt{n})}{\eta_1}\right) \quad (172)$$

□

Theorem 8 (Domino setting). *Under the assumption that η and $d(s_k) > 0$ are constant, in order to solve the setting,*

(i) U_{PG} updates require at least $2^{|\mathcal{S}|-1}$ steps.

(ii) U_{DI} updates require at most $1 + \frac{|\mathcal{S}|-1}{\eta}$ steps.

(iii) U_{CE} and U_{MCE} updates require at most:
 $\frac{32e^{8\eta+3}}{\eta^3} (|\mathcal{S}|-1) \log(|\mathcal{S}|-1)$ steps.

Proof. Let $h = |\mathcal{S}|$ be the horizon, or the number of dominos to flip. Let n_k be the time the flip domino k , starting from the last one. We obtain the following recurrence:

$$n_h = 1 \quad (173)$$

$$n_k = n_{k+1} + n'_k, \quad (174)$$

where n'_k is the time to recover once domino $k+1$ has flipped. n'_k is dependent on the realized updates as Theorem 6 describes, and on n_{k+1} : the time for domino $k+1$ to flip.

Proof of (i): We get $n'_k \geq n_{k+1}$:

$$n_k \geq 2n_{k+1} \geq 2^{h-k}, \quad (175)$$

hence at least a geometric dependency on h .

Proof of (ii): We get $n'_k \leq \left\lceil \frac{1}{\eta} \right\rceil$:

$$n_k \leq n_{k+1} + \left\lceil \frac{1}{\eta} \right\rceil \leq (h-k) \left\lceil \frac{1}{\eta} \right\rceil + 1 \quad (176)$$

hence at most a linear dependency on h .

Proof of (iii): We get $n'_k \leq 2 + \frac{1}{\eta} \log(1 + 2\eta n_{k+1})$:

$$n_k \leq n_{k+1} + 2 + \frac{1}{\eta} \log(1 + 2\eta n_{k+1}) \quad (177)$$

We rely on technical Lemma 15 to finish the proof and obtain:

$$n'_k \leq \frac{32e^{8\eta+3}}{\eta^3} (h - k) \log(h - k). \quad (178)$$

for any $h - k \geq 8$. □

A.5 Technical lemmas

Lemma 12 (Maximal update conservation through projection on the simplex). *Let $\theta \in \Delta_{\mathcal{X}}$ be a point on the simplex, and $\delta \in \mathbb{R}^{|\mathcal{X}|}$ be any vector. Then, the projection $\theta' \doteq \text{Proj}_{\Delta_{\mathcal{X}}}(\theta + \delta)$ on the simplex is such that:*

$$\theta'_{k_*} - \theta_{k_*} \geq \theta'_k - \theta_k, \quad \text{where} \quad k_* \in \underset{k \in [|\mathcal{X}|]}{\text{argmax}} \delta_k, \quad \forall k \in [|\mathcal{X}|]. \quad (179)$$

Proof. Let us assume without loss of generality that $\forall k, \delta_k \geq 0$ (by e.g. subtracting from each of them $\min \delta_k$). Then, $\theta + \delta$ is positive and sums to higher than 1.

In these conditions, an algorithm for finding the projection consists in removing mass equally across all indices with positive (non null) mass. This algorithm thus squeezes the low weights of the vector to zero until its ℓ_1 -norm reaches 1 and therefore belongs to the simplex. It is clear from this algorithm that $\theta'_k = \max\{0; \theta_k + \delta_k - \lambda\}$ for some constant $0 \leq \lambda \leq \delta_{k_*}$.

We may infer that:

$$\theta'_k - \theta_k = \max\{-\theta_k; \delta_k - \lambda\} \quad (180)$$

$$\leq \max\{-\theta_k; \delta_{k_*} - \lambda\} \quad (181)$$

$$= \delta_{k_*} - \lambda \quad (182)$$

$$= \max\{-\theta_{k_*}; \delta_{k_*} - \lambda\} \quad (183)$$

$$= \theta'_{k_*} - \theta_{k_*}, \quad (184)$$

which concludes the proof. □

Lemma 13. *Let us consider a sequence $(X_n)_{n \in \mathbb{N}}$ verifying:*

$$\begin{cases} X_0 = 0 \\ X_{n+1} \leq X_n + ce^{-X_n}. \end{cases} \quad (185)$$

Then, we have the following upper bound on X_n : For any $n \geq 0$, $X_n \leq c + \log(1 + cn)$.

Proof. We start by proving this result for the sequence $(Y_n)_{n \in \mathbb{N}}$ defined recursively as:

$$\begin{cases} Y_0 = 0 \\ Y_{n+1} = Y_n + ce^{-Y_n}. \end{cases} \quad (186)$$

As a first step, we introduce two functions on \mathbb{R}_+ , $y(t)$ and $z(t)$, respectively solutions on $[0, +\infty)$ of the ODEs:

$$\begin{cases} y(0) = 0 \\ y'(t) = ce^{-y(\lfloor t \rfloor)}, \end{cases} \quad (187)$$

and

$$\begin{cases} z(0) = 0 \\ z'(t) = ce^{-z(t)}. \end{cases} \quad (188)$$

Clearly, for any $n \in \mathbb{N}$, $Y_n = y(n)$ and for any $t \geq 0$, $z(t) \leq y(t)$. In other words, we have that $Y_n \geq z(n)$.

Solving the ode z verifies gives us: $z(t) = \log(1 + ct)$. Now, we move to upper-bounding Y_n for $n \geq 1$:

$$Y_n = c \sum_{i=0}^{n-1} e^{-Y_i} \leq c \sum_{i=0}^{n-1} e^{-z(i)} = \sum_{i=0}^{n-1} \frac{c}{1+ci} \leq c + \sum_{i=1}^{n-1} \int_{i-1}^i \frac{c}{1+ct} dt \quad (189)$$

$$= c + \int_0^{n-1} \frac{c}{1+ct} dt = c + \log(1 + c(n-1)). \quad (190)$$

Combined with the value of Y_0 , this guarantees that $\forall n \in \mathbb{N}, Y_n \leq c + \log(1 + cn)$.

We are left with proving that $X_n \leq Y_n$. Let us assume towards a contradiction that there exists n such that $X_{n+1} > Y_{n+1}$, and let us pick the smallest of such n (clearly, we have $n \geq 1$ as $X_0 = Y_0 = 0$ and $Y_1 = c \geq X_1$). Then:

$$X_{n+1} > Y_{n+1} \Leftrightarrow Y_n - X_n < c(e^{-X_n} - e^{-Y_n}) < ce^{-X_n}(1 - e^{X_n - Y_n}) < ce^{-X_n}(Y_n - X_n) \quad (191)$$

$$\Leftrightarrow 1 < ce^{-X_n} \quad (192)$$

$$\Leftrightarrow X_n < \log c, \quad (193)$$

which is not possible as $\forall n \geq 1, X_n \geq c$ and concludes the proof. \square

Lemma 14. *Let us consider a sequence $(X_n)_{n \in \mathbb{N}}$ verifying:*

$$\begin{cases} X_0 = 0 \\ X_{n+1} \leq X_n + \frac{c}{\sqrt{n+1}} e^{-X_n}. \end{cases} \quad (194)$$

Then, we have the following upper bound on X_n : For any $n \geq 0$, $X_n \leq c + \log(1 + 2c\sqrt{n})$.

Proof. We prove this result for the sequence $(Y_n)_{n \in \mathbb{N}}$ defined recursively as:

$$\begin{cases} Y_0 = 0 \\ Y_{n+1} = Y_n + \frac{c}{\sqrt{n+1}} e^{-Y_n}. \end{cases} \quad (195)$$

To do so, we introduce two functions on \mathbb{R}_+ , $y(t)$ and $z(t)$, respectively solutions on $[0, +\infty)$ of the ODEs:

$$\begin{cases} y(0) = 0 \\ y'(t) = \frac{c}{t+1} e^{-y(t)}, \end{cases} \quad (196)$$

and

$$\begin{cases} z(0) = 0 \\ z'(t) = \frac{c}{t+1} e^{-z(t)}. \end{cases} \quad (197)$$

Clearly, for any $n \in \mathbb{N}$, $Y_n = y(n)$ and for any $t \geq 0$, $z(t) \leq y(t)$. In other words, we have that $Y_n \geq z(n)$.

Solving the ode z verifies gives us: $z(t) = \log(1 - 2c + 2c\sqrt{t+1})$. Now, we move to upper-bounding Y_n for $n \geq 1$:

$$Y_n = c \sum_{i=0}^{n-1} \frac{e^{-Y_i}}{\sqrt{i+1}} \leq c \sum_{i=0}^{n-1} \frac{e^{-z(i)}}{\sqrt{i+1}} = \sum_{i=0}^{n-1} \frac{c}{1-2c+2c\sqrt{i+1}} \frac{1}{\sqrt{i+1}} \leq c + \int_1^n \frac{c}{(1-2c)\sqrt{t}+2ct} dt \quad (198)$$

$$= c + \log(1 - 2c + 2c\sqrt{n}). \quad (199)$$

Combined with the value of Y_0 , this guarantees that $\forall n \in \mathbb{N}, Y_n \leq c + \log(1 + 2c\sqrt{n})$.

We are left with proving that $X_n \leq Y_n$. Let us assume towards a contradiction that there exists n such that $X_{n+1} > Y_{n+1}$, and let us pick the smallest of such n (clearly, we have $n \geq 1$ as $X_0 = Y_0 = 0$ and $Y_1 = c \geq X_1$). Then:

$$X_{n+1} > Y_{n+1} \Leftrightarrow Y_n - X_n < c \frac{e^{-X_n} - e^{-Y_n}}{n+1} < ce^{-X_n} \frac{1 - e^{X_n - Y_n}}{n+1} < ce^{-X_n} \frac{Y_n - X_n}{n+1} \quad (200)$$

$$\Leftrightarrow 1 < \frac{ce^{-X_n}}{n+1} \quad (201)$$

$$\Leftrightarrow X_n < \log \frac{c}{n+1}, \quad (202)$$

which is not possible as $\forall n \geq 1, X_n \geq c$ and concludes the proof. \square

Lemma 15. *Let us consider a sequence $(X_n)_{n \in \mathbb{N}}$ verifying:*

$$\begin{cases} X_0 = 1 \\ X_{n+1} \leq X_n + 2 + \frac{1}{\eta} \log(1 + 2\eta X_n). \end{cases} \quad (203)$$

Then, we have the following upper bound on X_n :

$$\forall n \geq 8, \quad X_n \leq \frac{32e^{8\eta+3}}{\eta^3} n \log n. \quad (204)$$

Proof. We look for a function $f_\epsilon(x) = c_\epsilon x^\epsilon$ such that for all $x \geq 0$:

$$f_\epsilon(x) \geq \log(x) \quad (205)$$

$$\iff c_\epsilon x^\epsilon \geq \log x \quad (206)$$

$$\iff c_\epsilon y \geq \log(y^{\frac{1}{\epsilon}}) \quad \text{setting } y = x^\epsilon \text{ with } y \geq 0 \quad (207)$$

$$\iff c_\epsilon y \geq \frac{1}{\epsilon} \log y \quad (208)$$

Since $x \geq \log x$ for all $x > 0$, it suffices to choose $c_\epsilon = \frac{1}{\epsilon}$ to satisfy the desiderata. So we get:

$$\begin{cases} X_0 = 1 \\ X_{n+1} \leq X_n + 2 + \frac{1}{\eta} \log(1 + 2\eta X_n) = X_n + \frac{1}{\eta} \log(e^{2\eta}(1 + 2\eta X_n)) \\ \leq X_n + \frac{(e^{2\eta}(1 + 2\eta X_n))^\epsilon}{\eta\epsilon} = X_n + \frac{e^{2\eta\epsilon}(1 + 2\eta X_n)^\epsilon}{\epsilon\eta}. \end{cases} \quad (209)$$

As a first step, we introduce the function $y(t)$ on \mathbb{R}_+ solution on $[0, +\infty)$ of the ODE:

$$\begin{cases} y(0) = 1 \\ y'(t) = \frac{e^{2\eta\epsilon}(1 + 2\eta y(t))^\epsilon}{\epsilon\eta}. \end{cases} \quad (210)$$

Let $z(t) = 1 + 2\eta y(t)$, then $z'(t) = 2\eta y'(t)$ and we get:

$$\begin{cases} z(0) = 1 + 2\eta \\ z'(t) = \frac{2e^{2\eta\epsilon}}{\eta\epsilon} z(t)^\epsilon. \end{cases} \quad (211)$$

Solving the ode gives:

$$z'(t) = \frac{2e^{2\eta\epsilon}}{\eta\epsilon} z(t)^\epsilon \quad (212)$$

$$\iff \frac{z'(t)}{z(t)^\epsilon} = \frac{2e^{2\eta\epsilon}}{\eta\epsilon} \quad (213)$$

$$\iff \int_{z(0)}^{z(t)} \frac{dz}{z^\epsilon} = \int_0^t \frac{2e^{2\eta\epsilon}}{\eta\epsilon} du = \frac{2e^{2\eta\epsilon}}{\eta\epsilon} t \quad (214)$$

$$\iff \frac{1}{1-\epsilon} [z(t)^{1-\epsilon} - z(0)^{1-\epsilon}] = \frac{2e^{2\eta\epsilon}}{\eta\epsilon} t \quad (215)$$

$$\iff z(t)^{1-\epsilon} = (1 + 2\eta)^{1-\epsilon} + \frac{2(1-\epsilon)e^{2\eta\epsilon}}{\eta\epsilon} t \quad (216)$$

$$\iff z(t) = \left[(1 + 2\eta)^{1-\epsilon} + \frac{2(1-\epsilon)e^{2\eta\epsilon}}{\eta\epsilon} t \right]^{\frac{1}{1-\epsilon}}. \quad (217)$$

This results in:

$$y(t) = \frac{1}{2\eta} \left[\left[(1 + 2\eta)^{1-\epsilon} + \frac{2(1-\epsilon)e^{2\eta\epsilon}}{\eta\epsilon} t \right]^{\frac{1}{1-\epsilon}} - 1 \right]. \quad (218)$$

We can thus upper-bound X_n as follows:

$$X_n \leq \frac{1}{2\eta} \left[(1 + 2\eta)^{1-\epsilon} + \frac{2e^{2\eta}}{\eta\epsilon} n \right]^{\frac{1}{1-\epsilon}} \quad (219)$$

We choose $\epsilon = \frac{1}{\log n}$ for $n \geq 8 > e^2$ and obtain:

$$X_n \leq \frac{1}{2\eta} \left[(1 + 2\eta)^{1-\frac{1}{\log n}} + \frac{2e^{2\eta}}{\eta} n \log n \right]^{1+\frac{1}{\log n-1}} \quad (220)$$

$$\leq \frac{1}{2\eta} \left[1 + 2\eta + \frac{2e^{2\eta}}{\eta} n \log n \right]^{1+\frac{1}{\log n-1}} \quad (221)$$

$$\log X_n \leq -\log(2\eta) + \left(1 + \frac{1}{\log n - 1} \right) \log \left[1 + 2\eta + \frac{2e^{2\eta}}{\eta} n \log n \right] \quad (222)$$

$$\leq -\log(2\eta) + \left(1 + \frac{1}{\log n - 1} \right) \left[2 \log 2 + \log(1 + 2\eta) + \log \left(\frac{2e^{2\eta}}{\eta} n \log n \right) \right] \quad (223)$$

$$\leq -\log \eta - \log 2 + \left(1 + \frac{1}{\log n - 1} \right) [3 \log 2 + 4\eta - \log \eta + \log n + \log \log n] \quad (224)$$

$$= -2 \log \eta + 2 \log 2 + 4\eta + \log n + \log \log n + 1 + \kappa(n) \quad (225)$$

$$X_n \leq \exp(-2 \log \eta + 2 \log 2 + 4\eta + \log n + \log \log n + 1 + \kappa(n)) \quad (226)$$

$$\leq \frac{4e^{4\eta+1}}{\eta^2} n \log n \exp(\kappa(n)), \quad (227)$$

where $\kappa(n) \doteq \frac{3 \log 2 + 4\eta - \log \eta + 1 + \log \log n}{\log n - 1} \leq 2 + 3 \log 2 + 4\eta - \log \eta$ granted that $n \geq e^2$. $\kappa(n)$ is generally in $o(1)$ and therefore $\exp(\kappa(n))$ will converge to 1 asymptotically. For the all-time upper bound, we use:

$$X_n \leq \frac{32e^{8\eta+3}}{\eta^3} n \log n. \quad (228)$$

□

B Experiments

B.1 Domains

The Random MDP domain is taken from Laroche et al. (2019); Nadjahi* et al. (2019); Simão et al. (2020); Laroche and Tachet des Combes (2021), where it is fully described in Section B.1.3. We set the three hyperparameters as follows: number of states = 100, number of actions = 4, connectivity of transitions = 2.

The Chain domain is borrowed from Laroche and Tachet des Combes (2021), where it is fully described in Section C.1. We set hyper parameter β to 0.7. The number of states varies from one figure to another where it is always specified.

The cliff experiment is novel to this work and is a direct adaptation of the chain algorithm where a third action is added. This new action is terminal and yields a 0 reward. Intuitively, it should only marginally slow down the algorithms, but in practice some of the algorithms get significantly slowed down.

In all domains the discount factor γ is set to 0.99.

B.2 Algorithms

We use the Jekyll&Hyde algorithm from Laroche and Tachet des Combes (2021), which we recall below:

Input: Scheduling of exploration (ϵ_t), scheduling of off-policiness (o_t) and actor learning rate η .

- 1: Initialize Dr Jekyll’s replay buffer $\mathring{D} = \emptyset$, actor $\mathring{\pi}$, and critic \mathring{q} . ▷ exploitation agent
- 2: Initialize Mr Hyde’s replay buffer $\tilde{D} = \emptyset$, and policy $\tilde{\pi}$. ▷ exploration agent
- 3: Set the behavioural policy and working replay buffer to Dr Jekyll’s: $\pi_b \leftarrow \mathring{\pi}$ and $D_b \leftarrow \mathring{D}$.
- 4: **for** $t = 0$ to ∞ **do**
- 5: Sample a transition $\tau_t = \langle s_t, a_t \sim \pi_b(\cdot|s_t), s_{t+1} \sim p(\cdot|s_t, a_t), r_t \sim r(\cdot|s_t, a_t) \rangle$.
- 6: Add it to the working replay buffer $D_b \leftarrow D_b \cup \{\tau_t\}$.
- 7: **if** τ was terminal, **then** $(\pi_b, D_b) \leftarrow (\tilde{\pi}, \tilde{D})$ w.p. ϵ_t , $(\pi_b, D_b) \leftarrow (\mathring{\pi}, \mathring{D})$ otherwise.
- 8: **if** Update step, **then**
- 9: $\tau \doteq \langle s, a, s', r \rangle \sim \tilde{D}$ w.p. o_t , $\tau \doteq \langle s, a, s', r \rangle \sim \mathring{D}$ otherwise.
- 10: Update Mr Hyde’s policy $\tilde{\pi}$ on τ . ▷ with Q -learning trained on UCB rewards
- 11: Update Dr Jekyll’s critic \mathring{q} on τ . ▷ with SARSA update
- 12: Expected update of Dr Jekyll’s actor $\mathring{\pi}$ in state s . ▷ with Eq. (229)
- 13: **end if**
- 14: **end for**

Algorithm 1: Dr Jekyll & Mr Hyde algorithm. After initialization of parameters and buffers, we enter the main loop. At every time step, an action, chosen by the behavioral policy, is executed in the environment to produce a transition τ_t (line 5). τ_t is stored in the replay buffer of the personality in control (either Dr Jekyll or Mr Hyde, line 6). If the trajectory is done, the algorithm samples a new personality to be in control during the next one (line 7). Then, the updates of the models start (line 8). The update for Mr Hyde’s policy $\tilde{\pi}$ is performed with Q -learning trained on UCB rewards. Dr Jekyll’s critic \mathring{q} is trained with SARSA. When $(\epsilon_t) = 0$, Jekyll&Hyde amounts to on-policy expected updates from a single replay buffer.

Step 12 requires the use of approximate expected policy update (Silver et al., 2014; Lillicrap et al., 2016; Ciosek and Whiteson, 2018):

$$\hat{U}(\theta, s) \doteq \sum_{a \in \mathcal{A}} \mathring{q}(s, a) \nabla_{\theta} \pi(a|s). \quad (229)$$

We set Jekyll&Hyde hyperparameters as follows:

- Critic learning rate: $\eta_c = 0.1$, and critic initialization: $q_0 = 0$,
- Scheduling of exploration: $\epsilon_t = 0$ in NOEXPLO, $\epsilon_t = \frac{10}{\sqrt{t}}$ in LOWOFFPOL, and $\epsilon_t = \frac{10}{\sqrt{t}}$ in HI OFFPOL,

- Scheduling of off-policiness: $o_t = 0$ in NOEXPLO, $o_t = \frac{10}{\sqrt{t}}$ in LOWOFFPOL, and $o_t = 0.5$ in HIUFFPOL,
- In all the experiments, Mr Hyde is trained with Q -learning on a γ discounted objective from UCB rewards: $\tilde{r}(s, a) \doteq \frac{1}{\sqrt{n_{s,a}}}$. It has the same learning rate and initialization as the critic of Dr Jekyll.